

A Decentralized Approach to Dynamic Collaborative Driving Coordination

by

Thanh-Son Dao

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2008

© Thanh-Son Dao 2008

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Thanh-Son Dao

Abstract

This thesis presents a novel approach to several problems in intelligent transportation systems using collaborative driving coordination. With inter-vehicle communication and intelligent vehicle cooperation, important tasks in transportation such as lane position determination, lane assignment and platoon formation can be solved. Several topics in regard to inter-vehicle communication, lane positioning, lane assignment and platoon formation are explored in this thesis:

First, the design and experimental results of low-cost lane-level positioning system that can support a large number of transportation applications are discussed. Using a Markov-based approach based on sharing information among a group of vehicles that are traveling within the communication range of each other, the lane positions of vehicles can be determined. The robustness effectiveness of the system is shown in both simulations and real road tests.

Second, a decentralized approach to lane scheduling for vehicles with an aim to increase traffic throughput while ensuring the vehicles exit successfully at their destinations is presented. Most of current traffic management systems do not consider lane organization of vehicles and only regulate traffic flows by controlling traffic signals or ramp meters. However, traffic throughput and efficient use of highways can be increased by coordinating driver behaviors intelligently. The lane optimization problem is formulated as a linear programming problem that can be solved using the Simplex method.

Finally, a direction for cooperative vehicle platoon formation is proposed. To enhance traffic safety, increase lane capacities and reduce fuel consumption, vehicles can be organized into platoons with the objective of maximizing the travel distance that platoons stay intact. Toward this end, this work evaluates a proposed strategy which assigns vehicles to platoons by solving an optimization problem. A linear model for assigning vehicles to appropriate platoons when they enter the highway is formulated. Simulation results demonstrate that lane capacity can be increased effectively when platooning operation is used.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Christopher M. Clark, for giving me a chance to study in his team. His knowledge and his logical way of thinking have been of great value for me. With his enthusiasm, his inspiration, and his efforts to explain things logically and simply, he helped to make robotics fun for me. Throughout my time at the University of Waterloo, he provided sound advice, good teaching, good company, and lots of good ideas. I would have been lost without him. Besides of being an excellent supervisor, he also provided a friendly environment and was always willing to help me with my personal issues during his stay at Waterloo.

I would like to thank Professor Jan P. Huissoon for advising me and funding my research. He provided a steady stream of insights, suggestions, advice, and brainstorming. I am very grateful for his patience, motivation, as well as his detailed and constructive comments and important support throughout this work.

I wish to express my thanks to Professor Bruce Hellinga who gave me the first knowledge in transportation systems. I also want to thank him for his valuable advice and friendly help with transportation problems, and sat in both my defense and comprehensive committees.

I am grateful to the official referees, Professor Steve Lambert, Professor William W. Melek, and Professor Ata Khan. I appreciate their time and effort as well as their insightful comments to provide detailed review, constructive criticism and excellent advice during the preparation of this thesis.

I wish to thank Keith Leung and Luke Ng for being good team players and friends. The work described in Chapter 4 was done together with them. Without Keith's help with VISSIM and GPS stuff, and Luke's help with experiment, my research would be much more demanding and difficult.

I have never once regretted my decision to come to the University of Waterloo. It has been an honor to work in the department's friendly, supportive, and cooperative environment. Thanks in general to all my friends among the faculty, staff, and students.

It is a pleasure to express my gratitude wholeheartedly to my friends Cynthia, Mary, and Dong's family for their kind hospitality and help with my family issues and the birth of my son Andrew.

Thank you to my parents for their tireless dedication and absolute confidence in me, and to my brother, and my second parents for their perpetual care and support.

Finally, it is impossible to believe that I might have completed this work without the love, support, patience, devotion, and persistent confidence in me of my loving wife Giang.

She has always looked after my family and taken the load off my shoulders whenever I needed time to work on my research. One of the best experiences that we lived through in my PhD period was the birth of our son Andrew, who provided an additional and joyful dimension to our life mission. To my wife, and to my son Andrew, I owe my acknowledgment, my dedication, and my love.

Finally, I would like to thank NSERC and AUTO21 Canada for financial funding of this research.

Contents

List of Tables	x
List of Figures	xi
Nomenclature	xv
1 Introduction	1
1.1 Intelligent Transportation Systems	1
1.2 Thesis Objectives	3
1.3 Background to this Thesis	5
1.4 Contributions	7
1.4.1 Lane Positioning	7
1.4.2 Lane Assignment	8
1.4.3 Platoon Assignment	8
2 Collaborative Driving	9
2.1 Introduction	9
2.2 Inter-vehicle Communication	10
2.2.1 Overview	10
2.2.2 Ad-hoc Networks for Inter-vehicle Communication	10
2.2.3 Routing for Vehicular Networks	12
2.3 Collaborative Driving Simulator	13

3	Perception for Autonomous Driving	17
3.1	Introduction	17
3.2	Cameras	17
3.3	Laser Rangefinders	20
3.4	Global Positioning Systems	21
4	Lane Position Determination	27
4.1	Introduction	27
4.2	Problem Statement	27
4.3	Related Work	28
4.4	Approach to Lane Positioning	29
4.5	Markov Localization	30
4.5.1	Prediction	31
4.5.2	Correction	34
4.5.3	Generalization of Prediction and Correction	34
4.6	Simulation	35
4.7	Experiment Implementation	39
4.7.1	Software Architecture	39
4.7.2	GPS Receiver Noise Reduction	40
4.7.3	Results	44
4.8	Conclusions	55
5	Lane Assignment Optimization	58
5.1	Introduction	58
5.2	Related Work	58
5.3	Problem Statement	61
5.4	Linear Programming	62
5.5	System Architecture	62
5.6	Traffic Flow Estimation	64

5.6.1	Formulation	64
5.6.2	Simulation	65
5.7	Lane Assignment Optimization	66
5.7.1	Problem Formulation	66
5.7.2	Analysis of Maneuver Cost	70
5.7.3	Van Aerde’s Car-Following Model	74
5.8	Simulation	76
5.9	Selfish Lane Selection Versus System Optimality	80
5.10	Conclusions	82
6	Intelligent Platoon Formation	85
6.1	Introduction	85
6.2	Platoon Control Diagram	87
6.3	Strategy for Platoon Formation at Entrances	88
6.4	Problem Formulation	90
6.5	Simulation	92
6.6	Conclusions	94
7	Conclusions and Recommendations	98
7.1	Summary of the Thesis	98
7.2	Known Limitations	99
7.3	Future Research	100
7.3.1	Dynamic Capacity	100
7.3.2	Selfish Lane Selection	100
7.3.3	Fuel Consumption Estimation for Platooning	100
7.3.4	Lane Assignment for Vehicle Platoons	101
	Appendix	102
	A Simplex Algorithm	102

B VISSIM Simulation Setting	104
C Lane Capacity	108
Bibliography	109

List of Tables

1.1	Annual Total Costs of Congestion by City (millions of 2002 \$).	3
2.1	D-Link DWL-AG660 technical specifications.	14
3.1	Hokuyo URG-04LX technical specifications.	22
3.2	Garmin-18 technical specifications.	25
3.3	LocSense 40-CM technical specifications.	26
4.1	Percentage of time with correct lane estimation for simulations.	40
4.2	Percentage of time with correct lane estimation for experiments.	56
5.1	Comparison between VISSIM-controlled simulations and lane assignment simulations.	79
B.1	Highway parameters.	105
B.2	Vehicle following behavior parameters in VISSIM-controlled simulations (VISSIM default setting).	105
B.3	Lane change parameters in VISSIM-controlled simulations (VISSIM default setting).	105

List of Figures

1.1	Highway congestion is causing negative effects in many cities.	2
1.2	Examples of ITS technologies that can be deployed and integrated in urban areas.	4
1.3	Flow-chart of thesis layout.	6
2.1	Inter-vehicle communication.	11
2.2	Several localization techniques for VANETs.	12
2.3	D-Link DWL-AG660 802.11a/11g 108Mbps Wireless PCMCIA Adapter. . .	13
2.4	Accessing VISSIM vehicle parameters [31].	15
2.5	Software architecture for collaborative driving simulator module [31]. . . .	16
3.1	Some applications of computer vision in ITS.	18
3.2	Road-detection result for highways and the 3-D projected model with different conditions.	19
3.3	Obstacle detection example.	19
3.4	Hokuyo URG-04LX laser rangefinder.	21
3.5	The functioning of a GPS.	23
3.6	Garmin-18 GPS receivers.	24
3.7	LocSense 40-CM GPS.	24
4.1	Inter-vehicle communication.	29
4.2	Lane positioning diagram.	31
4.3	Prediction step.	32
4.4	Radius of curvature estimation.	33

4.5	Least squares method for approximating road curvature.	33
4.6	Correction step.	35
4.7	Highway 85, Waterloo, ON, Canada.	36
4.8	Probability distribution for simulation of three cars.	37
4.9	Simulation result for three cars.	38
4.10	Impact of oscillatory weaving.	38
4.11	Oscillatory weaving with filter.	39
4.12	Program architecture.	41
4.13	Screenshot of localization program.	41
4.14	Re-sampling: particles with higher weights are more likely to be selected and sampled.	43
4.15	A pictorial description of particle filtering.	44
4.16	GPS measurements for two computers.	46
4.17	Relative distance from raw GPS data vs. filtered data.	46
4.18	GPS measurements for two cars in experiment.	47
4.19	Probability distribution for experiment without GPS measurement filtering.	48
4.20	Lane estimation result for two cars.	49
4.21	Improved probability distribution for experiment.	50
4.22	Improved results with GPS measurement filtering.	51
4.23	Experiment apparatus.	51
4.24	Probability distributions for experiment 1.	52
4.25	Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).	53
4.26	GPS measurements for experiment 2.	53
4.27	Probability distributions for experiment 2.	54
4.28	Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).	55
4.29	Probability distributions for experiment 3.	56
4.30	Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).	57

5.1	Changeable lane assignment signs.	60
5.2	Highway system discretization in [78].	60
5.3	Control diagram for lane assignment	64
5.4	Highway system.	64
5.5	Flow rate estimation.	65
5.6	A screen shot of lane estimation application.	66
5.7	Estimated total flow.	66
5.8	Lane assignment strategy.	67
5.9	Vehicles traveling further tend to travel in a faster lane to reduce maneuver cost and vehicle travel time.	70
5.10	Inserting vehicle A into traffic flow with uniform average separation distance Δh between vehicles.	71
5.11	Velocity profile for a trailing vehicle in target lane.	71
5.12	Lane 2 before and after lane-change.	73
5.13	Determine adjustment spacing.	73
5.14	Number of affected vehicles.	74
5.15	Time cost versus average separation distance.	75
5.16	Speed-flow relationships for simulations.	78
5.17	Improvement over average travel time for 1000 cars with the input flow being 2000 <i>veh/hr</i> (top), 4000 <i>veh/hr</i> (middle), and 6000 <i>veh/hr</i> (bottom).	79
5.18	Trend of increased average travel time as input flow increases from 1000 <i>veh/hr</i> to 6500 <i>veh/hr</i>	80
5.19	Probability distributions of maneuver cost due to lane-change.	81
5.20	Average and standard deviation of maneuver cost due to lane-change.	81
5.21	Percentage of vehicles not complying to suggested lane assignment strategy with the input flow being (a) 2000 <i>veh/hr</i> , (b) 4000 <i>veh/hr</i> , and (c) 6000 <i>veh/hr</i>	83
5.22	Steady-state average travel time versus percentage of non-participating vehicles under different input flows.	84

6.1	(a) Three-Lumina platoon in the Dryden wind tunnel at USC, and (b) platoon operation at 4-meter spacing to measure aerodynamic drag and fuel consumption [95].	86
6.2	Control diagram for platoon assignment.	87
6.3	Platoon assignment strategy.	89
6.4	Highway used in simulations.	92
6.5	Screenshot of VISSIM simulation of platooning operation.	93
6.6	Platoon size probability distribution versus destination range r and input flow.	95
6.7	(a) Average and (b) standard deviation of platoon size versus destination range r and input flow.	96
6.8	(a) Average and (b) standard deviation of platoon size versus destination range r and input flow.	96
6.9	Lane capacity versus destination range r and input flow.	97
6.10	Platoon ratio versus destination range r and input flow.	97
A.1	Feasible region [69].	102
C.1	Lane capacity calculation.	108

Nomenclature

P : Probability

l : Lane index

σ : Standard deviation

t : Time [sec]

ΔD : The shortest distance from the vehicle position estimate from the GPS reading to the line passing through the two most recent position estimates [m]

z_t : Sensory measurement at time t

n_v : Number of vehicles

n_l : Number of lanes

w_t : Weight of particle at time t

u_t : Control input

f : Instantaneous flow rate estimate [veh/hr]

i : Road segment index

d : Estimated vehicle travel distance [m]

T : Lane change maneuver cost [sec]

n_d : Number of destinations

v_l : Nominal velocity of lane l

α : Percentage of vehicles

δh : Separation distance between two vehicles [m]

δh_{min} : Minimum separation distance between two vehicles [m]

s : Average vehicle length [m]

c_Σ : Total relative distance that trailing vehicles in target lane must slow down to create a gap for the lane changer [m]

τ : Lane change time constant [sec]

Δt : Time delay caused by lane change [sec]

Δt_{en} : Time delay caused by lane change when a vehicle enters highway [sec]

Δt_{ext} : Time delay caused by lane change when a vehicle exits highway [sec]

δ : Traffic density [*veh/km*]
 δ_j : Jam density [*veh/km*]
 v_f : Vehicle free speed [*km/hr*]
 C_{max} : Lane capacity [*veh/hr*]
 h : Intra-platoon distance [*m*]
 Υ : Maximum platoon size
 t_h : Time headway between platoons [*sec*]
 R : Communication range [*m*]
 ε_p : Size of platoon p
 r : Destination range [*m*]

Chapter 1

Introduction

1.1 Intelligent Transportation Systems

Increasing passenger and freight travel has led to growing congestion in many countries' transportation systems, which have posed a burden on the quality of life through wasted energy, time, and money; increased pollution; and threats to safety (see Fig. 1.1¹). According to transportation researchers, even with slow growth in jobs and travel, the cost of transportation in terms of extra fuel used and time spent in congestion was \$63 billion in 2003 in the US economy [1], and between \$2.3 billion and \$3.7 billion for the nine largest urban areas in Canada in 2002 [2]. Specifically, more than 90% of this cost was associated with the time lost in traffic to drivers and passengers; 7% occurred because of fuel consumed; and 3% was from increased greenhouse gas emissions [2].

The annual total costs of congestion by Canadian city in 2002 are summarized in Table 1.1 in which urban congestion levels were measured using a speed threshold that reflects a percentage of free flow speed along a roadway. This study also estimated an increase of 1.2 to 1.4 megatonnes of greenhouse gas due to congestion every year. Moreover, passenger and freight traffic are expected to grow substantially in the future, increasing the challenge of preventing congestion from overwhelming the transportation systems. Also according to the US Government Accountability Office (GAO) [1], by 2010, the US Department of Transportation forecasts that travel on roads will have increased by about 25% from 2000, while freight traffic will have increased by 43% from 1998.

It is important to note that these estimates of congestion costs are conservative because there was insufficient data on the costs associated with non-recurrent congestion (*i.e.*, congestion caused by random events, such as bad weather, accidents, stalled vehicles

¹Image from <http://www.urbancompass.net>

and other incidents), freight transportation, off-peak congestion, vehicle operating costs other than fuel, and other congestion-related costs such as noise and stress. Therefore, the actual costs due to congestion are much higher than the costs estimated.



Figure 1.1: Highway congestion is causing negative effects in many cities.

There have been a wide range of strategies employed with the effort to reduce the effect of increasing congestion, including building new infrastructure, corrective and preventive maintenance, rehabilitation, and managing system use through pricing or other techniques. However, although building new infrastructure can ease congestion, it is not always a viable solution due to constraints such as the cost of construction or limited availability of land.

Another tool available to help reduce congestion is the use of Intelligent Transportation Systems (ITS) in order to improve transportation system operations, management, and performance. Many transportation projects have been initiated to define the next generation of land transportation systems with the objective of improving road traffic efficiency and safety. Examples of these programs include: the California Partners for Advanced Transit and Highway (PATH) program in the US, the Automobile of the 21st Century (AUTO21) project in Canada, as well as the Prometheus and DRIVE projects in Europe, and the Super-smart Vehicle Project in Japan.

Improving system operations, management, and performance through the use of ITS technologies has the potential to reduce congestion without major capital investments.

Table 1.1: Annual Total Costs of Congestion by City (millions of 2002 \$).

Urban area	At 50% threshold	At 60% threshold	At 70% threshold
Quebec City	\$37.5	\$52.3	\$68.4
Montreal	\$701.9	\$854.0	\$986.9
Ottawa-Gatineau	\$39.6	\$61.5	\$88.6
Toronto	\$889.6	\$1,267.3	\$1,631.7
Hamilton	\$6.6	\$11.3	\$16.9
Winnipeg	\$48.4	\$77.2	\$104.0
Calgary	\$94.6	\$112.4	\$121.4
Edmonton	\$49.4	\$62.1	\$74.1
Vancouver	\$402.8	\$516.8	\$628.7
Total	\$2,270.2	\$3,015.0	\$3,720.6

ITS technologies range in complexity from ramp meters, which are small traffic light-like devices that control the traffic flow on ramps, to fully integrated systems in which several technologies work together to process information and respond to traffic conditions. Fig. 1.2² depicts some of ITS applications that can be employed and integrated to improve transportation system management.

Although the goals of the recent activities in ITS are quite broad and include increased traffic throughput, less accidents, reduced fuel consumption and better driving experience; less research has been done on developing appropriate technologies that allow cars to sense and intelligently affect the traffic flows that could result in more efficient use of highways. With that in mind, we address several issues in ITS with the objective to enhance the efficient use of the highway through the development of a system based on the coordination of vehicles.

1.2 Thesis Objectives

The overall objective of this thesis is to improve highway congestion through the development of a system based on dynamic collaborative driving. In order to assess the applications of ITS in different domains, this thesis has the following objectives:

The first objective of this thesis is to develop a localization system that determines

²Image from [1]

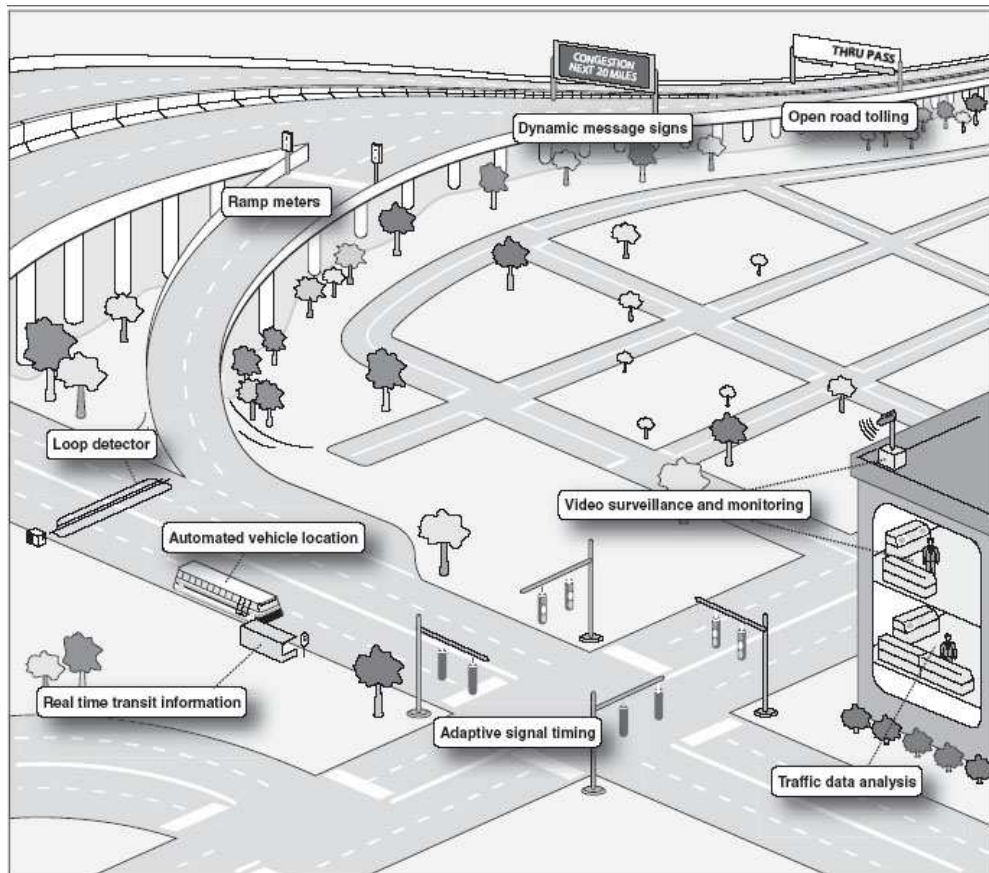


Figure 1.2: Examples of ITS technologies that can be deployed and integrated in urban areas.

lane positions of cars. A lane determination system is a necessary condition for the lane assignment system to work since the cars must be able to know which lanes they are occupying. Our fundamental assumption is that the cars are equipped with GPS receivers and processors. Each car has an ability to communicate with other cars within a certain radius (communication range) and be able to send its information to other cars via an ad-hoc network. The highway environment is chosen in this investigation to address the problem where traffic congestion and highway safety are among the most pressing problems. The results from this work show that lane positions for cars can be effectively estimated using only low-cost GPS receivers and a simple localization algorithm.

Minimizing the cost of sensors is beneficial because if the system becomes commercially available, this low-cost GPS sensor would help reduce the total cost of the products, and would reduce maintenance cost in case a sensor needs to be replaced. The drawback of minimizing the cost of sensors is a reduction of information available. Sensor aliasing may

also become more noticeable. Therefore, in general it becomes harder to localize. Vehicles that have been successfully employed in urban highways in the past rely on an array of different sensors to perceive the environment, such as the one presented in [3] which used multiple cameras, a laser scanner, proximity sensors, and GPS. Hence, an additional challenge is to achieve localization with less resources.

The second major objective of this thesis is to deal with lane routing for vehicles with an aim to enhance traffic system operation, reduce travel time, and improve traffic quality. The goal is to provide a distributed control strategy for routing cars to allocate lanes while satisfying practical constraints. Proposed is an algorithm that solves a minimization problem in order to allocate lanes to which cars should be sent in order to balance lane traffic flows and decrease the vehicle travel time. A proposed cost function to optimize is also discussed. Lane selection behaviors among cars could be coordinated to achieve greater traffic throughput. A challenge to this problem lies in the development of a cooperative driving strategy for traffic management. This research explores the problem of optimization of lane selection. A car's lane selections should consider not only the maintenance of its own performance, but also how the selection will affect the performance of other cars as well as the overall traffic flow of the highway system.

The end objective is to develop a strategy for optimal vehicle platoon formation, where vehicles travel in groups and are closely spaced. The platoon mode of operation was considered in the research community as a way of abbreviating the limitation of capacity and reducing fuel consumption that can be achieved by road vehicles. To maximize benefits, it is desirable to form platoons that are reasonably large (five or more vehicles), and it is also desirable to ensure that platoons remain intact for considerable distances [4].

1.3 Background to this Thesis

This thesis is divided into three main sections: lane position determination, lane assignment optimization for single vehicles, and cooperative vehicle platoon formation.

The Automobile for 21st Century (AUTO21), a research incentive initiated by the government of Canada to bring education institutes and industries together for the development of new automotive technologies, provided funding for this research. The first part of the thesis (Chapter 4) was carried out with an aim to develop a system for determining lane positions of cars on highways based on the transmitting and relaying of vital traffic information between vehicles. To expand on this idea, position localization and a communication network between vehicles will allow warning systems and driver aids to be implemented to improve driver awareness. In addition, traffic flows can improve as drivers

receive forecasts of road and traffic conditions ahead. This technology will be applicable to intelligent vehicles that are either semi-autonomous (*i.e.*, human driven vehicles that can make advisories) or fully autonomous capable of dynamic collaborative driving. Overall, this technology will be beneficial not only to the Canadian society, but to all drivers across the world.

The second and third parts of this thesis (Chapters 5 and 6) had their origins in the results of Chapter 4 and the general interests of the author and the project supervisors. It was found that through the reduction of unnecessary lane changes on the highway [5], the total traffic time for all cars can be minimized. This approach, however, assumed that cars can access information of all other cars on the highway. A distributed control strategy for individual cars was not presented. This thesis aims to provide a distributed control strategy to make the system possible using inter-vehicle communication.

A layout of this thesis is shown in Fig. 1.3.

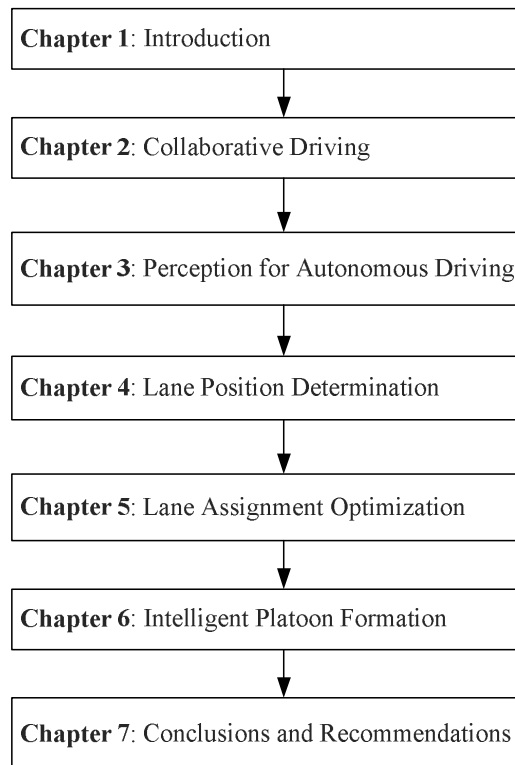


Figure 1.3: Flow-chart of thesis layout.

1.4 Contributions

In developing this approach to intelligent transportation systems, several research contributions were made that are summarized below:

1.4.1 Lane Positioning

1. Developed a strategy that allows for the determination of lane positions of cars on the highway using inter-vehicle communication. Our fundamental assumption is that the cars are equipped with GPS receivers and processors. Each car has an ability to communicate with other cars within a certain radius (communication range) and be able to send its information to other cars via an ad-hoc network.
2. Introduced a filtering strategy based on the fusion of particle and Butterworth filters that can effectively reduce GPS receiver noise. GPS receiver noise, which is dependent on the antenna design, the method used for the analogue to digital conversion, the correlation process, etc., is not common to all GPS receivers and can affect the lane estimation results by introducing a high-pitched noise to the GPS measurements.
3. Proposed an architecture to manage the communication and information sharing between vehicles across wireless ad-hoc networks. A multi-threaded software module was written in C++ to process GPS signals from satellites, convert GPS latitudes and longitudes into cartesian coordinates, eliminate GPS receiver error, and handle the sending and receiving of information between vehicles.
4. Demonstrated, through simulations in VISSIM [6], the lane positioning strategy. The inter-vehicle communication (IVC) simulator was designed on top of the existing commercial microscopic traffic simulator VISSIM. The IVC simulator module was programmed in C++. It worked in parallel with the VISSIM simulator engine by creating multiple threads for handling various tasks such as inter-vehicle communication and GPS coordinate reading.
5. Demonstrated, on hardware, the lane positioning system using low-cost GPS receiver units. Real road tests and simulations demonstrated that, in most situations, the lane positioning system can give correct lane positions more than 95% of the time, even when the vehicles were oscillatory weaving within their lanes (although this is unlikely to happen on real highways due to safety concern). The only data needed for lane estimation was the most recent GPS fixes, *i.e.*, the latitude and longitude (4 bytes of data in total) of the GPS measurements specifically, of the cars.

1.4.2 Lane Assignment

1. Proposed a decentralized control strategy for collaboratively sending cars to appropriate lanes with an aim to reduce vehicle traffic time.
2. Introduced a method for traffic flow estimation using inter-vehicle communication.
3. Demonstrated, through simulations in VISSIM the lane assignment algorithm. Simulation results showed that the strategy is viable and traffic throughput can be effectively increased through the coordination of cars.

1.4.3 Platoon Assignment

1. Developed a method for grouping cars into platoons to increase highway capacity. The optimization problem was solved by minimizing an objective function while ensuring that platoons remain intact for maximal distances.
2. Demonstrated, through simulations in VISSIM the platoon formation strategy. Simulation results demonstrated that lane capacity can be increased effectively when platooning operation is used.

Chapter 2

Collaborative Driving

2.1 Introduction

Collaborative driving has emerged as an important component of ITS that has the potential to significantly improve road safety and reduce human casualties by avoiding highway accidents. The concept of collaborative driving is based on the exchange of information between vehicles instrumented to perceive their surroundings and collaborating in dynamically formed groups. These vehicle groups, or ad-hoc networks, can form a collective driving strategy, which would require little if any operator intervention. The vehicles in these groups usually follow the same route while linked with each other through knowledge of relative positions and velocities and inter-vehicle communication. It is assumed in this work that no modification is made to the roadway infrastructure, and that the intelligence of the system is distributed over all of the vehicles in the automated highway system (AHS).

In order to create an efficient system that will meet the required objectives, a collaboration architecture must be developed that controls the interaction between vehicles (with the aid of a communication system) and between vehicles and their environment. The idea is the use of sensors and a wireless ad-hoc network to successfully replace human drivers or minimize human operator intervention.

In this work, inter-vehicle communication through an ad-hoc networking architecture are studied in highway traffic conditions. Collaborative driving strategies based on information exchange will be either tested on real experiments or simulated in a Virtual Reality (VR) environment.

2.2 Inter-vehicle Communication

2.2.1 Overview

The goal of inter-vehicle communication is to provide drivers with the ability to see further down the road and know if a collision has occurred, or if they are approaching a traffic jam. This can become possible if drivers and vehicles communicate with each other and with road-side base stations. If traffic information was provided to drivers, police, and other authorities, the road could be safer and traveling on them would become more efficient.

The communication can be made through a network of several vehicles that have communication devices. These vehicles would form a mobile ad-hoc network, and could pass information about road conditions, accidents, congestions, and vehicle states as illustrated in Fig. 2.1¹. A driver could be made aware of the emergency braking of a preceding vehicle, or the congestion due to an obstacle in the roadway. Such a network could also help vehicles determine their lane positions on a highway collaboratively as shown in Chapter 4 or help vehicle platoons utilize the roadway efficiently (Chapter 6). It can also help vehicles negotiate critical points like crossing intersections without traffic lights and entries to highways.

2.2.2 Ad-hoc Networks for Inter-vehicle Communication

An inter-vehicle communication network is a type of mobile ad-hoc network (MANET) in which high-speed vehicles send, receive and forward packets among other vehicles on the highways. A MANET is a self-configuring network of mobile routers (and associated hosts) connected by wireless links, the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily. A MANET may operate in a stand-alone fashion, or may be connected to the larger Internet.

To support inter-vehicle communication, Intelligent Vehicular Ad-hoc Networks (InVANETs) were developed to provide communications among nearby vehicles and between vehicles and roadside equipment. InVANETs are a form of MANET and use WiFi IEEE 802.11 and WiMAX IEEE 802.16 standards for easy and effective communication between vehicles with dynamic mobility.

More recent designs of InVANETs refer to the latest issues of IEEE 802.11p standard [7, 8] (also referred to as Wireless Access for the Vehicular Environment (WAVE)) which could

¹Image from <http://www.comnets.rwth-aachen.de>

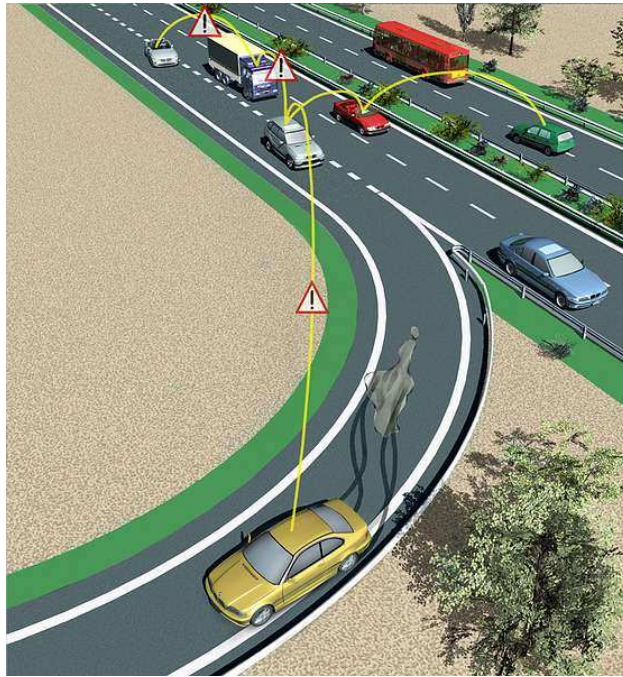


Figure 2.1: Inter-vehicle communication.

guarantee a maximum communication range of up to 1000 m under optimal conditions, or around 300 m for cars traveling at 200 km/hr .

Moreover, a new family of standards, referred to as IEEE 1609 suite [9], specifically for inter-vehicle communication built on IEEE 802.11 chipset is now under development. Three of the standards (IEEE Std. 1609.1 for Resource Management, IEEE Std. 1609.2 for Security Services for Applications and Management Messages, and IEEE Std. 1609.4 for Multi-Channel Operation) in the suite have been approved for trial use, and one (IEEE Std. 1609.3 for Networking Services) is pending. Once widely adopted, these standards will ensure that cars will have a communication range of 300-500 m on highways. Emergency vehicles will be equipped with longer-ranged (1- km) WAVE systems. For our lane positioning system, a conservative communication range estimate of 200 m is used for simulations and experiments, demonstrating our system is realizable in future highway systems.

Current research in the field of InVANETs are focusing on topology related problems such as range optimization, routing mechanisms, or address systems, as well as security issues such as traceability or encryption. In addition, there are also research interests such as the effects of directional antennas for InVANETs and minimal power consumption for sensor networks.

2.2.3 Routing for Vehicular Networks

Position-based routing (PBR) has appeared as a promising candidate for communication in vehicular ad-hoc networks (VANETs). PBR is a routing principle that relies on geographic position information. The idea of using position information for routing was first proposed in the 1980s in [10, 11]. Approaches to PBR include single-path [12, 13, 14], multi-path [15, 16, 17, 18, 19, 20, 21] and flooding-based [22, 23, 24, 25, 26] strategies. A survey on routing strategies for PBR can be found in [27, 28].

PBR routing in VANETs is built on top of a number of assumptions: (1) nodes can determine their own positions, (2) nodes can determine location of their neighbors, and (3) nodes can determine the position of the destination.

The first assumption is the most important one. This is a reasonable assumption since GPS receivers can be installed easily in vehicles to provide knowledge of real-time positions of nodes if they are outdoors. Alternatively, nodes can also use techniques based on signal strength information, available in IEEE 802.11 technology. However, as localization using GPS poses some undesired problems, several recent trends even fused a number of localization techniques such as dead reckoning, cellular localization, and video localization to overcome the limitations of GPS as shown in Fig. 2.2². The second assumption implies that nodes exchange small packets between neighboring nodes to make their positions available to others.

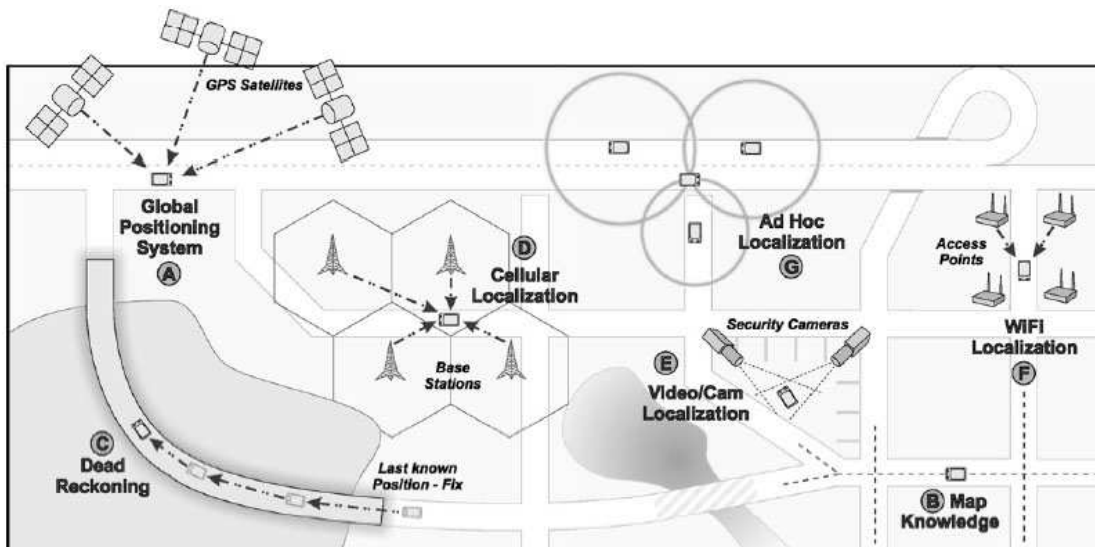


Figure 2.2: Several localization techniques for VANETs.

²Image from [29]

In our system, the vehicles are assumed to have access to information of their own state and can perceive states of surrounding vehicles, including current positions, driving speeds, etc., using a wireless ad-hoc network. The vehicles are equipped with standard GPS units which has the accuracy of 15 *m* (global average, 95%) and 36 *m* (worst site, 95%). The locations of vehicles are updated in real-time with 5 *Hz* update rate. To improve the position accuracy of the GPS units, a localization technique based on Markovian algorithm and particle filtering using relative measurements between vehicles was deployed. The communication between the notebooks/vehicles was then made through the IEEE 802.11a/11g standard D-Link DWL-AG660 108*Mbps* Wireless Cardbus Adapters which is equivalent to InVANET networks. These tri-mode, dualband adapters were inserted into three laptop PC's to allow them to wirelessly connect to each other. The main technical specifications for DWL-AG660 are shown in Table 2.1.



Figure 2.3: D-Link DWL-AG660 802.11a/11g 108*Mbps* Wireless PCMCIA Adapter.

2.3 Collaborative Driving Simulator

For simulation implementation, the VISSIM software package [6] was used. VISSIM is a microscopic, time-step, and behavior-based simulator that is developed to analyze the full range of functionally classified roadways. It is capable of modeling traffic with various control measures in a 3-D environment. VISSIM lets us communicate and control the behaviors of vehicles through a dynamic link library (DLL) file compiled from C/C++ code. Vehicle parameters from the external driver model DLL output function are stored within member variables of a designated vehicle class object.

The collaborative driving simulator was designed on top of the existing commercial microscopic traffic simulator VISSIM. The primary reason of selecting an existing software is to make use of the already established driver model which controls car following and lane

Table 2.1: D-Link DWL-AG660 technical specifications.

Specification	Description
Weight	54 <i>g</i>
Dimension	116 <i>mm</i> ×53 <i>mm</i> ×8 <i>mm</i>
Standards	802.11a/11g
Distance	Indoor: up to 100 <i>m</i> ; Outdoor: up to 300 <i>m</i>
Interface	32bit Cardbus
Maximum data rates	108 <i>Mbps</i>
Frequency band	2.4 <i>GHz</i> to 2.5 <i>GHz</i> ; 5.150 <i>GHz</i> to 5.850 <i>GHz</i>
Transmitted Power	15 <i>dBm</i> to 2 <i>dB</i>
Antenna	Dual antenna diversity switching
Temperature	0 °C to 55 °C
Features	Enhanced security features

changing. VISSIM also provides a mean of constructing the infrastructure of any traffic network. The two components listed above are essential to any traffic modeling simulators [30]. Other useful features include the ability to implement vehicle models that govern the dynamic characteristics (such as the engine power curve) for default or user defined vehicle types.

VISSIM has an application programming interface (API) but the control over vehicles within the simulator using this method is limited and therefore this method was not employed. Instead, vehicle parameter are accessed through a dynamic link library (DLL) that is used by VISSIM when it refers to an external driver model. This dynamic link library includes three functions which handles reading of vehicle parameters, writing of vehicle parameters, and the addition and deletion of vehicles as well as initialization configurations. Adjustable vehicle parameters include desired vehicle velocity, acceleration, and lane position.

In a sense the collaborative driving simulator integrates itself into VISSIM's external driver model. The collaborative driving simulator is essentially a controller that reads the relevant parameters for all vehicles within the simulation and make the appropriate outputs to control vehicle behavior as depicted in Fig. 2.4. It is also possible to allow VISSIM's internal driver model and external driver model to concurrently control different parameters by allowing vehicle parameters read from the output function to directly loop back through the vehicle parameters input function. For instance, the collaborative driving

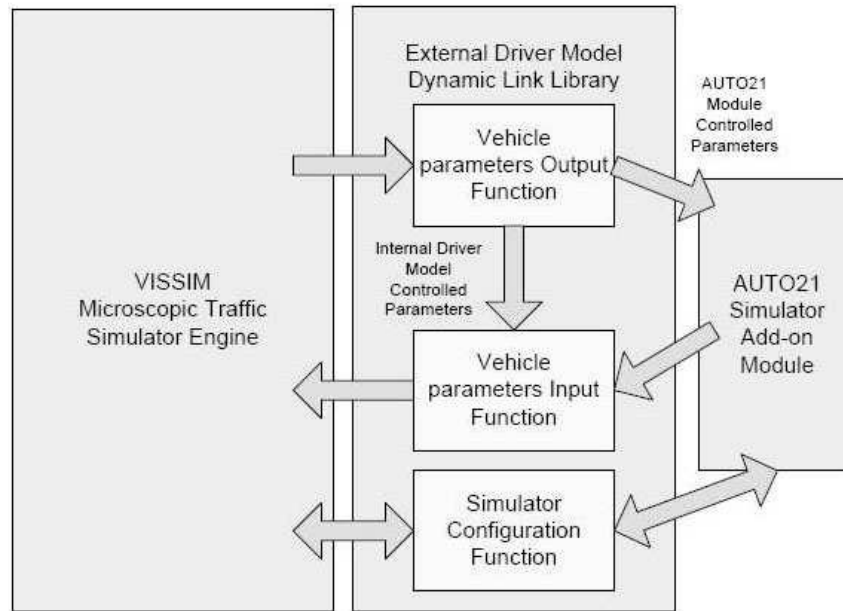


Figure 2.4: Accessing VISSIM vehicle parameters [31].

simulator add-on module can indicate a desired velocity but the acceleration of the vehicle to the desired velocity will be controlled by VISSIM's internal driver model which takes into account the presence of other neighboring vehicles to avoid collisions. Another example of internal driver model control is lane changing behavior. By allowing VISSIM control over this, overtaking of vehicles does not have to be handled by the collaborative driving simulator add-on unless it is desired to do so.

The collaborative driving simulator module is programmed in C++. It works in parallel with the VISSIM simulator engine by creating multiple threads for handling various tasks such as inter-vehicle communication and GPS coordinate reading. Multi-threading allows different parts of the program to run concurrently which is desirable as vehicle parameters and other variables can be updated without waiting for a simulation time step or iteration within the VISSIM simulator engine. One benefit of this approach is that vehicles exchanging information will always have the latest vehicle states as VISSIM continually updates vehicle parameters. All threads are created when a simulation starts and destroyed when the simulation is stopped.

Vehicle parameters from the external driver model DLL output function are stored within member variables of a designated vehicle class object. Instances of this class are created at the start of a simulation run and are stored in elements within an object derived from the standard template library (STL) vector class. The vehicle parameter storage class also contains member functions which are used to modify the received vehicle param-

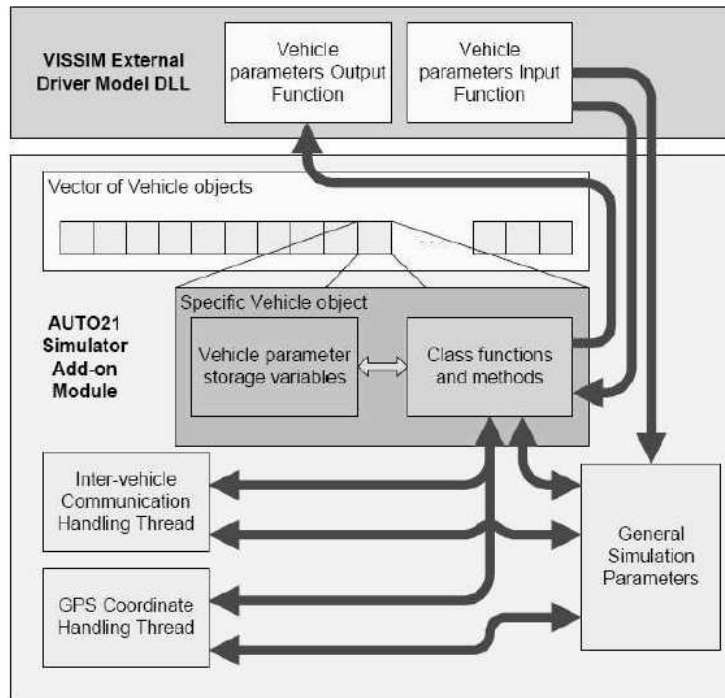


Figure 2.5: Software architecture for collaborative driving simulator module [31].

eters for output to the VISSIM simulator engine. Other parameters related to the general simulation environment such as simulation time and the identification of the vehicle currently being updated by VISSIM are stored and constantly updated as well. All vehicle parameters are reset at the start of a simulation run. Fig. 2.5 illustrates an overview of the software architecture for the collaborative driving simulator module.

More details about the collaborative driving simulator used in this work can be found in [31].

Chapter 3

Perception for Autonomous Driving

3.1 Introduction

Sensors are vital to an autonomous vehicle to allow it to perceive its operating environment in relation to its own position, whether the vehicle is trying to localize itself in order to navigate, create a map of the surrounding, or simply avoid obstacles such as buildings, cars, pedestrians, etc., in its path of travel. This thesis investigates the ability to perform autonomous localization using a GPS as the only available sensor.

The technical capabilities and safety benefits for ITS have been greatly enhanced by state-of-the-art sensor technologies. Sensing systems for ITS can be either infrastructure based, vehicle based, or some combination. Infrastructure sensors are devices that are installed on the road, or surrounding the road (*e.g.*, buildings, posts, and signs). These sensing technologies may be installed during preventive road construction maintenance or by sensor injection machinery for rapid deployment of road in-ground sensors.

Among a wide range of sensors used in ITS, cameras, laser rangefinders, and GPS are the most popular solutions for vehicle navigation. A brief overview of these three types of sensors will be discussed in the following sections.

3.2 Cameras

Human beings gather a vast majority of their sensory input visually. Much effort has been spent in enhancing visual acuity in humans. Lasik, eye-glasses, binoculars are examples of such effort which improve our quality of life by enhancing our experience of the world around us. Computer vision is a natural and logical extension of this effort. It is a

branch of artificial intelligence (AI) and image-processing technology that processes video content and identifies objects in the video stream. To date, computer vision has produced important applications in fields such as industrial automation, robotics, biomedicine, and satellite observation of Earth. In the field of ITS, its wide range of applications includes road detection [32] (see Fig. 3.2); surveillance systems for parking assistance; (2) landmark detection to assist the car in following the road; (3) traffic sign detection and recognition for route planning and alerting the driver; (4) obstacle detection, especially detecting the presence of pedestrians in a driver's blind spot [33] (see Fig. 3.3); and (5) driver condition monitoring for intelligent airbag deployment or driver distraction level monitoring [34, 35, 36, 37] (see Fig. 3.1).

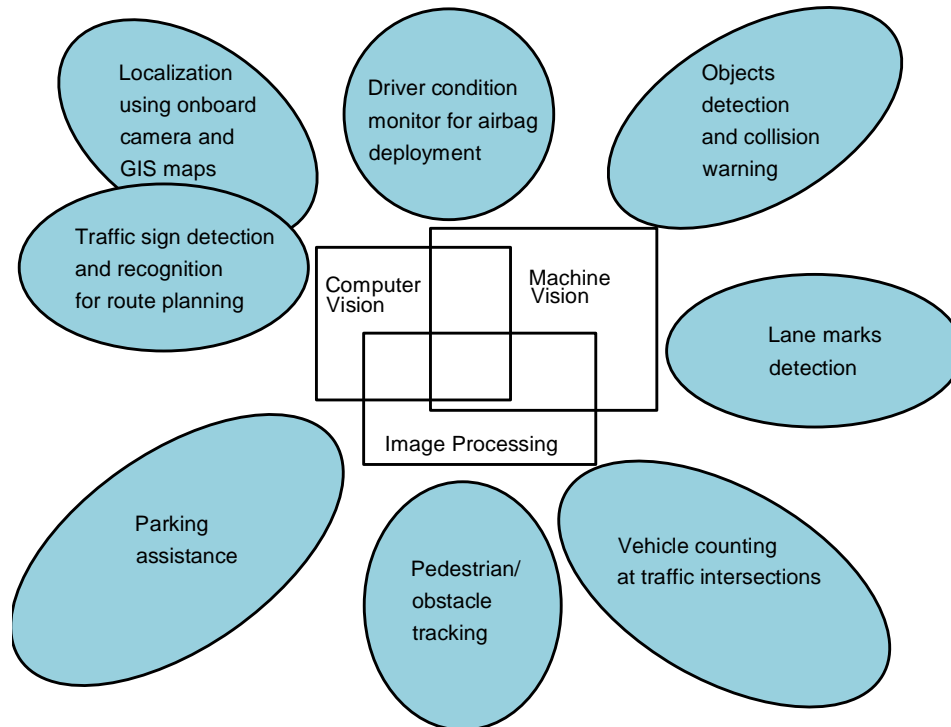


Figure 3.1: Some applications of computer vision in ITS.

With the development of new technologies and computer capabilities, the current computer vision enables the integration of views from many cameras into a single, consistent ‘super-image’. Such an image automatically detects scenes with people and/or vehicles or other targets of interest, classifies them in categories such as people, cars, bicycles, or buses, extracts their trajectories, recognizes limb and arm positions, and provides some form of behavior analysis. The benefit of vision is that it is a passive system and theoretically has infinite range as a camera or an eye receives light emitted from any source in

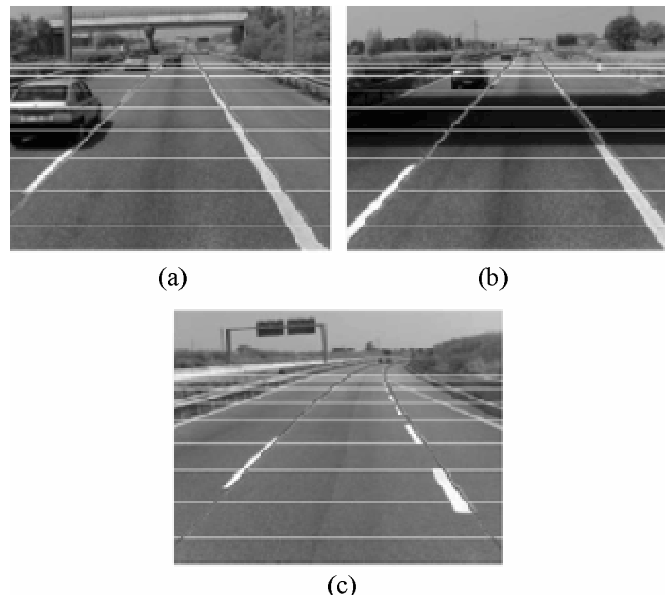


Figure 3.2: Road-detection result for highways and the 3-D projected model with different conditions.

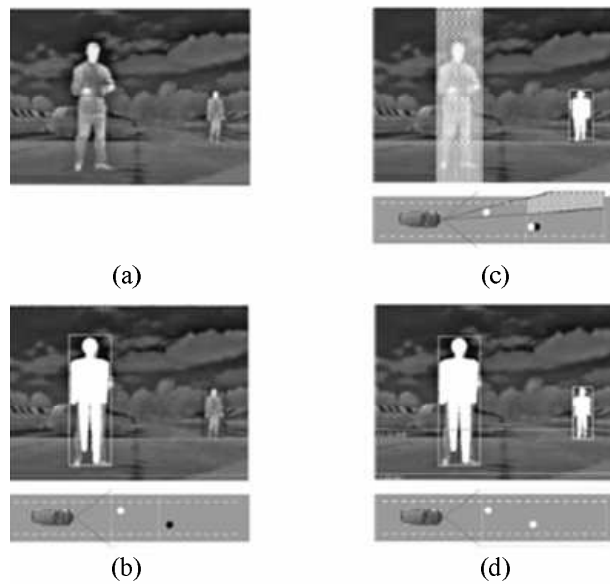


Figure 3.3: Obstacle detection example: fusion of the results from two resolutions. (a) Input image, (b) results of low-resolution processing, (c) results of original resolution processing, and (d) final fused results.

the three dimensional world and captures that information on a two dimensional image. Theoretically, this means a camera has longer range than most of other types of sensors.

Providing the equipment to give vision to a machine is not difficult. There are many charge coupled device cameras or CMOS (Complementary Metal-Oxide Semiconductor) cameras available on the market at low cost and they can easily be interfaced with a computer. The difficulty is interpreting the data that is retrieved from a camera frame. A machine must interpret this raw data, which is an array of numbers representing light intensity at each pixel.

3.3 Laser Rangefinders

Scanning laser rangefinders provide a relatively new and exciting high-resolution sensors for robotics and autonomous vehicles navigation. Common in high-end robotics for many years, these sensors are becoming more common on relatively inexpensive robotics applications due to the rich, high-resolution, and high-frequency data they generate.

The method a laser rangefinder uses to determine ranges can be based on time of flight measurement, phase-shift measurement, or optical triangulation [38]. The time of flight measurement calculates the distance to an object by using the speed of light propagation and the time it took for a laser beam to be reflected back to the sensor. For laser rangefinders that use phase-shift measurements, the phase of the transmitted signal is compared to the reflected signal to determine the distance traveled. However, the maximum range is limited by the wave length of the emitted signal. In optical triangulation sensors, the reflected light beam emitted by the sensor is focused by a lens. The position of this focus point will vary depending on the distance to the target object from which the light is reflected.

Laser range finders can provide data with pencil beam viewing at high data rates with roughly millimeter resolution. Power consumption varies but tends to be somewhat greater than smaller and less sophisticated sensors like infrared (IR) rangefinders, sonars, etc. The lasers tend to be self-contained in professional and sturdy enclosures for external mounting on mobile robots and other applications.

Lasers are often specified in resolution, frequency, and angular resolution. Resolution refers to how accurate the distance measurement is in a given direction and is typically around 1-3 *mm*. Frequency refers to how often the full 360° arc is swept by the scanning sensor and is typically 10 *Hz* or even more. Angular resolution refers to how many samples are taken with each full 360° scan of the sensor. This value is often over 500 steps of



Figure 3.4: Hokuyo URG-04LX laser rangefinder.

resolution meaning sub-degree wedges of the full circle are measured with each sample. Rangefinders typically also have a minimum and maximum range where they can measure. Minimums are often a few centimeters and range can be over a meter up to tens of meters and more.

Lasers are made by several manufacturers with some of the most common in robotics being SICK and Hokuyo. These lasers vary in range, performance, power consumption, and cost. The technical specifications of one of the typical laser scanners used in autonomous robots, the Hokuyo URG-04LX laser rangefinder, are shown in Fig. 3.4 and Table 3.1.

Lasers have drawbacks as well. One of the shortcomings of laser rangefinders is that they are relatively expensive compared to other types of sensors such as cameras and standard GPS units. A typical scanning laser rangefinder for autonomous robotics can cost between \$1500-\$6000.

3.4 Global Positioning Systems

The Global Positioning System (GPS) is a location system based on a constellation of about 24 satellites orbiting the earth at altitudes of approximately 20,000 *km*. GPS was developed by the United States Department of Defense (DOD), for its tremendous application as a military locating utility. The GPS developed by DOD, however, is not the only one in existence. Russia also has a satellite positioning system known as the global navigation satellite system (GLONASS). The European Union is also launching a satellite system called Galileo, which will be functional in the near future. The systems identified above are able to provide global coverage. Some countries are launching their own satellites to

Table 3.1: Hokuyo URG-04LX technical specifications.

Specification	Description
Weight	Approx. 160 <i>g</i>
Material	Polycarbonate
Light source	Semiconductor laser $\lambda = 785 \text{ mm}$, laser safety class 1 (IEC60825-1)
Power source	5 VDC, $\pm 5\%$
Current consumption	500 <i>mA</i> or less
Detectable distance and objects	20 <i>mm</i> to 4000 <i>mm</i> , white Kent sheet $70 \times 70 \text{ mm}$
Accuracy	Official 20 to 1000 <i>mm</i> : $\pm 10 \text{ mm}$, 1000 to 4000 <i>mm</i> : $\pm 1\%$ of measurement
Resolution	1 <i>mm</i>
Scanning angle	240°
Angle resolution	Approx. 0.36°
Scanning time	100 <i>ms</i> /scan

provide regional coverage. While there are military incentives in the installation of these satellite positioning systems, they have also benefited civilian applications, one of which is positioning for land vehicles. The remainder of this section will focus on GPS usage and performance.

GPS is based on satellite ranging to calculate the distances between the receiver and the position of 3 or more satellites (4 or more if elevation is desired). Assuming the positions of the satellites are known, the location of the receiver can be calculated by determining the distance from each of the satellites to the receiver. GPS takes these 3 or more known references and measured distances and trilaterates an additional position [39, 40, 41, 42] as shown in Fig. 3.5.

GPS signals are distorted by many factors, which manifest as psuedorange (the perceived satellite to receiver distance) errors [43]. The proper name for this error is user-equivalent range error (UERE), which can be considered the statistical sum of all error contributions associated with a particular satellite. To overcome this problem, a GPS receiver will lock on to the signals from more than four satellites to reduce the error in the calculated position using a least squares approach [41]. Commercially available GPS receiver units are usually capable of tracking up to twelve GPS satellites simultaneously and

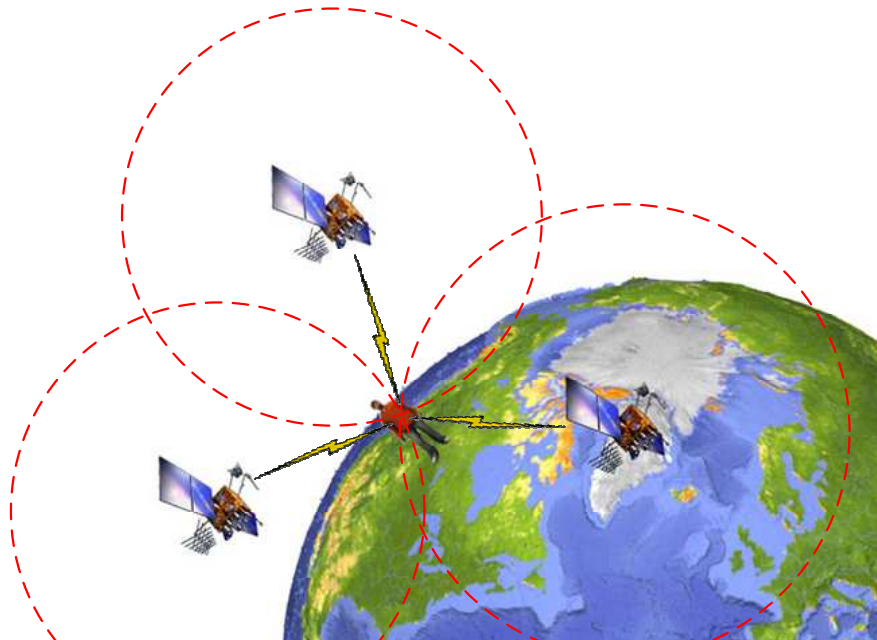


Figure 3.5: The functioning of a GPS.

will not provide a solution unless it is able to track at least four satellites. The solution to a position fix is given in geodetic coordinates (longitudes, latitudes, and elevation) according to the world geodetic system (WGS-84). A closed form solution exists to convert the geodetic coordinates to Cartesian coordinates in an Earth Centered Earth Fixed (ECEF) reference frame. For more information on this matter, refer to [39, 40].

There are many sources of error for a GPS solution, and the performance of a receiver depends on its pseudorange measurement quality. The model is used to compensate for certain effects, and the accuracy of the satellite ephemeris (trajectory) data which the receiver obtains. In general, GPS error is mathematically expressed as a product between geometric error (caused by the relative location of satellites and the receiver) and the pseudorange error.

Selective availability (SA) was a major source of pseudorange error. This feature involves intentionally inducing error in satellite clock and trajectory data that is broadcasted to receivers, and can cause errors of up to 70 *m* that oscillates every 4 to 12 minutes. This feature is currently disabled to allow improved SPS (Standard Positioning Service) accuracy, but when activated, its spatially correlated effect can also be overcome by differential-GPS services [42].

Multi-path is another major source of pseudorange error, and occurs when satellite signals reflect off objects in an environment. These objects may be buildings in an urban



Figure 3.6: Garmin-18 GPS receivers.



Figure 3.7: LocSense 40-CM GPS.

environment, or trees in a forest. This is a major concern for using GPS for localization in urban environments. Not only do the reflections cause timing errors, they also create multiple paths to the receiver and at times may cause a receiver to produce inaccurate position estimates. Multi-path effects are very location dependent but can cause positioning errors of up to 150 *m* for SPS and 15 *m* for PPS (Precise Positioning Service).

Other sources of pseudorange error include atmospheric errors that can delay satellite signals. The ionospheric (higher part of atmosphere 70 to 1000 *km* above the Earth's surface) effect is signal frequency dependent and arises from the interaction between solar activities and the Earth's geomagnetic field. This is very difficult to model and account for but there are models which are able to remove about 50% of ionospheric effects. Typically, signals from satellites that are low on the horizon with respect to a receiver are more prone to this type of error, which can vary from a few to 20 meters in a day. The tropospheric (lower atmosphere) error is a function of the tropospheric refractive index, which depends

Table 3.2: Garmin-18 technical specifications.

Specification	Description
Dimension	61 <i>mm</i> in diameter and 19.5 <i>mm</i> in height
Weight	(5 <i>m</i> cable) 161.6 <i>g</i>
Receiver	WAAS enabled, 12 parallel channel GPS receiver continuously tracks and uses up to 12 satellites to compute and update GPS fixes
Acquisition time	Reacquisition: less than 2 seconds (all data known) Warm: Approx 15 seconds Cold: Approx 45 seconds (ephemeris unknown)
Update rate	5 <i>Hz</i>
Accuracy	Position: < 15 <i>m</i> , 95% typical Velocity: 0.1 knot RMS steady state

on the temperature, humidity, and pressure at the receiver location. Most of this signal delaying effect can be accounted for and will only cause a few meters of pseudorange error, but the error can be as high as 20 *m* if the effect is uncompensated. Aside from atmospheric effects, other physical influences include relativistic effects due to the satellites' high speed of orbit, clock errors on satellites and receivers, and receiver noise. For further details and additional sources of GPS error, refer to [39, 40, 42].

Overall, the GPS available for civilian use is convenient for obtaining position measurements in any outdoor localization and navigation applications. With GPS, it is unnecessary to compare the position measurements to features on a map. All that is required is knowing the coordinate transformation between the global frame of reference used in GPS and the frame of reference used by the vehicle. However, the unpredictable behavior of GPS in urban settings due to multi-path makes it unreliable for autonomous navigation if it is the only available sensor. The procedures for effectively reducing GPS errors will be discussed in Chapter 4.

In this research, we employed the Garmin-18 GPS and LocSense 40-CM as shown in Fig. 3.6 and Fig. 3.7 for localization of vehicles. The main technical specifications are shown in Table 3.2 and Table 3.3, respectively. However, the Garmin-18's were used most of the time. The Garmin-18 tracks up to 12 satellites at a time while providing fast time-to-first-fix, precise navigation update (five times per second for Garmin-18-5Hz). The Garmin-18

Table 3.3: LocSense 40-CM technical specifications.

Specification	Description
Dimension	43mmL×42mmW×13mmH
Weight	24 g
Receiver	12 parallel channel, L1 C/A mode
Start-up time	Hot: Approx. 10 seconds Warm: Approx 35 seconds Cold: Approx 45 seconds
Update rate	1 Hz
Sensitivity	-137 dBm acquisition -145 dBm tracking
Accuracy	Position: < 5 m CEP Velocity: 0.1 m/second

used in this work interfaces to a serial port. The unit accepts TIA-232-F (RS-232) level inputs and transmits voltage levels that swing from ground to the positive supply voltage polarity. The cable contains wire for power, ground, receive, transmit, and measurement pulse output.

At the end of the Garmin GPS cable, the wires are terminated in a connector that can be removed and replaced with another connector such as a RS-232 serial connector and a PS/2 connector (for power) in our work.

Chapter 4

Lane Position Determination

4.1 Introduction

In this chapter, the development of a low-cost lane-level position determination is described. A large number of other transportation research applications would benefit from a lane-level positioning capability. For example, such a capability could be used for a lane keeping safety system in which an alarm will activate or autonomous driving will take over if the vehicle deviates from the center of the lane.

Another application for lane-level positioning is a lane assignment system, which advises the driver as to which lane should be chosen to reach the specified destination without requiring excessive last-minute lane changing. Last but not least, lane positioning system can be used in probe vehicles that measure lane-specific traffic conditions on a freeway [44].

In this chapter, a Markov-based approach which computes the lane position of transportation vehicles such as: cars, buses, etc, is introduced. The software architecture that enables the communication between vehicles using ad-hoc network is then described. Finally, simulation and real road test results are presented.

4.2 Problem Statement

The problem is to enable effective localization for multiple vehicles in a typical urban highway environment where infrastructure for highly accurate sensors such as differential GPS is not available.

In this problem, a group of n_v vehicles, each of which is located in a highway lane, must autonomously determine the lane positions they occupy. Each of them is equipped with a

standard GPS receiver, a processor, and a communication capability. The vehicles must also operate under the following conditions:

1. **Unknown environment:** prior knowledge of the vehicles' initial lane positions is unknown. The vehicles must also be able to recover from localization failures or GPS outage in which the positions of the vehicles are lost.
2. **Dynamic environment:** vehicles are moving in the environment.
3. **Limited communication:** the cars are equipped with limited communication ability. The cars can only communicate directly with other vehicles that are within a certain radius known as communication range. That is, a vehicle can communicate directly with any other vehicle that lies within the circle define by this radius, but cannot communicate directly with any vehicles outside this region. Because vehicles will move in and out of each other's communication range, they will only be able to communicate directly with one another for intermittent periods of time. Figure 4.1 illustrates the communication of cars in a specific situation. Cars 2, 3 and 4 are within communication range and they can inter-change information to one another while car 1 can only communicate with car 2. This means that the cars can only have a partial view of the overall traffic condition.
4. **Limited sensing:** vehicles are equipped with low-cost GPS receivers that can only provide their locations having an accuracy of 5 meters CEP (Circular Error Portable). This means that half of the data points fall within a circle of a 5-meter radius, half lie outside this circle. This accuracy is equivalent to 15 meters in 2dRMS, where 95% of the data points occur within a 15-meter radius.

4.3 Related Work

Much effort has been put in lane finding/positioning in the research community. Ieng *et al.* [45] dealt with the multi-lane detection by using multiple cameras. Pierre-Yves and Jeff [46] used lane-level navigation systems with a high level DGPS/DR sensor integration system and a map database. Their system was able to detect which lane the car is driving in and when a car is changing lanes. Another approach [47] integrated an Inertial Measurement Unit (IMU) with the GPS receiver to allow for accurate vehicle positioning. It also used Real-Time Kinematic DGPS (RTK-DGPS) receivers supported by DGPS base stations at the test site and a lane-level-detailed digital map. A variety of other research directions on lane detection and lane departure detection can also be found in [44, 48, 49, 50, 51, 52].



Figure 4.1: Inter-vehicle communication.

Most of these systems involved complicated image processing algorithms and/or costly equipments such as highly accurate sensors, high performance computers, etc. With that in mind, we approach the problem by using sharing information between cars to determine their lane positions. Inter-vehicle communication and co-operative driving systems have been in development for some time [53, 54, 55, 56]. With the availability of GPS systems, it is practical to locate a car within certain accuracy. However, GPS data are often off the road and do not provide the exact position of cars due to degradation or multi-path problem. Therefore, it is challenging to determine the exact lanes that cars are traveling in even when using the digital map of the road network.

4.4 Approach to Lane Positioning

As discussed in Section 3.4, an error from GPS generally comes from several sources, including satellite clock, ephemeris error, ionospheric effects, tropospheric effects, and the geometry of visible satellites. This set of errors from GPS will be common for vehicles. Other errors are local to the different receivers and include radio frequency noise from the environment, receiver noise and resolution, multipath, and receiver clock error. Aside from multipath, errors from this second set are generally smaller than those common to the

receivers. Hence, if multipath is not a problem (as is the case for many open highways), then receivers that are relatively close to one another will experience a similar GPS error. Although the absolute vehicle positions might have considerable errors, the relative position between vehicles should have a much lower error since many of the errors are common to the receivers. This is similar to what happens with differential-GPS, where the GPS solution is improved by removing the common errors. With this in mind, the problem is approached by using the relative distance between GPS measurements of vehicles to estimate their lane positions.

While GPS measurements can be very precise in some circumstances, they generally are not accurate enough in rural areas where infrastructure for DGPS is not available, making lane position determination difficult if using only GPS data from one vehicle. Even for future Geographic Information Systems (GIS) when information might be much improved to the point of determining where lanes are, it is still not clear to what extent the GIS information will be available. With an improved accuracy in mapping, there could be increased costs for access. Moreover, it is unclear how often this information will be updated to include changes in the roadway, including construction and traffic accidents, which divert traffic along different paths than recorded in the GIS database.

In this chapter, it will be shown that by sharing state estimates between vehicles that are traveling close to each other, the lane-level position for each vehicle can be determined. The architecture for the system is shown in Fig. 4.2. Each vehicle is equipped with a GPS receiver and a processor to implement the lane-positioning algorithm and to communicate with other vehicles. The GPS data for the vehicles can also be fed into a position filter to reject the measurement noise from receivers. In our simulations and experiments, a combination of a particle filter fused with a low-pass Butterworth filter (see Section 4.7.2) can satisfy this task.

4.5 Markov Localization

Markov localization addresses the problem of state estimation from sensor data. Instead of maintaining a single hypothesis as to which lane the vehicles might be in, Markov localization maintains a probability distribution over the space of all such hypotheses. A probabilistic representation allows it to weight these different hypotheses in a mathematically sound way.

The target of Markov localization within this work is to determine which vehicle is traveling in which lane using GPS data and inter-vehicle communication. To introduce the major concepts of Markov localization, let us begin with a simple case, followed by a

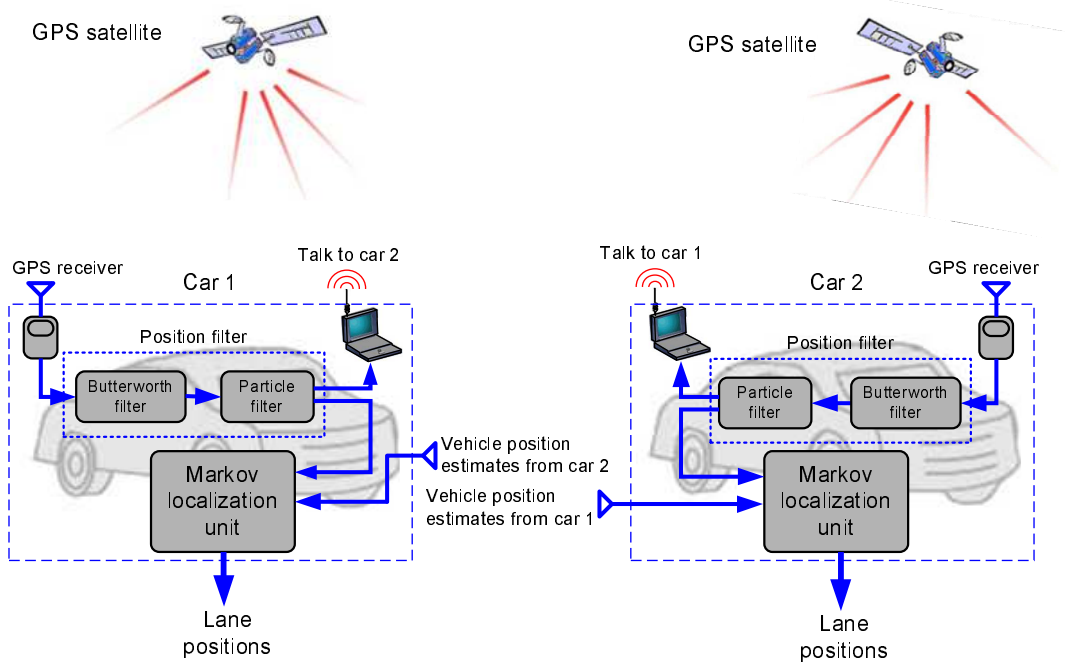


Figure 4.2: Lane positioning diagram.

mathematical description of the algorithm. The reader may notice that Markov localization is a special case of probabilistic state estimation [57, 58, 59].

Let us start with the simplest case: two vehicles traveling on a two-lane road. The two vehicles are assumed close enough to be able to communicate with each other. Within this Markov localization approach, call $P(v_{1,t} = l_a, v_{2,t} = l_b)$ the probability that car 1 is traveling in lane a and car 2 is in lane b at time t , where a and b are either 1 or 2. There are four possibilities in total, which are $P(v_1 = l_1, v_2 = l_1)$, $P(v_1 = l_1, v_2 = l_2)$, $P(v_1 = l_2, v_2 = l_1)$, $P(v_1 = l_2, v_2 = l_2)$. No initial estimates of these probabilities are required and they can simply be initialized with equal probability equal to 0.25. In the course of the vehicles' mission (*i.e.*, for $t > 0$), P is updated through two basic steps: (1) *prediction*, and (2) *correction*.

4.5.1 Prediction

In the prediction step, the state transition of a vehicle is modeled through the conditional probability $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$, which denotes the probability for a motion action that carries vehicle i from lane j to lane a (j and a can be equal). When the vehicle moves, $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$, which models the uncertainty in the vehicle's movement, is used to

compute the probability distribution at time t as

$$P(v_{i,t} = l_a) \leftarrow \sum_{j=1}^2 P(v_{i,t} = l_a | v_{i,t-1} = l_j) P(v_{i,t-1} = l_j), \quad (4.1)$$

where $P(v_i = l_a)$ is the probability that car i is traveling in lane a . This step is repeated for both vehicles and then uses the product to form the combined probabilities:

$$P(v_{1,t} = l_a, v_{2,t} = l_b) = P(v_{1,t} = l_a) P(v_{2,t} = l_b). \quad (4.2)$$

The conditional probabilities $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$ for a car are computed by comparing its current GPS position to the position at the last time step $t - 1$. For example, by calculating the shortest distance ΔD from the vehicle position estimate from the GPS reading at time t to the line passing through the two most recent position estimates, the probability at which vehicle 1 switches to lane a (see Fig. 4.3) can be computed. In this case by using zero as the mean value of the probability density function of a normal distribution $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\Delta D - 0)^2}{2\sigma^2}}$, if ΔD is close to zero, the probability at which car 1 still stays in the same lane, *i.e.*, car 1 does not switch lane, is close to 100%. In a similar fashion, one can develop all the conditional probabilities $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$ that cover all possible situations.

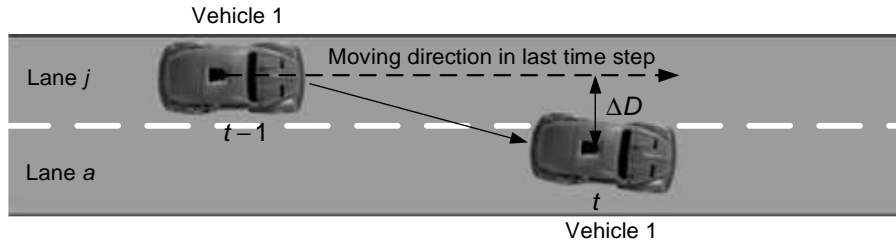


Figure 4.3: Prediction step: this figure illustrates a car switching lane and its position is unknown. The shortest distance ΔD from the GPS reading at time t to the line passing through the two most recent GPS readings is used to calculate the conditional probability.

It should be noted that a large ΔD can result from two possible situations: (1) the car is switching lanes or, (2) the car is simply following a curve in the road. For example Fig. 4.4 shows that the vehicle is still in lane 1, but a large value of ΔD due to the curve would falsely indicate that the vehicle is switching to lane 2.

To resolve this ambiguity, the lane-finding algorithm must estimate the radius of the lane curvature and compensate the drift caused by the curve. This can be done by using the least squares fitting algorithm based on successive vehicle positions. This algorithm allows us to fit an arc through the successive vehicle position estimates. When the vehicle changes lanes in the curved section of the road, its last position will not lie on the estimated arc, and

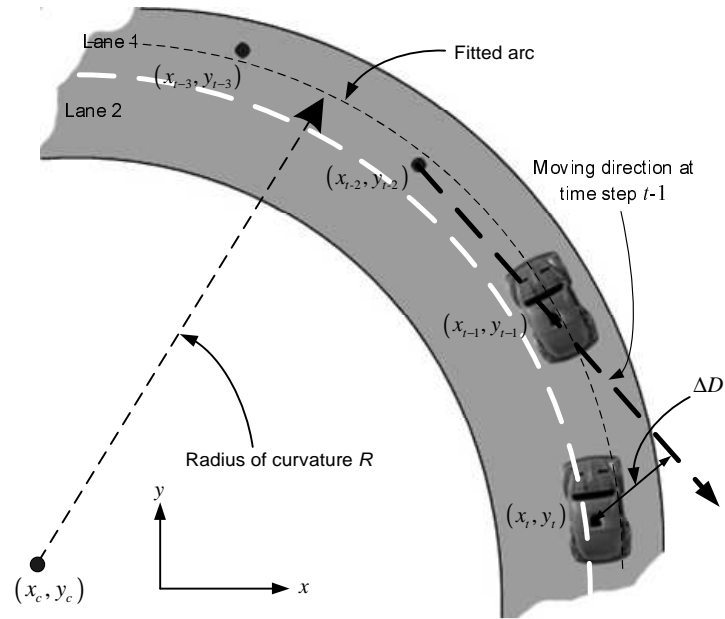


Figure 4.4: Radius of curvature estimation: this figure illustrates the GPS readings for a vehicle assumed to be traveling in lane 1. The position and moving direction of the vehicle are estimated from GPS data. The GPS measurements are not necessarily exactly in lane 1 and the actual vehicle does not necessarily overlay with the GPS measurements as the figure shows.

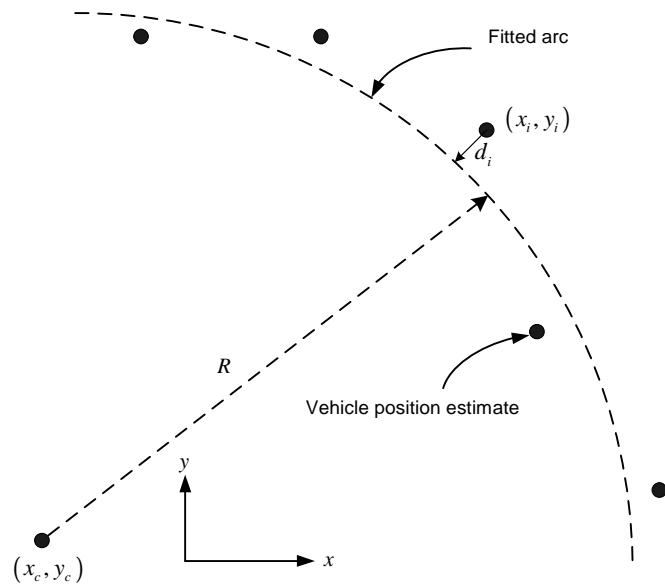


Figure 4.5: Least squares method for approximating road curvature.

the distance from the last position estimate to the arc will be used in the prediction step to predict if the vehicle is switching to the other lane. In our simulations and experiments, five successive vehicle position estimates were used to find the radius R and center (x_c, y_c) of the road curvature (see Fig. 4.5). The cost function to be minimized is

$$S = \sum_{i=t-4}^t d_i^2 = \sum_{i=t-4}^t (\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - R)^2, \quad (4.3)$$

where (x_i, y_i) are the coordinates of the vehicle position estimate number i (note that in 4.3, i takes on values from $t - 4$ to t). The following set of equations can be solved for R , x_c , and y_c using numerical method:

$$\begin{cases} \frac{\partial S}{\partial x_c} = 0; \\ \frac{\partial S}{\partial y_c} = 0; \\ \frac{\partial S}{\partial R} = 0. \end{cases} \quad (4.4)$$

4.5.2 Correction

Denote as z the GPS measurements that come in at time step t for both cars, and $P(z|v_1 = l_a, v_2 = l_b)$ as the probability of perceiving z when the two cars are in lanes a and b respectively. When the GPS measurements are taken into account, $P(z|v_1 = l_a, v_2 = l_b)$ is used to update the probability distribution at time t according to Bayes' rule

$$P(v_{1,t} = l_a, v_{2,t} = l_b | z_t) \leftarrow \frac{P(z_t | v_{1,t} = l_a, v_{2,t} = l_b) P(v_{1,t} = l_a, v_{2,t} = l_b)}{P(z_t)}, \quad (4.5)$$

where $P(z_t)$ has the purpose of normalizing the sum of all $P(v_{1,t} = l_a, v_{2,t} = l_b | z_t)$.

Suppose that at time t a new measurement z (see Fig. 4.6) which is the perpendicular distance from one vehicle to the other vehicles direction of motion in this case, is available. The conditional probability $P(z|v_1 = l_a, v_2 = l_b)$ can be calculated based on z using the probability density function $(1/\sigma\sqrt{2\pi})e^{-(z-\mu)^2/2\sigma^2}$. The mean $\mu = w \times (l_a - l_b)$, where w is the lane width. For example, in the case $l_a = 2$ and $l_b = 1$, the conditional probability $P(z|v_1 = l_a, v_2 = l_b)$ is closer to 100% when the measurement z is closer to the lane width w .

4.5.3 Generalization of Prediction and Correction

The result for the two-vehicle-and-two-lane-road can be extended to the general case. Denote the number of vehicles that are communicating with each other as n_v , and the

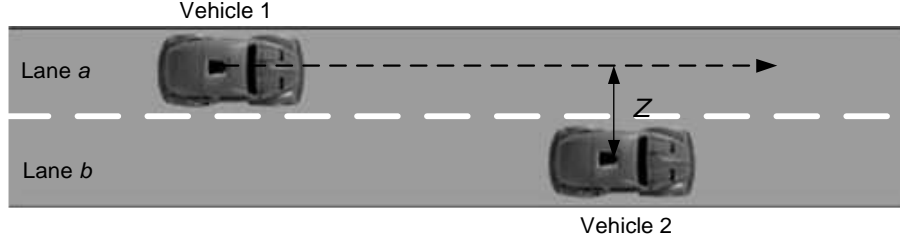


Figure 4.6: Correction step: this figure illustrates two cars traveling on a highway. The distance z is used to update the belief in prediction step.

number of lanes as n_l . Let $P(v_{1,t} = l_a, v_{2,t} = l_b, \dots, v_{n_v,t} = l_x)$ be the probability that car 1 is traveling in lane a , car 2 is in lane b , etc., at time t where a, b, \dots, x take on values between 1 and n_l . No initial estimates of the vehicle lane positions are required, and the initial probabilities are equally set to $1/(n_v \times n_l)$. The probability distribution at time t for the prediction step is given by

The probability distribution at time t for the prediction step is given by

$$P(v_{i,t} = l_a) \leftarrow \sum_{j=1}^{n_l} P(v_{i,t} = l_a | v_{i,t-1} = l_j) P(v_{i,t-1} = l_j), \quad (4.6)$$

and

$$P(v_{1,t} = l_a, v_{2,t} = l_b, \dots, v_{n_v,t} = l_x) = P(v_{1,t} = l_a) P(v_{2,t} = l_b) \dots P(v_{n_v,t} = l_x). \quad (4.7)$$

Bayes' rule for the correction step is

$$P(v_{1,t} = l_a, v_{2,t} = l_b, \dots, v_{n_v,t} = l_x | z_t) \leftarrow \frac{QS}{P(z_t)}. \quad (4.8)$$

where $Q = P(z_t | v_{1,t} = l_a, v_{2,t} = l_b, \dots, v_{n_v,t} = l_x)$ and $S = P(v_{1,t} = l_a, v_{2,t} = l_b, \dots, v_{n_v,t} = l_x)$.

With the developed Markov-based algorithm, we are able to implement simulations and experiments which will be discussed in the Sections 4.6 and 4.7.

4.6 Simulation

The lane positioning simulator is built on top of the traffic simulator described in Chapter 2.3. For a precise analysis of the algorithm, a map of a road based on an actual highway, as shown in Fig. 4.7, was built in VISSIM and a number of simulations with different situations were implemented. In all simulations, the GPS data are modeled by adding a random



Figure 4.7: Highway 85, Waterloo, ON, Canada.

walk bias of maximum amplitude 3 m and Gaussian noise having a standard deviation of 0.5 m to the position of each car.

Fig. 4.8 and Fig. 4.9 show a simulation with three cars and a two-lane road. The scenario is as follows: Initially, car 1 is in lane 1, and car 2 and car 3 are both in lane 2. After 7 seconds, car 2 switches to lane 1. Car 1 switches to lane 2 after 17 seconds, and car 3 moves to lane 1 after 26 seconds (Fig. 4.9). It can be seen that the algorithm accurately estimates the lane positions of the vehicles. The corresponding probabilities are given in Fig. 4.8. The algorithm also works when vehicles have frequent oscillatory weaving actions within their lanes (see Fig. 4.10), although this rarely happens on real highway systems due to safety concerns. Fig. 4.10 shows the lane estimation results without position filtering for three cars on a two-lane highway. It can be seen that car 1 and car 3 have wrong lane estimation by only a small fraction of time. The improved results when position filters consisting of a low-pass Butterworth filter and a particle filter for all three vehicles are added to reject the modeled GPS receiver noise (Gaussian noise) are shown in Fig. 4.11. Details about this filtering scheme will be described in Section 4.7.2. The speeds of the vehicles in all simulations are $80\text{-}120\text{ km/hr}$. Another case of interest is when the vehicles frequently change their lanes. The convergence rate of the algorithm in this case is faster since the localization algorithm has more frequent information update for both *prediction* and *correction* steps.

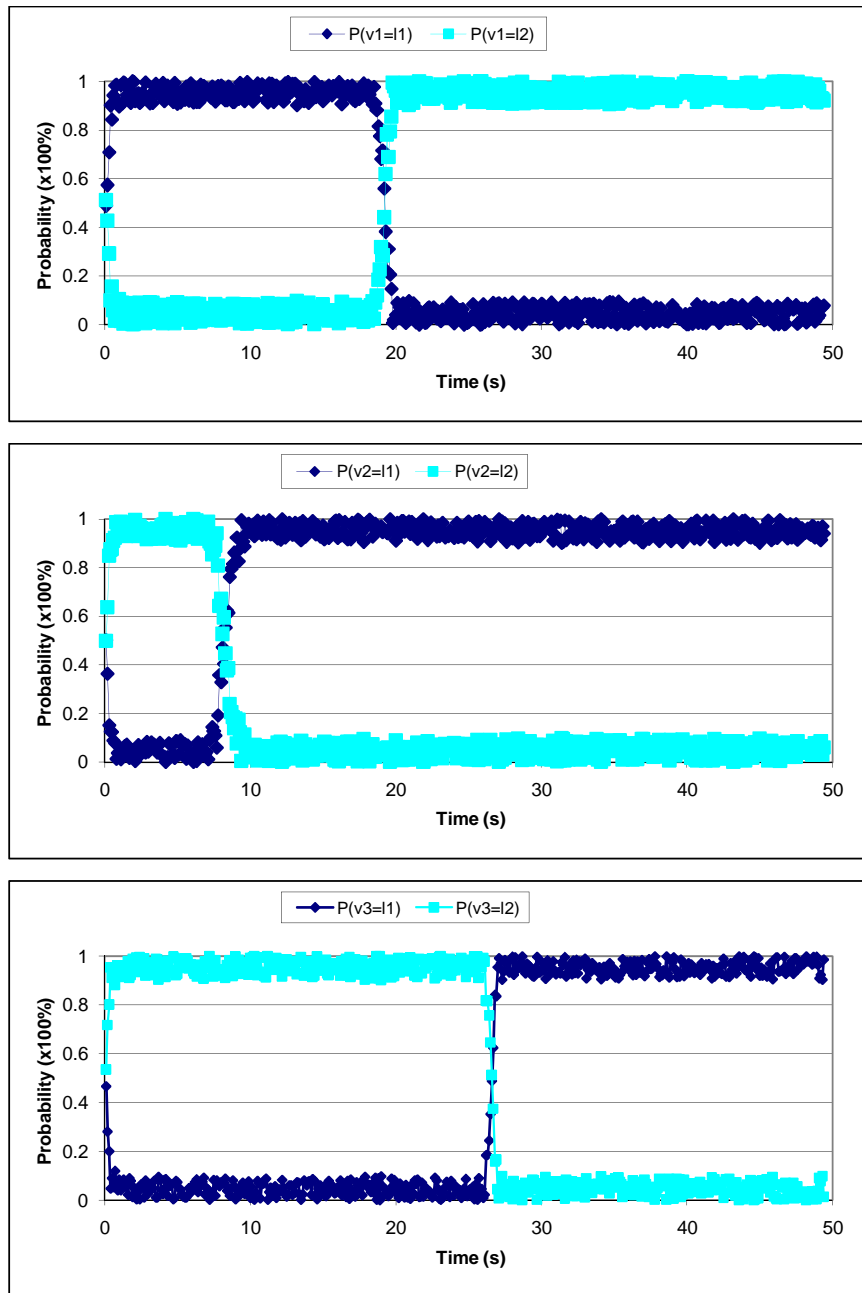


Figure 4.8: Probability distribution for simulation of three cars (car 1 to car 3 from top to bottom) on a two-lane highway.

Importantly, since the lane positions of vehicles are determined relative to each other, it is not necessary to know the number of lanes for the algorithm to work. The lowest lane position is always lane 1, and the other lanes are numbered relative to the first lane. In practice, as the number of vehicles increase, the algorithm will be able to estimate the lane positions faster and more reliably. The percentage of time with correct lane estimation for

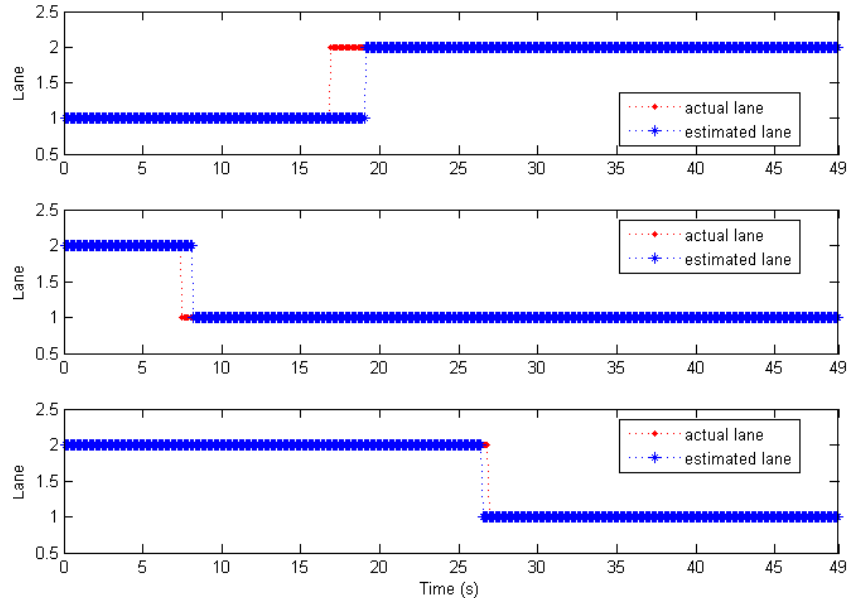


Figure 4.9: Simulation result for three cars (car 1 to car 3 from top to bottom) on a two-lane highway.

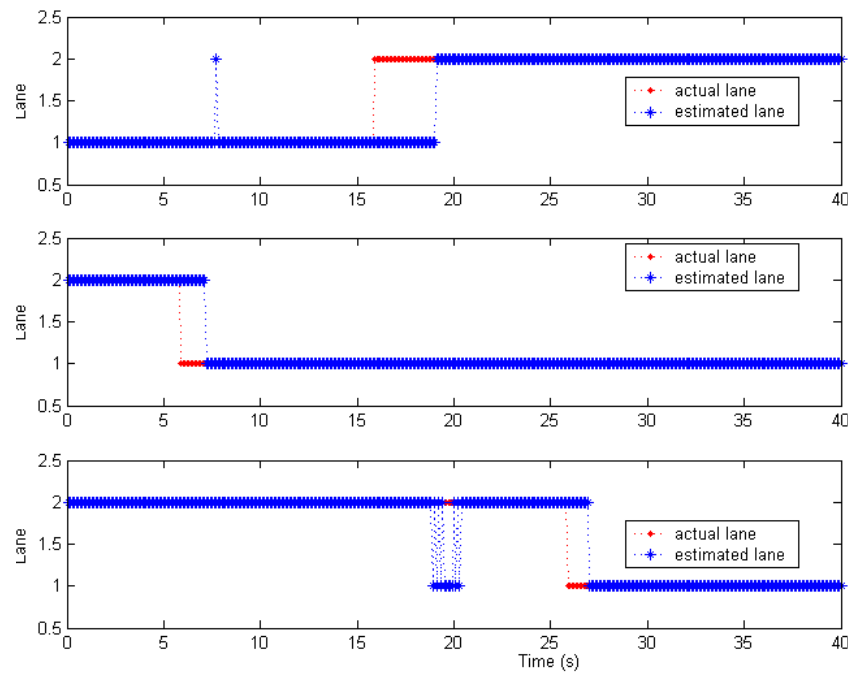


Figure 4.10: Impact of oscillatory weaving (car 1 to car 3 from top to bottom): This figure shows a simulation in which car 1, car 2, and car 3 follow sinusoidal paths with different amplitudes (-1 , -1 , and 1 m) and frequencies ($1/4$, $1/4.5$, and $1/5$ Hz).

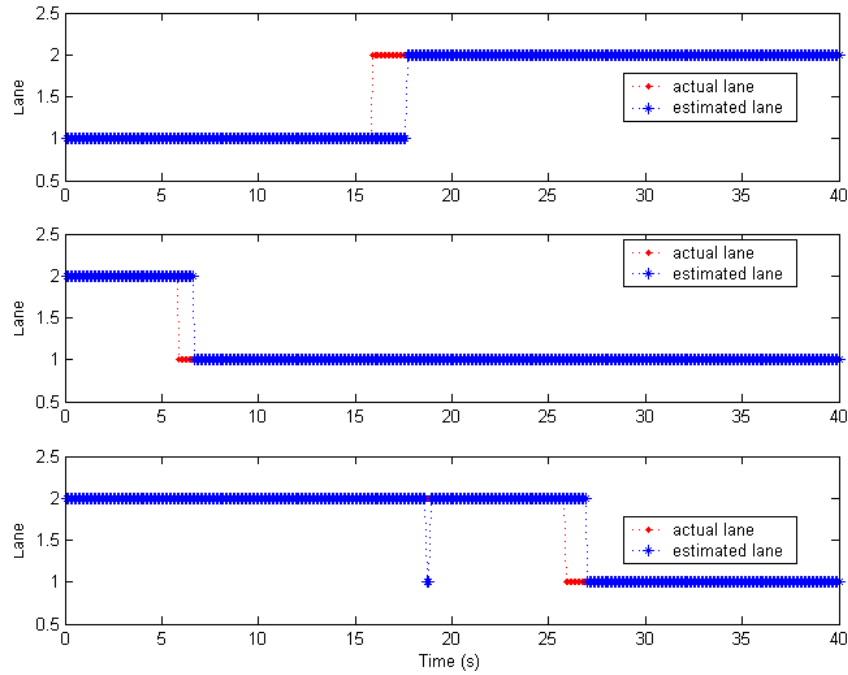


Figure 4.11: Improved lane estimation with position filters (car 1 to car 3 from top to bottom).

the simulations are summarized in Table 4.1.

4.7 Experiment Implementation

4.7.1 Software Architecture

To demonstrate the lane positioning algorithm, an in-car localization software was written in C++ which is responsible for reading GPS data from sensors, ensuring the smooth communication between the vehicles via wireless ad-hoc network, and estimating vehicle lane positions. Each vehicle is equipped with a GPS receiver and a processor to implement the lane positioning algorithm and to communicate with other vehicles. The GPS data for the vehicles is fed into a position filter to reject the measurement noise from receivers. The in-car localization software is organized in a multiple-threaded program. The functionality for the threads are as follows:

1. **TCP/IP socket server thread:** to send GPS data to other vehicles.
2. **Client threads:** to receive GPS data from other vehicles.

Table 4.1: Percentage of time with correct lane estimation for simulations.

Simulation	Correct lane estimation (%)
No oscillatory weaving, no position filters	Car 1: 93.1% Car 2: 96.3% Car 3: 98.9%
With oscillatory weaving, no position filters	Car 1: 91.8% Car 2: 96.8% Car 3: 94.5%
With oscillatory weaving and position filters	Car 1: 95.5% Car 2: 98.1% Car 3: 96.7%

3. **GPS thread:** to read GPS measurements from sensor, *i.e.*, GPS receiver.
4. **Filters thread:** to reject receiver measurement noise from raw GPS data.
5. **Localization thread:** to conduct localization algorithm for lane positions estimation.

The architecture for the system is illustrated in Fig. 4.12 and a screenshot of the localization software GUI is shown in Fig. 4.13.

4.7.2 GPS Receiver Noise Reduction

As described in Section 3.4, error from GPS generally comes from several sources including satellite clock, ephemeris error, ionospheric effects, tropospheric effects, and the geometry of visible satellites. This set of errors from GPS will be common for vehicles. Other errors are local to the different receivers and include RF noise from the environment, receiver noise and resolution, multi-path, and receiver clock error. Aside from multi-path, errors from this second set are generally smaller than those common to the receivers. Hence if multi-path is not a problem (as is the case for many open highways), then common errors from receivers that are relatively close to one another can be eliminated by utilizing relative distance between vehicles.

Other errors that need to be eliminated are the errors caused by receiver noise which is dependent on the design of antenna, the method used for the analogue to digital conversion, the correlation process, etc. These errors sometimes can falsely indicate a vehicle is

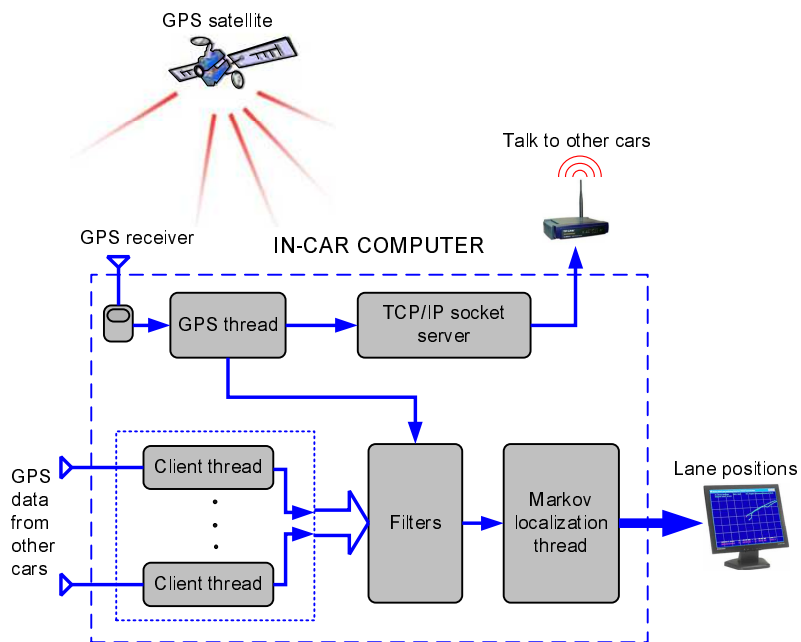


Figure 4.12: Program architecture.



Figure 4.13: Screenshot of localization program.

switching lanes. The high pitched receiver noise can be rejected using a low-pass filter. In the proposed lane positioning system, a combination of a low-pass Butterworth filter [60] fused with a particle filter can satisfy this task.

A second-order low-pass Butterworth filter is used to smooth the raw data distance measurements between two vehicles. More specifically, let the raw data measurement be $z_t = \text{distance}(GPS_t^{car1}, GPS_t^{car2})$ where GPS_t^{car1} and GPS_t^{car2} are raw GPS measurements from two vehicles. The distance z_t is then smoothed using a low-pass Butterworth filter producing the output z_t^b . However, the Butterworth filter causes time delay in the output due to phase-shift. The main objective of the particle filter is to compensate this delay while preserving the smoothed feature of the output curve.

Particle Filter

Particle filter, also known as Sequential Monte Carlo methods (SMC), is a sophisticated model estimation technique which approximates the sequence of probability distributions of interest using a large set of random samples. Particle filter is an implementation of the Bayes filter [61, 62, 63] which uses a finite number of parameters to describe the belief state distribution. Particle filtering is based on sequential importance sampling and Bayesian theory. It models the data distribution by random sample measures composed of particles, that are samples from the space of the unknowns, and their associated weights.

The spatial density of the particle set reflects the shape of the probability density function that it is sampled from, and hence the particle set is an approximation of the belief state probability distribution. A high density of the particles in the state space reflects the high likelihood of the true state in the vicinity [64]. In our problem, let the relative distance $x_{0:t} \in \mathfrak{R}$ that needs smoothing be the variable of interest at time t . The variable x_t is represented by a set of M particles $\mathbf{S}_t = (x_t[i], w_t[i]) : i = 1..M$, where the index i denotes the particle. Each particle consists of an estimate of the variable of interest and a weight $w_t[i]$ defining the contribution of this particle to the overall estimate of the variable [61, 64, 65, 66, 67].

The first step in implementing a particle filter is generating this set of M particles that is representative of the starting state $\mathbf{S}_0 = (x_0[i], w_0[i]) : i = 1..M$ (which may or may not be known). A random sample from a uniform distribution can be used if nothing is known of the starting state. On the other hand, if the exact starting state is known, all the particles can be initiated in the same state (thus occupying the same point in the state space).

Both the "act" and "see" steps of the Bayes filter are still carried out with the particle

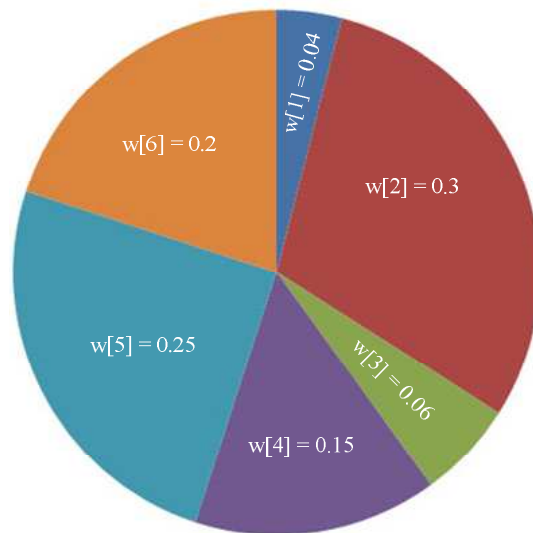


Figure 4.14: Re-sampling: particles with higher weights are more likely to be selected and sampled.

filter. The effects of motion control inputs (with noise) are applied to all particles using a state transition model [67]. Any type of function (linear or non linear) can be used in this case provided that it generates a propagated state based on the input of a previous state and a control input. For GPS noise reduction, the propagated particle set is a sample of the motion control propagated belief state distribution [61, 64, 65, 66] as

$$x_t[i] \sim P(x_t[i]|x_{t-1}[i], u_t), \quad (4.9)$$

where $u_t \in \mathfrak{R}$ is the control input which is, in this case, the output from Butterworth filter $u_t = z_t^b$.

The weight for each particle is updated based on the probability density function associated with the measurement model. Let $P(z_t|x_t[i])$ be the probability of perceiving z_t given $x_t[i]$. The weight $w_t[i]$ can be calculated by

$$w_t[i] = P(z_t|x_t[i]). \quad (4.10)$$

The calculated weights $w_t[i]$'s are then employed to generate a new set of particles for the next time step (re-sampling) using roulette wheel selection as shown in Fig. 4.14. In this step, a new particle set with the same number of particles as the current particle set is generated. Particles with a large weight are more likely to be sampled (maybe more than once). On the other hand, particles with a small weight may not be sampled and fail to reach the next iteration of the particle filter. The new particle set generated by this

survival of the fittest technique is representative of an approximation to the (posterior) state estimation after considering sensor measurements [61, 64, 65, 66, 67]. In the next iteration of the particle filter, the posterior particle set will represent prior belief state. The pictorial description of the particle filtering algorithm is shown in Fig. 4.15¹.

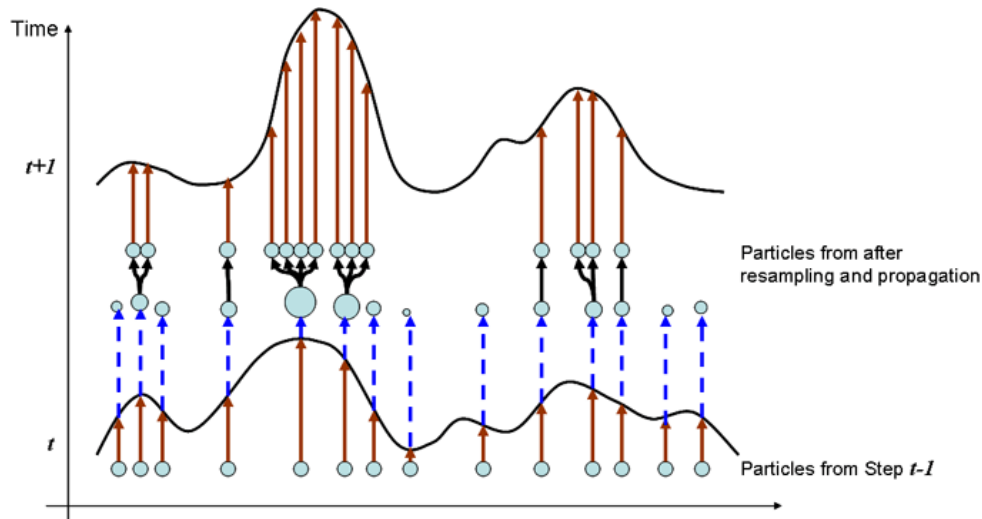


Figure 4.15: A pictorial description of particle filtering.

The pseudocode summarizing the particle filtering algorithm is shown in Algorithm 1 and the pseudocode for the re-sampling step using roulette wheel selection is shown in Algorithm 2.

Noise Rejection Results

In experiments, the number of particles was chosen to be 500 so that the filtering algorithm can work in real-time on Intel® Pentium®-M 1.4 GHz processors (computers used in the experiments). Fig. 4.16 shows the GPS plot for the two notebooks used in an experiment. The raw relative vs. filtered distances between the two computers (see Fig. 4.17) shows the effectiveness of the filtering scheme. It can be observed that the random walk caused by the GPS receiver noise was rejected effectively.

4.7.3 Results

A number of experiments with different scenarios were implemented on the same highway. Real GPS data were collected for the two cars used in the test. The approximately 4-km-

¹Image from [68]

Algorithm 1 Particle Filter

Require: A set of particles at time 0: $\mathbf{S}_0 = (x_0[i], w_0[i]) : i = 1 \dots M$

```

1: while  $t > 0$  do
2:    $z_t = \text{Distance}(GPS_t^{car1}, GPS_t^{car2})$ 
3:    $z_t^b = \text{Butterworth}(z_t)$  {Smooth raw distance}
4:    $u_t = z_t^b$  {Control input}
5:   for  $i = 1 \dots M$  do
6:     Draw  $x_t[i]$  with probability  $P(x_t[i]|x_{t-1}[i], u_t)$  {Propagation with  $u_t$ }
7:      $w_t[i] = P(z_t|x_t[i])$  {Update the weights}
8:   end for
9:   for  $i = 1 \dots M$  do
10:     $w_t[i] = \frac{w_t[i]}{\sum_{j=1}^M w_t[j]}$  {Normalize the weights}
11:   end for
12:    $\mathbf{S}_t = (x_t[i], w_t[i]) : i = 1 \dots M$ 
13:    $\mathbf{S}_t = \text{Resampling}(\mathbf{S}_t)$  {Generate new set of particles}
14: end while

```

Algorithm 2 Resampling

Require: The set of particles at time t: $\mathbf{S}_t = (x_t[i], w_t[i]) : i = 1 \dots M$

```

1: for  $i = 1 \dots M$  do
2:    $num = \text{Rand}(0,1)$  {Return a random number between 0 and 1}
3:   for  $j = 1 \dots M$  do
4:     if  $j == 1$  then
5:        $sum = 0$ 
6:     else
7:        $sum = sum + w_t[j - 1]$  {Sum of weights}
8:     end if
9:     if  $(sum \leq num < sum + w_t[j])$  then
10:       $x_t^*[i] = x_t[j]$  {Particle  $j$  is selected}
11:       $w_t^*[i] = w_t[j]$ 
12:      Add  $(x_t^*[i], w_t^*[i])$  to  $\mathbf{S}^*$ 
13:     end if
14:   end for
15: end for
16:  $\mathbf{S}_t = \mathbf{S}^*$  {New set of particles has been generated}

```

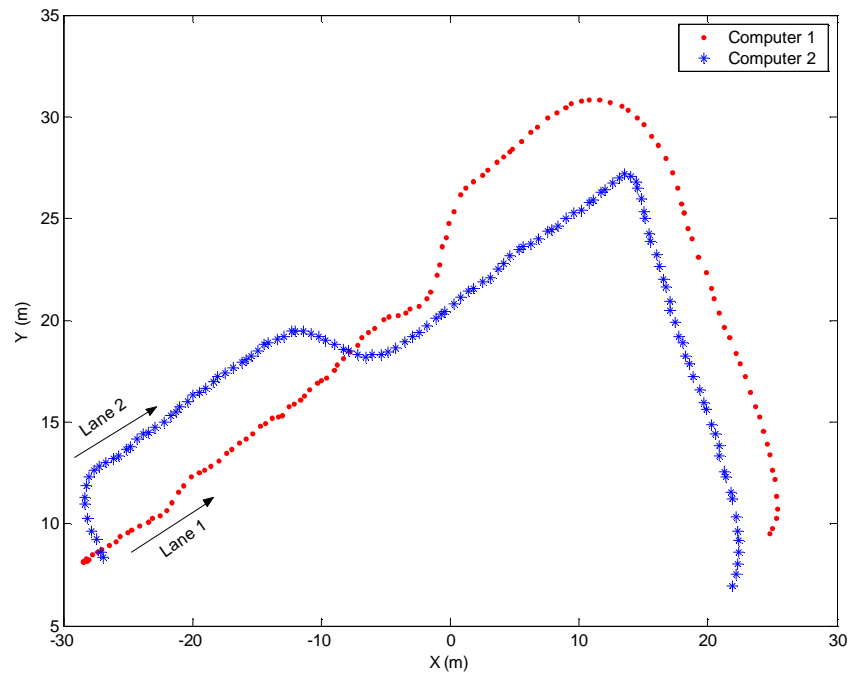


Figure 4.16: GPS measurements for two computers.

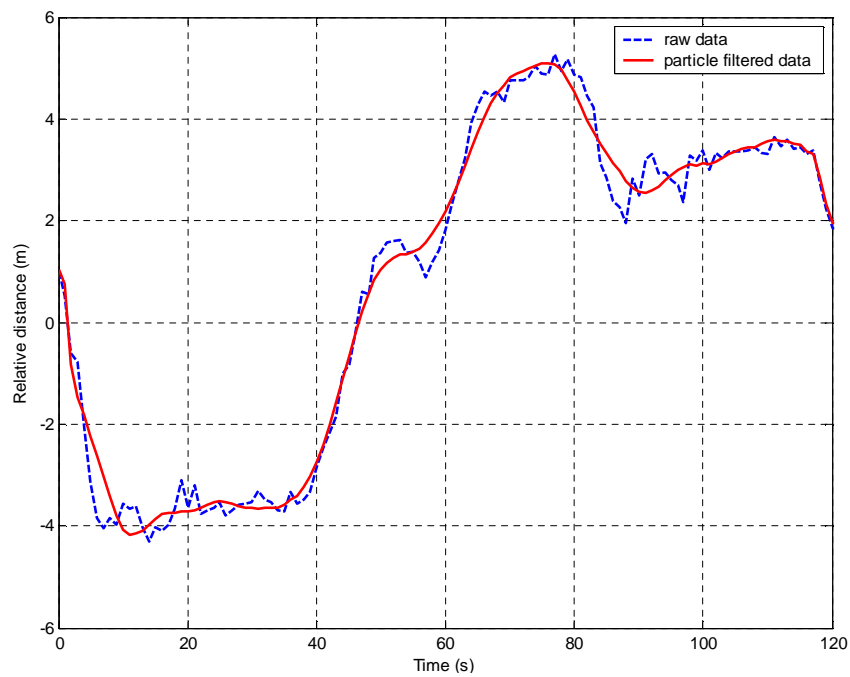


Figure 4.17: Relative distance from raw GPS data vs. filtered data.

highway section shown in Fig. 4.7 was used to conduct the experiments. The speeds of the vehicles used in the tests were between 90 and 110 km/hr .

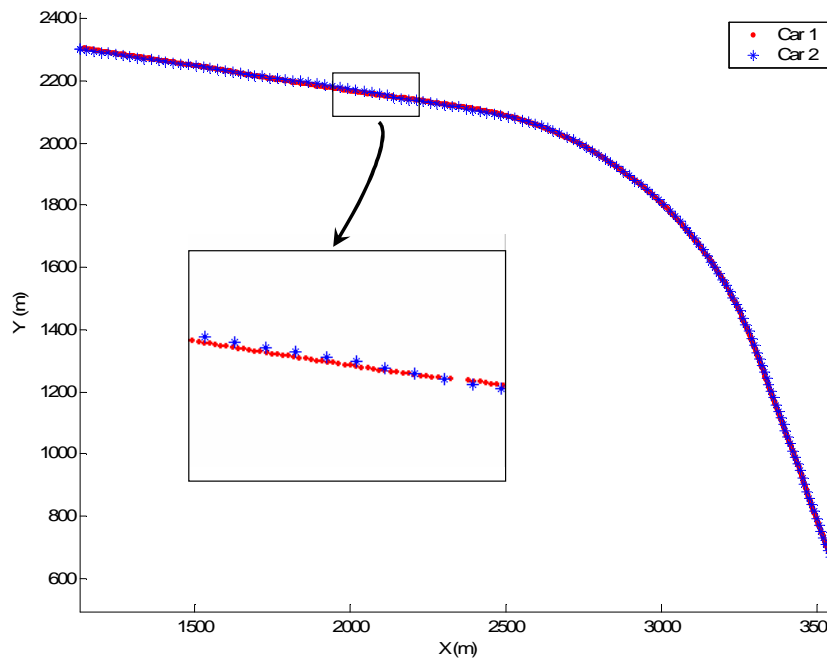


Figure 4.18: GPS measurements for two cars in experiment.

The GPS measurements and results from one of these experiments for both cars are plotted in Fig. 4.18, Fig. 4.19 and Fig. 4.20. These figures show the lane estimation results without using position filters to demonstrate the effect of GPS measurement noise. The strategy was as follows: Initially, car 1 was in lane 1, and car 2 was in lane 2. After 22 seconds, car 1 switched to lane 2. Car 2 switched to lane 1 right after car 1 completed its lane changing. Car 1 moved back to lane 1 after 40 seconds and finally moved to lane 2 after 60 seconds. Car 2 switched back to lane 2 at the fortieth second and maintained its lane position until the end of the experiment.

In this experiment, two cars equipped with low cost GPS receivers with different sampling rates and variances drove on the highway. The GPS receiver for the first car was a Garmin 18-5 which has the sampling rate of 5 Hz . The GPS receiver for the second car was a LocSense 40-CM whose sampling rate is 1 Hz . Both GPS receivers output National Marine Electronics Association (NMEA) 0183 standard messages. The second GPS receiver signal has a much higher noise when compared to the first receiver. This resulted in probability distributions with higher variability, as shown in Fig. 4.19. The performance of lane position estimation is thus reduced in that the lane the vehicle occupied was not always accurately predicted (see Fig. 4.20).

Fig. 4.21 and Fig. 4.22 show the effects of the improved noise rejection on the probability and lane estimates when a position filter for car 2 was used in comparison to the results

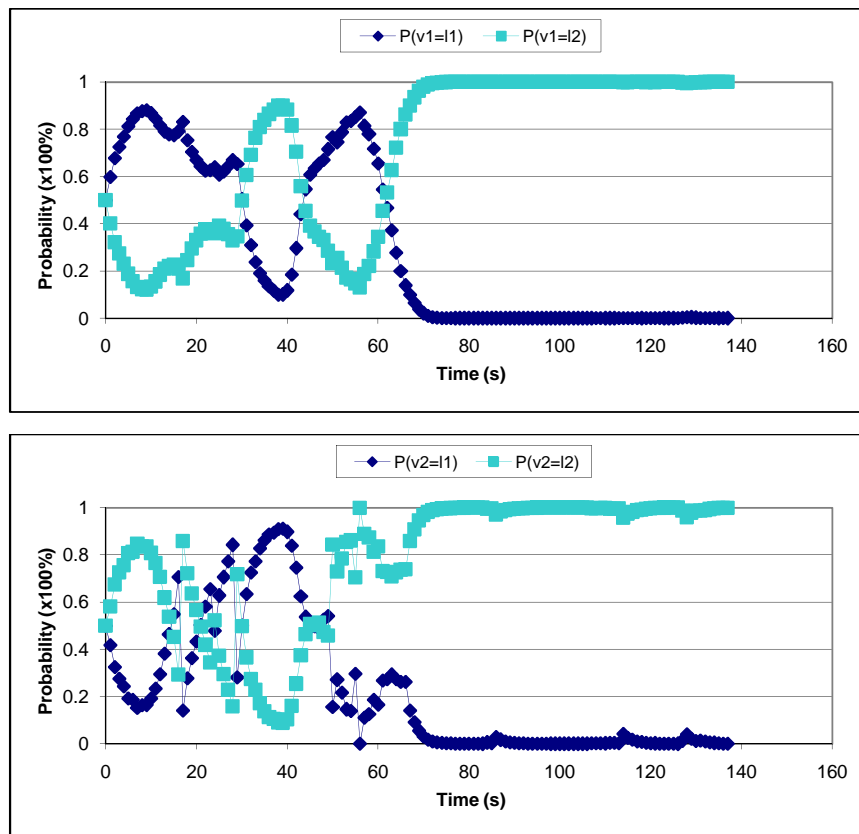


Figure 4.19: Probability distribution for experiment without GPS measurement filtering.

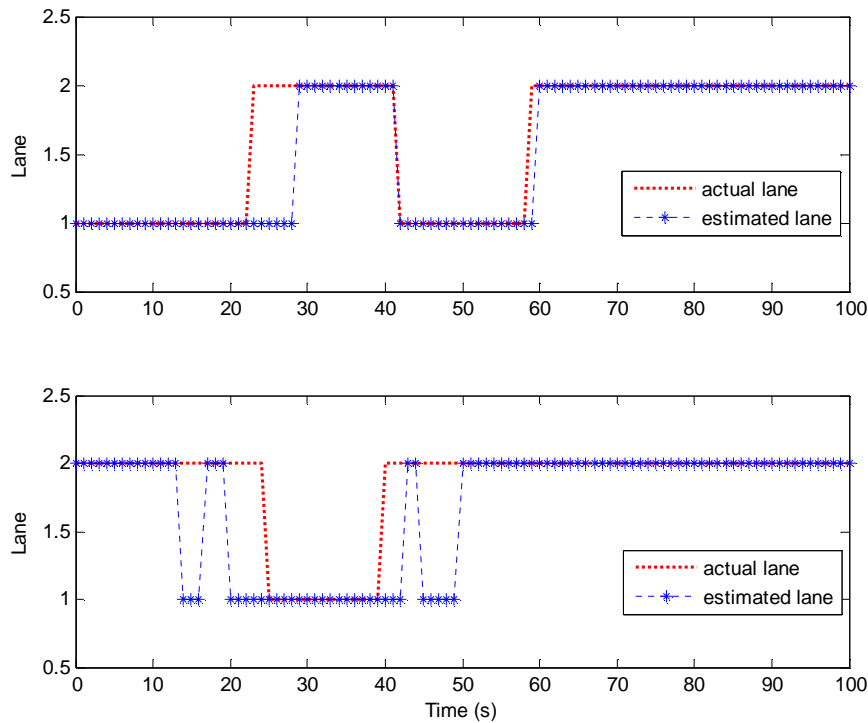


Figure 4.20: Lane estimation result for two cars (car 1 to car 2 from top to bottom) on highway without GPS measurement filtering.

shown in Fig. 4.19 and Fig. 4.20. This filtering resulted in more accurate and faster estimates of lane position. This improvement indicates that, in practice, low-cost GPS receivers can be effectively fused with a filter to obtain low-noise GPS measurements, rather than resorting to the use of more expensive sensors. It is also interesting to point out that the algorithm sometimes could anticipate the lane-changing action before the vehicles completely moved to the other lanes, as indicated in Fig. 4.22, at around the twenty-second second. This can be explained by the fact that when a vehicle starts making a lane change maneuver, its lateral displacement used in the *prediction* step of the localization algorithm anticipates the vehicle lane changing tendency before the lane changing is completed.

To demonstrate that the algorithm can work effectively at low speeds, a number of experiments were conducted on Wilhelm St in Kitchener, ON, Canada with two people, each holding a notebook computer. Each computer was a Compaq Evo N620c, and was equipped with a Garmin-18 GPS receivers. The communication between the notebooks was made through the IEEE 802.11a/11g standard D-Link DWL-AG660 Wireless Cardbus Adapters. The experimental apparatus are shown in Fig. 4.23. The walking speeds for both persons were estimated from GPS data to be approximately 5 km/hr .

The GPS measurements for both computers in the first experiment are plotted in

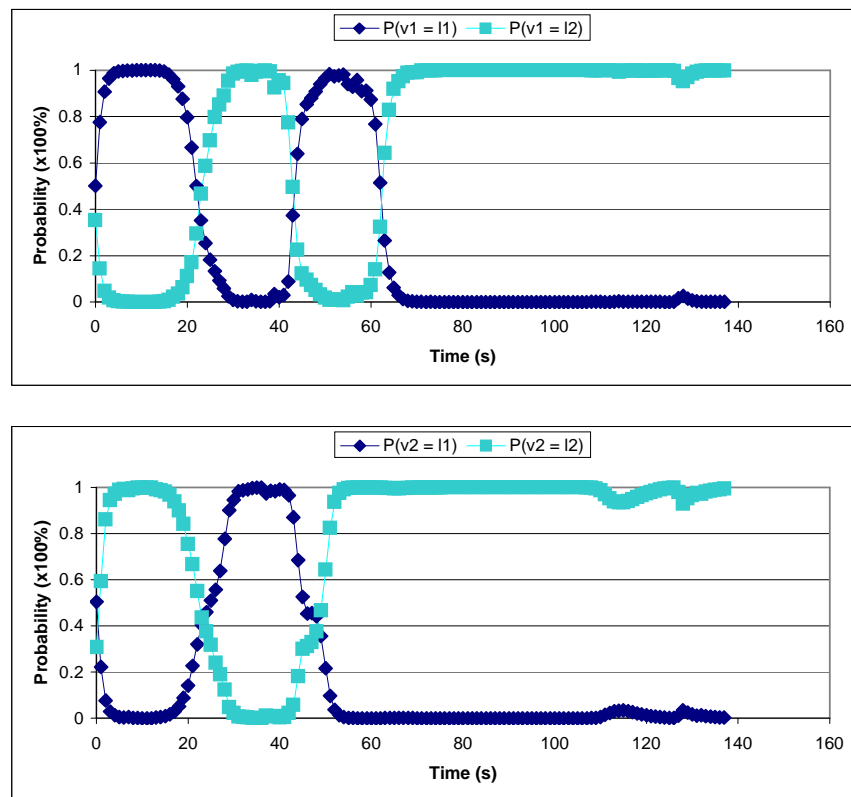


Figure 4.21: Improved probability distribution for experiment with GPS measurement filtering.

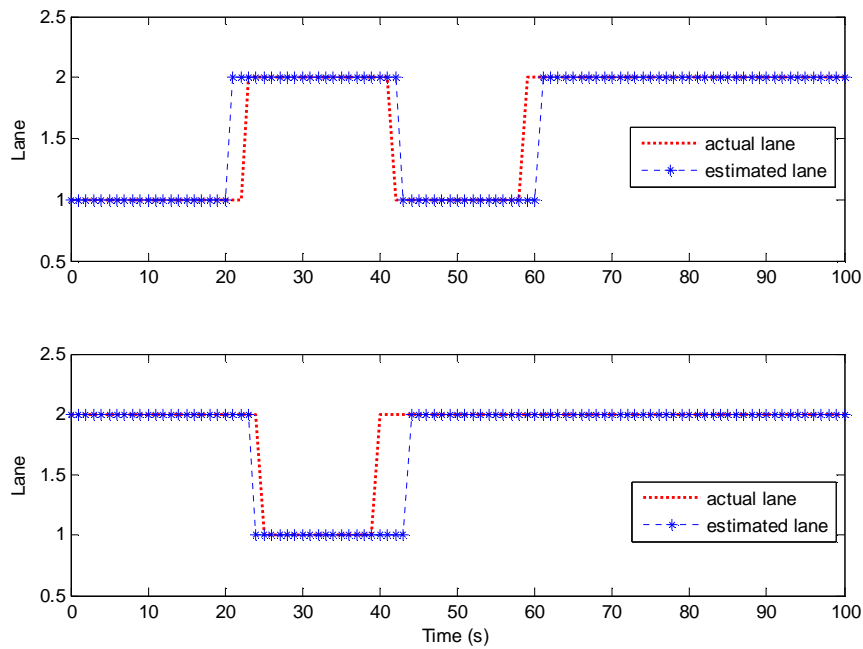


Figure 4.22: Improved results with GPS measurement filtering.

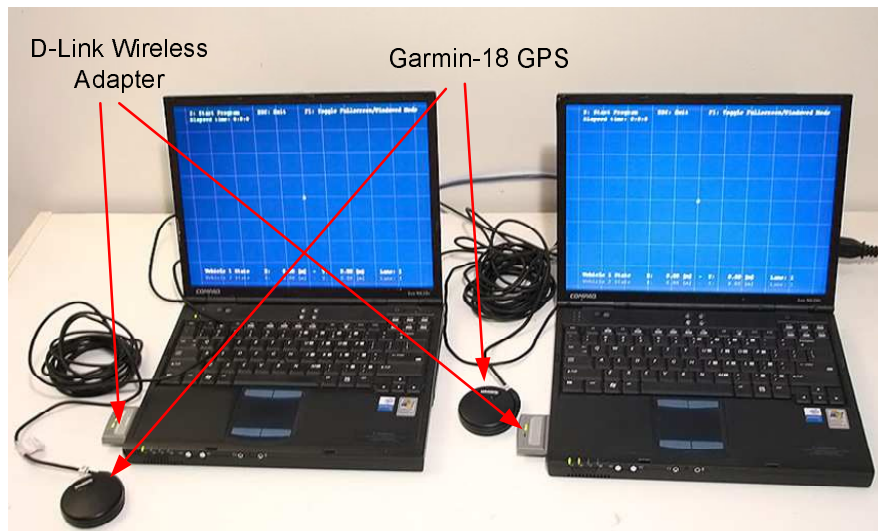


Figure 4.23: Experiment apparatus.

Fig. 4.16. The strategy was as follows: Initially, both person 1 (computer 1) and person 2 (computer 2) were in lane 1. After 6 seconds, person 2 switched to lane 2. Person 2 moved back to lane 1 after 44 seconds. Person 1 switched to lane 2 at the 63th second and stayed in lane 2 until the end of the test. The resulting probability distributions and estimated lane positions are shown in Fig. 4.24 and Fig. 4.25, respectively. The estimation appears to be quite good with the estimated lane positions following the actually values very closely.

Fig. 4.24 shows how confident the system was about its estimation.

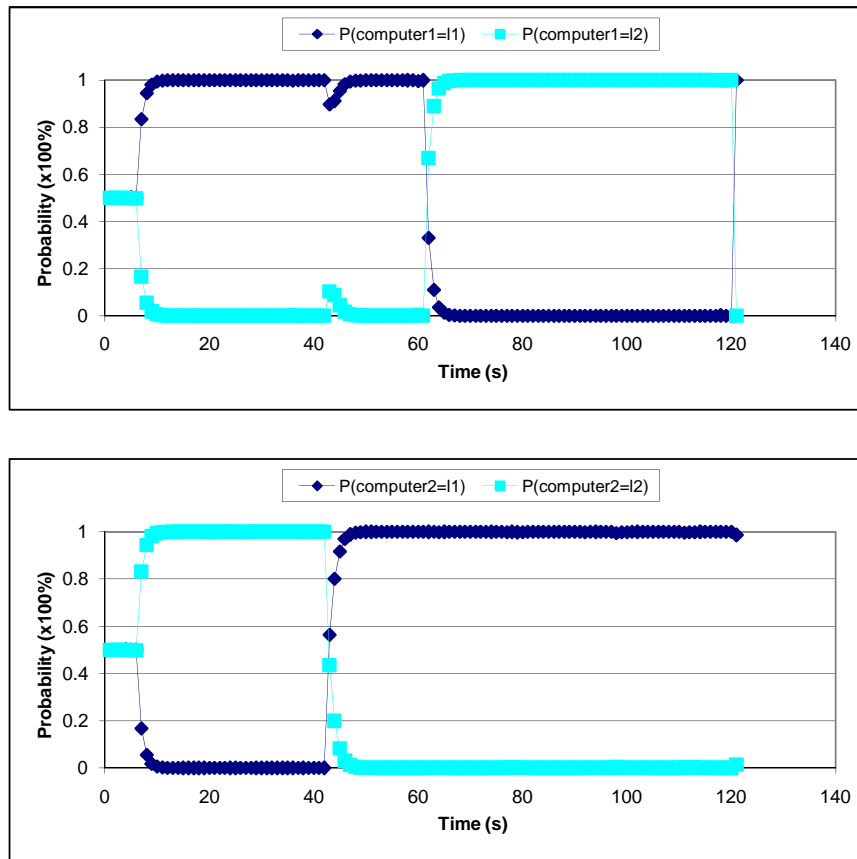


Figure 4.24: Probability distributions for experiment 1.

In the second experiment (see Fig. 4.26 for GPS plots) person 1 (computer 1) was initially in lane 1 and person 2 (computer 2) was in lane 2. After 12 seconds, person 1 switched to lane 2. Person 1 moved back to lane 1 after 28 seconds and finally moved to lane 2 after 63 seconds. Person 2 switched to lane 1 at the 45th second and stayed in lane 1 until the end of the test. Fig. 4.27 and Fig. 4.28 show the resulting probability distributions and estimated lane positions, respectively. It is interesting to note that the system sometimes could anticipate the lane changing action before the vehicles completely moved to the other lanes as indicated in Fig. 4.28 at around the 28th second. This can be explained by the fact that when a vehicle/person starts making a lane change maneuver, its lateral displacement used in the prediction step anticipates the vehicle lane changing tendency before the lane changing is completed. In this case, this happened when person 1 made a sharp turn at the 28th second as can be seen in Fig. 4.26.

The third experiment dealt with a three-lane situation on the same road section. The lane estimation results are given in Fig. 4.29 and Fig. 4.30. It can be seen from Fig. 4.30

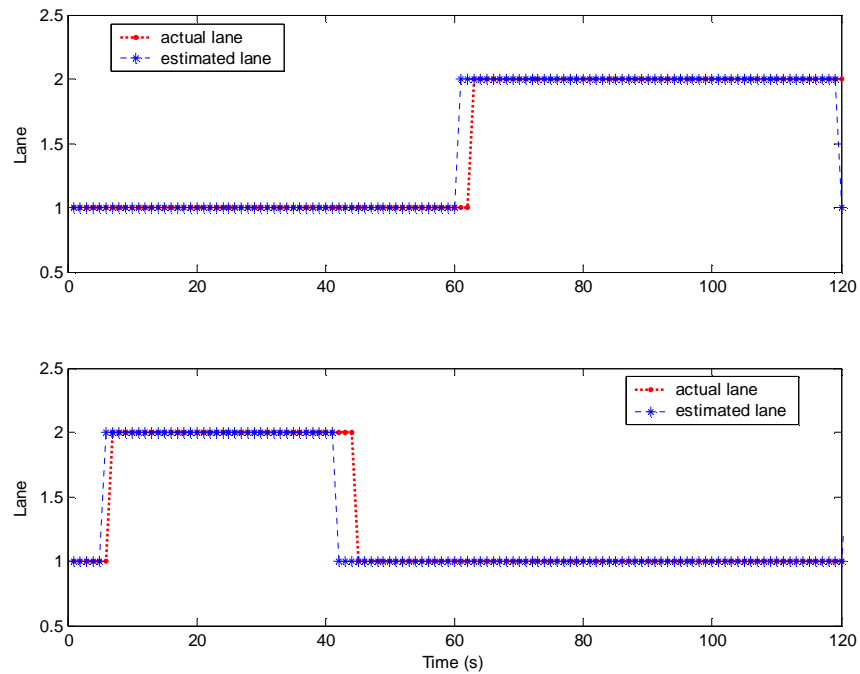


Figure 4.25: Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

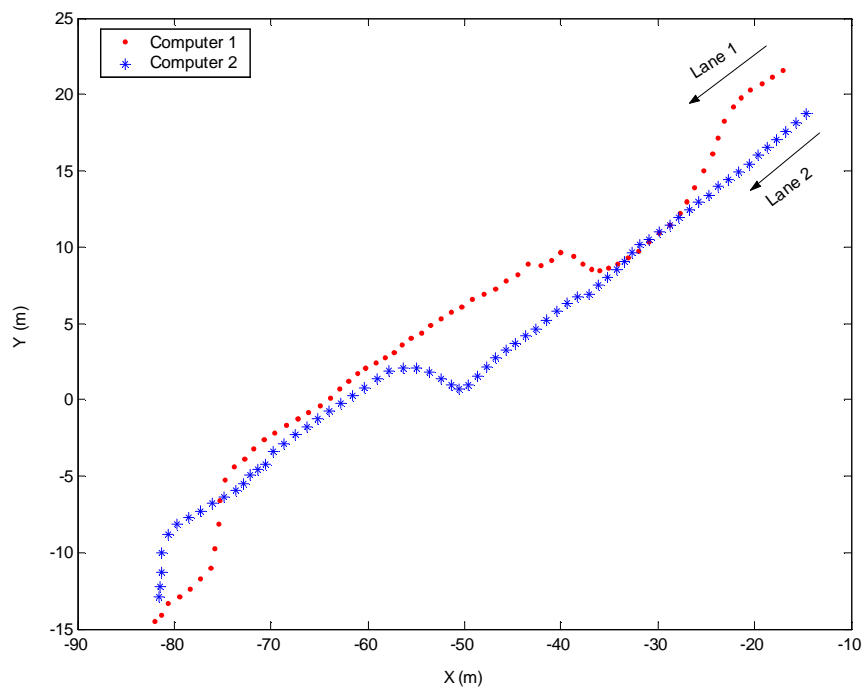


Figure 4.26: GPS measurements for experiment 2.

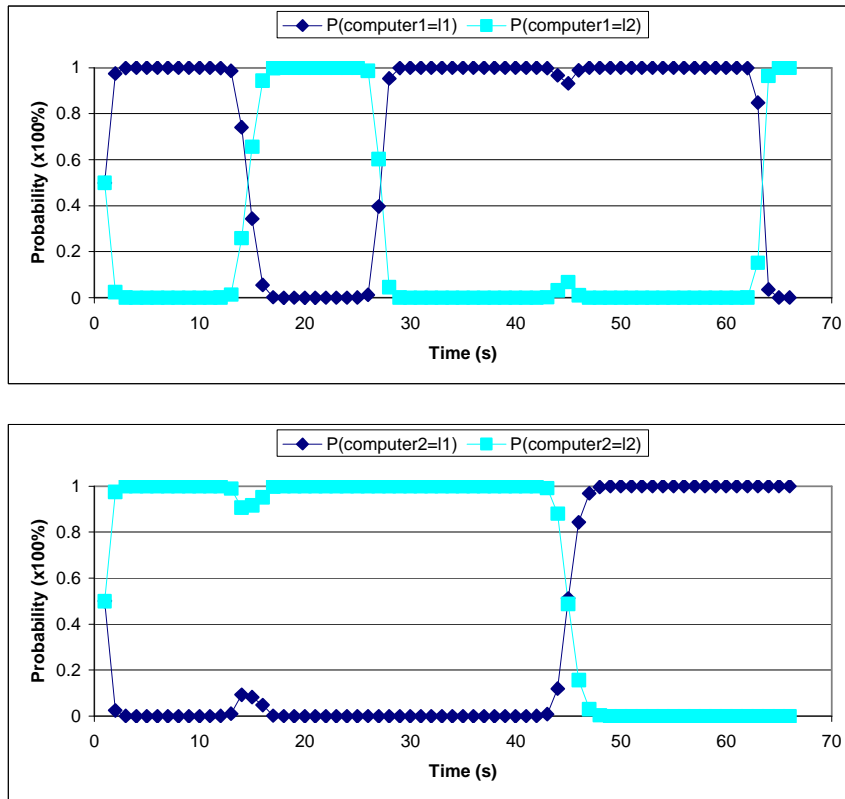


Figure 4.27: Probability distributions for experiment 2.

that both computers started off with the wrong estimation as the computers initially had no knowledge about the lanes they occupied and the initial lane positions were assumed to be the lowest lane, i.e., lane 1. Fig. 4.29 shows that the system was not very confident about the estimation in the first 7 seconds of the experiment. However, as soon as person 1 started moving to lane 1 at around the 10th second, the estimated lane positions for both computers quickly converge to the actual values. This experiment indicates that Markov localization algorithm can work well without prior knowledge about the vehicle's initial lane position. It is, therefore, possible to localize the vehicle from scratch and to recover from localization failures or GPS outage in which the positions of the vehicle are lost. The percentage of time with correct lane estimation for the simulations are summarized in Table 4.2.

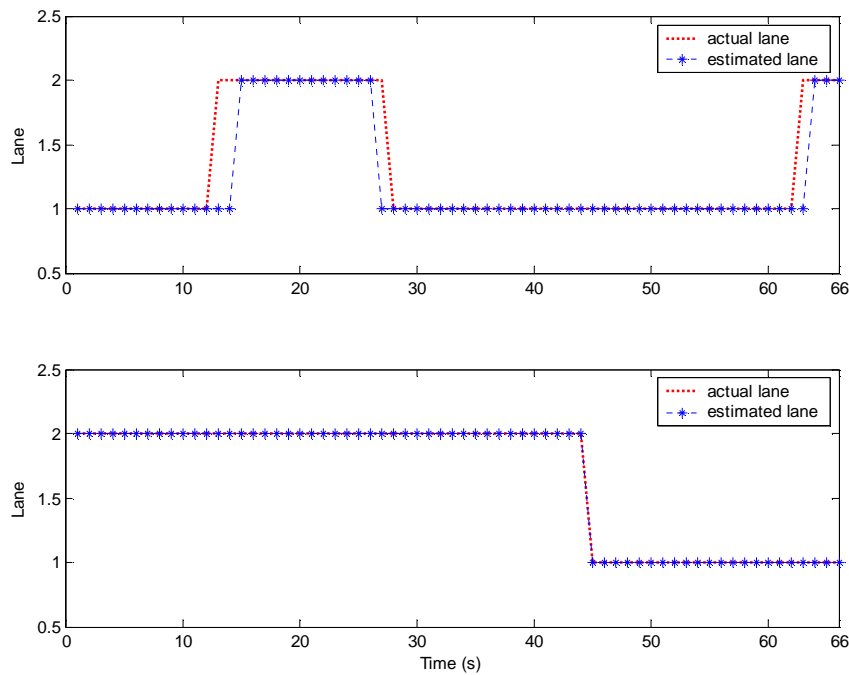


Figure 4.28: Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

4.8 Conclusions

A new lane position estimation algorithm that uses a Markovian approach based on cooperative driving models has been proposed in this chapter.

In comparison to conventional lane positioning methods which usually deal with complicated image processing techniques and/or expensive equipment, the proposed method only requires low cost GPS receivers, IVC, and a simple localization algorithm. Simulation and experimental results have shown the efficiency of the algorithm, even when the GPS data are significantly degraded.

The limitation of the proposed strategy lies in the fact that it only uses GPS data to estimate lane positions. This might be challenging where GPS data is not available or GPS signal is blocked completely by large obstacles like in a long tunnel. One possible solution to this problem is to fuse the GPS data with another type of sensor such as an Inertial Measurement Unit (IMU) until GPS data is again available.

Future research will continue with more complicated situations such as determining lane position on highways with intersections. Issues to be addressed include derivation of per-lane conditional probability models that are required at intersections. For example,

Table 4.2: Percentage of time with correct lane estimation for experiments.

Test	Correct lane estimation (%)
Two cars, no position filters	Car 1: 92.0% Car 2: 83.0%
Two cars, with position filters	Car 1: 93.0% Car 2: 95.0%
Walking 1	Computer 1: 95.0% Computer 2: 95.8%
Walking 2	Computer 1: 93.9% Computer 2: 100.0%
Walking 3	Computer 1: 94.3% Computer 2: 78.1%

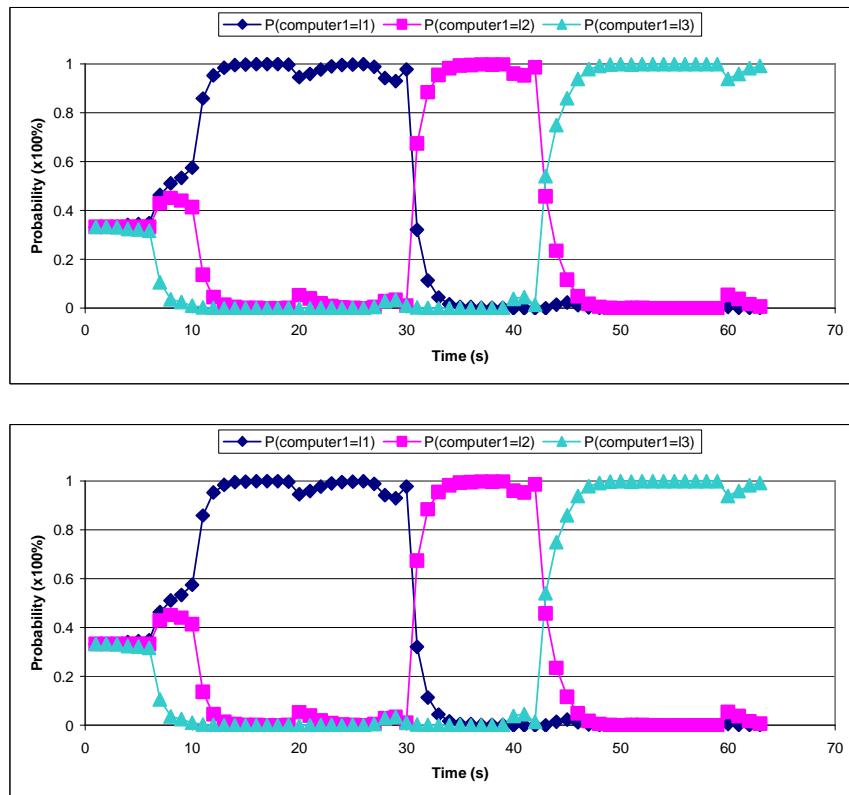


Figure 4.29: Probability distributions for experiment 3.

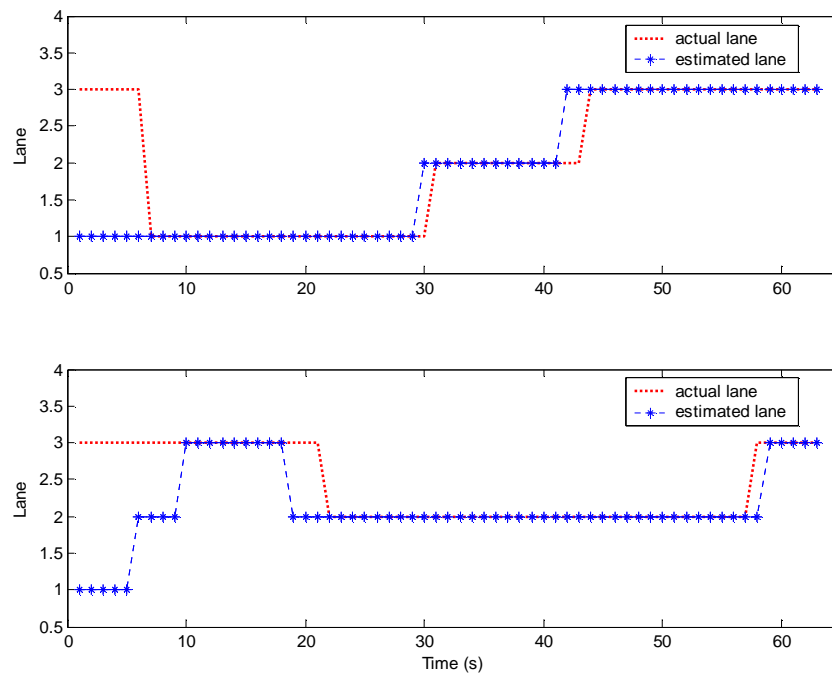


Figure 4.30: Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

the positioning system may indicate that 100% of drivers in the left lane at a particular intersection turn left, 50% of drivers in the right lane go straight, and 50% turn right.

Chapter 5

Lane Assignment Optimization

5.1 Introduction

Lane assignment is seen as one of the primary means of increasing traffic throughput on urban highways characterized by: (1) relatively small distances between adjacent entry and exit points from the highway, (2) a significant incidence of lane maneuvers undertaken by drivers attempting to reach their destinations quicker by utilizing the faster lanes, and (3) trip lengths which on the average span a significant number of sections between entry and exit points. It is on these highways that congestion is the most acute and lost resources including time, fuel, human lives, and negative effects on the environment are the greatest.

In this chapter, a direction for lane assignment is proposed with the aims of enhancing traffic system operation, enhancing safety, reducing travel time and improving traffic quality. The lane optimization problem is formulated here as a linear programming problem [69, 70] with the minimization of a cost function. The lane assignment strategy considered here assigns a fixed lane to a vehicle for the duration of a road segment starting from an entry point to the next nearest exit.

To implement lane assignment, a strategy based on inter-vehicle communication and collaborative driving model will be developed. This system will use GPS as the only available sensor and form an ad-hoc network as the vehicles travel within communication range of each other.

5.2 Related Work

Currently, almost all traffic management systems regulate traffic flows by controlling traffic signals or highway ramp meters. In these systems, the traffic is treated as a single mass and

the behaviors of individual cars are normally ignored [71][72][73]. Most recently, Goolsby *et al.* [74] even used changeable lane assignment signs at frontage road intersections to adapt to traffic conditions that change based on the time of day. When these interchanges experience high turning movement demands, permitted double turns are often used to increase traffic throughput (see Fig. 5.2). These previous approaches, however, miss an important component of traffic management: *coordination of cars themselves*.

One of the first works on collaborative lane assignment was conducted by Hall [75] who considered the lane-change maneuvers in deriving a min-max optimization problem. Hall adopted the total workload model combining lateral and longitudinal movements, so the form of the cost function to be optimized is the sum of the longitudinal requirements and adjustment factor representing the incremental workload for lane-change maneuver. Then, the objective is to equalize the workload and congestion across lanes. However, this work assumes that the vehicles flow into or out of highway at any point as in fluid flow (flux) systems and exact entrance and exit locations are not modeled. In other words, access and egress are assumed to occur continuously over the entire length of the highway. This work also does not account for variations in traffic flow. Optimization of such a problem would specify the locations where lane changes occur.

Medanic *et al.* [5] and Ramaswamy *et al.* [76][77] defined the lane assignment problem using an itinerary matrix, with the objective of reducing unnecessary crossings on the highway because it was assumed that maneuvers lead to path intercepts and are the main reason for delays and capacity reduction. Their simulation results show effects of maneuvers on costs, but a distributed control strategy for individual car was not presented.

Hall *et al.* [78] presented another approach to modeling highway systems (Fig. 5.2). They used network representation from network theory and represented a highway with arcs and nodes. Their paper does not deal explicitly with the lane change effect on the highway capacity. However, the network representation of a highway shows potential for providing a framework to which other methods can be applied.

Kim *et al.* [79] used the same approach by partitioning the highway systems. In their work, the lane assignment problem was formulated as an optimization problem to find proper positions of partitions on an itinerary matrix. Then the optimization problem was solved using genetic algorithms. The validation of the algorithm was conducted by comparing it with a random strategy in a realistic situation.

Lee *et al.* [80] handled the problem by using discrete event modeling. The model for the overall traffic system corresponding to a particular highway is decomposed into sub-models which may be recursively decomposed into smaller sub-models. As in [79], Lee *et al.* used roadside computers to receive information from vehicles, and assign them the lanes that



(a)

(b)

Figure 5.1: Changeable lane assignment signs.

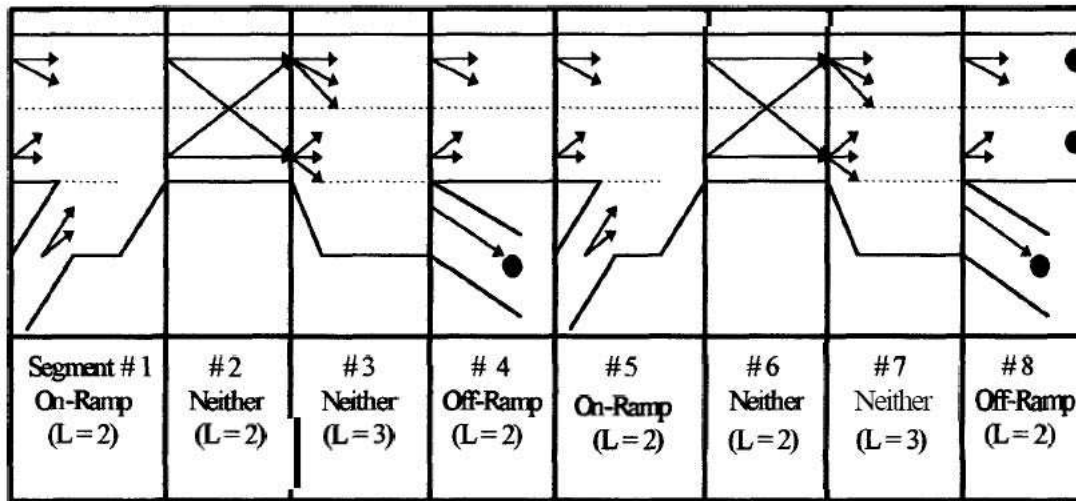


Figure 5.2: Highway system discretization in [78]: highway segments are indexed by location and defined by segment type (on-ramp, off-ramp, or neither), length of the segment, number of lanes, and ramp capacity. Nodes are assigned to the end of each lane in each segment, as well as to the start of each on-ramp and end of each off-ramp.

they should occupy for most of the trip. They also assumed that all cars move along the assigned lanes with the specified velocity of the lane. With this assumption, they neglected the fact that vehicle speeds are affected by traffic density in each lane. Therefore, their system was not really practical for current highway systems.

With that in mind, we approach the problem by taking the coordination and communication of vehicles into account. In our system, vehicles collaboratively select lanes with an effort to increase traffic throughput. Control is distributed such that each vehicle has a controller that allows it to coordinate with other vehicles to minimize the number of lane changes and avoid unpredictable situations based on inter-vehicle communication.

5.3 Problem Statement

Our aim is to coordinate maneuvers by sharing information between vehicles to schedule vehicle paths. The scheduling should try to maximize throughput by taking into account the origins and destinations (specified by drivers when entering a highway) of all vehicles, as well as the effects of maneuvers dictated by the proposed path scheduling strategy on time of travel. Our fundamental assumption is that the vehicles are equipped with microprocessors, GPS receivers, and wireless communication devices. A following assumption is that any two vehicles within a certain radius of each other can communicate. A vehicle should select lanes not only to improve its own travel time, but the travel time of other vehicles and the overall traffic flow of the highway system.

We can state the lane assignment problem to be: *Determine, for each origin-destination pair, the number of vehicles that should travel in each lane of the highway in order to minimize a performance index that is a function of total travel time.*

Several topics regarding inter-vehicle communication, lane finding and lane assignment are explored in this chapter: First, a brief introduction of linear programming is given in Section 5.4. The system architecture and the proposed steps to the solution are discussed in Section 5.5. Section 5.6 demonstrates a method for flow rate estimation in each lane using inter-vehicle communication. Section 5.7.1 discusses a distributed control strategy for routing vehicles to appropriate lanes while satisfying some constraint conditions. A proposed cost function to optimize is also discussed in Section 5.7.2. To accurately model traffic flow, a car-following model is proposed in Section 5.7.3. Currently, by solving a minimization problem, the algorithm is able to send vehicles to appropriate lanes in an effort to balance lane traffic flows and decrease the total vehicle travel time. Simulations to evaluate the algorithm are provided in Section 5.8, followed by some discussion about ‘selfish lane selection’ in which a portion or all of the vehicles do not comply with the suggested lane assignment strategy in Section 5.9. Finally, some concluding remarks are provided in Section 5.10.

5.4 Linear Programming

Linear programming problems are optimization problems in which the objective function and the constraints are all linear. The standard form of a linear programming problem consists of the following three parts:

- A linear cost function to be maximized or minimized:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

- Problem constraints:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2,$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m,$$

- Non-negative variables:

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

The problem can be expressed in matrix form as follows:

Optimize

$$\mathbf{c}^T \mathbf{x},$$

subject to

$$\mathbf{Ax} \leq \mathbf{b}, \text{ and } \mathbf{x} \geq \mathbf{0},$$

where the variables $\mathbf{x} \in \Re^n$ are called decision variables. These are the variables we need to determine so that the problem is optimized.

Linear programming problems can be solved by numerous methods, some of which are the Ellipsoid method [81] and the Simplex method (Appendix A). The next Sections will show how the lane optimization problem can be organized as a linear programming problem.

5.5 System Architecture

There are two architectures for implementing lane selecting schemes: centralized and decentralized. The key advantage of a decentralized over a centralized approach is scalability. It is easier to grow a decentralized system and to add new elements to it.

Two recent approaches taken for lane assignment are: lane assignment for single vehicles (*i.e.*, free-agent lane assignment), and lane assignment for groups of vehicles by organizing vehicles into platoons [82]. The idea of ‘platooning’ is to expand the limitation of capacity and safety that can be achieved by road vehicles.

In the framework of this chapter, the problem of lane assignment for single vehicles will be considered and investigated. The proposed implementation of the overall system can be presented as a closed-loop system and is broken into three main steps (see Fig. 5.3): (1) lane positioning (lane occupancy estimation), (2) lane flow estimation, and (3) lane assignment. Note that each of these three individual steps is accomplished through the collaboration of multiple vehicles communicating within ad-hoc networks. Lane positioning is necessary for vehicles to know where they are on the highway. In an automated highway system, vehicles are expected to know the lanes they occupy. This serves as a basis for the rest of the lane assignment algorithm. Lane positions of vehicles can be determined using Markov localization based on the exchange of GPS fixes between vehicles through a wireless ad-hoc network as discussed in Chapter 4. In order to determine if a lane capacity has been exceeded, the traffic flow rate in each lane must be estimated. A method for lane flow rate estimation will be discussed in Section 5.6. The lane assignment algorithm is implemented by individual vehicles with the objective of minimizing the total travel time of themselves and other neighboring vehicles.

In this system, let us consider a highway with n_e entry (on-ramp) and n_d exit (off-ramp) points. Each lane is characterized by a different nominal driving speed. The highway system used in this work is discretized into segments. For every on-ramp, a new segment, which contains one or more lanes, is created. The number of lanes can vary from segment to segment but must be constant along each segment. Lanes are numbered from right to left when facing in the direction of traffic flow, with the right-most lane numbered 1. On-ramps and off-ramps are designated as lane 0. Lane exits and lane entrances are assumed to occur on the right side of the highway. It is assumed that a typical vehicle would enter the segment in lane 0 and proceed gradually to the lane assigned to it.

As illustrated in Fig. 5.4, the highway is represented by a network. Nodes are placed at the start of each entry point. The objective is to maximize the flow across the highway. Mathematically speaking, the constraints of the problem are:

1. *Non-negativity*: the number of vehicles cannot be negative,
2. *Lane capacity*: the maximum rate at which vehicles can enter a lane cannot be exceeded.

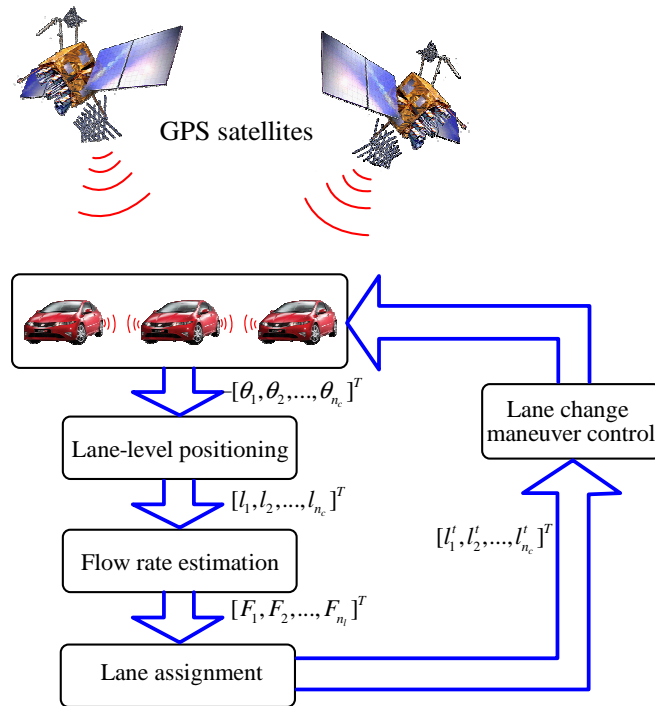


Figure 5.3: Control diagram: $[\theta_1, \theta_2, \dots, \theta_{n_c}]^T$ are the GPS measurements for car 1 through car n_c , $[l_1, l_2, \dots, l_{n_c}]^T$ are their lane positions, $[F_1, F_2, \dots, F_{n_l}]^T$ are the estimated flows in lane 1 through lane n_l , and $[l'_1, l'_2, \dots, l'_{n_c}]^T$ are target lanes for these cars.

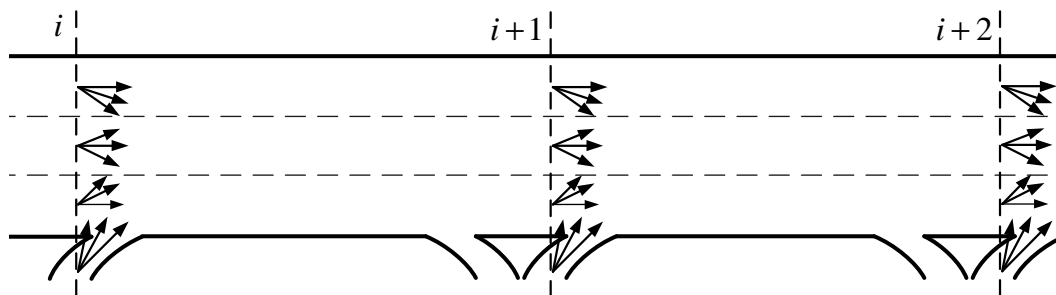


Figure 5.4: Highway system.

5.6 Traffic Flow Estimation

5.6.1 Formulation

The traffic flow rate in each lane must be estimated to determine if a lane capacity has been exceeded. This can be accomplished by counting the number of cars passing through

a certain point within a certain amount of time. Our approach is to use inter-vehicle communication. The strategy for flow rate estimation is illustrated in Fig. 5.5. t_1 and t_2 are the time instances that car 1 and car 2 hit point P , respectively. The instantaneous flow rate can be calculated as car 2 passes point P as: $f_{P,n} = \frac{1}{t_2 - t_1}$. The denominator $t_2 - t_1$ is called the time headway for vehicle 2. Since this calculation only considers 2 cars, it will not reflect the flow rate of all cars passing point P . This can be calculated recursively by fusing $f_{P,n}$ with the flow from the last vehicle

$$F_{P,n} = \frac{f_{P,n} + (n-1)F_{P,n-1}}{n} = \lambda f_{P,n} + \mu F_{P,n-1}, \quad (5.1)$$

where n is the number of cars passing through point P ; the factors $\lambda = \frac{1}{n}$ and $\mu = \frac{n-1}{n}$ are considered as weight factors.

Unfortunately, as the number of vehicles grows large, i.e., $n \rightarrow \infty$, Eq. 5.1 will no longer reflect the current flow rate, but a long term average of the cumulative flow in previous time steps. To avoid this, a sliding window that defines an amount of cars stored in the counting buffer can be used. As the application counts the cars, buffer space is freed up to accept more inputs.

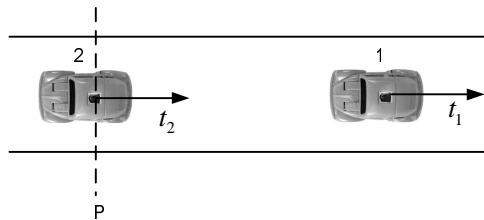


Figure 5.5: Flow rate estimation.

5.6.2 Simulation

To evaluate the algorithm, an application running on top of VISSIM was written in C to estimate the traffic flow rate in each lane, and hence the total flow rate. A map of a four-lane highway was also created. A screen shot of the application is shown in Fig. 5.6. In this figure, the flows of the four-lane highway are computed using Eq. (5.1) at three points: S_1 , S_2 and S_3 ; all are assumed to be the starts of the highway segments.

Figure 5.7 shows the time history of the estimated total flow of the highway at point S_1 . The input traffic volume is 4000 *veh/hr* and the cars are randomly generated by VISSIM. One can see that the estimated flow approaches the true value in approximately 50 seconds.

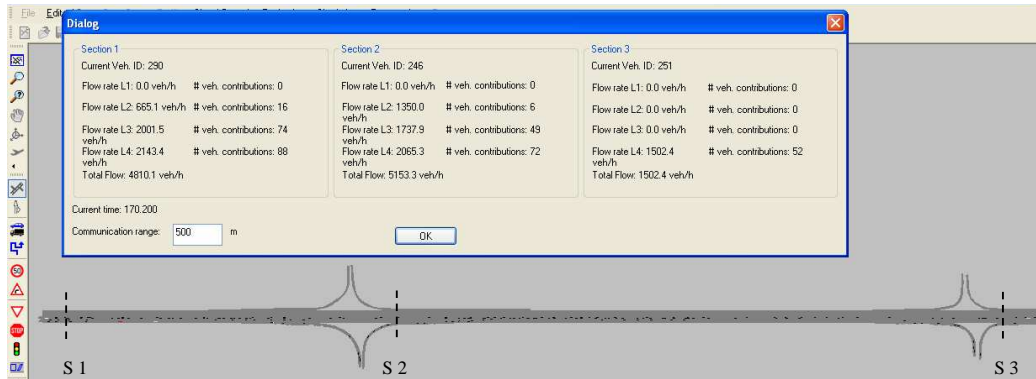


Figure 5.6: A screen shot of lane estimation application.

After that the estimated flow fluctuates around the actual value due to the randomness of the actual input flow.

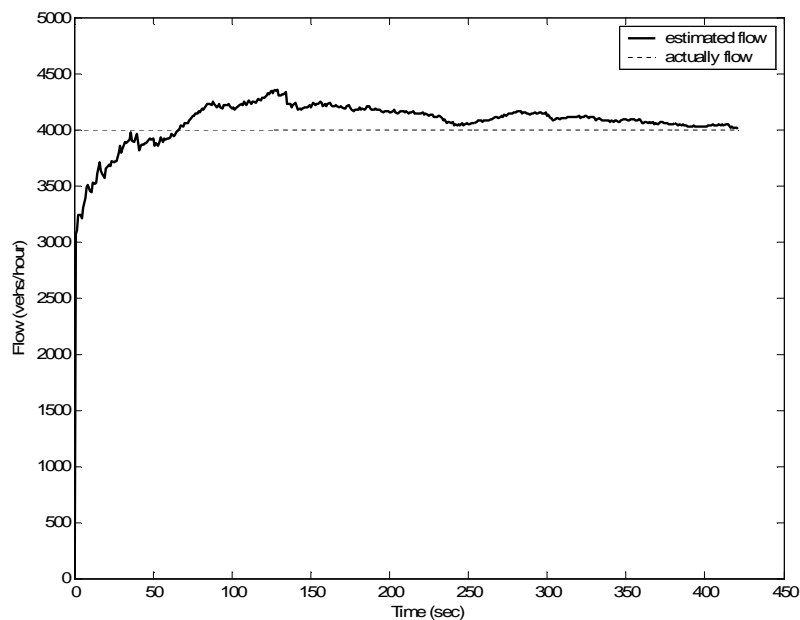


Figure 5.7: Estimated total flow.

5.7 Lane Assignment Optimization

5.7.1 Problem Formulation

Due to the limitation of communication, *i.e.*, it is not possible for vehicles to communicate with all other vehicles on the highway through an ad-hoc wireless network, and

the scalability in planning, it is necessary to pick out a subset of vehicles that are able to talk to one another for cooperative planning. The current approach is to define local clusters of vehicles near the vicinity of road segment boundaries. Here, each local cluster will use inter-vehicle communication to create a plan for optimizing traffic throughput in the upcoming road segment (segment i). The vehicle must be made aware of its entire path either when it enters the highway or must be told which lane to be in every time it moves to a new segment. Now, every time a vehicle enters the highway or hits the next segment, it forms a local area network (LAN) with its immediate neighbors in order to perform a coordinated maneuver. A cluster of vehicles is formed every time a vehicle hits a road segment (illustrated by a circle in Fig. 5.8). This vehicle communicates with all upstream vehicles within its communication range. For illustration, let us denote this local cluster *cluster A*, the vehicles inside this cluster *vehicles A* (shown in a bright color) and the trailing vehicles that are not in the cluster *vehicles B* (shown in a dark color). The lane assignment algorithm can then be implemented either by individual vehicles in cluster *A* or the leading vehicle in cluster *A*.

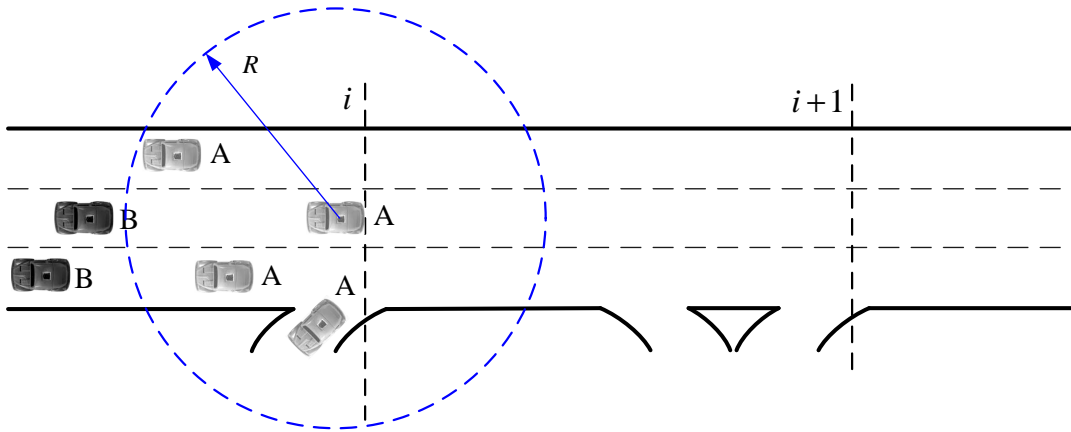


Figure 5.8: Lane assignment strategy.

The lane optimization problem in our work is formulated as follows: Let us consider an n_l -lane highway system as shown in Fig. 5.4 where $i, i+1, i+2, \dots$, indicate the road segment numbers starting at the entry point positions. Let the distance between entry i and exit j be $d_{i,j}$ and the nominal velocity on lane l be v_l . The estimated time that takes one vehicle currently in lane l_c to switch to lane l , then travel from i to exit j is $\frac{d_{i,j}}{v_l} + T_{\Sigma}^{l_c,l}$ where $T_{\Sigma}^{l_c,l}$ is the maneuver cost caused by lane changes. Detailed analysis of the maneuver cost will be discussed in Section 5.7.2. The total travel time for all vehicles in lane l_c starting at segment i is $\sum_{j=i+1}^{n_d} N_{i,j}^{l_c,l} (\frac{d_{i,j}}{v_l} + T_{\Sigma}^{l_c,l})$, where $N_{i,j}^{l_c,l}$ is the number of vehicles currently in lane l_c that will be traveling from the start of segment i to exit j in

lane l , and n_d is the number of exit points. Thus a candidate for the cost function to be minimized is the total travel time, $\sum_{l_c=0}^{n_l} \sum_{l=1}^{n_l} \sum_{j=i+1}^{n_d} N_{i,j}^{l_c,l} (\frac{d_{i,j}}{v_l} + T_{\Sigma}^{l_c,l})$.

The problem is to assign lanes to all vehicles within a local cluster A at the start of each segment. Let $\alpha_{i,j}^{l_c,l}$ be the percentage of vehicles (within cluster A) traveling from i to j that will be sent from lane l_c to lane l . The factor $\alpha_{i,j}^{l_c,l}$ relates the number of vehicles $N_{i,j}^{l_c,l}$ as $N_{i,j}^{l_c,l} = N_{i,j}^{l_c} \alpha_{i,j}^{l_c,l}$. The cost function for segment i becomes

$$\Phi_i = \sum_{l_c=0}^{n_l} \sum_{l=1}^{n_l} \sum_{j=i+1}^{n_d} N_{i,j}^{l_c} (\frac{d_{i,j}}{v_l} + T_{\Sigma}^{l_c,l}) \alpha_{i,j}^{l_c,l}. \quad (5.2)$$

The minimization problem can be cast as a linear programming problem to solve for $\alpha_{i,j}^{l_c,l}$'s with the cost function in Eq. (5.2) subject to the following constraints:

1. Non-negativity: $\alpha_{i,j}^{l_c,l} \geq 0$,
2. Lane capacities should not be exceeded: $\lambda_i^l f(\sum_{l_c=0}^{n_l} \sum_{j=i+1}^{n_d} N_{i,j}^{l_c} \alpha_{i,j}^{l_c,l}) + \mu_i^l \hat{F}_i^l \leq C_{max}^l$, where $f(\cdot) = (\sum_{l_c=0}^{n_l} \sum_{j=i+1}^{n_d} N_{i,j}^{l_c} \alpha_{i,j}^{l_c,l}) / \hat{t}_i^l$ is the instantaneous estimated flow which would be caused by the vehicles routed to lane l , \hat{t}_i^l is the estimated time taking the last vehicle in the group to travel from its current position to segment i , \hat{F}_i^l is the estimated flow in lane l , λ_i^l and μ_i^l are the weight factors (see Section 5.6), and C_{max}^l is the capacity of lane l ,
3. Percentages sum to 1: $\sum_{l=1}^{n_l} \alpha_{i,j}^{l_c,l} = 1$.

One should keep in mind that in all constraint conditions, the superscript l_c takes on values from 0 to n_l , i.e., $0 \leq l_c \leq n_l$; $1 \leq l \leq n_l$; and $i+1 \leq j \leq n_d$. Therefore, there exist $n_l(n_l+1)(n_d-i)$ equations for constraint 1, n_l equations for constraint 2, and $(n_l+1)(n_d-i)$ equations for constraint 3.

To summarize, the constrained minimization problem is of the form

$$\min \mathbf{c}^T \mathbf{x},$$

subject to

$$\mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq}, \quad \text{and } \mathbf{x} \geq \mathbf{0}.$$

The vector $\mathbf{c} \in \Re^{n_l(n_l+1)(n_d-i)}$ has the form

$$\mathbf{c} = \underbrace{\left[N_{i,i+1}^0 \left(\frac{d_{i,i+1}}{v_1} + T_{0,1} \right) \dots N_{i,i+1}^0 \left(\frac{d_{i,i+1}}{v_{n_l}} + T_{0,n_l} \right) \dots \right]}_{n_l \text{ components}} \dots \underbrace{\left[N_{i,n_d}^{n_l} \left(\frac{d_{i,n_d}}{v_1} + T_{n_l,1} \right) \dots N_{i,n_d}^{n_l} \left(\frac{d_{i,n_d}}{v_{n_l}} + T_{n_l,n_l} \right) \right]^T}_{n_l \text{ components}},$$

vector $\mathbf{x} \in \mathfrak{R}^{n_l(n_l+1)(n_d-i)}$ is

$$\mathbf{x} = \underbrace{\left[\alpha_{i,i+1}^{0,1} \dots \alpha_{i,i+1}^{0,n_l} \right]}_{n_l} \dots \underbrace{\left[\alpha_{i,n_d}^{n_l,1} \dots \alpha_{i,n_d}^{n_l,n_l} \right]^T}_{n_l},$$

matrix $\mathbf{A} \in \mathfrak{R}^{n_l \times [n_l(n_l+1)(n_d-i)]}$ has the form

$$\mathbf{A} = \underbrace{\left[\mathbf{A}_{i,i+1}^0 \mid \dots \mid \mathbf{A}_{i,n_d}^0 \right]}_{n_d-i} \dots \underbrace{\left[\mathbf{A}_{i,i+1}^{n_l} \mid \dots \mid \mathbf{A}_{i,n_d}^{n_l} \right]}_{n_d-i},$$

where $\mathbf{A}_{i,j}^{l_c} \in \mathfrak{R}^{n_l \times n_l}$ is

$$\mathbf{A}_{i,j}^{l_c} = \begin{bmatrix} N_{i,j}^{l_c} & & \\ & \ddots & \\ & & N_{i,j}^{l_c} \end{bmatrix},$$

and $\mathbf{A}_{eq} \in \mathfrak{R}^{[(n_l+1)(n_d-i)] \times [n_l(n_l+1)(n_d-i)]}$ is a band matrix

$$\mathbf{A}_{eq} = \begin{bmatrix} \overbrace{1 \dots 1}^{n_l} & & \\ & \ddots & \\ & & \underbrace{1 \dots 1}_{n_l} \end{bmatrix}.$$

Matrices \mathbf{A} and \mathbf{A}_{eq} correspond to constraints 2 and 3 respectively.

Vectors $\mathbf{b} \in \mathfrak{R}^{n_l}$ and $\mathbf{b}_{eq} \in \mathfrak{R}^{n_l(n_l+1)(n_d-i)}$ are

$$\mathbf{b} = \begin{bmatrix} \frac{(C_{max}^1 - \mu_i^1 \hat{F}_i^1) \hat{t}_i^1}{\lambda_i^1} \\ \vdots \\ \frac{(C_{max}^{n_l} - \mu_i^{n_l} \hat{F}_i^{n_l}) \hat{t}_i^{n_l}}{\lambda_i^{n_l}} \end{bmatrix},$$

and

$$\mathbf{b}_{eq} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

This control strategy can be implemented using the Simplex algorithm with the flow rate for each lane being estimated using the strategy given in the Section 5.6. The numbers of vehicles, which are integers, are obtained by rounding $N_{i,j}^{l_c} \alpha_{i,j}^{l_c,l}$ to the nearest integers after solving the optimization problem. For the greater reduction of the travel time, the system tends to assign a faster lane to vehicles which travel longer distances. That is to say, if two vehicles a and b enter the highway at the same node, the vehicle traveling further will travel in a faster lane or the same lane as the one assigned to the vehicle traveling a shorter distance. To put this in an equation, if the lane assigned to vehicle a is $l_a = g(i_a, j_a)$ and that assigned to vehicle b is $l_b = g(i_a, j_b)$, then

$$j_b > j_a \implies l_b \geq l_a. \quad (5.3)$$

where $l(p) > l(q)$ if $v_p > v_q$, g is the assigned lane as a function of a vehicle's entry node i and exit node j . The utility of the above result can be seen from Fig. 5.9. This reduces the number of interaction between longitudinal and lateral traffic flows when vehicles exit the highway. This result is similar to the 'destination monotone policy' described in [83].

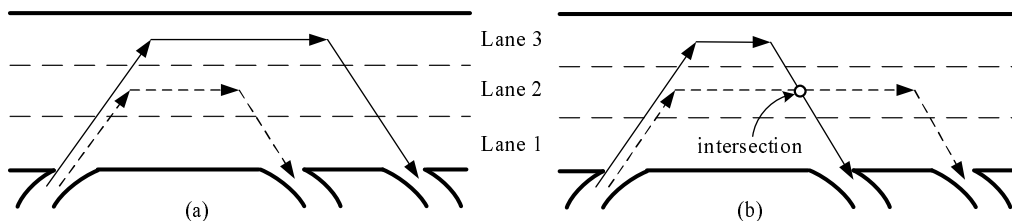


Figure 5.9: Vehicles traveling further tend to travel in a faster lane to reduce maneuver cost and vehicle travel time.

5.7.2 Analysis of Maneuver Cost

It is necessary to specify the maneuver cost $T_{\Sigma}^{l_c,l}$ considering the effect of lane changes between vehicles. Clearly, if lane changes and merges do not affect the longitudinal flow at all, then the maneuver cost is simply the time for the vehicle itself to change to the destination lanes. In fact, the difficulty of the problem comes mainly from the interaction between lane changes and longitudinal flow. A key attribute of any such interaction is whether the speed of traffic is affected. At one extreme, no speed change is necessary to accommodate lane changes, and successfully exiting at the desired exit is made possible primarily by starting the lane change attempts early enough, and merging of a vehicle into the existing traffic stream at on-ramps is performed by having the vehicle wait for a sufficiently large gap. On the other extreme is the policy that, to ensure a 100% exiting

success rate even though the lane-changing vehicles started the lane-changing attempt late, the lane-changing vehicles as well as the traffic on the destination lane must slow down, safety permitting, to accommodate the lane changes. This work deals with the latter extreme. We will identify the equations to represent the microscopic lane-change behaviors as well as the interaction between the lane changers and the longitudinal flow.

One of the possible candidates for $T_{\Sigma}^{lc,l}$ is to consider the time delay during which a vehicle slows down to free space for a vehicle changing into its lane as shown in Fig. 5.10. During this cooperative maneuver, the velocity profile of vehicle 2 is assumed to be as shown in Fig. 5.11.

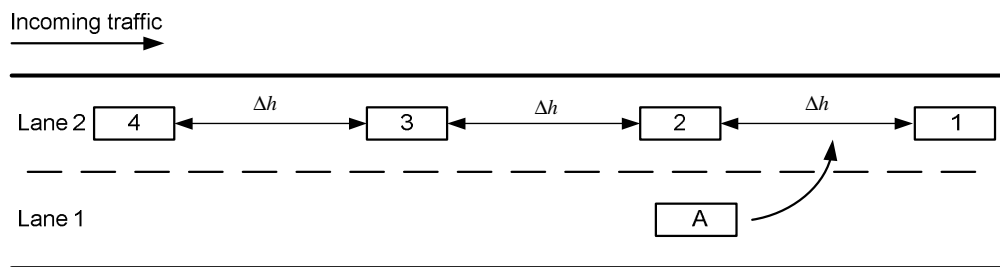


Figure 5.10: Inserting vehicle A into traffic flow with uniform average separation distance Δh between vehicles.

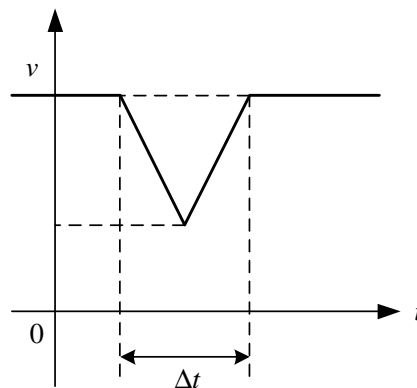


Figure 5.11: Velocity profile for a trailing vehicle in target lane.

To develop an expression for the maneuver cost $T_{\Sigma}^{lc,l}$, let us start with a two-lane lane changing scenario. To simplify the modeling, the following assumptions are made:

1. The vehicles are uniformly distributed along the highway. This is the case for congested highway conditions. The separation distance between the vehicles Δh can be

calculated from traffic density. The distance Δh_{min} , which can be calculated from traffic density at lane capacity, defines the minimum acceptance gap for lane changes. That is, if $\Delta h = \Delta h_{min}$, then the lane change is not allowed since the lane capacity would be exceeded.

2. Let s be the average length of the vehicles in both lanes. If $\Delta h \geq 2\Delta h_{min} + s$ (*i.e.*, the average gap between vehicles in the target lane is large enough to accommodate a lane change without having to force the trailing vehicles to slow down), then no upstream vehicles in the target lane are influenced, *i.e.*, they don't slow down, and vehicle A will be inserted in the middle between vehicle 1 and vehicle 2. If $\Delta h < 2\Delta h_{min} + s$, then the trailing vehicles in the target lane will have to slow down to create a gap large enough (Δh_{min}) for the lane changer A . The scenario is to have vehicle 2 decrease its speed so that the gap between it and vehicle A would be Δh_{min} , then vehicle 3, vehicle 4 and so on until vehicle $n + 1$ upstream.

When a vehicle changes lanes, it causes a speed disturbance to the other vehicles in the lane it moves into. The total time cost Δt needed for the trailing vehicles in the target lane to adjust can be calculated from c_{Σ} , the total relative distance that all the influenced vehicles in lane 2 have to slow down to create a gap for the lane changer A using $\Delta t = \frac{c_{\Sigma}}{v_2}$. To calculate c_{Σ} , we need to quantify how many vehicles are affected by the lane change.

Fig. 5.12 shows lane 2 before and after the lane-change. In the bottom figure, the lane changer A has been inserted into the sequence. Assume that n trailing vehicles in the target lane are affected by the lane change (note that vehicles 1 and 2 are not affected). The distance from the front bumper of vehicle 1 to the front bumper of vehicle $n + 2$ is

$$(n + 1)(\Delta h + s) = (n + 1)(\Delta h_{min} + s) + \Delta \bar{h} + s, \quad (5.4)$$

which gives

$$n = \frac{\Delta h_{min} + s}{\Delta h - \Delta h_{min}} - \frac{\Delta h - \Delta \bar{h}}{\Delta h - \Delta h_{min}}. \quad (5.5)$$

Note that since $\Delta h_{min} \leq \Delta \bar{h} < \Delta h$, we have $\frac{\Delta h_{min} + s}{\Delta h - \Delta h_{min}} - 1 \leq n < \frac{\Delta h_{min} + s}{\Delta h - \Delta h_{min}}$. Considering that n is an integer, we can always determine n from the above equation, which has a single solution.

To determine c_{Σ} , we need to calculate the distance each vehicle has to create to accommodate a lane change. Let c_1, c_2, \dots, c_n be the distances vehicle 2, 3, ..., $n + 1$ have to slow down, respectively. Referring to Fig. 5.13, c_1 through c_n can be determined as

$$c_1 = 2\Delta h_{min} - \Delta h + s$$

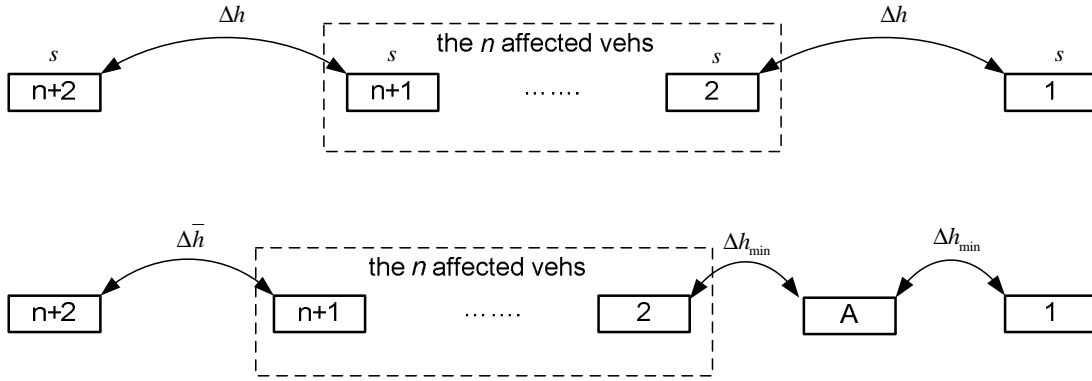


Figure 5.12: Lane 2 before and after lane-change.

$$c_2 = 3\Delta h_{min} - 2\Delta h + s$$

...

and in general

$$c_n = (n + 1)\Delta h_{min} - n\Delta h + s. \tag{5.6}$$

c_Σ is the sum of c_1 through c_n , which is

$$c_\Sigma = \sum_{i=1}^n [(i + 1)\Delta h_{min} - i\Delta h + s]. \tag{5.7}$$

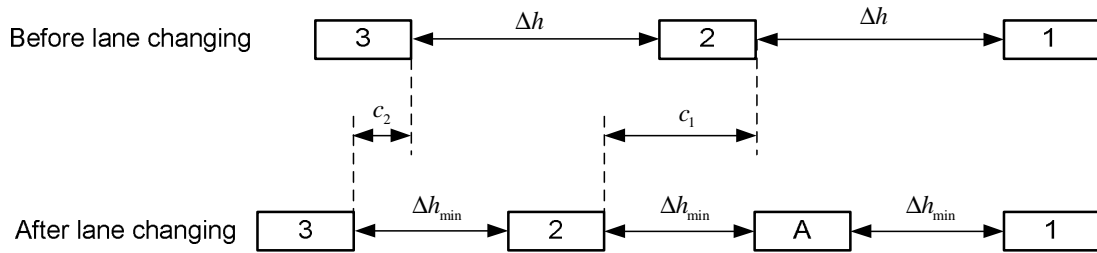


Figure 5.13: Determine adjustment spacing.

The total maneuver cost will be the sum of Δt and the time for vehicle A to move from lane 1 to lane 2

$$T_\Sigma^{1,2} = \Delta t + \tau, \tag{5.8}$$

where τ is the lane change time constant, *i.e.*, time that takes the lane changer to complete its lane change.

Given an average vehicle length $s = 4.5 \text{ m}$ and lane capacity $C_{max} = 2200 \text{ veh/hr}$, the numbers of affected vehicles and total time cost Δt due to a lane change for different average speeds are plotted in Fig. 5.14 and Fig. 5.15, respectively.

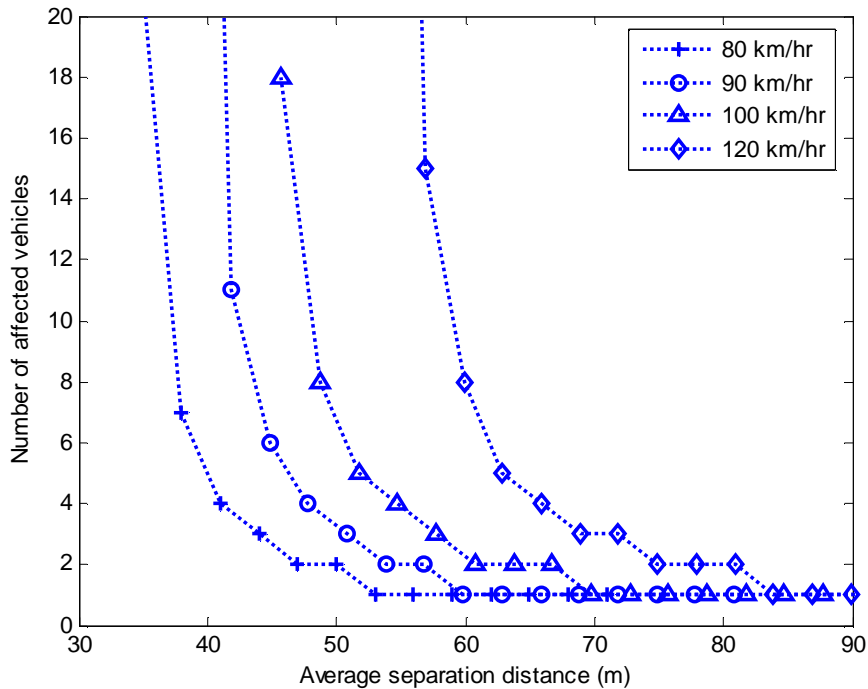


Figure 5.14: Number of affected vehicles.

Equation 5.9 can be generalized for the case of a vehicle A moving from lane l_c to lane l . Note that vehicle A , in order to reach its target lane, causes the delay Δt_{en} in all the intermediate lanes between l_c and l when it enters lane l and Δt_{ext} in all the intermediate lanes between l and the off-ramp (lane 0) when it exits the highway. Therefore the total maneuver cost is

$$T_{\Sigma}^{l_c, l} = \Delta t_{en} + \Delta t_{ext} + |l - l_c| \tau + l \tau. \quad (5.9)$$

5.7.3 Van Aerde's Car-Following Model

The total travel time can be better estimated by taking the speed-flow-density relationship into account. To formulate Eq. (5.2), the vehicle speeds are assumed to be some nominal

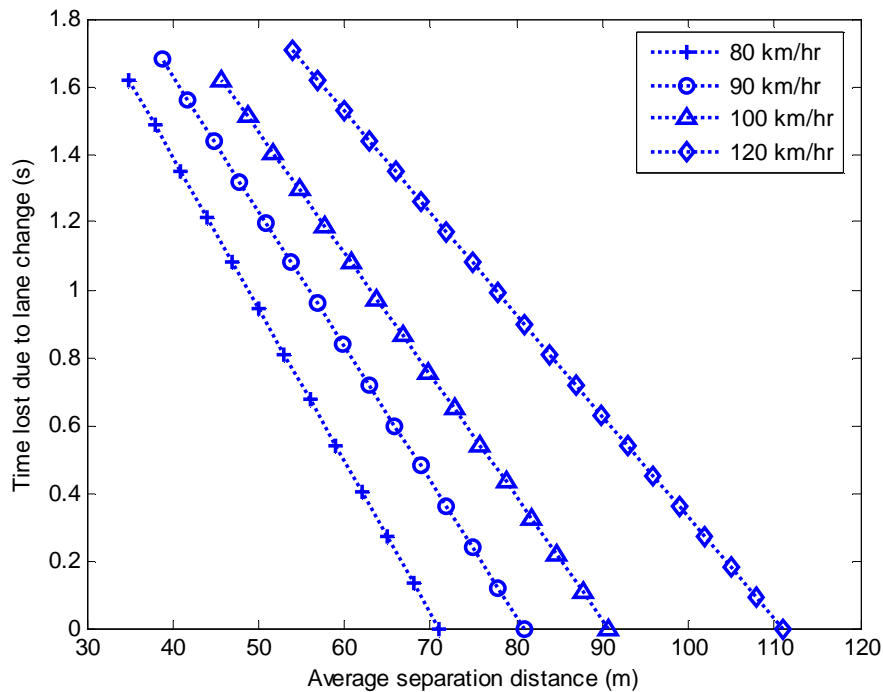


Figure 5.15: Time cost versus average separation distance.

lane velocities. However, on an actual highway, the vehicle speed is dependent on the traffic density (and hence the flow rate).

In this work, Van Aerde's model [84] is adopted and implemented in the traffic simulator to model the speed-flow-density relationship. This integration model uses a steady-state car-following model proposed by Van Aerde and Rakha, which combines the Pipes and Greenshields models [85] into a single-regime model. The model, which requires three input parameters, can be calibrated using field loop detector data. The efforts for calibrating Van Aerde's model were described in [84]. The Van Aerde single-regime model overcomes the shortcomings of the Greenshields and Pipes models which are often inconsistent with field data from a variety of highways [85].

Van Aerde's model can be described by a series of expressions as follows:

$$\delta = \frac{1}{c_1 + \frac{c_2}{v_f - v} + c_3 v}, \quad (5.10a)$$

$$m = \frac{2v_c - v_f}{(v_f - v_c)^2}, \quad (5.10b)$$

$$c_2 = \frac{1}{\delta_j \left(m + \frac{1}{v_f}\right)}, \quad (5.10c)$$

$$c_1 = mc_2, \quad (5.10d)$$

$$c_3 = \frac{-c_1 + \frac{v_c}{C_{max}} - \frac{c_2}{v_f - v_c}}{v_c}, \quad (5.10e)$$

where:

δ = traffic density (*veh/km*) or the inverse of the vehicle headway (*km/veh*),

v = vehicle speed (*km/hr*),

v_f = free-speed (*km/hr*),

v_c = speed at capacity (*km/hr*),

δ_j = jam density (*veh/km*),

c_1 = fixed distance headway constant (*km*),

c_2 = first variable headway constant (*km²/hr*),

c_3 = second variable headway constant (*hr⁻¹*).

In practice, the calibration of the car-following model requires the estimation of three parameters: v_f , v_c , and δ_j . The vehicle speed can be inferred from traffic density δ and flow rate F using the fundamental speed-flow-density relationship

$$v = \frac{F}{\delta}. \quad (5.11)$$

From Eq. (5.10a), the traffic density δ can be expressed as a function of flow by replacing v with Eq. (5.11). This gives the quadratic equation

$$\delta^2(c_1v_f + c_2) + \delta(-c_1F + c_3Fv_f - v_f) + (F - c_3F^2) = 0. \quad (5.12)$$

The estimated vehicle speed v_l in Eq. (5.2) (represented by v in Eqs. (5.10)-(5.11)) is a function of flow and can be obtained by solving Eq. (5.12) and substituting the resulting δ into Eq. (5.11). In summary, the procedure is to measure the flow rate F for each lane, then solve for v and use it in the cost function in Eq. (5.2) instead of the nominal lane velocities.

5.8 Simulation

The VISSIM [6] software package was used to simulate the proposed strategy. The lane routing simulator module works in parallel with the VISSIM simulator engine by creating

multiple threads for handling various tasks such as inter-vehicle communication, vehicle localization and GPS coordinate reading. In all simulations, the GPS data are modeled by adding a random walk bias of maximum amplitude $3 m$ and Gaussian noise having a standard deviation of $0.5 m$ to the position of each vehicle. The communication range used in simulations is $300 m$ which is consistent with the IEEE 802.11p standard for inter-vehicle communication [7, 8]. A description of the software architecture used in this research is provided in Section 2.3.

Based on the lane flows estimation strategy in Section 5.6 and the minimization of the cost function in Eq. (5.2), it is possible to assign appropriate lanes to vehicles. Up to this point, the system assigns lanes to vehicles once they come within the vicinity of a road segment. The average travel time of all the vehicles on the highway is also calculated to evaluate the effectiveness of the algorithm.

The four-lane highway used in the simulation has five entry and five exit points. The length of the highway is $10 km$. The capacity for each of the four lanes is $2200 veh/hr$. The highway starts with zero traffic and the vehicles are generated randomly by VISSIM. The speed-flow relationships for the four lanes of the highway are shown in Fig. 5.16, which are similar to the calibration results from a real freeway described in [84, 85, 86]. The free-speeds for lane 1 through lane 4 are $80 km/hr$, $90 km/hr$, $100 km/hr$ and $120 km/hr$, respectively. The speeds at capacity v_c in the four lanes are 81% of the free-speeds [84, 85, 86]. To ensure the consistency between simulations with and without lane assignment, the simulation parameters such as the parameters of the car-following models (calibrated parameters from [85, 86]), vehicle parameters, lane speeds, etc., are similar for all simulations. The main difference between the two strategies is that vehicles' lane changing behavior is set to 'free lane selection' (vehicles can overtake any lanes) mode in simulations without lane assignment (VISSIM-controlled simulations). In this case, vehicles change lanes whenever they need more room or higher speed. The comparison between VISSIM-controlled simulations and lane assignment simulations is summarized in Table 5.1. Other VISSIM parameters are listed in Appendix B.

When a vehicle enters a road segment, it communicates with other vehicles to plan for optimizing traffic throughput in the current road segment. The proposed strategy appears to be effective in simulations as shown in Fig. 5.17. These figures show the average travel time for 1000 vehicles on the highway with different total input flows and the lane change time constant τ of $5 s$. The total flows at the inputs in the three simulations are $2000 veh/hr$, $4000 veh/hr$ and $6000 veh/hr$, respectively. The simulation starts with no vehicles on the highway and approaches steady-state after approximately 5 minutes of simulated time (once the first cars have covered the $10-km$ stretch of highway and exit the model).

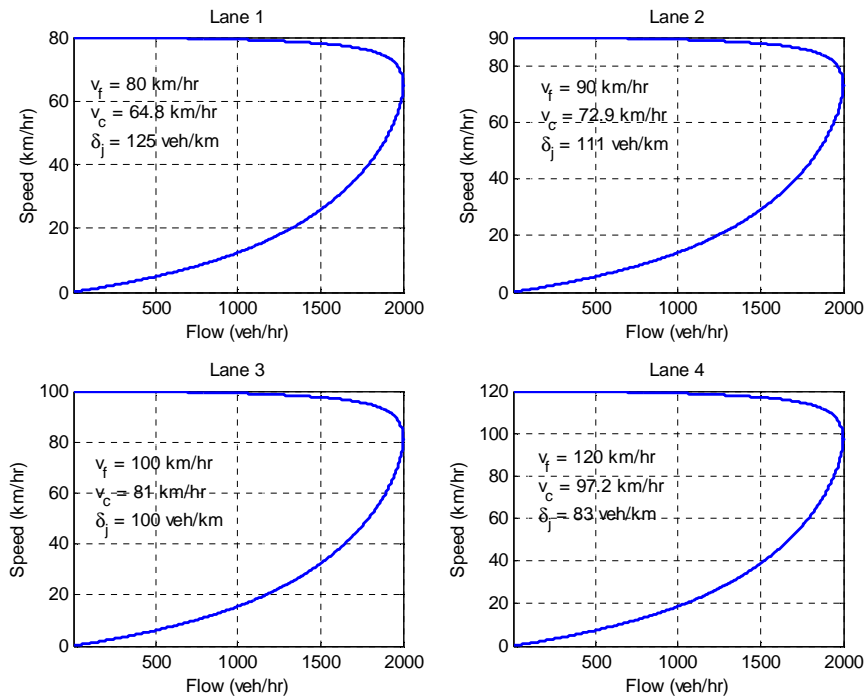


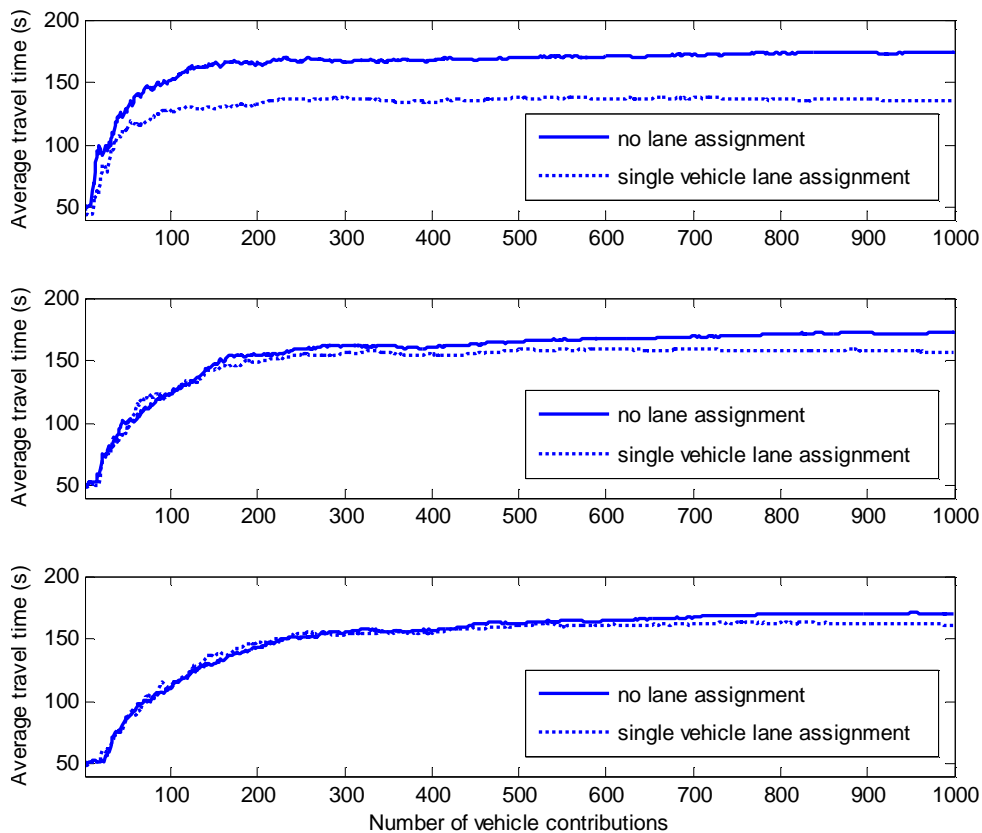
Figure 5.16: Speed-flow relationships for simulations.

The first simulation (Fig. 5.17 (top)) shows a large gap (and hence a big improvement) between the average traveling times with and without the implementation of the lane assignment, while the gaps are smaller in the next two simulations (Fig. 5.17 (middle and bottom)). This can be explained by the fact that when the traffic density increases, it is harder to find empty slots in the fast lanes which are filled up very quickly. The trend of increased average travel time as the input flow grows from 1000 *veh/hr* to 6500 *veh/hr* is shown in Fig. 5.18. It can be seen from Fig. 5.18 that the increasing rate of the average travel time is slow when the input volume is smaller than 2200 *veh/hr* since the fastest lane (lane 4) can assimilate most of the flow. The travel time in this case is only influenced by the traffic density in lane 4. As the input flow increases over 2200 *veh/hr*, the average travel time rises steadily as the system starts routing vehicles to slower lanes.

The probability distributions of lane-change maneuver cost are calculated and plotted in Fig. 5.19. The mean value and the standard deviation of lane-change penalty times are plotted against 21 input traffic flows in Fig. 5.20. As shown in Fig. 5.20, the average lane-change maneuver cost and the standard deviation increase with the input flow. At high flow rates, the average grows at a faster rate than the flow. This demonstrates the trade-off between the longitudinal flow and the lateral flow. From Fig. 5.19 and Fig. 5.20, it can be seen that at high longitudinal flows, both the average and the standard deviation are

Table 5.1: Comparison between VISSIM-controlled simulations and lane assignment simulations.

Type	VISSIM-controlled	Lane assignment
Vehicle speed	Vehicles are assigned a desired lane speed when they enter a lane but speed distributions are controlled by VISSIM.	Vehicles are forced to travel at lane speeds.
Lane changing behavior	Vehicles change lanes when they need more room or higher speed.	Controlled from external model.
Car-following model	Van Aerde's model.	Wiedemann's model.

Figure 5.17: Improvement over average travel time for 1000 cars with the input flow being 2000 *veh/hr* (top), 4000 *veh/hr* (middle), and 6000 *veh/hr* (bottom).

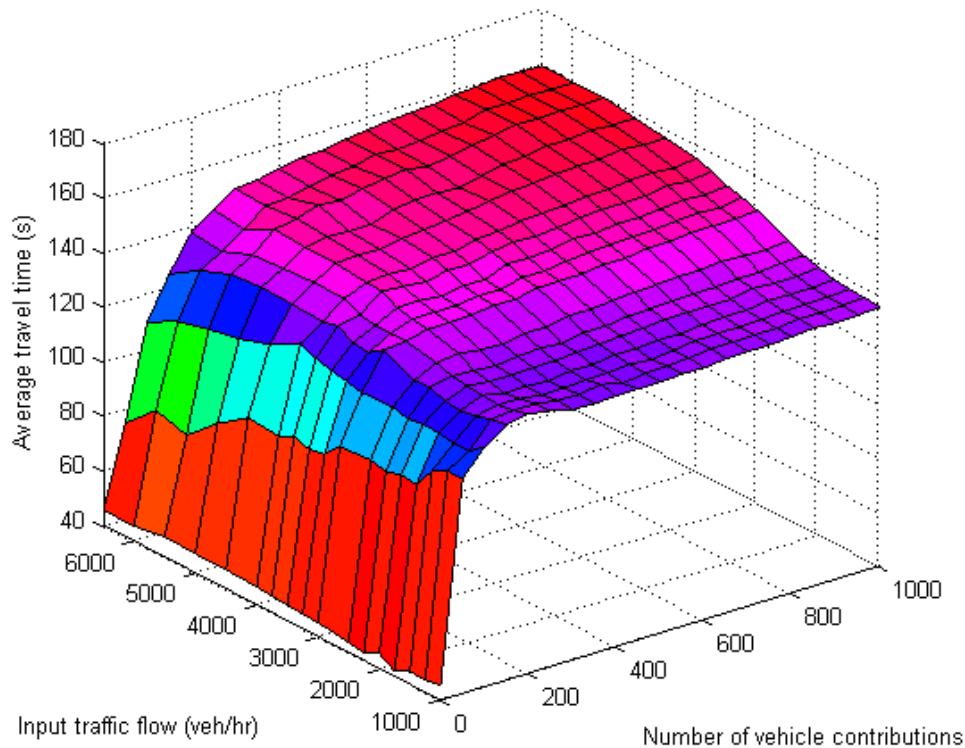


Figure 5.18: Trend of increased average travel time as input flow increases from 1000 *veh/hr* to 6500 *veh/hr*.

large due to high density in all four lanes. The standard deviation, however, is very small at low traffic conditions since the high-speed lane can accommodate all the longitudinal flow making the variance in gap sizes much smaller than at high traffic conditions where the vehicles spread all over the four lanes having very different traffic densities.

5.9 Selfish Lane Selection Versus System Optimality

In this chapter, the problem of lane routing to optimize the performance of a congested highway is considered. Given a network with associated traffic rates between each origin-destination pair, the objective is to route traffic such that the sum of all travel times is minimized.

However, in many situations, it may be difficult or impossible to regulate network traffic so as to implement an optimal assignment of lanes. In the absence of regulation by some central authority, some vehicles might route their traffic on the minimum cost available to

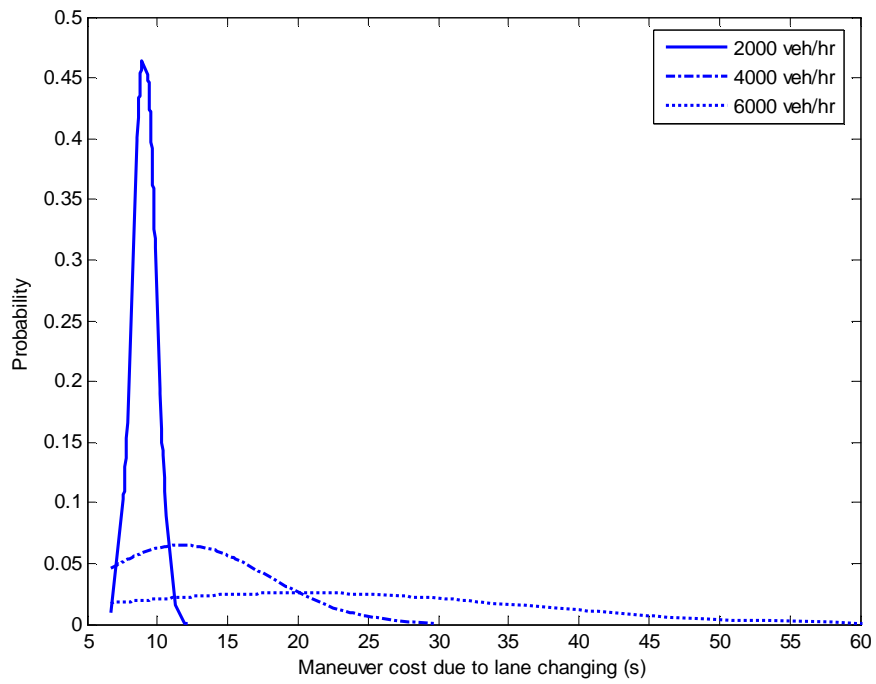


Figure 5.19: Probability distributions of maneuver cost due to lane-change.

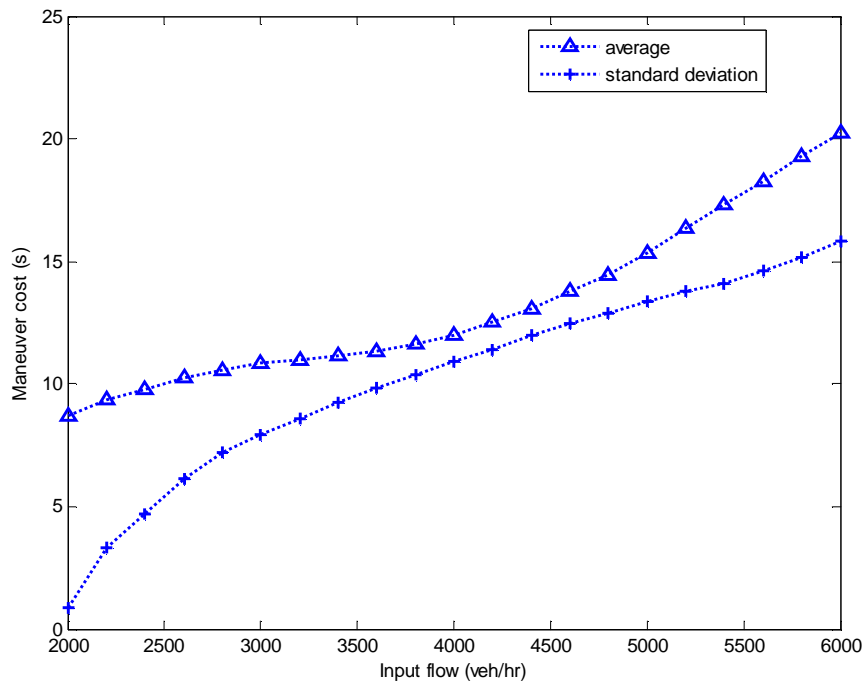


Figure 5.20: Average and standard deviation of maneuver cost due to lane-change.

them. In general such a ‘selfishly motivated’ assignment of traffic will not minimize the total cost.

Detailed analysis of such a system can be modeled using Nash equilibrium in classical game theory as proposed in [87] and will be considered in future work. Within the framework of this work, VISSIM was used to implement the simulations and show the effect of selfish lane selection. Vehicles that do not comply with the suggested lane routing decision are set to “free lane selection” mode in VISSIM, *i.e.*, a vehicle switches lanes to satisfy its own benefit. Fig. 5.21 shows the simulation results where the percentage of vehicles that do not comply with the suggested optimal lane assignment varies from 0% to 100% under different input traffic flows. Fig. 5.22 plots the steady-state average travel time versus percentage of non-participating vehicles under different input flow rates. The three dotted lines in Fig. 5.22 are cubic-spline smoothed fits for the three data groups. As could be expected, these figures show a steady rise in the average travel time as the number of non-participating vehicles increases.

5.10 Conclusions

We have proposed a system for optimization of lane assignment with the objective of increasing traffic throughput. This system is unique in that it uses GPS as the only form of sensing, which makes the system simpler in terms of hardware management and thus less costly. We have developed a linear model for lane assignment and presented results on the use of a linear programming algorithm in the solution of the problem. Simulation results demonstrate that intelligent lane selection can improve highway capacity.

The potential future direction for this work will be on the effect of selfish lane selection where a portion of the drivers do not comply with the suggested lane assignment strategy to help understand the underlying structure of such decision making situations. Such a system can be modeled using a non-cooperative game theory.

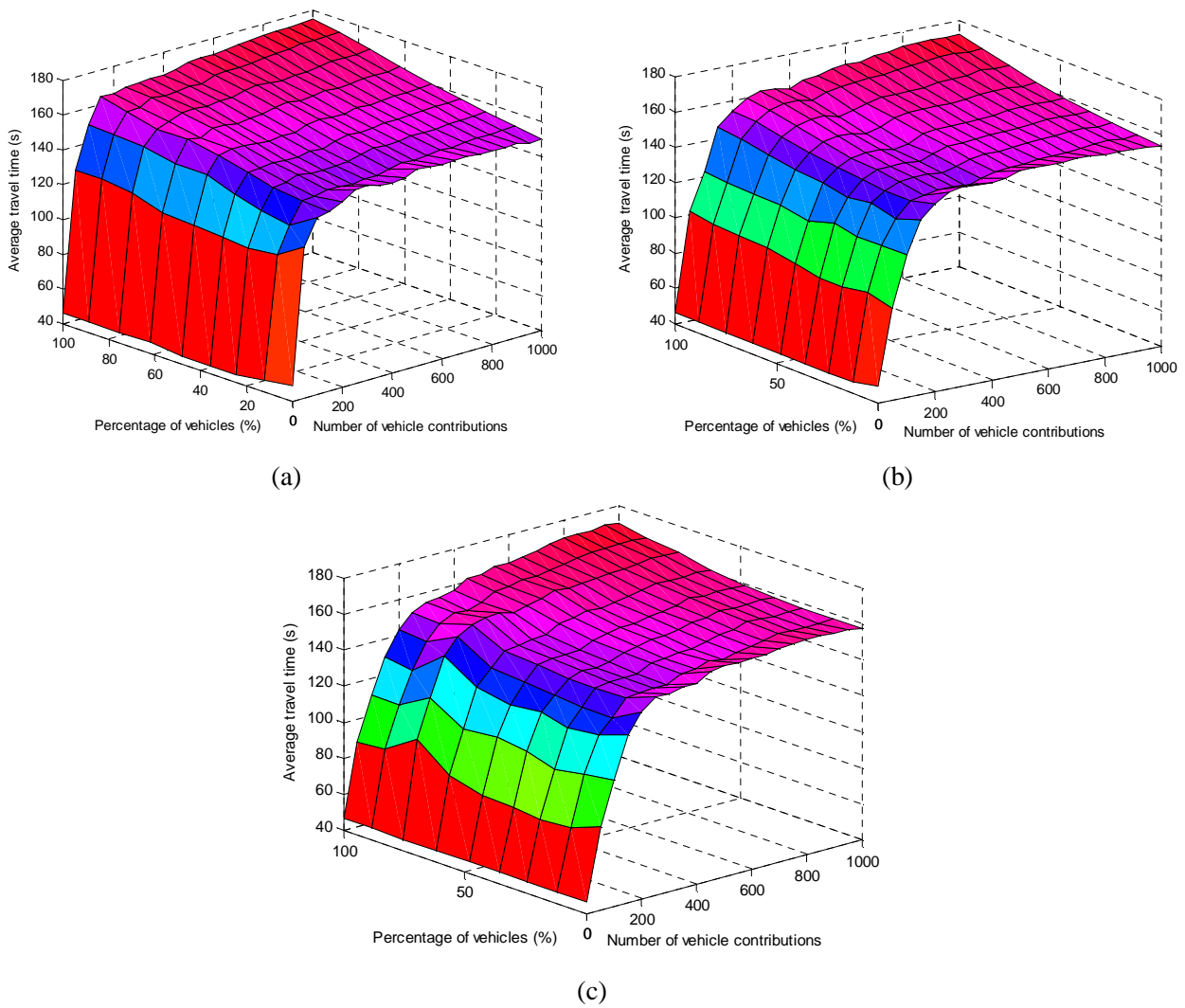


Figure 5.21: Percentage of vehicles not complying to suggested lane assignment strategy with the input flow being (a) 2000 *veh/hr*, (b) 4000 *veh/hr*, and (c) 6000 *veh/hr*.

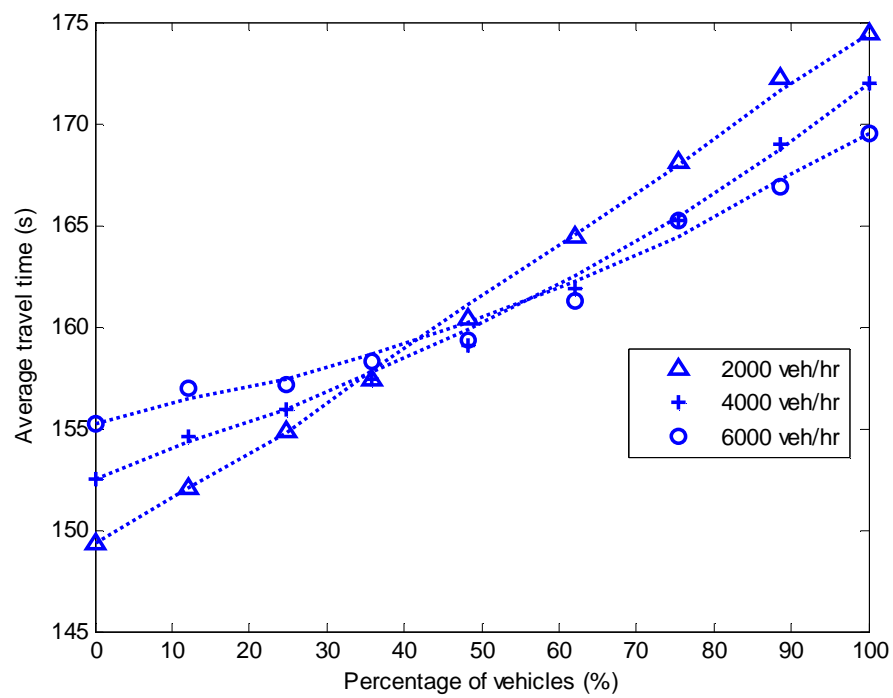


Figure 5.22: Steady-state average travel time versus percentage of non-participating vehicles under different input flows.

Chapter 6

Intelligent Platoon Formation

6.1 Introduction

This chapter targets the problem of traffic management using a ‘platoon’ concept with the objective of developing scheduling and control techniques to support autonomous driving on urban multi-lane highways with multiple entry (on-ramp) and exit (off-ramp) points. The concept behind platooning is to have vehicles in groups traveling in tight vehicle-string formations where the inter-vehicle spaces are on the order of 1-5 m (see Fig. 6.1). The group is controlled by a leader and multiple platoons would be separated by larger distances for a greater degree of safety [88, 89, 90]. A prototype of this concept was demonstrated during Demo’97 in San Diego on a reserved 7-mile highway lane guided by magnets embedded in the roadway [91, 103].

The idea of ‘platooning’ is to abbreviate the limitation of capacity and safety that can be achieved by road vehicles. The effect of platooning on capacity, which relates the capacity C_{max} in $veh/lane/hr$ to the steady state speed v in km/hr , the separation time between platoons t_h in seconds, the intra-platoon distance headway h in m , the average vehicle length s in m , and the maximum platoon size Υ is described by (see Appendix C):

$$C_{max} = \frac{3600\Upsilon v}{3.6[(\Upsilon - 1)h + s] + t_h v}. \quad (6.1)$$

For example, the lane capacity can be increased up to 7,600 vehicles per lane per hour when the maximum platoon size of 25 vehicles is used given that other parameters are: $h = 10 m$, $s = 4.5 m$, $v = 100 km/hr$, $t_h = 3$ seconds. With platooning, several-fold increase in roadway capacity can be achieved with minimal upgrades to infrastructure and relatively little public expense.

Under current highway driving conditions (without platooning), it is apparent that as the speed and density of vehicles increase, the likelihood and likely severity of the crashes will increase. The limitations of drivers are the primary causes. Driver errors are responsible for 90% of crashes that occur today, and the limited ability of drivers to follow other vehicles produces the limitation on lane capacity [93]. The limitation of drivers' ability to perceive changes in vehicle spacing, relative motion, and acceleration and their limited speed and precision of response ensure that lane capacity cannot generally exceed 2,200 vehicles per hour under manual control [93] or 2,400 vehicles per hour for a freeway with high-quality geometry [94]. In order to increase lane capacity, it is necessary to organize vehicles in platoons where vehicles are at closer average spacing (for the same speed). The platoon mode of operation was conceived as a way of abbreviating the limitation of capacity and safety that can be achieved by road vehicles.

Another reason for platooning is to reduce fuel consumption. It was shown in [95, 96] that platooning operation can save fuel consumption by reducing aerodynamic drag exerted on the trailing vehicles. In this work, the efforts to measure aerodynamic drag and fuel consumption on each vehicle in the platoon were conducted by Browand and Michaelian in both laboratory and real road tests as shown in Fig. 6.1. It was shown in this study that at 4-meter spacing and a travel speed of 96 *km/hr*, platooning can save between 5% and 12% in fuel, depending on whether the vehicle is a platoon leader, trailing or interior vehicle. The interior vehicles have the least fuel consumption and thus benefit the most from close-following. Trailing vehicles are intermediate in benefit, and leading vehicles benefit least.



Figure 6.1: (a) Three-Lumina platoon in the Dryden wind tunnel at USC, and (b) platoon operation at 4-meter spacing to measure aerodynamic drag and fuel consumption [95].

The benefits of platooning can be achieved by forming platoons at reasonably large sizes (five or more vehicles). It is also desirable to ensure that platoons remain intact for considerable distances [4].

6.2 Platoon Control Diagram

The control diagram for platooning used in this work is shown in Fig. 6.2. Each vehicle is assumed to be equipped with a GPS receiver and a processor to implement the lane-positioning algorithm and to communicate GPS data as well as other platoon information with other vehicles across an ad-hoc network.

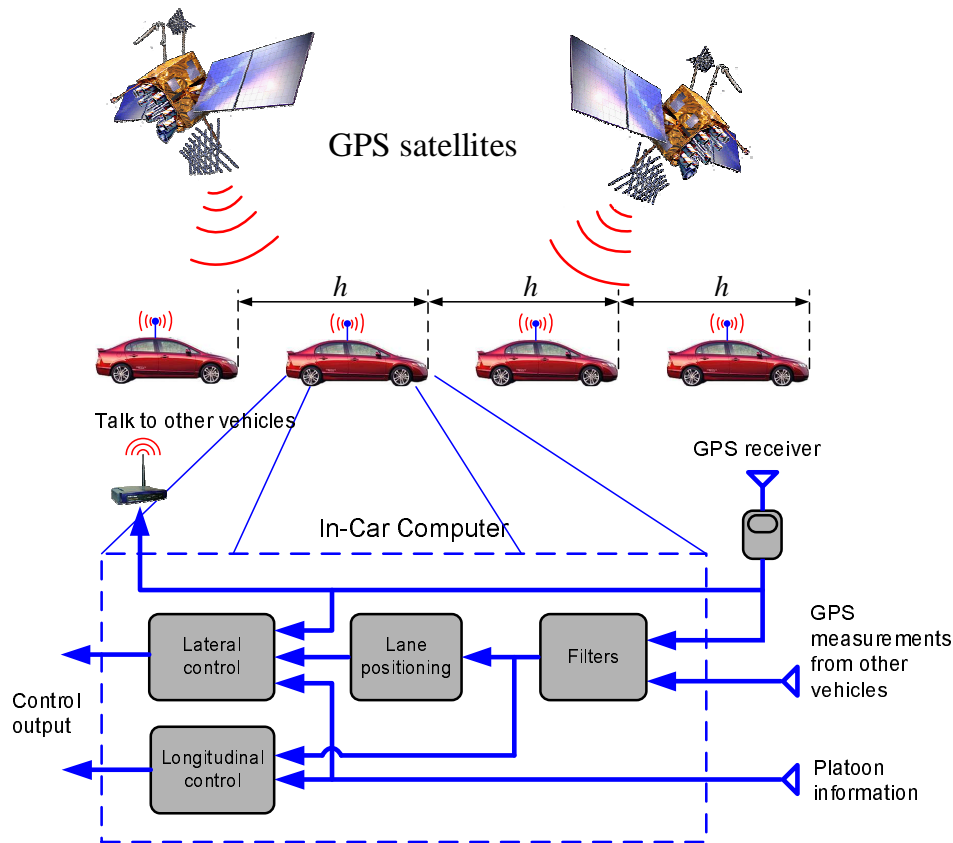


Figure 6.2: Control diagram for platoon assignment.

The most crucial capability for a vehicle to operate in any collaborative driving system is to be able to guarantee the following of a vehicle ahead both longitudinally and laterally. Once this basic skill is achieved, higher order commands with the aid of inter-vehicle communication can be issued to the vehicle to space the vehicles in a single lane formation, or

to enter/exit other platoons in other lanes. The lane positioning system in Chapter 4 works as a low-level input for platoon assignment. The GPS is again used for determining vehicle position in a global coordinate system. The error of filtered relative GPS measurements is in the order of 0.1 - 0.3 m and is good enough for platooning operation.

The lateral controller is necessary for each vehicle in platoon for lane following and a longitudinal controller gives a vehicle in a platoon the capability of maintaining an intra-platoon distance h between the vehicle in front of it if it is a follower or a distance between the front platoon if the vehicle is a leader. This topic has been extensively studied and demonstrated in the literature through the use of conventional control strategies to effectively achieve lateral control [97, 98, 99], longitudinal control [100, 101], and combined lateral and longitudinal control [102, 103].

Other platoon information a follower receives from other members includes: index numbers of platoon members, platoon split and platoon merge commands from the leader, platoon disband command from the leader, etc., while for a platoon leader, the information includes: index numbers of platoon members, platoon merger and split requests from other members and free-agent vehicles, destinations of platoon members, etc. A platoon leader must also broadcast commands such as merge, split, and disband to the following members.

6.3 Strategy for Platoon Formation at Entrances

In this section, a direction for forming platoons at highway entrances is proposed with an aim of increasing lane capacity and therefore enhancing traffic throughput. The platoon assignment will be formulated here as linear programming problems that can be solved using the Simplex algorithm.

The system works as follows: (1) When a vehicle hits an entry point as it enters the highway, it communicates with all the existing platoons (coming from highway upstream) and incoming free-agent vehicles (vehicles, in the entrance ramp, not assigned to any platoon yet) within its communication range R . A platoon is considered to be within communication range if its leader is in the communication range. (2) The platoon assignment is executed and decides which incoming vehicles go to which platoons based on the sharing information among the platoons and vehicles in the communication group. (3) Vehicles, now no longer are free-agent, are maneuvered to assigned platoons leaving the entrance ramp for the next group of incoming free-agent vehicles. A vehicle, which has been assigned to a platoon, will not call its planner again even if it hits the entrance point. Fig. 6.3 depicts the concept of this communication group in which vehicles and platoons within R are shown in a dark color.

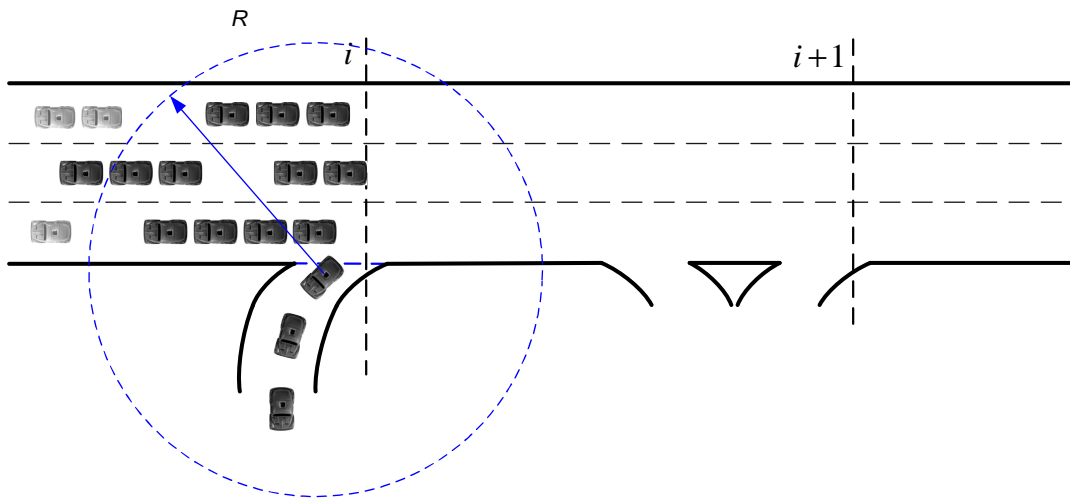


Figure 6.3: Platoon assignment strategy.

For assigning the incoming vehicles to appropriate platoons, the following policies are made to ensure the intactness of platoons [4]:

1. Once a vehicle has been assigned a platoon, it stays with the hosted platoon for its entire trip until it exits the highway,
2. Platoons are constrained to have the maximum platoon size Υ ,
3. The difference in index between the nearest and the furthest destinations of vehicles in a platoon cannot exceed a maximum number r , called the ‘range of destinations’ of a platoon,
4. Vehicles in platoons are sorted front to back in the order non-increasing destination so that the rest of platoon members remain intact after some vehicles split off. The meaning of this policy is that in a platoon the leader has the furthest destination and the last vehicle in the group has the nearest destination. This allows the same vehicle to remain as platoon leader through the platoon’s lifetime, while the platoon ‘drops off’ vehicles that have closer destinations. This also gives a platoon the flexibility of having a greater range of destinations, as long as the range does not exceed r ,
5. If an incoming vehicle cannot find a feasible platoon to join (*i.e.*, satisfying the range r and the maximum platoon size Υ), it initiates a new platoon.

6.4 Problem Formulation

This section develops a strategy for organizing vehicles arriving at entrance ramps into platoons, with the objective of maximizing the distance that platoons stay intact. Essentially, this entails grouping vehicles according to their destination.

Vehicles are appended to existing platoons at the beginning of each road segment i on the basis of their destinations. A vehicle enters the highway, is adjoined to an existing platoon coming from highway upstream and remains intact until it reaches its destination. At this point, the vehicle separates from the host platoon and travels to its exit.

Similar to lane assignment in Chapter 5, denote the distance traveled by a vehicle from entrance i to exit j as $d_{i,j}$. Let the distance that the last vehicle of platoon p travels from entry i to its destination be $\delta_{i,p}^l$, respectively. To satisfy policy 4 and to maximize the distance a vehicle at an entrance stays intact with platoon p , the goal is to determine platoon p so that $\delta_{i,p}^l - d_{i,j} \geq 0$ is minimized. Let $\chi_{i,j}^p$ be the number of vehicles at the entrance ramp traveling from i to j that will be sent to platoon p , and n_p be the number of platoons within the range R . A candidate for the cost function to be minimized is:

$$\Theta_i = \sum_{p=1}^{n_p} \sum_{j=i+1}^{n_d} (\delta_{i,p}^l - d_{i,j}) \chi_{i,j}^p. \quad (6.2)$$

Denote $\beta_{i,j}^p$ as the percentage of vehicles entering the highway and traveling from i to j . The factor $\beta_{i,j}^p$ relates the number of vehicles $\chi_{i,j}^p$ as $\chi_{i,j}^p = \chi_{i,j} \beta_{i,j}^p$ where $\chi_{i,j}$ is the number of vehicles within R traveling from i to j . Equation (6.2) therefore can be rewritten as

$$\Theta_i = \sum_{p=1}^{n_p} \sum_{j=i+1}^{n_d} \chi_{i,j} (\delta_{i,p}^l - d_{i,j}) \beta_{i,j}^p. \quad (6.3)$$

The minimization problem can again be cast as a linear programming problem to solve for $\beta_{i,j}^p$'s with the cost function in Eq. (6.3) subject to the following constraints:

1. Non-negativity: $\beta_{i,j}^p \geq 0$,
2. Maximum platoon size: $\varepsilon_p + \sum_{j=i+1}^{n_d} \chi_{i,j} \beta_{i,j}^p \leq \Upsilon$, where ε_p is the current size of platoon p , and Υ is the maximum platoon size (the number of vehicles a platoon can accommodate),
3. Percentages sum to 1: $\sum_{p=1}^{n_p} \beta_{i,j}^p = 1$.

To satisfy policies 3 and 4, it is also required that $\beta_{i,j}^p = 0$ (constraint 4) if either $des_p^f - j > r$ or $\delta_{i,p}^l - d_{i,j} < 0$ where des_p^f is the destination index of the leader of platoon p .

To summarize, the constrained minimization problem is of the form

$$\min \mathbf{c}^T \mathbf{x},$$

subject to

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, \quad \text{and } \mathbf{x} \geq \mathbf{0}.$$

The vector $\mathbf{c} \in \Re^{n_p(n_d-i)}$ has the form

$$\mathbf{c} = \underbrace{[\chi_{i,i+1}(\delta_{i,1}^l - d_{i,i+1}) \cdots \chi_{i,i+1}(\delta_{i,n_p}^l - d_{i,i+1}) \cdots]}_{n_p \text{ components}} \cdots \underbrace{[\chi_{i,n_d}(\delta_{i,1}^l - d_{i,n_d}) \cdots \chi_{i,n_d}(\delta_{i,n_p}^l - d_{i,n_d})]}_{n_p \text{ components}}]^T,$$

vector $\mathbf{x} \in \Re^{n_p(n_d-i)}$ is

$$\mathbf{x} = \underbrace{[\beta_{i,i+1}^1 \cdots \beta_{i,i+1}^{n_p}]}_{n_p} \cdots \underbrace{[\beta_{i,n_d}^1 \cdots \beta_{i,n_d}^{n_p}]}_{n_p}]^T,$$

matrix $\mathbf{A} \in \Re^{n_p \times n_p(n_d-i)}$ has the form

$$\mathbf{A} = [\mathbf{A}_{i,i+1} | \mathbf{A}_{i,i+2} | \cdots | \mathbf{A}_{i,n_d}],$$

where matrix $\mathbf{A}_{i,j} \in \Re^{n_p \times n_p}$ is

$$\mathbf{A}_{i,j} = \begin{bmatrix} \chi_{i,j} & & \\ & \ddots & \\ & & \chi_{i,j} \end{bmatrix},$$

and $\mathbf{A}_{eq} \in \Re^{(n_d-i) \times n_p(n_d-i)}$ is a band matrix

$$\mathbf{A}_{eq} = \begin{bmatrix} \overbrace{1 \cdots 1}^{n_p} & & \\ & \ddots & \\ & & \overbrace{1 \cdots 1}^{n_p} \end{bmatrix}.$$

Vector $\mathbf{b} \in \mathfrak{R}^{n_p}$ is

$$\mathbf{b} = \begin{bmatrix} \Upsilon - \varepsilon_1 \\ \vdots \\ \Upsilon - \varepsilon_{n_p} \end{bmatrix},$$

and vector $\mathbf{b}_{eq} \in \mathfrak{R}^{n_p(n_d-i)}$ is

$$\mathbf{b}_{eq} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

The idea is to assign a vehicle to a platoon that has the closest destinations to that of the vehicle. To illustrate this concept, suppose that one platoon currently has destinations $\{4,3\}$, another currently has $\{6\}$ and destinations are equally spaced. Also, suppose that $r = 3$. If the newly arriving vehicle has destination 3, it is assigned to the first platoon, even though it would be feasible to assign it to the second.

6.5 Simulation

VISSIM was used to simulate the proposed platoon assignment strategy. The four-lane highway used in the simulation has ten entrances from e_1 to e_{10} and ten exits from x_1 to x_{10} as shown in Fig. 6.4. The distance from one entry to the next exit is 2 km. The highway starts with zero traffic and the vehicles are generated randomly by VISSIM. A screenshot of platooning operation in VISSIM is shown in Fig. 6.5.

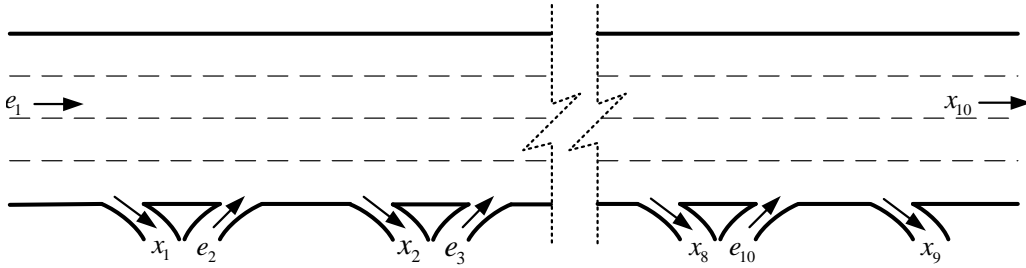


Figure 6.4: Highway used in simulations.

Let the platoon ratio be the ratio of vehicle distances traveled to platoon distances traveled. To evaluate the proposed strategy, platoon sizes and platoon ratios at the road segment immediately after the third entrance are collected under different total input flows from the first three entrances and different destination ranges r . Eq. (6.1) was used

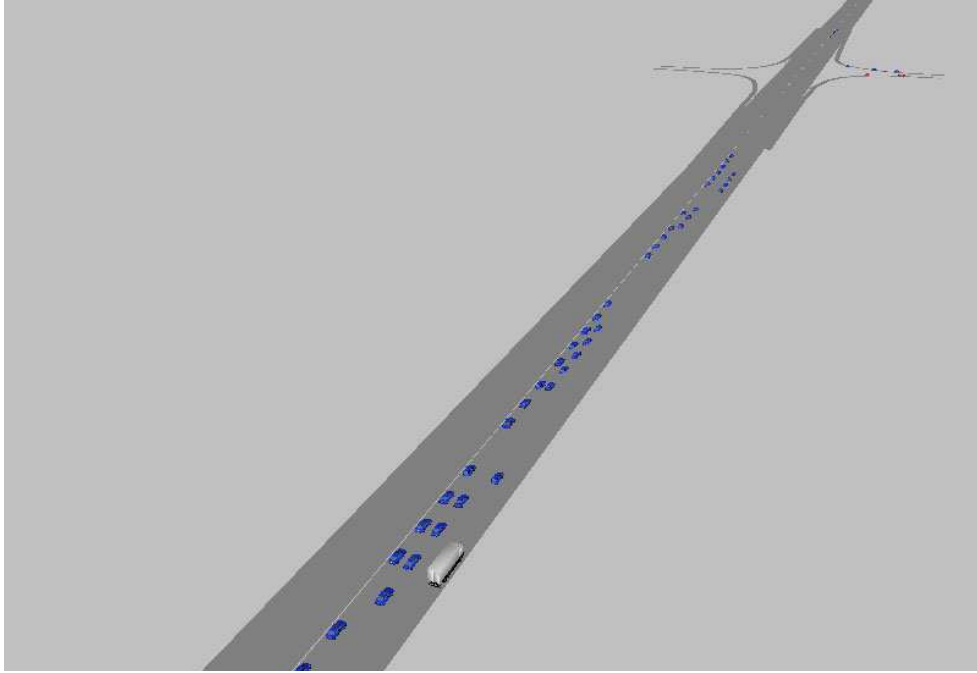


Figure 6.5: Screenshot of VISSIM simulation of platooning operation.

to calculate the lane capacity based on the average platoon size. The parameters used in this equation are: $t_h = 3 \text{ sec}$, $v = 100 \text{ km/h}$, $h = 10 \text{ m}$, $s = 4.5 \text{ m}$, and $\Upsilon = 10 \text{ veh}$.

The probability distributions of platoon sizes versus destination range and total input flow are plotted in Fig. 6.6 and the associated means and standard deviations are plotted in Fig. 6.7. Fig. 6.8 shows the relationship between mean/standard deviation of platoon size, destination range and total input flow in three dimensional space. It can be seen that for each destination range, the average platoon size is small at low longitudinal input flow and grows as the volumes of vehicles at entries get higher. We can also see that no platoon exceeds the maximum platoon size of 10 vehicles when the input flow is 1000 veh/hr even at $r = 5$. The average platoon size is also larger when the destination range is large which increases lane capacity as shown in Fig. 6.9. This is reasonable since at high longitudinal flow, the chance of getting a feasible vehicle/platoon to join is higher.

Fig. 6.9 shows the lane capacities at 100 km/hr calculated from Eq. (6.1) based on the average platoon sizes. It is possible to draw several conclusions from this plot. Over the ranges of destinations tested, a larger range has greater benefit on lane capacity, but at the cost of reducing platoon ratio as shown in Fig. 6.10. The values for lane capacity are significant, at $r = 5$, the lane capacities range from 2360 veh/hr to 4880 veh/hr depending upon the total input flow. A higher longitudinal input flow results in a larger capacity.

On the other hand, there is no big improvement in terms of lane capacity for small

destination range at low input flows. For example, the lane capacity is only 1610 *veh/hr* when the input flow is 1000 *veh/hr* and $r = 1$. A small destination range, however, means that most of the vehicles in a platoon have same destination and they can therefore exit the highway at the same time. Fig. 6.10 shows that at $r = 1$ the vehicles stay intact with the platoons up to 88% of the highway on average while this number drops to 66%, 69% and 70% for the longitudinal flow of 1000 *veh/hr*, 5000 *veh/hr* and 10,000 *veh/hr*, respectively.

6.6 Conclusions

To maximize highway throughput and reduce fuel consumption, it is desirable to create platoons that are large in size, and that remain intact over long distances. Sorting vehicles by destination at the entrance is one way to accomplish this objective. Toward this end, this work evaluated a proposed strategy which assigns vehicles to platoons by solving an optimization problem.

A linear model for effectively assigning vehicles to appropriate platoons when they enter the highway was formulated. The simplex algorithm was used to solve this optimization problem and simulation results in VISSIM were presented.

Simulation results demonstrated that lane capacity can be increased to between 2360 *veh/hr* and 4880 *veh/hr* when the destination range, representing the difference in index between the closest and the furthest exits for vehicles in the platoon, of 5 exits was used, depending on the total input traffic volume. Simulation results also suggested that while a smaller range of destination can lead to a lesser increase in lane capacity, it ensures that vehicles stay intact with the platoons for longer distances. For example, at the total input traffic volume is 5000 *veh/hr*, vehicles remain with their host platoons up to 88% when the destination range is 1 and only 69% when the destination range is 5. This results can be used to balance the tradeoff between highway capacity and distances platoons remain intact.

The negative side of the proposed platoon assignment strategy is that it increases the frequency of lane changes and therefore might reduce the level of highway safety. However, this issue can be overcome with automatic control and is far outweighed by many other advantages of platooning.

A potential future direction for this work will be on quantifying the effect of platoon assignment on the reduction of fuel consumption. This can be done by using the field test results from the study of Browand and Michaelian [95, 96].

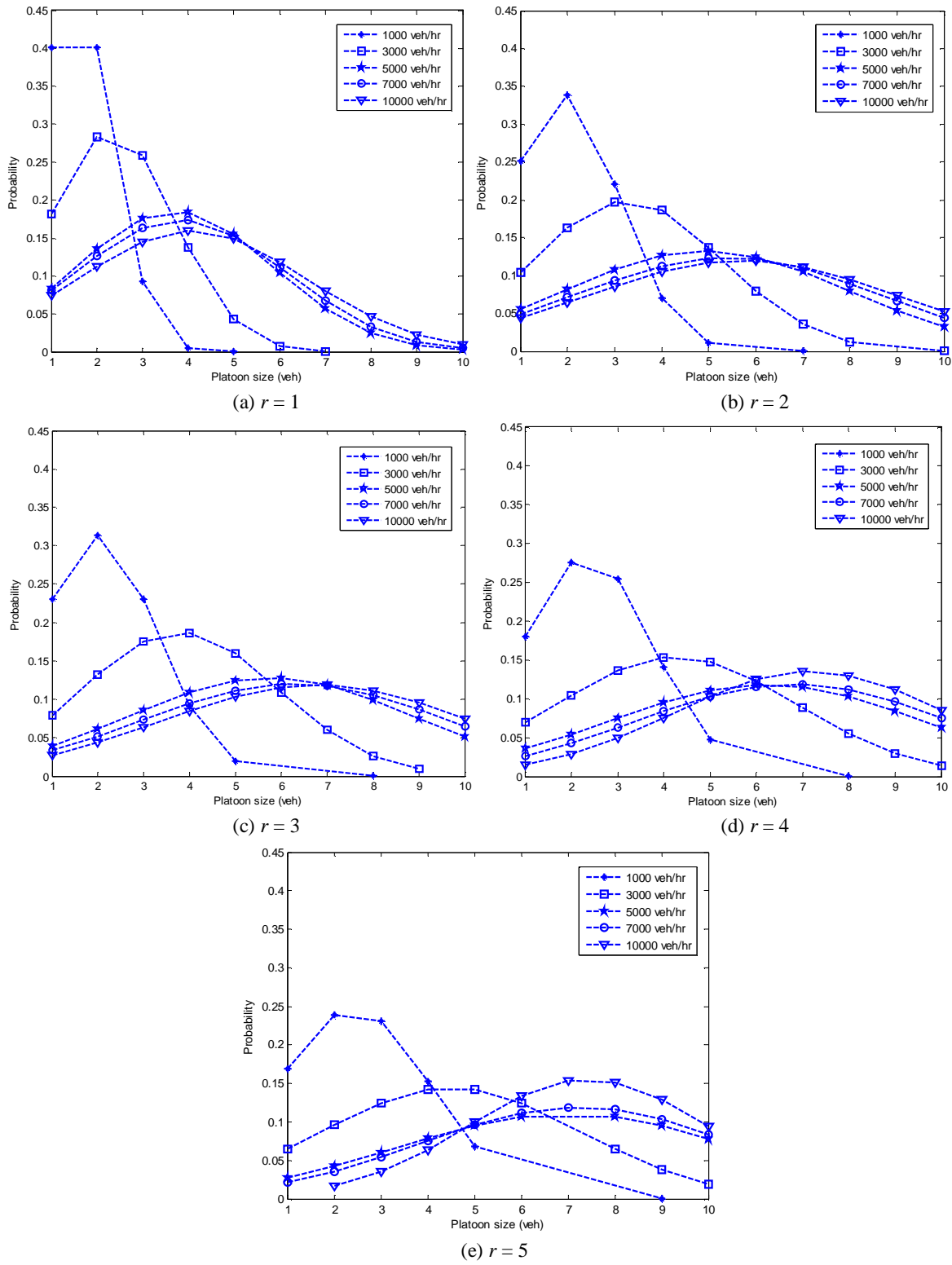


Figure 6.6: Platoon size probability distribution versus destination range r and input flow.

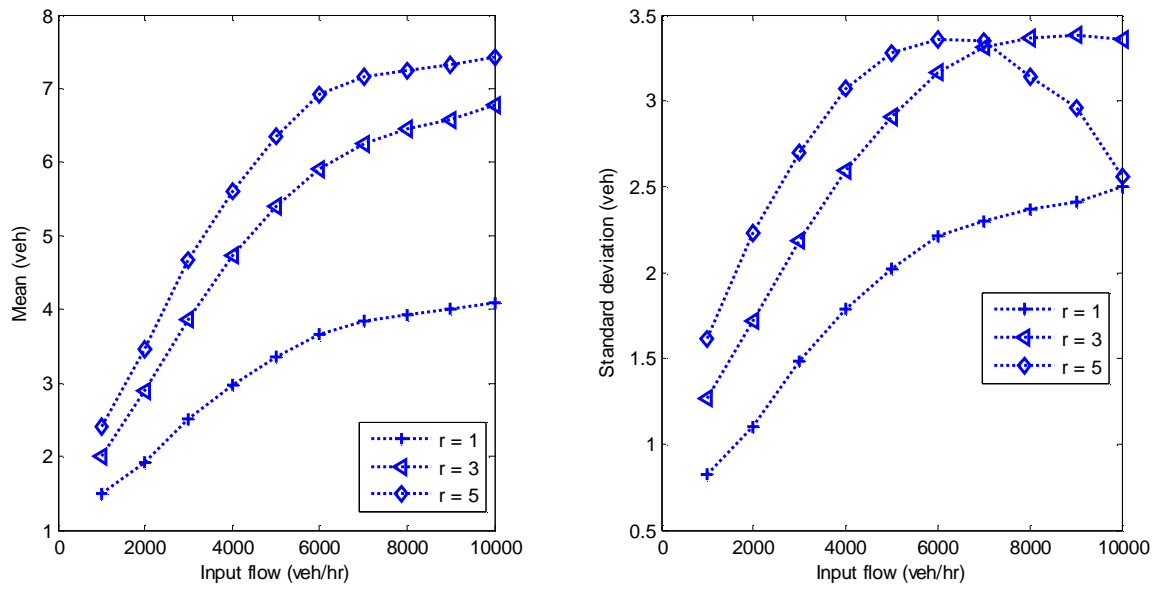


Figure 6.7: (a) Average and (b) standard deviation of platoon size versus destination range r and input flow.

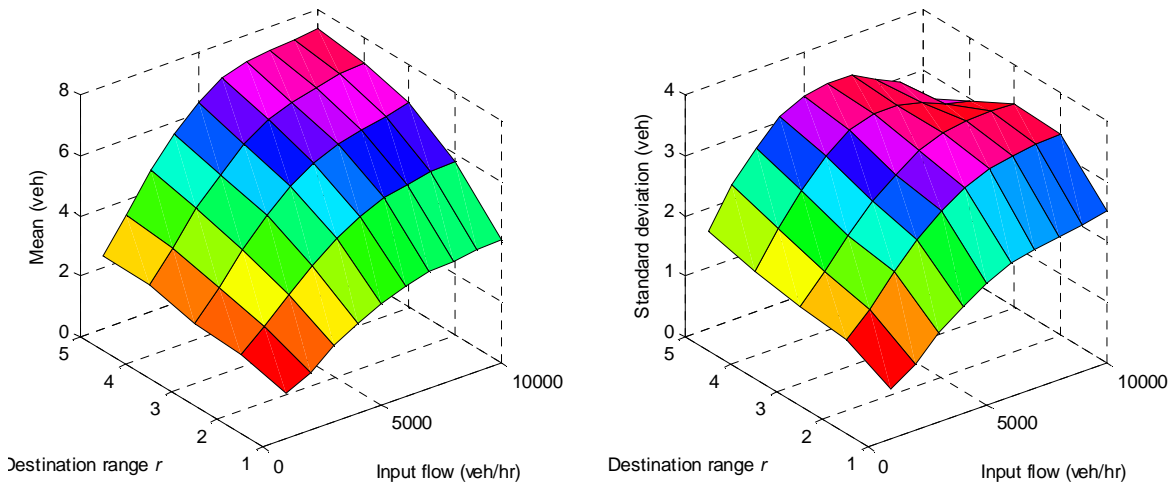


Figure 6.8: (a) Average and (b) standard deviation of platoon size versus destination range r and input flow.

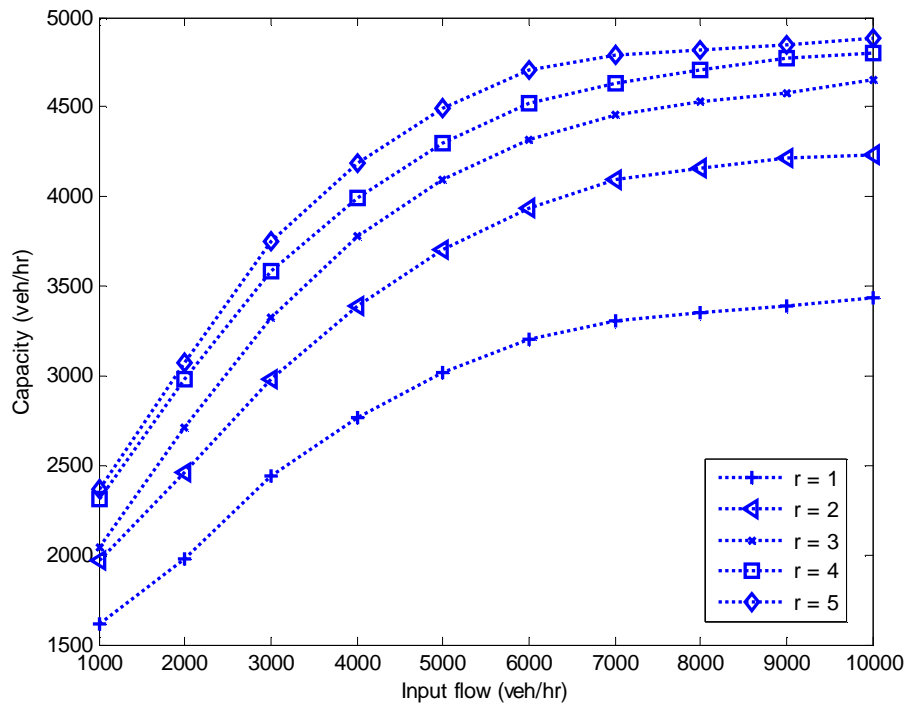


Figure 6.9: Lane capacity versus destination range r and input flow.

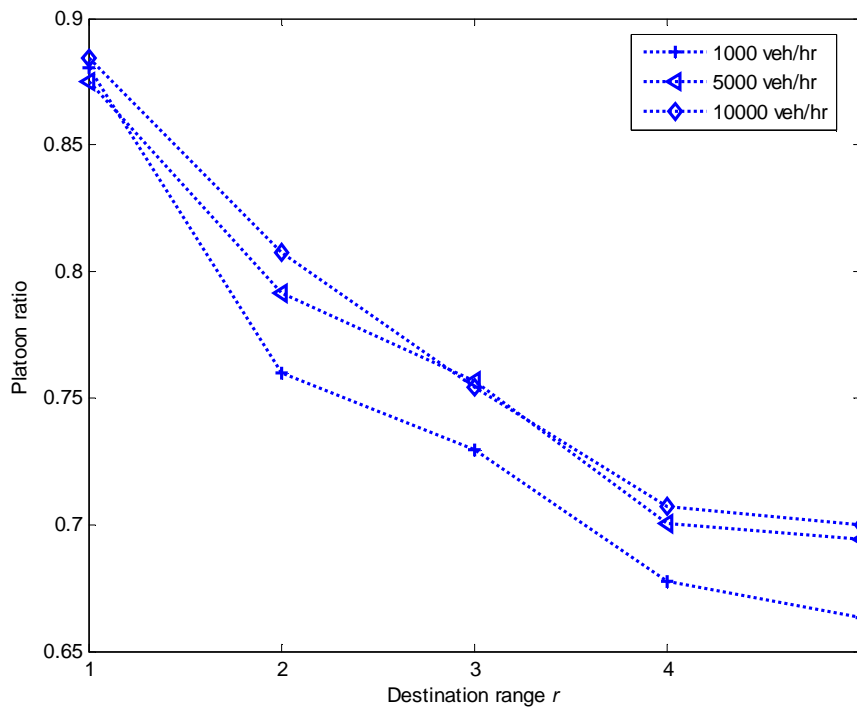


Figure 6.10: Platoon ratio versus destination range r and input flow.

Chapter 7

Conclusions and Recommendations

7.1 Summary of the Thesis

The work presented in this thesis addresses several issues in ITS by way of a dynamic collaborative driving approach. The problems of lane positioning, lane assignment optimization, and platoon formation have been addressed and appropriate strategies to solve these problems have been developed and tested in both real experiments and simulations.

The main scientific achievements can be summarized as follows:

In Chapter 4 a novel lane position estimation system using a Markovian approach based on cooperative driving was developed. The robustness and effectiveness of the system was shown in both real road-tests and VISSIM simulations. For hardware implementation, an experimental setup consisting of notebook PC's, equipped with standard GPS units for vehicle locations and a wireless ad-hoc network for communication capability was designed and tested in real-time in outdoor conditions.

In comparison to conventional lane positioning methods which usually deal with complicated image processing techniques and/or expensive equipment, the proposed method only requires low cost GPS receivers, a peer to peer communication system and a simple localization algorithm. Simulation and experimental results have shown the efficiency of the algorithm, even when the GPS data were significantly degraded. This lane positioning system also serves as a basis for all subsequent strategies such as traffic flow estimation, lane assignment, and platoon assignment.

Chapter 5 presents a strategy for optimization of lane assignment with the objective of increasing traffic throughput on highway systems. To solve the problem, a linear model for lane assignment was proposed and results on the use of a linear programming algorithm in

the solution of the problem were presented. This system is unique in that it used GPS as the only form of sensing, which made the system simpler in terms of hardware management and thus less costly. Simulation results demonstrated that intelligent lane selection can improve highway capacity.

Also in Chapter 5, a method for flow rate estimation was proposed with a formula being given in Eq. 5.1. The simulation results showed that the strategy was reliable to estimate the flows under different traffic conditions and can be used together with lane positioning scheme for the subsequent lane optimization algorithm.

Chapter 6 presents an approach to increasing highway capacity and reducing fuel consumption by organizing vehicles into platoons with the objective of maximizing the travel distance that platoons stay intact. To quantify the effectiveness of platoon assignment, platoon sizes and ratios were collected under different longitudinal input flows and destination ranges. Simulation results showed that lane capacity can be increased effectively when platooning operation is used.

7.2 Known Limitations

For the lane positioning system, the limitation of the proposed strategy lies in the fact that it only uses GPS data to estimate lane positions. This might be challenging where GPS data is not available or the GPS signal is blocked completely by large obstacles such as in a long tunnel. One possible solution to this problem is to fuse the GPS data with another type of sensor such as an Inertial Measurement Unit (IMU) until GPS data is again available.

In the formulation of the lane assignment problem, it was assumed that the capacities of lanes are fixed and are not affected by lateral flows due to lane changes. This is a shortcoming of the system since in practice, the actual lane capacities depend on the number of lane changing maneuvers that occur and hence on the lane assignment. For a better assessment of the system, lane capacities need to be estimated by taking into account lane change maneuvers.

For platooning, the negative side of the proposed platoon assignment strategy is that it increases the frequency of lane changes and therefore might reduce the level of highway safety. Additional work is required to ensure that lane changes are safe either by using autonomous control of vehicles, or by including a term in the cost function that increases the cost based on lane changes required.

Another limitation in platoon assignment is that the complex interaction between platoons and free-agent vehicles when vehicles approach their assigned platoons was not considered in the objective function. This leaves room for further investigation in future research.

7.3 Future Research

The future direction of this work has four logical aspects: (1) modeling dynamic capacity as a function of lane change maneuvers, (2) quantifying the effect of selfish lane selection on lane assignment, (3) formulating the effect of lane assignment and platooning operations on the reduction of fuel consumption, and (4) developing a lane assignment algorithm for vehicle platoons.

7.3.1 Dynamic Capacity

One of the shortcomings in the formulation of the lane assignment problem is that the capacities of lanes are assumed to be fixed. For better assessment of the system, the lane changing maneuvers need to be taken into account to estimate realistic lane capacities. In order to do so, the lateral flow caused by lane changes when the vehicle move to target lanes as well as when they exit the highway must be formulated.

7.3.2 Selfish Lane Selection

Another potential future direction for lane assignment will to study the effect of selfish lane selection, where a portion of the drivers do not comply with the suggested lane assignment strategy. This will increase our understanding the underlying structure of such decision making situations. Such a system can be modeled using a non-cooperative game theory.

7.3.3 Fuel Consumption Estimation for Platooning

A potential future direction for platoon assignment can be on quantifying the effect of platoon assignment on the reduction of fuel consumption. It was recognized that close-following would likely decrease the average vehicle drag and therefore also decrease the average fuel consumption shown in the field test results from the study by Browand and Michaelian [95, 96]. However, we are aware of no information in the open literature quantifying the fuel savings for vehicle platoons as a function of vehicle spacing and platoon size

and determining the effect of platoon assignment on fuel savings. A study on this issue will be beneficial for future research on platooning.

7.3.4 Lane Assignment for Vehicle Platoons

Vehicle platooning is of interest to a number of researchers for increasing traffic throughput by packing vehicles together. Algorithms for handling lane changing, lane assignment for platoons of vehicles and other associated problems within vehicle platoons will also be considered in future research.

Appendix A

Simplex Algorithm

In mathematical optimization theory, the simplex algorithm of George Dantzig is a popular technique for numerical solution of the linear programming problem defined by maximizing

$$\mathbf{c}^T \mathbf{x},$$

subject to

$$\mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0.$$

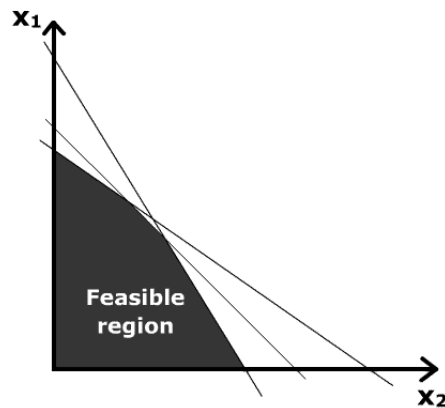


Figure A.1: Feasible region: a series of linear constraints on two variables produces a region of possible values for those variables. Solvable problems will have a feasible region in the shape of a simple polygon [69].

Geometrically, the linear constraints define a convex polyhedron, which is called the feasible region (see Fig. A.1). Since the objective function is also linear, all local optima are automatically global optima. The linearity of the objective function also implies that an optimal solution can only occur at a boundary point of the feasible region, unless the objective function is constant, when any point is a global minimum.

Let H be the halfspaces defined by the constraints. If the constraints contradict each other ($\bigcap H = \emptyset$) then the feasible region is empty and there is no optimal solution, since there are no solutions at all. In this case, the linear programming problem is said to be infeasible. The polyhedron can also be unbounded in the direction of the objective function. In this case, there is also no optimal solution.

Linear programming problems must be converted into augmented form before being solved by the simplex algorithm. This form introduces non-negative slack variables to replace non-equalities with equalities in the constraints. The problem can then be written in the following form

$$\begin{bmatrix} 1 & -\mathbf{c}^T & 0 \\ 0 & \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} Z \\ \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix},$$

where Z is the variable to be maximized and \mathbf{x}_s are the newly introduced slack variables.

Algorithm 3 Simplex(H)

Require: set of halfspaces H

Ensure: the lowest vertex in the intersection of H

```

1: if  $\bigcap H = \emptyset$  then
2:   return INFEASIBLE
3: end if
4:  $x \leftarrow$  any feasible vertex
5: while  $x$  is not locally optimal do {pivot downward, maintaining feasibility}
6:    $x \leftarrow$  any feasible neighbor of  $x$  that is lower than  $x$ 
7:   if  $x = \infty$  then
8:     return UNBOUNDED
9:   end if
10: end while
11: return  $x$ 

```

The simplex algorithm solves linear programming problems by constructing an admissible solution at a vertex of the polyhedron, and then walking along edges of the polyhedron to vertices with successively higher values of the objective function until the optimum is reached. The pseudo-code of the algorithm is given in Algorithm 3 [70].

Appendix B

VISSIM Simulation Setting

Highway parameters used in the simulations are listed in Table B.1. The main difference in VISSIM-controlled simulations and lane assignment simulations is the way vehicle speeds and lane change behavior were controlled.

In VISSIM-controlled simulations, a vehicle will travel at its desired speed (with a small stochastic oscillation) if it is not hindered by other vehicles. A vehicle with a higher desired speed than its current speed will check for the opportunity to pass. The desired speed in each lane was set using Desired Speed Decisions. The s-shaped distribution of the desired speed distribution is concentrated around the median value, which is the nominal driving speed assigned for a lane. Vehicle following and lane change parameters in VISSIM-controlled simulations are listed in Table B.2 and Table B.3, respectively.

In lane assignment simulations, vehicles were assigned a driving speed plus a small speed oscillation once they enter a lane. The driving speed for each lane was obtained using Van Aerde's model based on the traffic density and calibrated parameters (see the source code below). The parameters of Van Aerde's model used in this work are similar to VISSIM's Wiedemann's car-following model as calibrated in [85, 86].

```
// Member function name: van_aerde_speed
// Purpose: determine lane speed using Van Aerde's model
// Input:  f = traffic flow rate; vf = free speed; C = capacity
// Output: double value of vehicle speed [km/hr]
double van_aerde_speed(double f, double vf, double C) {
    if (f > C) f = C;

    double Alpha = 0.81;
    double vc = Alpha*vf; //speed at capacity
    double dj = 5*vc/vf; //jam density

    double m = (2*vc - vf)/((vf - vc)*(vf - vc));
    double c2 = 1/(dj*(m + 1/vf));
```


Table B.1: Highway parameters.

Type	Parameters
Number of entrances and exits	5
Number of lanes	4
Highway length	10 <i>km</i>
Lane free speeds	80, 90, 100, 120 <i>km</i>
Lane capacity	2,200 <i>veh/hr</i>
Highway type	Freeway
Vehicle length	4.5 <i>m</i>

Table B.2: Vehicle following behavior parameters in VISSIM-controlled simulations (VISSIM default setting).

Type	Parameters
Car-following model	Wiedemann's model
Look ahead distance	min: 0.00 <i>m</i> , max: 250 <i>m</i>
Temporary lack of attention	duration: 0.00 <i>s</i> , probability: 0.00%
Speed distribution	s-shaped distribution

Table B.3: Lane change parameters in VISSIM-controlled simulations (VISSIM default setting).

Type	Parameters
General behavior	Free lane selection
Lateral behavior	Observe vehicles in next lane(s)
Waiting time before diffusion	60 <i>s</i>
Min. headway	0.5 <i>m</i>
Max. deceleration	Own: -4.00 <i>m/s</i> ² , Trailing vehicle: -3.00 <i>m/s</i> ²
-1 <i>m/s</i> ² per distance	Own: 100 <i>m</i> , Trailing vehicle: 200 <i>m</i>
Accepted deceleration	-1.00 <i>m/s</i> ² , Trailing vehicle: -0.50 <i>m/s</i> ²

```

    double c1 = m*c2;
    double c3 = (-c1 + vc/C - c2/(vf - vc))/vc;

    double a = c1*vf + c2;
    double b = -c1*f + c3*f*vf - vf;
    double c = f - c3*f*f;
    double delta = b*b - 4*a*c;

    double d = (-b + sqrt(delta))/(2*a); //density

    return f/d;
}

// Member function name: give_da
// Purpose: Gives vehicle desired acceleration to VISSIM
// Input: Null
// Output: double value of desired acceleration [m/s^2]
double CVehicle::give_da() const {
    const double speedOscil = 0.1; // velocity oscillation [m/s]
    const double accStep = 2;

    long lane = this->give_lane(); // get lane position from VISSIM
    double v = this->give_v(); // get speed from VISSIM

    // calculate lane speed using Van Aerde's model
    // from estimated flow rate, lane free-speed, and lane capacity
    double v1 = van_aerde_speed(estimatedFlow[0], SPEED_LANE1, CAPACITY_LANE1)/3.6;
    double v2 = van_aerde_speed(estimatedFlow[1], SPEED_LANE2, CAPACITY_LANE2)/3.6;
    double v3 = van_aerde_speed(estimatedFlow[2], SPEED_LANE3, CAPACITY_LANE3)/3.6;
    double v4 = van_aerde_speed(estimatedFlow[3], SPEED_LANE4, CAPACITY_LANE4)/3.6;

    switch (lane)
    {
    case 1: // lane 1
        if (v > (v1 + speedOscil))
            return -accStep;
        else if (v < (v1 - speedOscil))
            return +accStep;
        else
            return 0;
    case 2: // lane 2
        if (v > (v2 + speedOscil))
            return -accStep;
        else if (v < ((v2 - speedOscil))
            return +accStep;
        else
            return 0;
    case 3: // lane 3
        if (v > (v3 + speedOscil))
            return -accStep;
        else if (v < (v3 - speedOscil))
            return +accStep;
        else
            return 0;
    case 4: // lane 4
        if (v > (v4 + speedOscil))

```

```
        return -accStep;
    else if (v < (v4 - speedOscil))
        return +accStep;
    else
        return 0;
default:
    return 0;
}

return 0;
}
```

Appendix C

Lane Capacity

Let v be the steady state speed of the vehicles [km/hr], t_h be the separation time between platoons [seconds], s be the average length of vehicles [m], h be the intra-platoon distance headway [m], and Υ be the maximum platoon size (see Fig. C.1). The total distance from the front bumper of a platoon leader to the rear bumper of the last vehicle in the following platoon is $(\Upsilon - 1)h + s + \frac{t_h v}{3.6}$ [m]. The total time in hr taking all the Υ vehicles to pass a given point is

$$\frac{3.6[(\Upsilon - 1)h + s] + t_h v}{3600v}. \quad (C.1)$$

Therefore the maximum capacity is

$$C_{max} = \frac{3600\Upsilon v}{3.6[(\Upsilon - 1)h + s] + t_h v}. \quad (C.2)$$

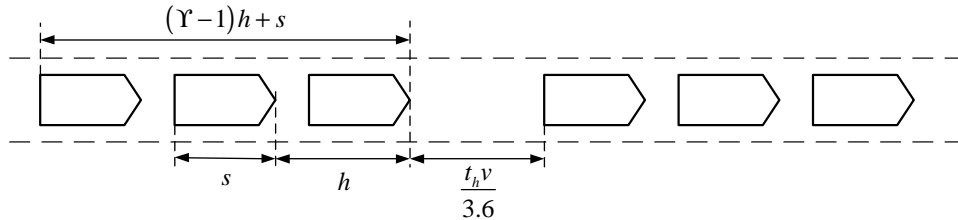


Figure C.1: Lane capacity calculation.

Bibliography

- [1] US Government Accountability Office, “Highway congestion: Intelligent transportation systems’ promise for managing congestion falls short, and DOT could better facilitate their strategic use,” *Report to Congressional Committee*, Sep 2005. 1, 3
- [2] Transport Canada, “The cost of urban traffic congestion in Canada,” March 2006. 1
- [3] S. Thrun *et al.* , “Stanley: The robot that won the darpa grand challenge,” *Journal of Robotic Systems*, vol. 23, no. 9, pp. 661-692, 2006. 5
- [4] R. Hall and C. Chin, “Vehicle sorting for platoon formation: impacts on highway entry and throughput,” *California PATH research report*, March 2002. 5, 87, 89
- [5] J. V. Medanic, D. Ramaswamy, W. R. Perkins, and R. Benekohal, “Partitioned Lane Assignment Strategies for Balancing Excess Lane Capacity on AHS,” in *Proc. 1995 Amer. Contr. Conf., Seattle, Washington*, 1995, pp. 3581-3585. 6, 59
- [6] PTV Planung Transport Verkehr AG, *VISSIM 4.10 user manual*, 2005. 7, 13, 76
- [7] ASTM E2213-03, *Standard specification for telecommunications and information exchange between road-side and vehicle systems - 5GHz band dedicated short-range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications*. ASTM International, Jul 2003. 10, 77
- [8] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, and M. Gerla, “How do you quickly choreograph inter-vehicular communications? A fast vehicle-to-vehicle multi-hop broadcast algorithm, explained,” in *Proc. of the 3rd IEEE CCNC International Workshop on Networking Issues in Multimedia Entertainment (CCNC/NIME 2007)*, Las Vegas, NV, USA, IEEE Communications Society, Jan 2007. 10, 77
- [9] I. Berger, *Standards for car talk*, [Online]. Available: www.theinstitute.ieee.org, Mar 2007. 11

- [10] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246257, 1984. 12
- [11] G. G. Finn, *Routing and addressing problems in large metropolitan-scale Internetworks*. University of Southern California, ISI/RR-87-180, 1987. 12
- [12] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *Proceedings 2nd IEEE Workshop, Mobile Computing Systems and Applications*, pp. 90100, 1999. 12
- [13] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, October 1994, pp. 234244. 12
- [14] D. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing (T. Imielinski and H.F. Korth, eds.)*, ch. 5, pp. 153-181, The Netherlands: Kluwer Academic Publishers, Feb. 1996. 12
- [15] Y. B. Ko and N. H. Vaidya, "GeoTORA: A protocol for geocasting in mobile ad hoc networks," *Proc. of 2000 International Conference on Network Protocols*, Osaka Japan, pp. 240250, Nov. 2000. 12
- [16] S. J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," *Wireless Communications and Networking Conference*, vol. 3, Chicago, USA, 2000, pp. 13111316. 12
- [17] M. Jiang and R. Jan, "An efficient multiple paths routing protocol for ad-hoc networks," *Proc. of the 15th International Conference on Information Networking*, Oita, Japan, 2001, pp. 544549. 12
- [18] S. R. Das *et al.* , "On-demand multipath distance vector routing for ad hoc networks," *IEEE ICNP 2001*, Nov, 2001. 12
- [19] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," *INFOCOM 2003*, vol. 1, 2003, pp. 270280. 12
- [20] A. Nasipuri, R. Castaneda, and R. R. Das, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," *Mobile Networks and Applications*, pp. 339349, Aug. 2001. 12
- [21] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," *ICC 2001*, Helsinki, Finland, 2001, vol. 10, pp. 32013205. 12

- [22] K. Obraczka and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless Networks*, vol. 7, no. 6, pp. 627-634, 2001. 12
- [23] S. Leng, L. Zhang, L. W. Yu, and C. H. Tan, "An efficient broadcast relay scheme for MANETs," *Computer Communications*, 2004. 12
- [24] C. S. Hsu and Y. C. Tseng, "An efficient reliable broadcasting protocol for ad hoc networks," *IASTED Networks, Parallel and Distributed Processing, and Applications (NPDPA)*, Japan, pp. 93-98, 2002. 12
- [25] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," *Proc. of MOBIHOC 2002*, Lausanne, Switzerland, 2002, pp. 194-205. 12
- [26] S. Y. Wang, C. C. Lin, Y. W. Hwang, K. C. Tao, and C. L. Chou, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," *Proc. of the 2005 IEEE Intelligent Transportation Systems*, pp. 161-166, 2005. 12
- [27] I. Stojmenovic, "Position based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128-134, 2002. 12
- [28] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023-1032, 2001. 12
- [29] A. Boukerchea, H. Oliveiraa, E. F. Nakamurab, and A. Loureiro, "Vehicular ad hoc networks: a new challenge for localization-based systems," *Computer Communications*, Elsevier, 2008. 12
- [30] Y. Hormann, H. P. Großmann, W. H. Khalifa, M. Salah, and O. H. Karam, "Simulator for inter-vehicle communication based on traffic modeling," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 991-104. 14
- [31] K. Y. K. Leung, T.-S. Dao, C. M. Clark, and J. P. Huissoon, "Development of a Hardware-in-the-loop Simulator for Inter-Vehicle Communication Application Research," in *Proc. of IEEE ITSC 2006*, Toronto, 2006, pp. 1286-1291. xi, 15, 16
- [32] C. Demonceaux, A. Potelle, and D. Kachi-Akkouche, "Obstacle detection in a road scene based on motion analysis," *IEEE Trans. On Vehicular Technology*, vol. 53, no. 6, Nov. 2004, pp. 1649-1656. 18
- [33] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M. -M. Meinecke, "Pedestrian detection for driver assistance using multiresolution infrared vision," *IEEE Trans. On Vehicular Technology*, vol. 53, no. 6, Nov. 2004, pp. 1666-1678. 18

- [34] J. Manigel and W. Leonard, "Vehicle control by computer vision," *IEEE Trans. On Industrial Electronics*, vol. 39, no. 3, June 1992, pp. 181-188. 18
- [35] R. Cucchiara, D. Lovell, A. Prati, and M. M. Trivedi, "Special section on in-vehicle computer vision systems," *IEEE Trans. On Vehicular Technology*, vol. 53, no. 6, Nov. 2004. 18
- [36] GM Collaborative Research Lab at Carnegie Mellon University [Online], Available: <http://gm.web.cmu.edu>. 18
- [37] M. M. Trivedi, S. Y. Cheng, E. M. C. Childers, and S. J. Krotosky, "Occupant posture analysis with stereo and thermal infrared video: algorithms and experimental evaluation," *IEEE Trans. On Vehicular Technology*, vol. 53, no. 6, Nov. 2004, pp. 1698-1712. 18
- [38] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. USA: The MIT Press, 2004. 20
- [39] J. Farrel and M. Barth, *The Global Positioning System and Inertial Navigation: Theory and Practice*. McGraw-Hill, 1999. 22, 23, 25
- [40] E. Kaplan, *Understanding GPS: principles and applications*. USA: Atech House Inc., 1996. 22, 23, 25
- [41] J. Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach*. Canada: John Wiley and Sons, Inc., 2000. 22
- [42] G. Xu, *GPS Theory, Algorithms and Applications*. Germany: Springer Verlag, 2003. 22, 23, 25
- [43] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. Hoboken, NJ: Wiley, 2000. 22
- [44] J. Du, J. Masters, and M. Barth, "Lane-level positioning for in-vehicle navigation and automated vehicle location (AVL) systems," in *Proc. Of IEEE Intelligent Transportation Systems Conference*, Washington, USA, 2004, pp. 35-40. 27, 28
- [45] S. S. Ieng, J. Vrignon, D. Gruyer, and D. Aubert, "A new multi-lane detection using multi-camera for robust vehicle localtion," in *Proc. of IEEE Intelligent Vehicles Symposium*, Nevada, USA, 2005, pp. 700- 705. 28
- [46] G. Pierre-Yves and K. Jeff, "Enhanced navigation system for road telematics," *3rd Swiss Transportation Research Conference*, 2003, pp. 1-16. 28

- [47] *Enhanced digital mapping project final report*. U.S. Department of transportation, Washington, 2004. 28
- [48] K. Y. Chiu and S. F. Lin, "Lane detection using color-based segmentation," in *Proc. Of the IEEE Intelligent Vehicle Symposium*, 2005, pp. 706-711. 28
- [49] B. D. Stewart, I. Reading, M. S. Thomson, T. D. Binnie, K. W. Dickinson, and C. L. Wan, "Adaptive lane finding in road traffic image analysis," in *Proc. of the 7th IEEE Int. Conference on Road Traffic Monitoring and Control*, London, UK, 1994, pp. 133-136. 28
- [50] J. Gangyi, C. Yanhua, Y. Mei, and Z. Yi, "Approach to lane departure detection," in *Proc. of IEEE 5th Int. Conference on Signal Processing, ICSP'2000*, pp. 971-974. 28
- [51] A. H. S. Lai and N. H. C. Yung, "Lane detection by orientation and length discrimination," *IEEE Trans. On Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 30, no. 4, August 2000, pp. 539-548. 28
- [52] T. K. Lim and M. A. Do, "Camera detection, classification, and positioning of vehicles on a multi-lane road," in *Proc. of IEEE Singapore International Conference on Information Engineering*, vol. 1, Singapore, 1993, pp. 374-378. 28
- [53] M. M. Artimy, W. Robertson, and W. J. Phillips, "Connectivity in intervehicle ad-hoc networks," *Engineering Canadian Conference on Electrical and Computer*, vol. 1, May 2004, pp. 293-298. 29
- [54] J. Blum, A. Eskandarian, and L. Hoffman, "Performance characteristics of inter-vehicle ad hoc networks," in *Proc. Of the IEEE 6th International Conference on Intelligent Transportation Systems*, China, 2003, pp. 114-119. 29
- [55] Z. D. Chen, H. T. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proc. Of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001. 29
- [56] W. Enkelmann, "FleetNet applications for inter-vehicle communication," in *Proc. Of the IEEE Intelligent Vehicles Symposium*, Ohio, USA, 2003, pp. 162-167. 29
- [57] S. J. Russell, and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995. 31
- [58] D. Fox, *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. Ph.D. thesis, Dept. of Computer Science, University of Bonn, Germany, December 1998. 31

- [59] S. Koenig and R. Simmons, "A robot navigation architecture based on partially observable markov decision process models," in *Kortenkamp et al. AI-ROBOTS*. 31
- [60] B. A. Sheno, *Introduction to digital signal processing and filter design*, John Wiley & Sons, 2005. 42
- [61] I. M. Rekleitis, "A particle filter tutorial for mobile robot localization TR-CIM-04-02," *Centre for Intelligent Machines*, McGill University, Canada. 42, 43, 44
- [62] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, 2001. 42
- [63] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, 2000. 42
- [64] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. USA: The MIT Press, 2005. 42, 43, 44
- [65] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P. -J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425-437, 2002. 42, 43, 44
- [66] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002. 42, 43, 44
- [67] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. USA: The MIT Press, 2005. 42, 43, 44
- [68] T. Chen, H. Haussecker, A. Bovyryn, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, "Computer vision workload analysis: case study of video surveillance systems". *Intel Technology Journal*, vol. 9, no. 2, May 2005, pp. 109-118. 44
- [69] Linear Programming [Online]. Available: http://en.wikipedia.org/wiki/Linear_programming. xiv, 58, 102
- [70] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. 58, 103
- [71] J. F. Gilmore, K. J. Elibiary, and H. C. Forbes, "Knowledge-based Advanced Traffic Management System," in *Proc. of IVHS America Atlanta*, 1994, GA. 59

- [72] K. Kagolanu, R. Fink, H. Smartt, R. Powell, and E. Larson, "An Intelligent Traffic Controller," in *Proc. of the Second World Congress on Intelligent Transportation Systems*, Japan, 1995, pp. 259-264. 59
- [73] F. .J. Pooran, P. .J. Tarnoff, and R. Kalaputapu, "RT-TRACS: Development of the Real-Time Control Logic," in *Proc. of the 1996 Annual Meeting of ITS America*, 1996, pp. 422-430. 59
- [74] M. E. Goolsby, D. W. Fenno, and A. P. Voigt, "Project Summary Report 2910-S: Changeable Lane Assignment System (CLAS) on Frontage Roads," *Texas Transportation Institute*, 2000. 59
- [75] R. W. Hall, "Longitudinal and lateral throughput on an idealized highway," *Transportation Science*, vol. 29, 1995, pp. 118-127. 59
- [76] D. Ramaswamy, J. V. Medanie, W. R. Perkins, and R. F. Benekohal, "Lane assignment on an automated highway," in *Proc of the 1994 American Control Conference*, 1994, pp. 413-417. 59
- [77] D. Ramaswamy, J. V. Medanie, W. R. Perkins, and R. F. Benekohal, "Lane assignment on automated highway systems," *IEEE Trans. on Veh. Tech.*, 1997, vol. 46, pp. 755-759. 59
- [78] R. W. Hall and C. Caliskan, "Design and evaluation of an automated highway system with optimized lane assignment," *Trans. Rev. - part C*, vol. 7, 1999, pp. 1-15. xiii, 59, 60
- [79] K. Kim, D. .I. Cho, and J. .V. Medanic, "Lane assignment using a genetic algorithm in the automated highway systems," in *Proc. of the 8th International IEEE Conference on Intelligent Transportation Systems*, Austria, 2005, pp. 332-337. 59
- [80] J. K. Lee and J. J. Lee, "Discrete event modeling and simulation for flow control in an automated highway system," *Transportation Research - Part C: Emerging Technologies*, 1997, vol. 5, no. 3-4, pp. 179-195. 59
- [81] L. Khachian, *A polynomial algorithm in linear programming*. Soviet Math. Dokl., pp. 191-194, 1979. 62
- [82] T. -S. Dao, C. M. Clark, and J. P. Huissoon, "Distributed platoon assignment and lane selection for traffic flow optimization," in *Proc. of IEEE IV 2008*, Eindhoven, Netherlands, June 2008. 63

- [83] D. Ramaswamy, J. V. Medanic, W. R. Perkins, and R. Benekohal, "Lane assignment on an automated highway," in *Proc. 1994 Amer. Contr. Conf., Maryland*, June 1994, pp. 413-417. 70
- [84] M. V. Aerde and H. Rakha, "Multivariate calibration of single regime speed-flow-density relationships," in *Proc. of the Vehicle Navigation and Information Systems (VNIS) conference*, Seattle, Washington, August, 1995. 75, 77
- [85] H. Rakha and B. Crowther, "A comparison of the Greenshields, Pipes, and Van Aerde car-following and traffic stream models," *Transportation Research Record No. 1802, Traffic Flow Theory and Highway Capacity 2002*, pp. 248-262, 2002. 75, 77, 104
- [86] M. Fellendorf and P. Vortisch, "Validation of the microscopic traffic flow model VISSIM in different real-world situations. Transportation Research Board," 2001. 77, 104
- [87] J. Medanic, "Multi-person games for infrastructure exploitation," in *Proc. of the 10th Mediterranean Conference on Control and Automation - MED2002*, 2002. 82
- [88] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, February 1993. 85
- [89] S. Halle, B. Chaib-draa, and J. Laumonier, "Car platoons simulated as a multiagent system," in *Proc. of Agent Based Simulations*, 2003. 85
- [90] S. Halle, J. Laumonier, and B. Chaib-draa, "A decentralized approach to collaborative driving coordination," in *Proc. of IEEE Conference on Intelligent Transportation Systems*, 2005. 85
- [91] H. S. Tan, R. Rajamani, and W. B. Zhang, "Demonstration of an automated highway platoon system," In *Proceedings of the American Control Conference 1998*, Philadelphia, PA, USA, vol.3, pp. 1823-1827, 1998. 85
- [92] R. Rajamani, H. S. Tan, B. K. Law, and W. B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," In *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, July, pp. 695-708, 2000. 85, 88
- [93] P. A. Ioannou, *Automated highway Systems*. Springer, 1997. 86
- [94] Transportation Research Board, *Highway capacity manual*. 2000. 86
- [95] F. Browand and M. Michaelian, "Platoon travel saves fuel...how much?," *Intellimotion*, vol. 9, no. 2, pp. 1-5, 11, 2000. xiv, 86, 94, 100

- [96] M. Michaelian and F. Browand, "Field experiments demonstrate fuel savings for close-following," *PATH Research Report UCB-ITS-PRR-2000-14*, University of California, Berkeley: Institute of Transportation Studies, California PATH Program, 2000. 86, 94, 100
- [97] G. Meier, G. Roppenecker, and C. Wurmthaler, "Automatic lateral vehicle guidance using tracking control - a modular approach to separate driver- and vehicle-dependent dynamics," In *2004 IEEE Intelligent Vehicles Symposium*, pp. 145-149, June 2004. 88
- [98] M. S. Netto, S. Chaib, and S. Mammar, "Lateral adaptive control for vehicle lane keeping," In *Proc. of the 2004 American Control Conference*, pp. 2693-2698, Boston, USA, 2004. 88
- [99] H. Peng, W. B. Zhang, A. Arai, Y. Lin, T. Hessburg, P. Devlin, M. Tomizuka, and S. Shladover, "Experimental automatic lateral control system for an automobile," In *Research Reports: UCB-ITS-PRR-92-11*, Institute of Transportation Studies California Partners for Advanced Transit and Highways (PATH), University of California, Berkeley, USA, 1992. 88
- [100] D. Swaroop and J. K. Hedrick, "Direct adaptive longitudinal control for vehicle platoons," *IEEE Conference on Decision and Control*, December, 1994. 88
- [101] H. Raza and P. Ioannou, "Vehicle following control design for automated highway systems," in *IEEE Control Systems Magazine*, vol. 16, no. 6, pp. 43-60, Dec, 1996. 88
- [102] H. Fritz, "Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the CHAUFFEUR project," In *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast, HI, USA, vol. 2, pp. 1348-1352, 1999. 88
- [103] R. Rajamani, H. S. Tan, B. K. Law, and W. B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," In *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695-708, July, 2000. 85, 88
- [104] T. S. Dao, K. Y. K. Leung, C. M. Clark, and J. P. Huissoon, "Co-operative Lane-Level Positioning Using Markov Localization," in *Proc. of IEEE ITSC 2006*, Toronton, 2006, pp. 1006-1011.
- [105] T. S. Dao, K. Y. K. Leung, C. M. Clark, and J. P. Huissoon, "Markov-based lane positioning using intervehicle communication," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 641-650, December 2007.