Proactive Traffic Control Strategies for Sensor-Enabled Cars

Ziyuan Wang

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

Department of Computer Science and Software Engineering
THE UNIVERSITY OF MELBOURNE

November 2009
Abstract

Traffic congestions and accidents are major concerns in today’s transportation systems. This thesis investigates how to improve traffic throughput by reducing or eliminating bottlenecks on highways, in particular for merging situations such as intersections where a ramp leads onto the highway. In our work, cars are equipped with sensors that can measure distance to neighboring cars, and communicate their velocity and acceleration readings with one another. Sensor-enabled cars can locally exchange sensed information about the traffic and adapt their behavior much earlier than regular cars.

We propose proactive algorithms for merging different streams of sensor-enabled cars into a single stream. A proactive merging algorithm decouples the decision point from the actual merging point. Sensor-enabled cars allow us to decide where and when a car merges before it arrives at the actual merging point. This leads to a significant improvement in traffic flow as velocities can be adjusted appropriately. We compare proactive merging algorithms against the conventional priority-based merging algorithm in a controlled simulation environment. Experimental results show that proactive merging algorithms outperform the priority-based merging algorithm in terms of flow and delay.

More importantly, the imprecise information (errors in sensor measurements) is a major challenge for merging algorithms, because inaccuracies can potentially lead to unsafe merging behaviors. In this thesis, we investigate how the accuracy of sensors impacts merging algorithms, and design robust merging algorithms that tolerate sensor errors. Experimental results show that one of our proposed merging algorithms,
which is based on the theory of time geography, is able to guarantee safe merging 
while tolerating two to four times more imprecise positioning information, and can 
double the road capacity and increase the traffic flow by 25%.
Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Ziyuan Wang, November 2009
Acknowledgements

First and foremost, I would like to thank my supervisors Dr. Lars Kulik and Prof. Rao Kotagiri for their support, encouragement, views and comments during my PhD candidature. At many occasions, they provided me early ideas and made incisive comments and constructive suggestions on my research. Without their supervision, this thesis would not have been completed. I am grateful to them for providing the opportunity to work in a group of brilliant researchers in the Sensing, Ubiquity, and Mobility (SUM) Lab.

I acknowledge National ICT Australia (NICTA) and the University of Melbourne for providing scholarships and travel supports to pursue doctoral studies and attend international conferences.

I wish to give heartful thanks to my parents for their support, dedication and love at all times. Without them it would not be possible to come through the various stages of my life.

I am thankful to Christian Vecchiola, Marcos Dias de Assunção, Marco Netto, and Chee Shin Yeo for their help in my research. On several occasions, they provided excellent advice and insights by reading the early drafts of my papers, and helped to improve the structure, presentation and correctness of those manuscripts.

I would like to thank all the past and present members of the SUM Lab. In particular, I thank Archana Sathivelu, Adeel Zafar, Hairuo Xie, Muhammad Umer, Eunus Ali, Tanzima Hashem, and Parvin Asadzadeh for their support and comments on my work. I also had great time with them outside research. I enjoyed a number of barbeques, picnics, outdoor activities, sightseeing, and sports with them and my other
friends from the CSSE department, Lei Ni, Andreas Schutt, Jason Lee, and Jubaer Arif, to name a few. Special thanks to my ex-housemate Jenny Lin for her friendship and company. I also extend my gratitude to the staff from the CSSE department and School of Engineering, especially, Binh Phan, Pinoo Bharucha, Julien Reid, and Madalain Dolic for their support during my candidature.

Finally, I would like to thank Mukaddim Pathan for his support and encouragement. I sincerely thank him for the patience and understanding when I was too busy writing my thesis.
To my parents.
# Contents

1 Introduction 1
   1.1 Motivation ............................................. 2
   1.2 Research Problem and Objectives ....................... 5
      1.2.1 Research Objectives ............................... 5
   1.3 Contributions ......................................... 6
   1.4 Structure of the Thesis ................................ 7
   1.5 List of Publications .................................... 8
      1.5.1 Published Papers .................................. 9
      1.5.2 Papers under Review ............................... 9

2 Literature Review 11
   2.1 Vehicular Traffic Models ............................... 11
      2.1.1 Longitudinal Models ............................... 13
      2.1.2 Lane-Changing Models ............................. 16
      2.1.3 Vehicular Traffic Theory ......................... 20
   2.2 Sensors ................................................ 24
      2.2.1 Position ............................................ 25
      2.2.2 Velocity and Acceleration ......................... 26
      2.2.3 Communication ...................................... 27
      2.2.4 Applications ....................................... 28
   2.3 Flow Control Strategies ............................... 30
      2.3.1 Centralized and Decentralized Systems .......... 32
      2.3.2 Automation Control .................................. 35
      2.3.3 Driver Assistance Systems ........................ 37
      2.3.4 Ramp Metering ...................................... 38
      2.3.5 Diversion ........................................... 40
   2.4 Conclusion ............................................. 41

3 Traffic Model and Evaluation Measures 43
   3.1 Introduction ........................................... 43
   3.2 Employed Traffic Model ............................... 44
      3.2.1 IDM ................................................. 44
      3.2.2 Impact of Parameter Values ......................... 47
      3.2.3 MOBIL ................................................ 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Evaluation</td>
<td>50</td>
</tr>
<tr>
<td>3.3.1 Safety</td>
<td>50</td>
</tr>
<tr>
<td>3.3.2 Efficiency</td>
<td>51</td>
</tr>
<tr>
<td>3.4 Conclusion</td>
<td>54</td>
</tr>
<tr>
<td>4 Microscopic Traffic Simulator Incorporating Sensor-Enabled Cars</td>
<td>55</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>55</td>
</tr>
<tr>
<td>4.2 Overall Design</td>
<td>57</td>
</tr>
<tr>
<td>4.3 Implementation</td>
<td>57</td>
</tr>
<tr>
<td>4.3.1 Road Networks</td>
<td>58</td>
</tr>
<tr>
<td>4.3.2 Vehicle Movements</td>
<td>60</td>
</tr>
<tr>
<td>4.4 Recommendation System</td>
<td>65</td>
</tr>
<tr>
<td>4.4.1 Tolerance for Drivers’ Mistakes</td>
<td>65</td>
</tr>
<tr>
<td>4.4.2 Personalization</td>
<td>65</td>
</tr>
<tr>
<td>4.5 Graphical User Interface</td>
<td>66</td>
</tr>
<tr>
<td>4.6 Conclusion</td>
<td>66</td>
</tr>
<tr>
<td>5 Proactive Merging Algorithms</td>
<td>69</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>69</td>
</tr>
<tr>
<td>5.2 Design Procedure</td>
<td>70</td>
</tr>
<tr>
<td>5.3 Single Lane Scenario</td>
<td>71</td>
</tr>
<tr>
<td>5.3.1 Algorithms for the Main Road with a Single Lane</td>
<td>71</td>
</tr>
<tr>
<td>5.3.2 Pseudocode for Merging Algorithms</td>
<td>74</td>
</tr>
<tr>
<td>5.4 Multiple Lane Scenario</td>
<td>78</td>
</tr>
<tr>
<td>5.4.1 Lane-Changing Model</td>
<td>78</td>
</tr>
<tr>
<td>5.4.2 Algorithms for the Main Road with Multiple Lanes</td>
<td>78</td>
</tr>
<tr>
<td>5.5 Simulation Results</td>
<td>80</td>
</tr>
<tr>
<td>5.5.1 Results of the Single-Lane Scenario</td>
<td>80</td>
</tr>
<tr>
<td>5.5.2 Results of the Multi-Lane Scenario</td>
<td>91</td>
</tr>
<tr>
<td>5.6 Conclusion</td>
<td>102</td>
</tr>
<tr>
<td>6 Robustness for Merging Algorithms</td>
<td>103</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>103</td>
</tr>
<tr>
<td>6.2 Time Geography</td>
<td>104</td>
</tr>
<tr>
<td>6.3 Robustness of Merging Algorithms</td>
<td>106</td>
</tr>
<tr>
<td>6.3.1 Factors of Robustness</td>
<td>106</td>
</tr>
<tr>
<td>6.3.2 Criteria of Robustness</td>
<td>107</td>
</tr>
<tr>
<td>6.3.3 Design of Robust Merging Algorithms</td>
<td>107</td>
</tr>
<tr>
<td>6.4 Simulation Results</td>
<td>114</td>
</tr>
<tr>
<td>6.4.1 Impact of Accuracy Level on Safety</td>
<td>114</td>
</tr>
<tr>
<td>6.4.2 Impact of Accuracy Level on Efficiency</td>
<td>118</td>
</tr>
<tr>
<td>6.5 Higher Level of Realism</td>
<td>123</td>
</tr>
<tr>
<td>6.5.1 Sensor Coverage</td>
<td>123</td>
</tr>
<tr>
<td>6.5.2 Heterogeneity of Vehicles</td>
<td>125</td>
</tr>
<tr>
<td>6.5.3 Impact of Speeding</td>
<td>125</td>
</tr>
</tbody>
</table>
List of Figures

1.1 A vehicular ad-hoc network (VANET) ........................................ 3
1.2 Comparison of traffic merging strategies .................................... 4

2.1 Lane-changing types ............................................................... 17
2.2 Fundamental diagram of traffic flow ......................................... 21
2.3 Alternative shapes of the flow-density relationships ................. 22
2.4 Phase transitions in three-phase traffic theory ......................... 23
2.5 PATH architecture ................................................................. 33
2.6 Dolphin architecture ............................................................... 34

3.1 Acceleration and velocity ......................................................... 45
3.2 Relationship between the velocity, the distance to the front car and the velocity difference $\Delta v$ between two cars .................................................. 46
3.3 Lane-changing model: MOBIL .................................................. 49

4.1 Structure of our traffic simulator ............................................. 57
4.2 Class diagram of the traffic simulator (\( \star \) denotes one or more instances and 1 denotes exactly one instance.) ......................... 59
4.3 Example of sensor error distribution: positioning accuracy within 1 m 61
4.4 Flow chart of the main road cars’ update process ....................... 62
4.5 Flow chart of the ramp cars’ update process ............................. 64
4.6 Graphical user interface of our simulator .................................. 67

5.1 Notations of proactive merging algorithms ............................... 72
5.2 Notations for CPV .................................................................. 79
5.3 Illustration of lane switch in CPV ............................................ 79
5.4 Light initial density of the main road ........................................ 83
5.5 Medium initial density of the main road ................................... 84
5.6 Heavy initial density of the main road ...................................... 85
5.7 Medium initial density of the main road with low incoming rate of the ramp ........................................................................ 87
5.8 The decision point varies from 50 to 400 m away from the merging point 88
5.9 The performance of merging algorithms with a short ramp .......... 89
5.10 Incoming number of cars following the Poisson distribution with $\lambda = 6$ 90
5.11 Incoming number of cars following the Poisson distribution with $\lambda = 12$ 91
<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.12</td>
</tr>
<tr>
<td>5.13</td>
</tr>
<tr>
<td>5.14</td>
</tr>
<tr>
<td>5.15</td>
</tr>
<tr>
<td>5.16</td>
</tr>
<tr>
<td>5.17</td>
</tr>
<tr>
<td>5.18</td>
</tr>
<tr>
<td>6.1</td>
</tr>
<tr>
<td>6.2</td>
</tr>
<tr>
<td>6.3</td>
</tr>
<tr>
<td>6.4</td>
</tr>
<tr>
<td>6.5</td>
</tr>
<tr>
<td>6.6</td>
</tr>
<tr>
<td>6.7</td>
</tr>
<tr>
<td>6.8</td>
</tr>
<tr>
<td>6.9</td>
</tr>
<tr>
<td>6.10</td>
</tr>
<tr>
<td>6.11</td>
</tr>
<tr>
<td>6.12</td>
</tr>
<tr>
<td>7.1</td>
</tr>
<tr>
<td>7.2</td>
</tr>
<tr>
<td>7.3</td>
</tr>
<tr>
<td>7.4</td>
</tr>
<tr>
<td>7.5</td>
</tr>
<tr>
<td>7.6</td>
</tr>
<tr>
<td>7.7</td>
</tr>
<tr>
<td>7.8</td>
</tr>
<tr>
<td>7.9</td>
</tr>
<tr>
<td>7.10</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Traffic models overview</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Automotive sensors</td>
<td>27</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of automotive sensors</td>
<td>27</td>
</tr>
<tr>
<td>2.4</td>
<td>Overview on current applications</td>
<td>31</td>
</tr>
<tr>
<td>2.5</td>
<td>Overview of merging strategies</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>IDM parameters</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>Traffic merging algorithms overview</td>
<td>74</td>
</tr>
<tr>
<td>5.2</td>
<td>Traffic merging algorithm CPV overview</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>Experiment settings for main road with a single lane</td>
<td>82</td>
</tr>
<tr>
<td>5.4</td>
<td>Experiment settings for main road with multiple lanes</td>
<td>92</td>
</tr>
<tr>
<td>6.1</td>
<td>Traffic merging algorithms overview</td>
<td>114</td>
</tr>
<tr>
<td>6.2</td>
<td>Experiment settings</td>
<td>114</td>
</tr>
<tr>
<td>6.3</td>
<td>The level of robustness and capacity</td>
<td>118</td>
</tr>
<tr>
<td>7.1</td>
<td>Experiment settings</td>
<td>131</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

Road traffic congestion is a major challenge nowadays. Congestion leads to negative effects in terms of road safety, travel times, fuel consumption, and air pollution [43, 128]. According to a 2005 US study [45], the major cause of traffic congestion is the bottlenecks on highways, which attributes to 40% of congestions. Typical examples of bottlenecks are intersections where an on-ramp leads to the main highway, a blocked lane section caused by obstacles, and construction activities on the road (i.e., work zones). Bottlenecks have a common characteristic: several traffic streams merge to fewer streams, hence the road capacity reduces.

Sensor-enabled cars provide an opportunity to improve road capacity at bottlenecks and reduce traffic congestion. The integration of sensing, communication and local computing within cars facilitates to enhance the efficiency of transportation systems. Currently, a variety of automotive sensors are available to collect data related to a vehicle and its surroundings.

This thesis focuses on improving capacity at merging bottlenecks with the use of sensor-enabled cars. We explore the benefits of simple traffic rules at merging areas when all cars are equipped with sensors. These sensor-enabled cars exchange sensed information such as position, velocity and acceleration for making merging decisions. The traffic rules include car-following and lane-changing rules, which are designed to increase the traffic flow. For example, if a car in front speeds up, the following car will also accelerate while maintaining a safe distance. A lane change on a multi-lane main road is ideal to make space for a merging car, which can increase the merging
capacity and reduce the delay. We address the problem by the following three research questions:

1. how to use local spatial information in traffic merging strategies;
2. how to address the adverse impact of inaccuracies in sensor measurements; and
3. how local decisions impact global performance.

Information and communication technologies assist the development of the intelligent transportation system (ITS). A range of applications aim to use the road more efficiently, such as traffic monitoring, traffic signal control, vehicle navigation, and variable message signs (VMS). A more recent research area of ITS is the vehicular ad-hoc network (VANET), which is an autonomous system of vehicles connected by wireless links without the need for separate infrastructures. In addition, communication techniques, especially, the Dedicated Short Range Communications (DSRC) support the information exchange between cars. For example, General Motors (GM) is developing sensor-enabled cars with sensing and Vehicle-to-Vehicle (V2V) [1] communication abilities. Current studies primarily focus on: safety by reducing traffic accidents, e.g., based on collision warnings [40]; traffic flow control by adapting traffic lights and speed limits [110]; and automation to reduce a driver’s burden, e.g., using adaptive cruise control (ACC) [82]. ITS and VANETs contribute to the improvement of road utilization, however, our analysis of the previous work has revealed that there is a lack of research on traffic merging strategies.

1.1 Motivation

The ultimate goal of traffic control is to mitigate traffic congestions and ensure road safety. Building more roads is often not a viable solution to deal with traffic congestion, in particular in metropolitan areas where space is limited. Moreover, the congestion does not occur only because of the excessive traffic demand over the road capacity but also because of the inefficient utilization of roads and other factors such as traffic
1.1 Motivation

accidents, road repairs, and temporary blocks on certain roads, which decrease the traffic throughput. There are two types of imbalances that affect the throughput adversely: (1) global factors impacting the entire network. For instance, considering the entire road network, some roads are underutilized while others are heavily congested. (2) the local factors that are responsible for local perturbations. Local perturbations can result from an improper driving car whose impact may get amplified along the road, thus leading to a reduced traffic throughput. For example, if a car pushes into a small gap to change lane, the impact may appear small locally but is globally significant as the following cars may need to slow down considerably. This is known as “slinky-type effect” [16], which can lead to traffic congestions and car accidents.

In this thesis, we argue that traffic control strategies integrating the on-board sensor information can reduce merging bottlenecks. We propose collision-free strategies for efficient traffic merging, which significantly reduces the bottlenecks. Figure 1.1 illustrates a VANET consisting of cars on a highway and ramp merging section. The dashed lines with arrows indicate that sensor-enabled cars can exchange their sensed spatial information locally.

![Figure 1.1: A vehicular ad-hoc network (VANET)](image)

As an example of motivating the concept in this thesis called *proactive* traffic merging strategy, we consider the following highway merging scenario shown in Figure 1.2, where a ramp car merges onto a main road. In this merging scenario, we consider two cases: merging strategy with and without sensor information. In both cases we assume the same initial configuration (middle figure), such as the positions and velocities of the cars. In the upper figure we assume a priority-based merging algorithm in
which a car does not adapt its velocity before it arrives at the merging section. This requires car $x$ to slow down considerably in order to merge onto the main road. In the lower figure, we assume that car $x$ adapts its velocity before its arrival at the merging section and can merge immediately when it arrives at this section. This is the core idea of the proposed proactive merging strategies. Using sensed information, the merging process will be “smoother” because a car on the ramp can adjust its velocity earlier to adapt to the gaps between the cars on the main road and their velocities. This leads to smaller interruptions for the main road flow. A positive side-effect of our strategy is a small number of local perturbations, i.e., a smaller number of velocity changes, which reduces fuel consumption and air pollution.

![Diagram of traffic merging strategies](image)

Figure 1.2: Comparison of traffic merging strategies

Traditional traffic management strategies, such as adaptive speed limits, dynamic route guidance, and ramp metering use centralized system design. However, a centralized system design suffers from a single point of failure and the lack of robustness. In particular, the following challenges for traditional traffic management strategies motivate us to take an alternative approach:
1.2 Research Problem and Objectives

- The number of cars for a whole road network is very large. Therefore, the scalability of traffic control strategies is a challenge.

- Measurements of traffic conditions are limited in locations with inductive loop detectors.

- Many unpredictable and hardly measurable disturbances (e.g., incidents, illegal parking, pedestrian crossings, and intersection blocking) may interrupt the traffic flow.

The first challenge is the main reason that we adopt a decentralized system design, which is inherently scalable. The second and the third challenges can take advantage of sensor-enabled cars that facilitate sensing and communicating an extensive set of information.

1.2 Research Problem and Objectives

This thesis addresses the challenge of how to reduce or eliminate merging bottlenecks. Our research develops merging algorithms that will assist drivers in making more informed and efficient decisions, such as when to accelerate or brake, to change lane or to enter a multi-lane road, under the constraints imposed by physical limitations and traffic rules. The proposed algorithms are based on the fact that sensor-enabled cars have readily available spatial information, such as location, velocity, and acceleration or deceleration. In this thesis, we make the following two assumptions: all cars are sensor-enabled and the communication is reliable.

1.2.1 Research Objectives

Based on the research problem described above, we outline the following objectives:

1. Develop algorithms that proactively merge different traffic streams aiming at optimizing the traffic throughput. We study what benefits we get from using the sensor-enabled cars.
2. Investigate our merging algorithms in terms of robustness considering the accuracy of sensor measurements.

3. Investigate the scalability of traffic merging algorithms, in particular, at multiple intersections of ramps and the main road.

1.3 Contributions

In this thesis, we make the following contributions:

1. We develop proactive traffic control algorithms that use the current road facilities efficiently, including the improvement of overall traffic throughput, decrease of the merging delay and fuel consumption. The proposed proactive merging algorithms separate the decision point, and the actual merging point to adjust the velocity of a merging car in advance. Our observations reveal that conventional road traffic control strategies, such as intersection signal control and ramp metering will benefit from this work.

2. We investigate different merging algorithms under inaccurate sensor measurements in terms of distance and explicitly develop robust algorithms that minimize the adverse effect of positioning accuracies. We apply the concept of space-time prisms or cones in the developed time geography-based merging algorithm to compute the “smoothest” acceleration for involved cars and guarantee safe merging while tolerating four times more imprecise positioning information.

3. We investigate the criteria to better evaluate the performance of traffic control algorithms. We evaluate a range of algorithms using the performance criteria from safety and efficiency perspectives and show how to assess merging algorithms appropriately.

4. We design a controlled simulation environment to test various traffic control strategies. Using this simulator, we conduct extensive experiments and evaluation of the performance of the proposed algorithms. We also study different
traffic patterns such as Poisson arrivals and highly bursty traffic in order to understand traffic behaviors.

1.4 Structure of the Thesis

The rest of this thesis is structured as follows:

Chapter 2 – Literature Review provides the background and analysis of existing literature on methods and approaches of traffic flow control with the assistance of new technologies. It surveys the state of the art in the area in terms of vehicular traffic theories, traffic models, automotive sensors, and merging strategies. A classification of flow control strategies based on their architecture, as well as the properties of these approaches are detailed in this chapter. This chapter establishes the foundation for basic understanding of the existing literature and lay out a guideline for the following chapters.

Chapter 3 – Traffic Model and Evaluation Measures describes the microscopic traffic model and evaluation criteria for traffic merging algorithms. Based on the extensive survey of traffic models, we present how we modify an existing traffic model to satisfy the purpose of studying merging algorithms. Furthermore, we show how to assess merging algorithms appropriately by identifying a range of performance criteria from safety and efficiency perspectives, such as number of violations, delay, flow, capacity, and fuel efficiency.

Chapter 4 – Microscopic Traffic Simulator Incorporating Sensor-Enabled Cars presents the development of our traffic simulator, which is specialized for the investigation of merging scenarios on highways, such as on-ramps, obstacle avoidance, and lane closure at work zones. We describe the design and implementation of the simulator. Initially, we build a single loop main road with one on-ramp and one off-ramp as a foundation for extensive study of merging algorithms locally. Then the simulator is extended to multiple loops connected
with multiple on and off-ramps, which can construct arbitrary road networks to explore the global influence of merging strategies.

Chapter 5 – Proactive Merging Algorithms proposes proactive strategies for cooperatively merging different traffic flows at highway ramps. The aim of the proposed strategies is to decrease the merging delay and improve the overall traffic flow. We first study merging strategies for the main road with a single lane and then we extend our work to the scenario where the main road consists of multiple lanes.

Chapter 6 – Robustness for Merging Algorithms investigates how the accuracy of sensors impacts merging algorithms. The accuracy level of sensors is a major challenge for merging algorithms, since inaccuracies can potentially lead to unsafe merging behaviors. In this context, we develop robust algorithms that are tailored to minimize the adverse effect of positioning inaccuracies.

Chapter 7 – Merging Algorithms for Traffic Diversion at Multiple Ramps explores the modeling of multiple merging locations. The main focus of this chapter is to divert traffic at multiple merging points to reduce congestions. We study traffic diversion strategies when there are two ramps. It is an essential step to study large scale road networks.

Chapter 8 – Conclusions and Future Directions concludes this thesis and highlights a number of future research directions.

Appendix A provides all the notations used in this thesis.

1.5 List of Publications

The following publications are the outcomes during the PhD candidature.
1.5 List of Publications

1.5.1 Published Papers


1.5.2 Papers under Review

Chapter 2
Literature Review

The main aim of the thesis is to integrate sensing data and communication with traffic flow control algorithms. Consequently, this chapter includes three components: vehicular traffic models, sensors, and flow control strategies. We first review research on traffic models and theories. These models provide the foundation to explore the effects of implementing new traffic control strategies. We highlight some basic vehicular traffic theory concepts, with a focus on the traffic congestion on highways to provide a better understanding of traffic flow and design of traffic flow control strategies. Then, we review automotive sensors that can be implemented in the sensor-enabled cars. Those sensors collect traffic information as the input for the traffic model applied in traffic control strategies. Finally, we survey the flow control strategies in the area of automation control, driver assistance systems, ramp metering, and diversion approaches.

2.1 Vehicular Traffic Models

Traffic models and simulations provide an understanding of traffic phenomena to manage traffic in such a way that traffic flow is optimized and congestion is alleviated. From a computational perspective, various tools and techniques such as car-following theories and cellular automata have been applied to traffic modeling.

Traffic models can be classified according to two criteria. First, according to the use of space, time and state variable, traffic models can be categorized as continuous or discrete. Earlier theoretical approaches, in particular, fluid dynamic theory, kinetic
theory, and car-following theories are continuous. Cellular automata (CA) models are discrete. Second, depending on the level of detail of the simulation, there are macroscopic and microscopic models. Macroscopic models are based on equations to describe the collective characteristics of traffic, such as average velocity and traffic density. Researchers often build such models by adapting physical models similar to the behavior of fluid and gases. Recognizing that unlike molecules, cars move under the control of people, drivers’ behaviors have been taken into account. Macroscopic models are sufficient to predict the general patterns of the cumulative number of vehicles. In general, they are suitable to describe large-scale traffic networks such as an entire traffic of a city. They are widely used in the design of freeway facilities, and they form the basis of nearly all model-based on-ramp metering designs. However, they fail to represent the interactions among vehicles. They cannot describe variations in velocities, as velocities vary in overtaking situations or within platoons. Thus, macroscopic models are not adequate to evaluate the performance of local merging algorithms.

We first discuss a classical mobility model that is widely used: random waypoint model (RWP). RWP was the first macroscopic model used by Johnson and Maltz [70], and it has become the de facto standard in mobile computing research, including vehicular ad hoc networks (VANETs). In the RWP model, the waypoints are uniformly distributed over a given area. The mobile nodes move randomly from one waypoint to the next. At the start of each trip a random velocity is chosen from the velocity distribution.

RWP has several limitations as a model for VANETs. First, it describes the movement pattern of independent nodes, however, the movement of vehicles is not independent as vehicles interact with other vehicles in their vicinity. Second, RWP assumes that node movement occurs in an open field, whereas the movement of vehicles is constrained by the roads. Third, the node chooses a random velocity from the uniformly distributed velocity range in RWP. In contrast, a vehicle’s velocity is constrained by many factors, such as the road conditions, traffic lights, the number and type of intersections, and the existence of obstacles. Therefore, mobility models based on real
street maps have been proposed for VANETs [25,124]. In Saha and Johnson’s model (SJM) [124], street map data is converted to \((x, y)\) coordinates. The road segments are represented as the edges and the intersections are represented by the nodes in an undirected graph. The STRAW approach [25] includes intersection management, such as the coordination of traffic lights and stop signs. Saha and Johnson point out that RWP is a good approximation of car movement in many cases. While it is often sufficient for evaluating the communication performance of a VANET, it is not a good approximation to evaluate a flow control strategy for a real road. Although Saha and Johnson [124] applied real street maps for simulating vehicle movement, some assumptions restrict its application to flow control strategies, which rely on realistic vehicular traffic characteristics. In this regard, models with varying velocity are highly needed. As a result, vehicular traffic models have attracted extensive research.

In contrast, microscopic models overcome these limitations by describing the individual behavior of vehicles. Microscopic approaches include car-following models [9] and cellular automata (CA) models [26]. We detail microscopic models from the longitudinal and the lane-changing aspects in Section 2.1.1 and 2.1.2.

### 2.1.1 Longitudinal Models

Longitudinal models describe the acceleration and deceleration of vehicles. Generally, there are two types: car-following models and cellular automata (CA) models.

In car-following models, the assumption is that the behavior of a driver depends only on its predecessor. In general, the basic idea of car-following models can be regarded as the response of the vehicle to the stimulus it receives [26,62], i.e.,

\[
[Response]_n \propto [Stimulus]_n
\]  

(2.1)

for the vehicle \(n (n = 1, 2, \ldots)\) and \(\propto\) could represent linear or nonlinear relationship.

Each driver responds to the surrounding traffic conditions by maintaining the velocity, accelerating or decelerating the vehicle. Different car-following models repre-
sent the function of stimulus differently, which can be summarized by the following equation [26]:

$$\dot{v}_n = f_{sti}(v_n, \Delta x_n, \Delta v_n)$$  \hspace{1cm} (2.2)

where the function $f_{sti}$ is the stimulus received by the vehicle $n$, $\Delta x_n$ is the distance, and $\Delta v_n$ is the velocity difference. We categorize car-following models into three groups based on the major parameters: velocity-dependent, distance-dependent, and velocity-difference-dependent models.

In velocity-dependent models, also known as follow-the-leader models (FTL), a car tends to the same velocity as its front car. In the following we label vehicles in the opposite of driving direction such that the vehicle $n - 1$ is in front of the vehicle $n$:

$$\dot{v}_n(t) = \frac{1}{\tau}[v_{n-1}(t) - v_n(t)]$$  \hspace{1cm} (2.3)

where $\frac{1}{\tau}$ is the sensitivity of the driver responding to the stimulus. Most of such models and their modifications were proposed in the 1950s and 1960s. They have two major limitations. First, such models cannot capture the clustering characteristic of real traffic resulting from the ignorance of the distance. Second, as there is no parameter related to density, these models cannot derive fundamental relation between flow and density. Pipes [117] concludes that: (1) a higher velocity requires a larger distance; and (2) cars must maintain a safe distance to avoid collisions. Herman et al. [63] and Gipps [53] investigate driver’s strength of response, or, sensitivity. They point out that a smaller distance leads to higher sensitivity.

To overcome the limitations of follow-the-leader models, distance-dependent models have been introduced recently. To maintain a safe distance, in follow-the-leader models the leader’s velocity is chosen as the follower’s desired velocity, i.e., $v_{n-1}(t) = V_{n}^d(t)$. From (2.3) we can derive:

$$\dot{v}_n(t) = \frac{1}{\tau}[V_{n}^d(t) - v_n(t)]$$  \hspace{1cm} (2.4)

where $V_{n}^d(t)$ is the desired velocity of the vehicle $n$ at time $t$. In contrast, Bando et
al. [9] propose optimal velocity model (OVM), where desired velocity, also called optimal velocity, depends on the distance between two cars, i.e., $V_d^n(t) = V^{opt}(\Delta x_n(t))$. Therefore,

$$\dot{v}_n(t) = \frac{1}{\tau} [V^{opt}(\Delta x_n(t)) - v_n(t)]$$

(2.5)

Its main drawback is that it produces unrealistic acceleration when a fast car approaches a stopped car in front of it.

To avoid the drawback of OVM, velocity-difference-dependent models are proposed. We highlight the intelligent driver model (IDM) [135], which considers relative velocity to the front car. A car tends to approach the maximum permitted velocity and maintains at least the safety distance to its front car. The safety distance depends on the car’s velocity and velocity difference to its front car. The acceleration and deceleration depend on its own velocity, safety distance and actual distance to the front car. Thus, the velocity difference between two cars becomes the key parameter instead of the velocity or distance in the previously discussed models. IDM has the following advantages [134]: it is a realistic description of both the individual driving behavior and collective patterns of the traffic flow, such as stop-and-go waves; and many aspects of traffic control strategies can be simulated by representing different driving styles, which are easy to implement. Therefore, we adopt IDM in our traffic simulator. We will discuss IDM in detail in Chapter 3.

Nevertheless, the computational cost of such detailed models is very high and it is a challenge to apply them to real large networks. Cellular automata (CA) are designed for large-scale simulations, therefore, they can be efficiently used to describe traffic flow. In CA models [26], the position, velocity, acceleration, and time are discrete variables. In this approach, a lane is represented by one-dimensional cells. Each cell can be empty or occupied by at most one vehicle at an instant of time. At each discrete time step, the state of the system is updated according to predefined rules. The computational efficiency of the CA models is the major advantage compared to car-following models. In 1992, Nagel and Schreckenberg [101] proposed a simple microscopic CA model that describe the motion of vehicles using a set of update rules. The
rules include 1) acceleration, 2) deceleration, 3) randomization, and 4) vehicle movement. The first two rules describe an optimal driving strategy, the driver accelerates if the velocity is lower than the maximum velocity and brakes to avoid accidents. Such a CA model is deterministic and the stationary state depends only on the initial conditions. However, drivers do not always exhibit optimal behaviors. Driving behaviors change without any obvious reasons. Therefore, it is necessary to present the parameter "slowdown probability" $p$, which is essential for a realistic representation of traffic flow. It imitates the complex interactions between vehicles and is also responsible for the spontaneous formation of traffic jams. Although it is one of the simplest traffic models, it is capable of reproducing important properties of traffic flow, such as the density-flow relation and the spatio-temporal evolution of traffic jams. Kerner-Klenov model [77, 78] is the first CA model attempted to represent phase transitions in traffic flow. CA models are oversimplified for our work because they abstract the acceleration and deceleration process, which is an essential study subject in our work.

2.1.2 Lane-Changing Models

Besides longitudinal models, realistic models of lane changing behaviors are critical for investigating traffic patterns at merging sections. Based on car-following models, a number of lane-changing models are proposed. Two types of lane changes: mandatory and discretionary, are classified in most models, for example, CORSIM [57], MITSIM [148] and SITRAS [64, 65]. A discretionary lane change is executed when a driver wants to improve driving conditions, for example, to gain speed. A mandatory lane change occurs when a driver must leave the current lane in order to maintain the route to the destination (for example, changing lane from a closed lane or changing lane to exit). Figure 2.1 illustrates typical lane changing scenarios.

The discretionary and mandatory lane changing is based on the critical gap acceptance behavior, which means drivers search for a minimum acceptable gap, called critical gap. Traditional gap acceptance models cannot capture lane changing behaviors in congested traffic, because there are not sufficient acceptable gaps. Several stud-
ies consider lane-changing in congested traffic. Besides discretionary and mandatory lane-changing, Ahmed et al. [3] identify forced lane changing, where a driver forces other drivers to create an acceptable gap. Hidas [64, 65] proposes cooperative lane-changing, where the back car on the target lane slows down to allow the subject car to enter.

Gipps [54] introduces a model that includes a hierarchy of decision-making that determines the necessity and desirability of lane changes.

Kesting et al. [81] propose to minimize the overall braking induced by a lane change (MOBIL) to derive lane-changing rules for discretionary and mandatory lane changes for a wide class of car-following models. Both the advantage of a given target lane and the disadvantage associated with lane changes are determined in terms of longitudinal accelerations calculated with car-following models. The minimum safety distance prevents critical lane changes and collisions. The incentive criterion takes into account the advantages and disadvantages of other drivers affected by a lane change via the “politeness factor”. This parameter offers a variety of motivation for lane changing.
from purely selfish to more cooperative.

Toledo [132] proposes a framework for modeling driving behaviors, which integrates lane changing and car-following. This integration is achieved by a concept of “short-term plan”. Drivers who want to change lanes but cannot accomplish immediately, set a short-term plan to perform the desired lane change. Short-term plans are defined by a set of gaps in the target lane. Drivers adapt their acceleration behavior to facilitate the lane change using the target gap. Therefore, inter-dependencies between lane changing and acceleration are represented. Compared to independent lane-changing and car-following models in a microscopic simulator, the integrated model achieves better approximation for the observed velocity, travel time, and distribution of vehicles on a four-lane highway. Similar to the “short-term plan”, we implement merging algorithms considering alternative gaps within a car’s communication range and adjust a car’s velocity according to the target gap.

Bunker and Troutbeck [18] present a gap acceptance model for highway merging, which gives main road limited priority. In the limited priority model, drivers in the main road at a merging area may incur delay to restore small gaps to larger gaps between them and their front vehicles. This allows ramp cars to accept smaller gaps. They focus on the ramp delay varying arrival patterns and find out that constant incoming rate controlled by ramp metering leads to less delay than when the ramp cars arrive in bunches. The results indicate the importance of arrival patterns that impact the merging process. We will later show that our merging algorithms optimize merging not only for the constant arrival but also for the incoming following the Poisson distribution.

A widely adopted macroscopic merging model, which is of great importance for the design of microscopic merging model, is proposed by Daganzo [31]. The following merging rules are defined to satisfy three conditions: (1) the discharge flow of one stream $d_M^i$ should not exceed the capacity of this stream; (2) the total discharge flow of two streams should not exceed the merge capacity $d_M^j$; and (3) when there is a queue in one stream, the merge capacity should be shared by the two streams in a ratio of $d_M^i / d_M^j \geq \lambda^i$, where $\lambda^i$ is a “merge-priority” constant. For two streams A and
\( B, \lambda^A + \lambda^B = 1 \). If both streams are queued, then \( \frac{dA}{dt} = \lambda^A \). The rules of the model represent traffic in agreement with empirical observations by [19], which observed that queued vehicles from the on-ramp and highway streams merge in congested situations at some fixed ratio. Chevallier and Leclercq [24] examine whether microscopic merging models reproduce the priority sharing ratio. They indicate that observation cannot be represented due to the constraints imposed by car-following models on gap-acceptance models. They point out that more complicated gap-acceptance models considering cooperation or forced merging can overcome this problem. Similarly, our merging algorithms enable cooperation among vehicles on the main road and the ramp, thus, merging at fixed ratio is reflected in our simulation as well.

Another research direction of lane-changing models is the generalization of CA models in multi-lane traffic. A variety of models are proposed based on different lane-changing rules. These rules all include two aspects. First, there must be an incentive to change lane. Two typical incentives are 1) the driving conditions on the other lane are more convenient; and 2) the need to enter or exit the highway through ramps. Second, safety criteria must be satisfied. In general, safety criteria are defined by minimum gaps in front and back on the other lane. Nagel et al. [102] summarize a large number of lane-changing rules in the literature and present a general scheme to develop lane-changing rules. In spite of differences among several lane-changing rules, their simulations show that the scheme generates similar and realistic results. Although our model is not discrete as CA models, we still follow the incentive and safety rules of lane-changing.

Based on the Nagel-Schreckenberg CA model, Pedersen and Ruhoff [114] develop a method to include ramps. The idea is to place “shadow cars” on a highway parallel to cars on ramps. Thus, drivers on the highway can react with ramp cars. We also use this idea in our work. Similar to this model, several other work also considers the extension of CA models at ramps to study the impact of merging traffic [12, 105].

The traffic model is a bridge between the communication and the traffic control algorithms. It should reflect vehicle movements as well as the impact of communica-
tion on the traffic flow. Härrt et al. [60] provide a comprehensive survey on mobility models for VANETs. A thorough analysis of the topological properties of a vehicular network can be found in [47]. It shows the physical reasons underlying the connectivity dynamics generated by a variety of mobility models. A survey of microscopic models can be found in [131].

Table 2.1 summarizes the models we have discussed. We categorize them according to three criteria: microscopic (Mic) or macroscopic (Mac), discrete(Dis) or continuous (Con), and deterministic (Det) or stochastic (Sto).

<table>
<thead>
<tr>
<th>Model</th>
<th>Microscopic /Macroscopic</th>
<th>Discrete /Continuous</th>
<th>Deterministic /Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWP [70]</td>
<td>Mac</td>
<td>Con</td>
<td>Sto</td>
</tr>
<tr>
<td>SJM [124]</td>
<td>Mac</td>
<td>Con</td>
<td>Sto</td>
</tr>
<tr>
<td>STRAW [25]</td>
<td>Mac</td>
<td>Con</td>
<td>Sto</td>
</tr>
<tr>
<td>FTL [53,117]</td>
<td>Mic</td>
<td>Con</td>
<td>Det</td>
</tr>
<tr>
<td>OVM [9]</td>
<td>Mic</td>
<td>Con</td>
<td>Det</td>
</tr>
<tr>
<td>IDM [135]</td>
<td>Mic</td>
<td>Con</td>
<td>Det</td>
</tr>
<tr>
<td>CA [101]</td>
<td>Mic</td>
<td>Dis</td>
<td>Sto</td>
</tr>
</tbody>
</table>

### 2.1.3 Vehicular Traffic Theory

Road efficiency is measured by the traffic flow. It is the product of the number of cars and the velocity. The number of cars or velocity separately is not sufficient to describe efficiency. For example, in a traffic congestion there could be a huge number of cars on a road section but they cannot move. In contrast, if there is only one car on a road, although its velocity can be as high as possible, the road is underutilized. One of the first studies that attempt to demonstrate the relationship among velocity, density, and flow was by Greenshields in 1935 [55]. He employed photographic images to estimate aggregate velocities and densities on a straight two-lane road, and found that they could be well approximated by a straight line. Lighthill and Whitham in 1955 [90] were the first to propose a macroscopic traffic model using Greenshields’ hypothesis of a static flow-density relationship.
2.1 Vehicular Traffic Models

The density-flow relation forms the basis of the so-called *fundamental diagram*. The fundamental diagram is of major importance for the design of road systems and traffic management systems. It can be used to predict the capacity of a road, or its behavior when applying ramp metering or speed limits. The origin of the typical shape of the fundamental diagrams (see Figure 2.2) can be understood as follows. When the density $D$ is sufficiently small, the average velocity $V$ is independent of $D$ since the vehicles are too far away to interact. However, for larger values of $D$, $V$ decreases more rapidly with increasing $D$ because the motion of vehicles is significantly influenced by preceding vehicles due to the reduced average distance between them. As a consequence, the flow $Q = DV$ peaks at $D = D_c$, and then increased $D$ leads to a decrease of the flow. Finally, in a completely congested situation, no vehicle can move and the flow is zero.

![Figure 2.2: Fundamental diagram of traffic flow](image)

Many researchers suggest alternative shapes that provide a better fit to the measured data. Some examples are shown in Figure 2.3. Newell [107] proposes a triangular shaped curve with two velocities, one for free flow and the other for congested traffic. Daganzo [31] uses trapezoidal relations. Reported in several field studies [20, 58],
the maximum flow in congested traffic is sometimes lower than the capacity, which is achieved in free flow conditions. This is known as capacity drop, ranging from 6% to 10% [20, 58], during the transition of free flow to congestion. Therefore, some studies consider a discontinuity at $D_c$ (as in Figure 2.3).

Kerner [75] attributes the capacity drop to the increased time headway. Treiber et al. [137] explain capacity drop from safety perspective. The issue of capacity drop is important to merging control strategies, especially ramp metering in the literature, and will be discussed in Section 2.3.4.

Classical theories based on the fundamental diagram of traffic flow consider only two phases—free flow and congestion. In contrast, Kerner [75] distinguishes congestion as two different phases. He proposes three-phase traffic theory considering the following phases:

- free flow (F), where the velocity is not constrained by neighboring vehicles.
- wide moving jam (J), \(^1\) where traffic flow reduces significantly.
- synchronized flow (S), where the velocity reduces but the flow is close to free flow. Same velocity at all lanes without lane changes. It is called “one-pipe flow” by Daganzo [32].

\(^1\)Width in the “wide moving jam” is measured by the longitudinal extension of a jam along the road.
Kesting et al. [82] identify five traffic situations: free traffic, congested traffic, upstream jam front, downstream jam front, and bottleneck sections. They associate those traffic situations with a specific set of intelligent driver model (IDM) parameters. Therefore, their adaptive cruise control (ACC) strategy adapts the driving characteristics (that is, the parameters of the ACC controller) to the traffic conditions.

A better understanding of the cause of congestion assists in developing strategies to reduce congestion. A 2005 US study [45] states that there are seven major causes of congestion. They are bottlenecks (40%), traffic incidents (25%), bad weather (15%), work zones (10%), poor signal timing (5%), and special events/other (5%). Empirical observations show that congestion exhibits a “probabilistic nature” [41,91,115], which means congestion may or may not occur at the same traffic conditions. Daganzo [33] explains in detail how queuing begins at merging traffic near on-ramps. Chen et al. [22] believe that congestion occurs not because demand exceeds capacity rather because of the inefficient operation of highways.
2.2 Sensors

Traffic dynamics measured by sensors are input variables to the traffic models underlying traffic control strategies. We review the major automotive sensors and techniques. It has been found that the assumed sensor-enabled cars are realistic for current technology. Among different types of sensors, such as position, velocity, or acceleration sensors, we identify what is the most critical type of sensor. We also investigate how different levels of sensors’ accuracy influence their applications.

Automotive sensors have been developed since the early 1970s. Recent advances in micro-electro-mechanical systems (MEMS) technology and wireless communications have enabled the development of low-cost, multifunctional sensor nodes that are small in size and with short-range wireless communication capabilities [5]. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, can be installed in modern cars to facilitate transport applications. This new type of sensor-enabled cars is able to sense information about its own position and local traffic conditions, process the information, and communicate the information to other vehicles in its neighborhood.

There are two kinds of automotive sensing: out-vehicle environment sensing and vehicle-state sensing, summarized in [87]. Out-vehicle environment sensing collects information about the driving environment. For example, this includes sensors for extracting lane boundaries, especially when not clearly marked or in bad weather conditions; detecting nearby vehicles and estimating their position, velocity, and acceleration; and detecting the unexpected traffic participants (such as pedestrians) and obstacles. Vehicle-state sensing focuses on measuring a vehicle’s movement and monitoring its actuators. For example, researchers have studied how to detect a vehicle’s position, velocity, and acceleration; an engine’s pressure and temperature; and a tire’s pressure, temperature, and friction coefficients.

The accuracy of the information obtained by sensors impacts parameter settings in the traffic model. For example, if the position or distance information is provided with an accuracy of 20 meters, the safety distance calculated by the merging algorithm
would be of little value. Besides distance and velocity, there are several factors that impact the minimum safety distance, such as a tire’s pressure, temperature, and the friction coefficient for braking. Currently, most work on traffic control neglect these factors and focus on position and velocity. A complete model that integrates these factors will lead to significant improvement in road safety and efficiency of control strategies, but is beyond this thesis.

2.2.1 Position

Most vehicular ad-hoc networks (VANETs) applications consider the availability of real-time updated position information. They differ, however, on the position accuracy required to function properly. For example, some applications such as navigation, can work with position information with errors in the range of 10–20 m, while other applications, especially critical safety applications such as collision warning systems, require more accurate and reliable positioning systems with sub-meter precision.

In cooperative driving applications, vehicles in a VANET exchange messages among them to drive and share the available space on the road cooperatively. In these applications, the vehicles can assume partial control over driving. In order to deal with safety, position information reliability and accuracy are crucial. Accurate positioning ensures localization within a meter or sub-meter precision in order to estimate accurately the distances between vehicles, while a reliable localization will ensure that updated information will always be available.

A number of localization techniques have been proposed for computing the position of mobile nodes. Most localization techniques can be applied easily to VANETs. Global positioning system (GPS) receivers have a position error in the range of 1–10 m [66]. Unfortunately, GPS receivers are not the best solution in some cases since they cannot work in the indoor parking or dense urban areas where there is no direct visibility to satellites. As a result, GPS information is likely to be combined with other localization techniques such as dead reckoning, cellular localization, and image/video localization [17]. In merging scenarios, GPS is not sufficient with an accuracy of 1–
10m. Although the use of accelerometers can also estimate the position of a vehicle, the estimation is not precise. Therefore, it is required to develop merging algorithms that cope with imprecise positioning information.

Only a few studies have addressed inaccurate sensor information. Treiber et al. [136] propose a microscopic traffic model including estimation errors. Shladover and Tan [129] report that a collision warning system requires a position accuracy of 50 cm. Rajamani and Shladover [118] compare constant-time-gap autonomous control systems and cooperative control systems with inter-vehicle communication (IVC). Experimental results from typical sensors and actuators show that in practice autonomous vehicle following achieves a time gap no less than 1 second. By replacing radar range rate in the autonomous control algorithm with the velocity difference between the two cars (a rudimentary form of cooperation). They present a cooperative platoon of 8 cars maintaining 6.5 m gaps with an accuracy of 20 cm.

Farrell et al. [44] present implementation of a vehicle control system, which uses control state information. This information is obtained from a carrier phase (CP) differential global positioning system (DGPS) which is integrated with inertial navigation system (INS). DGPS achieves centimeter-level position accuracy. Du and Barth [39] propose a lane-determining system including an on-board DGPS receiver with a 2–3 m positioning accuracy. Lane-level positioning facilitates a number of new ITS applications such as better fleet management, lane-based traffic measurements from probe vehicles, and lane-level navigation.

### 2.2.2 Velocity and Acceleration

Conventional methods measure the velocity of a vehicle by wheel speed sensors. The drawbacks of wheel speed sensors are inaccuracy and low resolution. Radar speed measurement systems are commonly used due to their reasonable cost and acceptable accuracy. A GPS based velocity sensor [13] has been recently developed that can be used instead of radar sensors and wheel speed sensors.

The acceleration and deceleration is measured by accelerometers and gyroscopes.
Currently, acceleration sensors are used in applications such as safety systems, vehicle stability systems and electronic suspension. Recent trends of automotive acceleration sensors are to employ MEMS technology. Their advantages over traditional sensors are low cost, robustness, self diagnostics, and multiplex network connectivity [48].

The traffic model in our work relies on the availability of the following sensed information: velocity, acceleration, forward distance and backward distance. To increase the reliability of sensed data, communication is used to check the sensed data, thus each car also has a transceiver to communicate with cars in its vicinity. In table 2.2, we summarize three types of automotive sensors, and report their range and accuracy. In table 2.3, we compare different types of sensors and their limitations.

### Table 2.2: Automotive sensors

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position /Distance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPS [66]</td>
<td>worldwide</td>
<td>1-10 m</td>
</tr>
<tr>
<td>Laser Range Finder [14]</td>
<td>50 m</td>
<td>5 cm</td>
</tr>
<tr>
<td><strong>Velocity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speckle Velocimeter [14]</td>
<td>0.4-400 km/h</td>
<td>0.1%</td>
</tr>
<tr>
<td>Velocimeter [142]</td>
<td>0.53-96.6 km/h</td>
<td>1%</td>
</tr>
<tr>
<td>Dickey-John radar [69]</td>
<td></td>
<td>1-3%</td>
</tr>
<tr>
<td><strong>Acceleration</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASCO CI-6558 [28]</td>
<td>-5 to +5 g</td>
<td>0.01 g</td>
</tr>
</tbody>
</table>

### Table 2.3: Comparison of automotive sensors

<table>
<thead>
<tr>
<th>Technique</th>
<th>Function</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical</td>
<td>lane detection, traffic sign recognition</td>
<td>sensitive to undesired variables</td>
</tr>
<tr>
<td>Radar</td>
<td>obstacle detection</td>
<td>angular reach and resolution</td>
</tr>
<tr>
<td>Laser</td>
<td>risk detection (e.g., pedestrians, cyclists)</td>
<td>sensitive to adverse weather (e.g., rainy)</td>
</tr>
<tr>
<td>GPS/DGPS</td>
<td>navigation ACC, collision warning</td>
<td>signal may be blocked</td>
</tr>
</tbody>
</table>

### 2.2.3 Communication

Communication among cars is of great importance because sensing data need to be exchanged. However, the communication protocols or underlying networking layers
are not in the scope of the thesis. We only highlight the communication range of cars because it is highly related to the assumptions we make in our work. For example, a key parameter in our proposed algorithms to achieve the adaptiveness to dynamic traffic conditions is the “decision point”, where cars of conflicting traffic streams cooperatively make plan for smooth merging. The maximum value of decision point is set to be 400 m away from the actual merging point. This is based on the fact that Dedicated Short Range Communications (DSRC) standard communication range is 400 m. We assume single-hop inter-vehicle communications. For the area larger than 400 m protocols for multi-hop communications are required.

A comprehensive survey on inter-vehicle communication systems is provided in [130]. In addition, Bai et al. [8] characterize and classify communication-based automotive applications from a wireless networking perspective.

2.2.4 Applications

We provide an overview of sensor applications in this section to position our work among other related sensor applications.

- Traffic monitoring: Traditionally, traffic monitoring is performed depending on inductive loop detectors and video cameras. An alternative approach is to use wireless sensor networks (WSNs). For example, Coleri et al. [29] use WSNs to achieve high accuracy and lower cost for vehicle detection by magnetometer sensor nodes attached in the center of a lane. In their prototype, the sensor nodes collect traffic information such as the number of cars, velocity, and vehicle length. The information is transferred to an access point over radio, and then to traffic management center for further analysis. Similarly, Bathula et al. [11] design and implement a WSN for monitoring traffic at temporary construction work zones on highways. A variety of traffic statistics are collected, including flow, density and average velocity. An array of nodes with infrared ranging sensors are deployed at a work zone. Each sensor node transmits the sensed data to a local base-station. The base-station is connected to a centralized server that
is responsible for data archival and analysis. One of the main challenges of applying WSNs is the highly constrained battery power of the sensor nodes. These studies focus on developing energy-efficient routing protocols.

Unlike the above work, which focuses on dedicated sensors at the roadside, some studies use on-board sensors for traffic monitoring. Battery power of the sensors is not a constraint assuming that vehicles can provide the sensors unlimited power. Instead the mobility becomes a key challenge. CarTel [68] is a mobile sensor system that collects traffic information such as average velocity and delays. Nericell et al. [99] utilize smartphones carried with drivers to monitor road and traffic conditions. Smartphones with GPS, accelerometer sensors and communication, as well as computing capabilities can detect various situations, such as a bump on the road, a sharp brake, and stop-and-go traffic. Each node processes sensed data locally before delivering to the central server in both CarTel and Nericell.

In contrast to the centralized systems, Schönhof et al. [127] propose a completely decentralized system for detection of traffic jams on highways and prediction of the position of a traffic jam evolved over time based on messages exchanged by inter-vehicle communication (IVC). We also adopt a decentralized architecture in our work, more detailed discussion on centralized and decentralized systems will be in section 2.3.1.

- Vehicle control: Naranjo et al. [104] propose an adaptive cruise control (ACC) system which is able to function well in stop-and-go traffic. This is a significant improvement because conventional ACC systems cannot work when the velocity is below 30 km/h. Their proposed ACC system requires a differential global positioning system (DGPS) with positioning accuracy of 2 cm and wireless local area network (WLAN) to continuously send the position to the following car. An on-board computer calculates the variables that control the pressure on the throttle and brake pedals. In another work, Naranjo et al. [103] present further improvement of an ACC system, which enables autonomous vehicles to perform
the overtaking operation. The challenge is that during the time of overtaking, keeping the vehicle either at the right velocity or at a safe distance from the front vehicle is not possible by ACC system. Lane-changing is achieved by specifying GPS coordinates. The information required for navigation is supplied by a DGPS and wireless communication.

A review on ACC systems can be found in [116], which focuses on comparing between autonomous systems (sensor-based ACC) and cooperative systems (IVC-based ACC) in terms of technologies used, impacts on capacity and stability, as well as implementation.

- Variable speed limit (VSL) control: Kang and Chang [73] propose a model for optimal speed control at highway work zones. Their basic idea is to divide a day into different control periods based on historical data and compute optimal control values for each. Simulation results show an increase of work-zone throughput and a reduction of total vehicle delay. Moreover, their speed control leads to lower velocity variances. They consider the velocity variance as a measurement for traffic safety. Lower velocity variance can improve the overall traffic safety in work zones. The limitation is that real-time traffic data may significantly differ from historical data.

Table 2.4 summarize the discussed applications from their objectives, sensing component, communication, infrastructure, adaptable to dynamic traffic conditions and robustness.

### 2.3 Flow Control Strategies

In this section, we discuss different techniques to improve traffic flow control, in particular, we focus on the context of freeways. The main aim of the flow control is to improve the efficiency of road usage. Optimization of freeways is based on a homogenization of vehicle traffic [62]. Stop-and-go traffic, jams, and congested traffic are
Table 2.4: Overview on current applications

<table>
<thead>
<tr>
<th>Project</th>
<th>Objective</th>
<th>Sensors</th>
<th>Communication</th>
<th>Infrastructure</th>
<th>Adaptable</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathula et al. [11]</td>
<td>monitoring work zone</td>
<td>ranging sensors</td>
<td>multihop wireless</td>
<td>Centralized</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CarTel [68]</td>
<td>traffic monitoring, information query</td>
<td>mobile sensors, GPS</td>
<td>WLAN</td>
<td>Centralized</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Nericell [99]</td>
<td>traffic monitoring</td>
<td>smartphone with accelerometer, GPS</td>
<td>GSM</td>
<td>Centralized</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Kang and Chang [73]</td>
<td>speed control at work zone</td>
<td>historical data</td>
<td>N/A</td>
<td>Centralized</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Schönhof et al. [127]</td>
<td>congestion detection</td>
<td>N/A</td>
<td>IVC</td>
<td>Decentralized</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Naranjo et al. [103,104]</td>
<td>ACC</td>
<td>DGPS, speedometer</td>
<td>WLAN</td>
<td>Decentralized</td>
<td>Yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>
associated with reduced efficiency due to a capacity drop, often triggered by perturbations in the traffic flow. For example, if a car pushes into a small gap to change lane, the impact may appear small locally but is globally significant as the following cars may need to slow down sharply. This is known as “slinky-type effect” \[16\], which can lead to traffic congestions and car accidents. Therefore, perturbations should be suppressed by control strategies such as variable speed limits, dynamic route guidance, driver assistance systems, and ramp metering.

2.3.1 Centralized and Decentralized Systems

In theory, traffic control systems can be categorized into central\textit{ized} and decen\textit{tralized} systems. In practice, there are no pure centralized systems. The present research mainly adopts a hybrid system architecture, i.e., a combination of a centralized and decentralized approach \[6,139\]. Varaiya \[139\] discusses pure centralized and decentralized designs. A centralized approach has a tremendous computation and communication cost. On the other hand, a decentralized approach requires local computation. In his approach, cars are grouped into tightly spaced train-like \textit{platoons} and the first car of the platoon is controlled centrally whereas within platoons cars are decentralized.

California Partners for Advanced Transit and Highways (PATH) \[67,139\] propose a multilayer automated highway systems (AHS) control architecture to handle the control problems. Starting at the top, the layers are called the network, link, coordination (planning), regulation, and physical layers (see Figure 2.5). The network layer assigns a route to each vehicle entering the system. The link layers task is to assign a section to each vehicle and a target velocity for each section in order to increase traffic flow. The planning layer coordinates the movement of neighboring vehicles. The regulation layer executes feedback control laws to finish certain tasks such as join, split, and lane change. The physical layer uses nonlinear differential equations for lateral and longitudinal control.

Similar to the architecture of PATH, Dolphin \[138\] consists of three layers: the vehicle control layer, the vehicle management layer, both of which are on each vehicle, and
Estrin et al. propose that in sensor networks localized algorithms [42], where sensors only interact with other sensors in a restricted vicinity, collectively achieve a desired global objective. Localized algorithms have three advantages. First, the communication scales well as the number of vehicles grows. Second, the algorithms are robust to network partitions and node failures. Third, unlike centralized systems, localized algorithms provide much greater local details.
Lygeros et al. [93] introduce a methodology for designing hybrid controllers for large scale, multiagent systems. The system is initially designed fully decentralized where each vehicle has only access to local information obtained by sensors. If the level of performance is unsatisfactory, some centralization is introduced by communication of information between vehicles. Moreover, they suggest what the most important pieces of information are. The effect of this more global knowledge is to reduce the disturbance generated by other vehicles. The optimum strategy for each vehicle does not coincide with the global optimum. Ideally, to achieve the global optimum a centralized control scheme computes the global optimum and commands the vehicles accordingly. A centralized controller may be undesirable, however, as the design process will probably be very complicated, it is likely to be computationally intensive, may be less reliable and the information that needs to be exchanged may be too large for the available communication capabilities. In their opinion, a completely decentralized solution is too inefficient and a completely centralized solution is prohibitively complex and/or expensive. Therefore, a compromise will feature semi-autonomous vehicle operation. In [94], a method is proposed to deal with multiple control requirements. To achieve the goal of improving the throughput of the highway system while maintaining the safety, their method is to prioritize the multiple requirements, i.e.,

![Traffic control layer](image_url)
safety is more important than throughput.

Researchers have agreed that the traffic control problem on a global network level is practically unsolvable by traditional optimization techniques (see, e.g., [51, 84]) because of the exponential complexity of the underlying algorithms. Hence, a number of decentralized optimal strategies whose actions are coordinated heuristically by a superior control layer have been proposed [50]. Lammer et al. [84] present a self-organizing, decentralized control method for global coordination of traffic signal control. They map the problem to phase-oscillator models. By synchronizing these oscillators, the desired global coordination is achieved. The concept applies to networks where time sharing mechanisms between conflicting flows in nodes are required and where a coordination of these local switches on a network level can improve the performance. Gershenson [51] proposes self-organizing methods using simple rules to coordinate traffic lights to improve traffic flow. Kesting et al. [79] present a decentralized approach for self-organized traffic lights control and an ACC system that adapts driving strategies to different traffic situations.

Table 2.5: Overview of merging strategies

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Group</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall and Tsao [59]</td>
<td>H</td>
<td>P</td>
</tr>
<tr>
<td>Lu et al. [92]</td>
<td>H</td>
<td>P</td>
</tr>
<tr>
<td>Anda et al. [6]</td>
<td>H</td>
<td>P</td>
</tr>
<tr>
<td>Davis [36]</td>
<td>D</td>
<td>I</td>
</tr>
<tr>
<td>Our approach</td>
<td>D</td>
<td>I</td>
</tr>
</tbody>
</table>

Table 2.5 summarizes the traffic control strategies we discussed from the aspects of system architecture (H denotes hybrid, D denotes decentralized), group driving or individual (P denotes platoon, I denotes individual) and control strategy (FA denotes full automation, PA denotes partial automation).

2.3.2 Automation Control

Automated highway systems (AHS) have been proposed based on automated vehicles. Fully automated vehicle control [139] is able to increase the capacity dramatically. Dif-
Different traffic automation strategies are classified according to the communication:

- Autonomous: automation is concerned within individual vehicles and no communication between vehicles;
- Cooperative: vehicles cooperate by communicating selected information; and
- Platooning: vehicles form tightly spaced groups by communicating detailed information at a high frequency.

There are basically two types of control. One is to control the behaviors of cars on the highway. The other is to control how the cars enter the highway, which is the merging control. Merging control algorithms for automated vehicles focus on calculating the proper velocity and time to get to the merging point [74, 92], as well as the minimum safety distance [72].

Hall and Tsao [59] examine the impact of merging strategies on the capacity of AHS. They present three merging strategies integrated with ramp metering for cooperative and/or platooning vehicles:

- Release-to-gap: when the ramp vehicle reaches the merge point, the intended gap also reaches the merge point so that the ramp vehicle and the gap are properly aligned at the merge point;
- Preplatooning: ramp vehicles into platoons before merging; and
- Release-to-tag: when the ramp vehicle or platoon reaches the merge point, the tail of the mainline platoon reaches the merge point so that the ramp vehicle or platoon tag along the mainline platoon.

Their simulation results show that the release-to-tag strategy is the most efficient merging strategy, with 20% reduction from the theoretical maximum road capacity. Other strategies lead to as much as 70% reduction. They point out that the main cause of the capacity reduction is the first-in last-out phenomenon, where an available gap for merging may be taken by a following vehicle rather than the vehicle that arrived first.
However, their merging strategies and simulations assume that automated vehicles must stop at a check-in facility and then proceed to the merging area.

In contrast, Lu et al. [92] show that ramp vehicles merging into the middle of a platoon provides higher efficiency in high-density traffic conditions. They propose merging control algorithm with the concept of virtual platooning, which forms a platoon of the ramp vehicle and main road vehicles before the ramp vehicle arrives at the merging point. This idea emphasizes four factors that must be considered simultaneously: vehicle relative positions, timing, velocity and acceleration.

At present no approach has proven itself superior to all its competitors in every respect. Since there are many different criteria to define performance, and so many parameters that determine performance, one approach may be the winner in several respects but inferior in other respects. Although the concept of AHS improves safety, road utility, and driver comfort, the requirement of major changes to the existing highway infrastructure such as dedicated lanes, and a high percentage of automated vehicles, makes AHS unlikely to come to reality in the foreseeable future.

2.3.3 Driver Assistance Systems

Partial automation adaptive cruise control (ACC) systems are already available on the market. ACC systems only control longitudinal driving with the aim of improving safety and driver comfort. Merging, overtaking, or lane-changing still need the intervention of drivers.

In the literature, there is research on how the ACC systems impact traffic flow, especially in relation to congestion. There is an agreement that ACC systems can achieve a smoother traffic flow, which leads to an increase of the road capacity and reduction in congestion [35,82,89]. Kesting et al. [82] show that traffic congestion occurs due to increasing incoming traffic at the ramp without ACC vehicles in their simulation. When there are 25% ACC vehicles, with the same value of ramp flow 12.5 vehicles/min, congestion is eliminated. Similarly, Davis [35] reports that with 20% ACC vehicles no jams are formed in simulations. Furthermore, Davis identifies a potential prob-
lem with merging. ACC vehicles maintain desired headways and do not “give way” when a vehicle enters from a ramp. The ability to merge may be hindered by the lack of minimum safety gaps. In another work, Davis [36] focuses on the merging problem and proposes cooperative merging strategy to increase throughput and reduce travel times. In his strategy, an ACC equipped car adjusts its position according to the front car on the other road to create a safety gap without slowing down sharply. However, it only considers the nearest front car and thus the benefit is limited.

2.3.4 Ramp Metering

Ramp metering has been successfully implemented since the mid-1960s as an efficient freeway management strategy. All ramp metering strategies calculate suitable ramp flow that is allowed to the freeway. The ramp flow is converted to a green-phase duration of the traffic light.

Ramp metering strategies are typically classified as follows:

- Fixed-time strategies are based on constant historical demands and static models to derive optimized settings. One example of fixed-time strategies is ideal ramp metering (IMP) [22]. To find out when a freeway section is most efficient, i.e., when there is maximum flow on the highway section, Chen et al. analyze Los Angeles highways data from the PeMS (Performance Measurement System) database [23]. They show that the maximum flow occurs near 60 mph (≈ 100 km/h), which is also the velocity under free flow conditions. They assume that the pattern of demand is unchanged. Therefore, to maintain the velocity of 60 mph, IMP stops the ramp flow whenever the occupancy exceeds the critical occupancy (at which the freeway flow is maximum). The main drawback of fixed-time strategies is the absence of real-time data. Fixed-time strategies are oversimplified, because traffic demands are not constant and sometimes fluctuate strongly.

- Reactive strategies, also known as traffic-responsive or feedback controllers,
based on real-time measurements of the freeway flow upstream of the ramp and the freeway occupancy downstream of the ramp. They aim to keep the freeway traffic conditions close to pre-specified set (desired) values. Reactive strategies can be performed independently for each ramp or extended to several ramps. The ALINEA strategy outperforms other strategies in field trials [108]. ALINEA can typically achieve 20% decrease of the total time spent. The major limitation of reactive strategies is that appropriate set of values are hard to get. ALINEA has become a benchmark that most other studies compare with their ramp metering strategies, such as ANCONA [76], FLOW [61], BOTTLENECK and ZONE [27].

- Nonlinear optimal ramp metering are based on macroscopic dynamic models to coordinate ramp metering in freeway networks. The total time spent by the vehicles in a freeway network can be reduced by up to 50% in simulations [108]. This type of strategies has not been implemented in the field. [109] According to [27], two area-wide coordinated algorithms BOTTLENECK and ZONE can outperform the local reactive algorithm ALINEA. Papamichail et al. [111] present a nonlinear model-predictive hierarchical control approach for coordinated ramp metering. Their structure consists of three layers: the estimation/prediction layer, the optimization layer and the direct control layer.

In spite of the reduced congestion at the merge area by ramp metering, the length of queue on ramps becomes longer. Lee et al. [85] investigate ramp metering from the safety perspective. They consider the long delay in ramp traffic as a trade-off to the crash potential reduction.

In summary, ramp metering is an efficient control approach. On the other hand, it presents several drawbacks:

- Ramp metering is a sensitive technique. If ramp metering is not accurate enough, then it can lead to traffic congestion by weak metering that overloads the mainstream flow, or prevent full utilization of the mainstream capacity by overly strong metering.
• Ramp metering may increase the waiting time for vehicles on the ramp, create ramp queues and transfer the congestion problem to the ramp and local street networks.

• Ramp metering including those reactive strategies still have the limitation of not being adaptive to the highly variable and unpredictable traffic conditions.

2.3.5 Diversion

Diversion is a traffic flow control strategy intended to mitigate congestion by proposing alternative routes to drivers in order to avoid bottlenecks.

Daganzo et al. [34] classify three ways of diversion based on the different user subsets:

• diversion of upstream traffic of a bottleneck: recommend highway drivers who are approaching a congested section to take exits before the bottleneck by delivering messages.

• diversion of entering traffic: restrict drivers who are going to enter a congested highway section by constraining ramp metering or closing the ramp.

• diversion by pricing: set tolls depending on the distance of origin or destination to the bottleneck.

Davis [37] investigates how to assign a given demand (i.e., incoming flow) between two on-ramps to reduce congestion. They compare two approaches for diversion: (1) drivers choice based on the information of average travel times received and (2) vehicles are diverted based on the average velocity of the main road near the ramps. Their simulation results show that the second controlled diversion approach generally performs better than the first one.

Banks [10] considers three cases with ramp metering used to reduce delay by diverting traffic around bottlenecks. They are: (1) travel times on an alternate route,
bypassing the bottleneck, are insensitive to flow, (2) the alternate route is undersaturated but travel times are sensitive to flow, and (3) the alternate route is oversaturated. He proposes a ramp metering algorithm which decides the sequence of metering different ramps to minimize traffic delay.

Muñoz and Laval [100] study dynamic traffic assignment to achieve the system optimum in a network consisting of a surface street grid and a congested highway section. In their approach, vehicles can be diverted through off-ramps, and on-ramps can be metered. They suggest that controlled diversion is still a complex technological problem, their results provide a benchmark for future ITS applications.

We investigate traffic diversion particularly on the two-ramp scenario. With local spatial information on individual cars obtained by sensors and inter-vehicle communication, our simulation results show the potential benefit of combining merging algorithms with diversion strategies.

2.4 Conclusion

In this chapter we presented a comprehensive overview of various methods and approaches for traffic models, automotive sensors, and flow control strategies. This overview provided a complete understanding of the essential components to develop proactive traffic merging strategies for sensor-enabled cars.

Our study of the vehicular traffic models suggests that the existing models can be categorized from two perspectives: 1) continuous or discrete models depending on the use of space, time and state variables and 2) macroscopic and microscopic models depending on the level of simulation details. We reviewed representative traffic models in each category and discussed their characteristics, comparative advantages and disadvantages. We reviewed automotive sensors in terms of their types, accuracies and applications. The accuracy of sensed information impacts traffic control strategies. Finally, we studied several flow control strategies that improve the efficiency of road usage. We covered centralized and decentralized traffic control strategies including autonomic vehicle control, driver assistance systems, ramp metering, and diversion.
In the next chapter, we describe our traffic model and our simulator, which integrates sensor-enabled cars into the framework of traffic merging algorithms. The aim is to use the current road networks more efficiently and ultimately help to alleviate traffic congestions.
Chapter 3
Traffic Model and Evaluation Measures

Analysis of the traffic models, introduced in Chapter 2, reveals that microscopic traffic models are more suitable for investigating merging algorithms. Based on this finding, we adapt a microscopic traffic model in our work. This chapter details the underlying traffic model we use as the foundation of our work. Equally important, this chapter provides the methodology of how to evaluate merging algorithms. In this context, we present evaluation measures from both safety and efficiency perspectives.

3.1 Introduction

Longitudinal and lane-changing models are essential components of any research on traffic control strategies. Based on the survey on traffic models, we choose the intelligent driver model (IDM) [135] as our longitudinal model, and the minimizing overall braking induced by lane change model (MOBIL) [81] as our lane-changing model. We detail the lane-changing model adopted in our work since merging is a special case of mandatory lane-changing. However, applying a lane-changing model directly to the merging case faces two problems. First, there are vehicles that fail to merge. A vehicle on the ramp has to decelerate as it approaches the end of the ramp if it cannot merge. A larger safety distance is required as the velocity difference between the ramp vehicle and the main road vehicles increases. Consequently, the chance of a safe merge decreases. Second, the acceleration calculated by the car-following model is
not suitable to represent the acceleration of merging process. In a merging scenario, a link is required to integrate the car-following model and lane-changing model via the acceleration.

3.2 Employed Traffic Model

3.2.1 IDM

In IDM [135], the acceleration or deceleration of a car depends on its velocity $v$ and its front car, specifically, the distance $s$ to the front car, and the velocity difference $\Delta v = v - v_f$ ($v_f$ is the velocity of the front car).

The acceleration $\dot{v}(t)$ in IDM is given by

$$\dot{v}(s, v, \Delta v) = a \left[ 1 - \left( \frac{v}{v_0} \right)^4 - \left( \frac{s^*(v, \Delta v)}{s} \right)^2 \right]$$  \hspace{1cm} (3.1)

The acceleration consists of an acceleration component $a[1 - (v/v_0)^4]$ on a free road and a deceleration component $-a(s^*/s)^2$ when a vehicle is too close to its front car. In light traffic, since the distance $s$ to the front car is large, the deceleration component is negligible and the acceleration decreases from the maximum acceleration $a$ to 0 when the velocity approaching the desired velocity $v_0$. In heavy traffic, the deceleration component becomes relevant. It is based on the ratio between the desired minimum gap $s^*$ and the actual distance to the front vehicle $s$. Specifically, $s^*$ is expressed as

$$s^*(v, \Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}$$  \hspace{1cm} (3.2)

$s_0$ is the minimum distance between two cars at standstill. The contribution of $s_0$ to $s^*$ is significant only in congested traffic when the velocity $v$ is very low. $T$ is the desired time gap that a vehicle maintains from its front vehicle. In addition, $s^*$ increases when approaching a slower car and decreases when the front car is faster. In nearly all situations the braking component limits deceleration within the comfortable deceleration
b. A car brakes stronger than $b$ when $s$ is too small.

Note that in any microscopic model, the velocity of a vehicle fluctuates around the desired velocity rather than remaining constant even in the free-flow traffic or in the absence of disturbance from the surrounding traffic [26].

![Graph](image.png)

Figure 3.1: Acceleration and velocity
To explain the equations we use the following figures that illustrate the typical relationships rather than specific values of parameters. Figure 3.1 shows the relationship between acceleration and velocity. Figure 3.1(a) shows how a vehicle accelerates from standstill in the free-flow condition. Acceleration equals to the maximum acceleration $a$ when the velocity is low and acceleration decreases until 0 as velocity increases to the desired velocity $v_0$. Figure 3.1(b) shows how a vehicle brakes when it encounters an obstacle in front. The deceleration increases to the maximum deceleration $b$ and then decreases to 0 until the vehicle stops.

Figure 3.2 shows the relationship between a car’s velocity and the distance to the front car, as well as the velocity difference to the front car. Velocity increases as distance increases. When the distance is sufficiently large the velocity approaches the desired velocity $v_0$, which corresponds to the free-flow situation.

Figure 3.2: Relationship between the velocity, the distance to the front car and the velocity difference $\Delta v$ between two cars
3.2.2 Impact of Parameter Values

All IDM parameters should have realistic values. The setting of values can characterize different driving styles or different vehicle types. In this section, we discuss parameter values in detail.

- **Desired velocity**: the desired velocity $v_0$ is the maximum velocity that a vehicle tries to achieve. It is generally set as the free flow velocity or the speed limit by legislation. It may vary according to the road. For example, in a highway the speed limit is normally 100–120 km/h. In the city, the speed limit is typically 50 km/h. In addition, the desired velocity may vary according to the vehicle types. Cars usually have a higher value of desired velocity than trucks.

- **Desired time headway**: the desired time headway $T$ determines the minimum safe distance $vT$ which a vehicle driving at velocity $v$ needs to keep from its front car. The time headway is the time available to reach the same deceleration as the front car in case it brakes. $T$ is independent of $v$. Empirical studies find that typical $T$ ranges from 1s to 2s [15, 106]. Very small $T$ (below 1s) is sometimes observed as well [83].

- **Maximum acceleration**: The maximum acceleration should be lower than the physical limit of a vehicle. The acceleration depends on the driving situations. For example, a smaller acceleration is applied in a relaxed car-following situation, whereas a higher acceleration is applied in an overtaking maneuver.

- **Comfortable deceleration**: The comfortable deceleration determines the process of approaching a slower vehicle in front or a traffic light. A lower value of deceleration is applied when approaching a visible obstacle, while a higher value is applied for emergency brakes, such as the front car suddenly decelerates or a car changes lane.

For certain parameter values IDM guarantees collision-free driving. Our IDM parameter values for the simulations are given in Table 3.1. These values are chosen in
such a way that they reflect realistic traffic conditions. We settle for a high velocity (100 km/h) because a higher velocity can distinguish a good merging algorithm from a less good one. We set the maximum acceleration and the comfortable deceleration smaller than the physically possible values to avoid collisions. Furthermore, more restricted values make it easier to identify an effective merging strategy and smaller values for acceleration and deceleration can decrease the fuel consumption.

Table 3.1: IDM parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired velocity $v_0$</td>
<td>100 km/h</td>
</tr>
<tr>
<td>Desired time headway $T$</td>
<td>1.5 s</td>
</tr>
<tr>
<td>Maximum acceleration $a$</td>
<td>1 m/s$^2$</td>
</tr>
<tr>
<td>Comfortable deceleration $b$</td>
<td>3 m/s$^2$</td>
</tr>
<tr>
<td>Minimum distance $s_0$</td>
<td>2 m</td>
</tr>
</tbody>
</table>

3.2.3 MOBIL

MOBIL [81] is employed as the lane-changing model in our work, mainly due to its implicit compatibility with the IDM. This model adopts a game theoretical approach to address the lane changing problem, allowing a vehicle to move to a different lane if the lane change minimizes the overall braking of vehicles. Such requirement is fulfilled when two conditions in terms of safety and incentive are verified.

To explain the notations we use in the following text, let us consider the general lane-changing scenario illustrated in Figure 3.3, where vehicle $i$ wishes to change lane. The following vehicles on the current lane and target lane are denoted by $x$ and $y$, respectively. Let the current acceleration of $i$ be $a_i$, the acceleration of $i$ in the target lane after the lane change be $a'_i$. Similarly, $a'_x$ and $a'_y$ denote the new accelerations of $x$ and $y$ respectively, after $i$ changes lane.

Now we formulate the following constraints and explain how the lane-changing model MOBIL is related to the car-following model IDM.
Safety constraint

The safety criterion requires that after the lane change, the deceleration \( a'_y \) of the back car \( y \) in the target lane should not exceed the safe deceleration limit \( b_{safe} \):

\[
a'_y \geq -b_{safe}
\]  

(3.3)

The lane-changing model and the car-following model is integrated by enforcing the above constraint on the acceleration. In particular, the velocity difference plays an important role. A larger gap is required to satisfy the safety constraint if vehicle \( y \) is faster than \( i \). In contrast, a smaller gap is allowed if \( y \) is slower than \( i \).

\( b_{safe} \) should be below the maximum possible deceleration \( b_{max} \), which is approximately \( 9m/s^2 \). The braking reaction of the back car in the target lane is always limited by the value of \( b_{safe} \). Collisions due to lane changes are automatically avoided as long as the underlying car-following model is collision-free.

Incentive criterion

The incentive of a lane change is also formulated in acceleration. The lane change is executed when the sum of acceleration of all influenced vehicles increases compared to without lane-changing:

\[
a'_i + a'_x + a'_y > a_i + a_x + a_y
\]  

(3.4)
This is the core idea of minimizing overall braking induced by lane changes (MOBIL). MOBIL provides basic underlying physics but it does not tell a driver when and where to merge. The process of making a merging decision is given in Chapter 5.

### 3.3 Evaluation

We evaluate merging algorithms from two perspectives: safety and efficiency. Although all merging algorithms have to ensure road safety our main aim is to improve the traffic flow while maintaining the safety requirement. In addition, we consider other factors to measure efficiency. In this section, we introduce criteria for efficient traffic merging algorithms and show how to measure the performance. We identify which criteria depend on each other and how optimized traffic flow implies higher velocity coupled with tighter gaps, while safety requirements imply lower velocity and larger gaps.

#### 3.3.1 Safety

Most studies use the concept of “times-to-collision” (TTC) as a safety measure [52, 95, 126, 143]. If a vehicle is faster than the vehicle in front and the velocity remains, then after a while the vehicle would collide with its front vehicle. TTC is defined as the time that two vehicles would collide if they maintain their current velocity. If the value of TTC is infinite, the vehicles will not collide. If the value of TTC is finite and decreases during a period of time, a collision may occur unless the driver takes any action to avoid the collision. A collision occurs when TTC equals to 0. An alternative approach is to use the number of small gaps or emergency braking. Vogel [143] compares two safety indicators headway and TTC and finds that for vehicles in a car following situation headway and TTC are independent of each other. Small headways generate potentially dangerous situations. On the other hand, TTC indicates the actual occurrences of dangerous situations.

To evaluate the safety characteristics of a merging strategy we distinguish two
types of events:

- **Collision**: there is more than one car at the same position on the road at the same time.

- **Near-collision**: the gap between two successive cars is smaller than the minimum safety gap but if the front car does not brake a collision would not occur.

Near-collision indicates a traffic hazard. To guarantee collision-free driving, cars should maintain a safe distance. Since the basic requirement of a merging strategy is to avoid collisions, we expect mergings in our algorithms are collision-free. Therefore, we count the number of near-collisions (violations of safety distance) relative to the total number of cars in a period of time. This measurement is defined as the number of violations in our experiments in Chapter 6.

### 3.3.2 Efficiency

A good merging strategy utilizes the road resources more efficiently. We use the following criteria to evaluate efficiency.

- **Flow**: the average number of cars passing a certain position over a period of time. It equals to the product of velocity and density \( f(v, \rho) = v\rho \), where density \( \rho \) is the number of cars per unit distance. To evaluate the overall traffic flow we measure the number of cars and the average velocity over the entire main road every minute. The largest flow would occur if cars are bumper to bumper (maximum density) and move at the speed limit (maximum velocity). Clearly, this is not achievable in reality for safety reasons. If the density is maximal then cars are not moving, yielding a minimum traffic flow of zero. Therefore, the maximum of traffic flow occurs at a certain density with a corresponding velocity.

- **Capacity**: the maximum number of cars passing a certain position over a period of time, i.e., the maximum traffic flow. Traffic congestion occurs when the
demand exceeds the capacity, thus achieving a higher capacity can effectively avoid traffic congestion. However, the real capacity is typically lower than the theoretical maximum capacity because it is not only determined by the road configuration (e.g., the number of lanes, the lane width) but also by perturbations in the traffic flow, such as lane changing or merging. We expect that a merging strategy has a significant impact on the real capacity of a road. Therefore, the observed reduction of the real capacity from the theoretical capacity is an important criteria for merging strategies investigated in our work. Neglecting the length of cars, the theoretical capacity $C_{th}$ mainly depends on the time headway $T$ (i.e., a time gap between two cars): $C_{th} \approx 1/T$ (see Eq. (3.9)).

When traffic flow achieves its maximum, the cars travel at the same velocity (3.5), the distance between each car equals to the safety distance (3.6) and is evenly distributed (3.7).

$$\Delta v = 0 \quad (3.5)$$

$$s = s^* \quad (3.6)$$

$$n(l + s) = L \quad (3.7)$$

where $n$ is the number of cars, $l$ is the length of a car, and $L$ is the length of the road.

Using $f(v, \rho) = v\rho$, we can calculate the capacity as

$$C_{th} = \frac{n}{L} \quad (3.8)$$

We can derive from (3.2), (3.5), (3.6), (3.7) and (3.8) the following:

$$C_{th} = \frac{1}{T} \left[ 1 - \frac{l + s_0}{l + s_0 + vT} \right] \quad (3.9)$$

- **Stability**: the ability to adapt to a new traffic situation. It impacts the capacity and the total number of merged cars. We expect that a stable merging algorithm
achieves higher capacity by accommodating more merged cars. A merging car impacts the stability of two traffic streams because in one traffic stream a new car is added and in the other stream this car is removed. As a result, the cars of both streams adjust their velocities and gaps to achieve approximate equilibrium conditions. As the density of one stream increases the gaps between the cars of this stream decrease until no farther cars can merge without violating the safety distance. The gaps between cars distribute more evenly in the stable traffic than the unstable traffic. For the same traffic density, there is higher merging capacity in the stable traffic. Thus, traffic flow and capacity are also higher in the stable traffic.

- **Delay**: the total time required for a merging algorithm to fill the main road with a certain number of merged ramp cars. It describes how quickly the system absorbs merging cars. In merging scenarios, a ramp car locates a gap on the main road and adjusts its velocity to fit in the gap in order to merge. A less efficient merging algorithm may cause more delay for the merging car because the car has to wait if it can not adjust its velocity appropriately to merge. Normally, it is more difficult for a car to merge in higher traffic flow and hence, higher traffic flow leads to more delay. However, for a good merging algorithm, the delay should be less impacted by the traffic flow. We use an accumulative time to measure the delay.

- **Total Travel Time**: the time duration for a certain number of cars to complete merging to the main road and traveling a certain distance before exiting the main road. This measurement reflects the merging delay, the capacity and the average velocity on the main road.

- **Fuel consumption**: varies as much as 45% [140] between different drivers using identical cars. Multiple studies show that accelerating slower will dramatically improve fuel efficiency: the author in [121] found fuel savings up to 37% with an average of 31% simply by accelerating a car slowly. Similar findings [113] show
slower acceleration saving between 14% and 21% of fuel use over aggressive acceleration. Different merging strategies have a significant impact on the acceleration. Therefore, we use the average acceleration as a substitute for assessing the fuel consumption.

Guaranteeing safety is a fundamental requirement of proposed merging algorithms. It is not measured in Chapter 5 since all proposed algorithms satisfy the safety requirement. However, when the accuracy of sensor measurements is considered in Chapter 6, safety becomes the most important criterion. A safety evaluation is performed accordingly in Chapter 6, and an efficiency evaluation is performed in Chapter 5, 6 and 7.

3.4 Conclusion

In this chapter we described two microscopic traffic models adapted in our work: the car-following model IDM and the lane-changing model MOBIL. They have the advantage that they are capable of capturing both individual and collective traffic properties as discussed in Section 2.1. We analyzed the impact of parameter values for IDM and observed that for certain sets of parameter values IDM represents different driving styles and vehicle types. As merging is a special case of lane-changing, we integrated MOBIL with IDM via the acceleration. To evaluate the traffic merging strategies we presented performance measurement methodologies. We identified several performance criteria with respect to safety and efficiency. We also highlighted the correlation of some criteria, for example, increased traffic flow implies higher velocity coupled with tighter gaps, while improved safety implies lower velocity difference and larger gaps.

To realize the goal of developing and evaluating traffic merging algorithms, in the next chapters we detail the traffic simulator and present proactive and robust traffic merging algorithms.
Chapter 4
Microscopic Traffic Simulator
Incorporating Sensor-Enabled Cars

The microscopic traffic model including the car-following model and the lane-changing model described in Chapter 3 is the foundation of the traffic simulator that we design and implement. We extend the microscopic simulator [133], developed by Treiber, to incorporate the explicit modeling of the spatial information obtained by sensor-enabled cars and the cooperation among drivers. The main advantages of our simulator are the following: it is scalable and robust; it provides a user interface (showed later in a screenshot), which is used to configure the traffic model and simulation parameters, and visualize the dynamic traffic properties; and it provides the capability to evaluate traffic management algorithms under realistic scenarios. In this chapter, we first identify the importance of microscopic traffic simulators and introduce a few widely used simulators. Then, we present the overall design and implementation of our simulator.

4.1 Introduction

Simulation is an indispensable tool for the optimization of complex traffic systems. It provides a controllable environment for repeatable experiments. Microscopic traffic simulation tools are increasingly being applied to design and evaluate transportation applications. A range of applications such as adaptive traffic management, traveler information, and incident management systems [38] are difficult to evaluate using tra-
ditional analytical tools due to the complex nature of the underlying system dynamics.

This section discusses some commonly used simulation tools: Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks (AIMSUN) [4], Microscopic Traffic SIMmulator (MITSIM) [97, 148, 149], PARAllel MICroscopic Simulation (PARAMICS) [112], and Verkehr In Stadtten–SIMulation (VISSIM) [141].

AIMSUN is used for the assessment of intelligent transportation systems (ITS), advanced transport analysis and advanced traffic management systems (ATMS). AIMSUN is based on Gipps car-following model [53] and lane-changing model [54] discussed in Section 2.1. MITSIM supports the evaluation of dynamic traffic management systems: advanced traffic control, route guidance and surveillance systems. Hasan et al. use MITSIM for evaluating ramp control algorithms [61]. MITSIM is based on Herman’s car-following model [63] (see Section 2.1.1) and Gipps lane-changing model [54]. PARAMICS is used for simulations of traffic congestion in large-scale road networks and based on the psychophysical model proposed by Fritzsche [49]. VISSIM offers a wide variety of urban and highway applications, integrating public and private transportation. VISSIM [46] is based on the psychophysical model proposed by Wiedemann [145, 146]. Psychophysical models assume that a driver only reacts to the front car when perceptual thresholds are exceeded in terms of velocity difference and distance to the front car. Psychophysical models are complex high-fidelity models, which try to represent traffic as realistic as possible at the cost of a large number of parameters. On the other hand, Gipps-based models are simple and show realistic driver behavior, but they fail to capture collective traffic properties, such as traffic instabilities [135].

We choose Treiber’s Dynamic Traffic Simulation (DTS) [133] to develop our own traffic simulator because it has the following advantages: DTS is open source, well-documented, and it is easy to extend. More importantly, it accurately describes both the individual driving behavior and collective patterns of the traffic flow, such as stop-and-go waves and traffic phase transitions (see Section 2.1.3) identified in empirical observations [135]. We need to extend DTS and integrate it with sensors and cooperation among drivers.
4.2 Overall Design

The traffic simulator that we design captures the relationship among the three components: traffic merging algorithms, the underlying traffic model and sensors. Figure 4.1 shows that traffic merging algorithms provide rules or recommendations for the traffic model in determining the movements of vehicles. The sensors collect traffic data that are used by the merging algorithms.

![Figure 4.1: Structure of our traffic simulator](image)

4.3 Implementation

We use object-oriented programming to implement the main components and represent relationships such as inheritance and composition as classes. A hierarchical view of these classes is given in Figure 4.2. Their functions are detailed in the following:

- **TraSimGUI**: represents the graphical user interface through which a user can define some parameters of a simulation, such as the length of a road, the number of cars, the incoming rate of ramp cars, the distribution of incoming cars, the merging algorithm applied, the outgoing rate of cars. The GUI displays the evolution of the system at run time by showing the movements of individual cars on the road network.
• **SimOutput**: this class displays the results. It can collect aggregated traffic information as the measures to evaluate algorithm performance, which we have discussed in Section 3.3. Currently, the available metrics are: the number of violations, traffic flow, the average velocity, the merging delay, total travel time, and the level of acceleration.

• **MicroStreet**: this class represents road sections that consist of a single-lane or two-lane for one direction. Two subclasses are **OutRoad**: the off-ramp via which cars go out of the main road; and **OnRamp**: the on-ramp for cars to enter the main road. The main component of this class is a range of merging algorithms that determine the behavior of how to merge. We detail those algorithms in Chapter 5 and Chapter 6.

• **InOutCars**: it connects an off-ramp with an on-ramp, which means after a car goes out of the main road it can enter again by the on-ramp. This class is of great importance in simulating large scale road networks, which consist of multiple main roads and ramps. We perform multi-ramp simulations in Chapter 7.

• **Car**: this class models a sensor-enabled car. We detail the sensing component in Section 4.3.2.

The longitudinal and lane-changing model are implemented in the class **IDM** and **LaneChange**. Two classes are inherited from IDM: **IDMloop** and **IDM-ramp**. They define the parameters of the car-following model for the main road cars and the ramp cars, respectively. These parameters include the speed limit, the desired time headway, the maximum acceleration and deceleration, which can model different types of cars.

### 4.3.1 Road Networks

The road network consists of connected road sections such as main roads, on-ramps and off-ramps implemented as the class **MicroStreet**. The classes **OnRamp** for on-ramps and **OutRoad** for off-ramps are derived from **MicroStreet**. The main attributes
4.3 Implementation

are the road length, number of lanes, speed limit, etc. The main elements of **MicroStreet** are:

- **street**, a vector representing the vehicles.
- An **update** method, which determines a vehicle’s movement, is invoked at every step. It calls all methods mentioned below.
- Methods for moving the vehicles: **translate**, accelerating them: **accelerate** and performing the lane changes: **changeLanes**.
- A sorting routine **sort** for rearranging the vehicle order in **street** in the order of
decreasing longitudinal positions.

- A method `ioFlow` for implementing the upstream and downstream boundary conditions (inflow of `OnRamp` and outflow of `OutRoad`).

The road network is one of the fundamental building blocks for modeling traffic. In our simulator, we first model the road network consisting of a single main road loop. Then we extend it to multiple loops.

We initially design the main road as a loop for the following reasons. First, the increasing load shows when a strategy fails if we only allow cars to merge without an opportunity to leave the main road. The loop simulates certain densities of freeway merging. This allows us to focus on a main parameter to describe the behavior of the entire system. For example, the main parameter could be the density of one stream or the incoming rate from the other stream. Second, in a closed loop we can examine the road capacity (how many cars the road can contain without congestion). In order to test when a congestion occurs, the initial input number of cars needs to change for every simulation run if it is not a closed loop, whereas in the loop the number of cars simply increases. Third, it is a completely controlled environment and can be extended easily to simulate current highways such as the ring-road (Périphérique) in Paris, France.

After investigating the merging algorithms under scenarios such as a single loop with one on-ramp and one off-ramp, we extend the simulator to generate scenarios with two loops, on-ramps, and off-ramps. Since a loop has two inputs and two outputs, we can use the loop as a basic building block to construct arbitrary road networks.

### 4.3.2 Vehicle Movements

We assume that all cars in the simulator are equipped with sensors for position and velocity information. To represent the sensor-enabled cars, a sensing component is implemented in the `Car` class using the following methods: `getPosition`, and `getVelocity`. 
which return a car’s position and velocity, respectively.

The accuracy of sensors is a key parameter that influences traffic merging algorithms because merging algorithms rely on the sensed data. The method `getPosition` takes the sensor accuracy as an argument to generate deviations from the actual position. We investigate a variety of merging algorithms based on perfect sensor measurements in Chapter 5. In Chapter 6, we study the impact of inaccurate positioning data on merging algorithms and design robust merging algorithms. We assume that the error $e$ in the positioning measurements follows a truncated normal distribution [123], i.e., the probability distribution of a normally distributed random variable whose value is bounded within a range. Therefore, the positioning measurement errors are within the sensor’s accuracy level. If the actual position is $p$, the position measurement is $p + e$. Figure 4.3 provides an example of the error distribution. The dotted line is a normal distribution and the solid line is the truncated normal distribution, which is used in generating sensor measurement errors.

![Figure 4.3: Example of sensor error distribution: positioning accuracy within 1 m](image)

Figure 4.3 shows the update process for main road cars. In the update procedure, a car on the main road does the following: checks whether it is interacting with a ramp car; if yes, then it computes the acceleration based on the merging algorithms; otherwise, it computes the acceleration based on the car-following model IDM. Then,
Figure 4.4: Flow chart of the main road cars’ update process
4.3 Implementation

A main road car checks whether it needs to change lane and whether the gap between neighbors on the new lane is feasible for the desired lane change. A car’s velocity is then updated and it moves to the new position based on

\[ v_{t+\Delta t} = v_t + a_t \Delta t \]  

\[ p_{t+\Delta t} = p_t + v_t \Delta t + \frac{1}{2} a_t \Delta t^2 \]

where \( v \) is the velocity, \( a \) is the acceleration, \( p \) is the position, and \( \Delta t \) is the update time interval. Although the acceleration functions are generally nonlinear, car-following models assume constant accelerations within each update time interval \( \Delta t \). A typical \( \Delta t \) is between 0.1 s and 0.2 s [80]. The value of \( \Delta t \) is based on the update rate of sensors, for example, if equipped with a 10 Hz GPS receiver [2], the update rate is 0.1 s.

A ramp car updates its velocity and position in the same way as a main road car before the ramp car passes the decision point, where it starts to compute when and where to merge into the main road based on the merging algorithm. The details of merging algorithms will be presented in Chapter 5 and Chapter 6. If a ramp car has passed the decision point, it conducts pre-merge adjustment in cooperation with the neighboring cars on the main road in order to merge safely. After a ramp car arrives to the merging point, it merges into the main road if it is safe to merge. Otherwise, in the following time steps it either merges in the safe condition or it stops at the end of the ramp and waits to merge. For a new car arriving at the ramp, whether or not it is added to the ramp list, depends on the distance to the last ramp car. If there is sufficient space, the new car is added to the end of the ramp car list. Otherwise, the new car waits for the next time step. Figure 4.5 shows the process of how the ramp cars update.
Figure 4.5: Flow chart of the ramp cars’ update process
4.4 Recommendation System

The system provides recommendations to a human driver rather than forcing a merging decision. However, there can be situations when a human driver does not follow the system’s recommendations, for example in an attempt to drive faster. Assume the system assigns the best slot for the driver to merge, but the driver tries to speed up and merges in the slot in front of the recommended slot. This behaviour is accepted if it does not violate the road safety requirements. Otherwise, if the system recognizes an unsafe driving situation, it takes control of the driver by slowing down or performing an emergency brake to guarantee safety and prevent a potential accident.

4.4.1 Tolerance for Drivers’ Mistakes

In our study, sensor inaccuracies or human misjudgements have similar impact on the developed merging algorithms. This thesis addresses the robustness issue of sensor measurements, which implicitly captures human drivers’ mistakes.

Similar to work in this thesis, other driver-assisted systems also compute an optimal way for driving and provide recommendations. In these systems, drivers are usually allowed to disobey the recommendations as long as the driving behaviour does not endanger others. Our work is not focused on a human-centered system. However, we acknowledge human-system interactions and provide tolerance to drivers’ mistakes, in contrast to the concept of platooning in automated highway systems.

4.4.2 Personalization

The system can be personalized as it allows a driver to specify individual input for parameter settings regarding safety measures, such as the minimal safety distance and time to reach the merging point. However, the system does not allow any setting outside physically safe distance/time headway. For the sake of simplicity, in this thesis the simulations use a single parameter to unify the safety measurements.

It is possible to extend our work to accommodate additional human factors and
more human-system interactions, which will increase the acceptance of the system in real traffic conditions. An extensive study of human activities and interactions during a driving scenario, however, is outside the focus of the thesis.

4.5 Graphical User Interface

Figure 4.6 shows a screenshot of our simulator. The road network consists of a two-lane main road with two on-ramps and one off-ramp. The cars on the main road are moving in a counterclockwise direction. The aim of our work is to design traffic merging algorithms that can mitigate or even eliminate the traffic congestion. Figure 4.6(a) shows a traffic breakdown at one of the on-ramps caused by increasing local traffic near the ramp (highlighted by a circle), enlarged in Figure 4.6(c). The ramp area becomes a bottleneck. Our simulator provides a graphical user interface. We explain the user input by using Figure 4.6(b). There are two buttons for starting and stopping a simulation. One group of radio buttons allow to select the distribution of arrival ramp cars. Currently, the arrival pattern could be either constant or Poisson distributed. The other group of radio buttons are for choosing a traffic merging algorithm. The scroll bars are for adjusting key parameters: initial number of cars on the main road; outgoing percentage of cars that exit the main road by the off-ramp; incoming rate of cars on each of the two on-ramps; simulation speed, which is the update rate of the GUI; and positioning accuracy of sensors.

4.6 Conclusion

This chapter presented the design and implementation of the traffic simulator we used for evaluating the proposed traffic merging algorithms. Our simulator enables us to study traffic merging strategies under realistic scenarios. It explicitly models the spatial information exchange among sensor-enabled cars and driver cooperation. We detailed the implementation of road networks, vehicle movements, and the graphical user interface of our simulator. We highlighted the sensing component, which is in-
Figure 4.6: Graphical user interface of our simulator
dispensable for integrating sensor-enabled cars. Our simulator is flexible and easy to extend, for example, to simulate different car-following models and incorporate a number of cars without sensors. The graphical user interface helps to observe cars’ behaviors and identify errors.

In the subsequent chapters, we investigate traffic merging algorithms incorporating the sensed information. We perform extensive experiments to evaluate different merging algorithms using the simulator.
Chapter 5
Proactive Merging Algorithms

We assume that cars are equipped with sensors that can measure distance to neighboring cars and communicate their velocity and acceleration readings among each other. In this chapter, we propose proactive traffic control algorithms for merging different streams of sensor-enabled cars into a single stream. A proactive merging algorithm decouples the decision point from the actual merging point. Sensor-enabled cars allow us to decide where and when a car merges before it arrives at the actual merging point. This leads to a significant improvement in traffic flow as velocities can be adjusted appropriately. We compare proactive merging algorithms against the conventional priority-based merging algorithm in an extensive set of experiments in our simulator described in Chapter 4 and using the measurements presented in Chapter 3. Results show that proactive merging algorithms outperform the priority-based merging algorithm in terms of flow and delay. Our experiments demonstrate that the traffic flow can be increased by up to 100% and the delay can be reduced by 30%.

5.1 Introduction

The merging section of highways is a bottleneck that influences the traffic throughput significantly. There is already a considerable amount of research, in particular based on queuing theory and statistics. However, these approaches such as the approach in [30] neither integrate different information sources available from sensor-enabled cars nor do they compare different merging strategies under realistic experiments. Our work, therefore, focuses on optimizing traffic throughput when merging different
traffic flows at intersections by using sensor-enabled cars. We set out to explore the benefit of applying simple traffic control rules at the merging section if all cars are sensor-enabled.

The key insight is that dissociating the decision point and the actual merging point can optimize traffic flow. Currently, most of the literature [7, 74, 92] assumes that the decision point and the actual merging point coincide based on the fact that traditionally, a driver arrives at the merging point and makes a decision of how to merge at that point. The reason that we are able to make proactive decision is that sensor-enabled cars can obtain necessary information for safe merging much earlier than normal cars.

To compare proactive merging strategies, we outline a priority-based merging strategy, which is currently applied in merging sections (referred to as R). In the strategy R, the cars from one stream always have the right of way, e.g., cars from the ramp must give way to the cars on the main road. A ramp car only merges when the gap between two cars on the main road is large enough. Meanwhile, cars on the main road completely ignore the ramp car and do not create gaps for the ramp car to merge. The main drawback of this strategy is the lack of coordination between two traffic streams, which results in excessive delays for the merging traffic.

5.2 Design Procedure

Proactive merging algorithms aim to

(a) maximize the number of ramp cars that can merge to the main road;

(b) maximize the traffic flow on the main road; and

(c) minimize the merging delay.

These goals are achieved by minimizing the disturbance of merging in the subsequent traffic on the main road. The reasons are detailed as follows. The traffic flow \( Q \) is defined as the number of vehicles \( n \) that pass a fixed position in a time interval \( t \): \( Q = n/t \). For a large \( n \), the time interval \( t \) equals to \( \sum_{i=1}^{n} T_i \), where \( T_i \) is the time headway,
which is the arrival time difference between the vehicle \( i \) and \( i - 1 \) \((T_i = t_i - t_{i-1})\). Thus, the traffic flow is

\[
Q = \frac{n}{t} = \frac{1}{\frac{1}{n} \sum_{i=1}^{n} T_i} = \frac{1}{T}
\]

i.e., the reciprocal of the average time headway \( T \).

The theoretical maximum traffic flow occurs under the equilibrium condition. A smaller average time headway leads to a higher traffic flow. Anticipating future traffic situations or checking the traffic situation several cars ahead can reduce \( T \). In practice, the maximum traffic flow is typically lower than the theoretical value, since it depends on traffic stability, in particular due to transient traffic disturbances (e.g., emergency stops, sudden lane changes, or merging). Merging events influence an equilibrium condition due to local perturbations, such as a car that merges from a ramp into the main road. As a result, subsequent cars on the main road adjust their velocities and distances to adapt to the new traffic situation. To achieve a higher traffic flow, merging algorithms should minimize the impact of ramp merging on the main road traffic, by creating larger inter-vehicle distances on the main road when a ramp car merges. The maximal velocity that ensures collision-free braking increases with larger inter-vehicle distances.

Therefore, we design proactive merging algorithms following two steps: first, compute an appropriate merging order for cars on the main road and on the ramp prior to the merging point; second, calculate accelerations or decelerations for those involved cars to ensure accurate velocity and position when a ramp car merges onto the main road.

### 5.3 Single Lane Scenario

#### 5.3.1 Algorithms for the Main Road with a Single Lane

The basic idea behind the algorithms is to use the knowledge of position, velocity, and acceleration received beforehand in making merging decisions. Each car knows
about a small number of cars in the neighborhood. When a car arrives at the decision point, which is before the actual merging point, it chooses a proper gap to prepare for merging. Along the path of approaching to the merging point, it adjusts velocity to catch that gap when it arrives at the actual merging point. In such way, the velocity change is smaller compared to the strategy R.

With the aid of Figure 5.1, we explain the notations we use in the following text and pseudocode (Algorithm 1–4). The main road and the ramp meet in the merging area. We denote the start point by $O$ and the end point by $E$, which coincides with the end of the ramp. The decision point is denoted by $S$, which indicates the point at which the driver decides where and when to merge between $O$ and $E$. The decision point $S$ should be chosen to optimize traffic merging. In simulations we have varied the position of $S$ and found that the distance between $S$ and $O$ of 100 m to 200 m achieves the best performance for merging algorithms with current traffic safety constraints. If a ramp car decides to merge too early (that is the distance $SO$ is large), variations of traffic situations may lead to inaccurate pre-computed merging point since the prediction becomes invalid. On the other hand, if the decision is made too late (that is the distance $SO$ is too small) then the chance to compute the ideal velocities becomes less as there is not sufficient time for cars to adjust their velocities for optimal merging. Although it is theoretically possible to check the available gap continuously and advice a new gap when traffic conditions have changed, the cost is likely to be prohibited. We should balance the complexity of computation and the safety of traffic merging.

![Figure 5.1: Notations of proactive merging algorithms](image)

We assume that communication cost is not an issue for sensor-enabled cars as its energy cost is negligible compared to transportation cost. Therefore, sensor-enabled
cars can listen continuously. Once a ramp car arrives at S, it initiates the communication and exchanges position, velocity, and acceleration information among cars within communication range (see Section 2.2.3).

A car on the ramp approaches the merging point as if there is a car stopping at the end of the merging section. Therefore, its velocity is composed of a decreasing tendency and the adjustment to the gap on the other stream. The requirement for the merging algorithm is that a safety distance should be guaranteed not only along each road before merging but also at the point of and after the merging maneuver. The objective of proactive merging is to make decision early so that a ramp car can merge to the main road without slowing down significantly.

The proactive merging algorithm works as follows:

**phase 1** make decision of which car merges first at the decision point considering the traffic condition rather than which road the car is on, in contrast to the strategy R.

**phase 2** adjust the velocity prior to the merging point. This is the key feature of proactive merging algorithm.

All proactive algorithms assume that there is a decision start point S before the merging start point O. They use the segment between S and O to decide, based on different types of spatial information, which car of the two traffic streams should merge first. All algorithms are initiated by the ramp car once it arrives at S. Proactive algorithms essentially determine which car has the right of way by using either the shortest distance to O or the shortest arrival time at O. The first strategy is a distance-based merging algorithm (referred to as D) that measures the distance from the main road car to the merging point O and compares it with the distance between S and O (denoted as SO). If it is greater than SO, then the ramp car merges first. Otherwise, we repeatedly compare the distance δ of the subsequent main road car and the distance δ' of the ramp car to O. We continue until δ > δ' in order to find a slot for the ramp car. The second strategy is a velocity-based algorithm (referred to as V) that also takes the velocity into account. It computes the time for a car to arrive at the merging point and
compares the time for cars from the two traffic streams. The car with shortest arrival
time merges first. The third strategy is a platoon velocity-based merging algorithm
(referred to as PV). It is a refined version of the velocity-based merging algorithm V.
Instead of comparing the ramp car against one main road car as in V, PV compares
the ramp car with a group of main road cars on the segment $SO$ and assigns a slot
between two main road cars for the ramp car. The slot is chosen based on the arrival
time calculated at $S$.

In Table 5.1, we summarize proactive merging algorithms D, V, and PV, which we
implement and compare against the priority-based algorithm R.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Information</th>
<th>Right of Way</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority-based</td>
<td>No info. in advance</td>
<td>Main road car</td>
<td>Main road priority</td>
</tr>
<tr>
<td>(R)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance-based</td>
<td>Position</td>
<td>Closest to the merging</td>
<td>Velocity does not vary much</td>
</tr>
<tr>
<td>(D)</td>
<td></td>
<td>point</td>
<td></td>
</tr>
<tr>
<td>Velocity-based</td>
<td>Position, velocity</td>
<td>Arrival time to the</td>
<td>Acceleration does not vary much</td>
</tr>
<tr>
<td>(V)</td>
<td></td>
<td>merging point first</td>
<td></td>
</tr>
<tr>
<td>Platoon-velocity-based</td>
<td>Position, velocity</td>
<td>Assign a slot in a group of main road cars</td>
<td>Acceleration does not vary much</td>
</tr>
<tr>
<td>(PV)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.3.2 Pseudocode for Merging Algorithms

A car is characterized by the following attributes: position, velocity, and acceleration. A car list is a logical concept for a number of cars that share common characteristics. For example, the three car lists: $RampList$, $MainList$, and $OutList$, respectively means the ramp cars that have not arrived at the merging point, the main road cars that have not arrived at the merging point, and the potential sequence of those two groups of cars to pass the merging point computed by merging algorithms. Based on the definition, $RampList$ is $\{x, y\}$, and $MainList$ is $\{c, d, e\}$ in Figure 5.1. $OutList$ varies depending on the merging algorithm. It may be $\{c, d, x, e, y\}$ in priority-based merging algorithm, $\{c, x, d, y, e\}$ in distance-based merging algorithm, $\{x, c, d, y, e\}$ in velocity-
based algorithm if car x with much higher velocity than car c. In Algorithm 1–4, we
detail how the algorithm R, D, V and PV work.

The main steps for the priority-based algorithm R (Algorithm 1) are the following:
when the ramp car r arrives at the merging area between O and E (see Figure 5.1) it
searches for a gap on the main road (line 1.1 – 1.2). We assign a virtual position for each
ramp car on the main road. The function getNeighborsOnMain returns the front car f
and the back car b on the main road. Then the function changeOK (line 1.3) computes
whether the safety criterion is satisfied, which requires the distances between r and f,
b and r larger than the minimum safe distance. If the function changeOK returns true, r
merges into the main road (line 1.4), otherwise r waits for the next time step (line 1.6).

Algorithm 1: R – the priority-based merging algorithm

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>r is at the merging area between O and E</td>
</tr>
<tr>
<td>1.2</td>
<td>getNeighborsOnMain(r, MainList) /* Find the front car f and the back car b on the main road */</td>
</tr>
<tr>
<td>1.3</td>
<td>if changeOK(r, f, b) then</td>
</tr>
<tr>
<td>1.4</td>
<td>r.merge() /* r can merge in front of b */</td>
</tr>
<tr>
<td>1.5</td>
<td>else</td>
</tr>
<tr>
<td>1.6</td>
<td>r.wait() /* Wait for next step */</td>
</tr>
</tbody>
</table>

Different from R (Algorithm 1), the proactive merging algorithm (Algorithm 2 –
4) is called before r arrives at the merging point O. Algorithm 2 makes a distance-
based decision on whether the ramp car r has the right of way. Once the ramp car
r has passed the decision point S (line 2.1), the distances of the ramp car r and main
road car m from the merging point O are calculated (line 2.2 – 2.3). If r is closer to the
merging point O, it merges in front of m. Otherwise, r waits to find the next possible
gap to merge (line 2.4 – 2.7). Algorithm 3 is similar to Algorithm 2, except that the
decision of the right of way is based on the arrival time depending on the distance as
well as the velocity (line 3.2 – 3.4).
Algorithm 2: D – Distance-based decision of the right of way

Input: \( r \): the ramp car; 
\( m \): the main road car before the merging point \( O \)

1. if \( r.\text{position} > S \) then
   2. /* \( r \) has passed the decision point \( S \) */
   3. \( d_m \leftarrow m.\text{position} \) /* the distance of \( m \) and \( O \) */
   4. \( d_r \leftarrow r.\text{position} \) /* the distance of \( r \) and \( O \) */
   5. /* decide which car gets the right of way: \( m \) or \( r \) */
   6. if \( d_m > d_r \) then
      7. /* \( r \) is closer to \( O \) than \( m \) */
      8. \( r.\text{merge}() \) /* \( r \) can merge in front of \( m \) */
   9. else
      10. /* \( m \) is closer or at the same distance to \( O \) */
      11. \( r.\text{wait}() \) /* Find next possible gap */

Algorithm 3: V – Velocity-based decision of the right of way

Input: \( r \): the ramp car; 
\( m \): the main road car before the merging point \( O \)

1. if \( r.\text{position} > S \) then
   2. /* \( r \) has passed the decision point \( S \) */
   3. \( t_m \leftarrow m.\text{position}/m.\text{velocity} \) /* time when \( m \) arrives at \( O \) */
   4. \( t_r \leftarrow r.\text{position}/r.\text{velocity} \) /* time when \( r \) arrives at \( O \) */
   5. /* decide which car gets the right of way: \( m \) or \( r \) */
   6. if \( t_m > t_r \) then
      7. /* \( r \) arrives at \( O \) earlier than \( m \) */
      8. \( r.\text{merge}() \) /* \( r \) can merge in front of \( m \) */
   9. else
      10. /* \( r \) arrives later or at the same time at \( O \) */
      11. \( r.\text{wait}() \) /* Find next possible gap */

Algorithm 4 is a refinement over Algorithm 3 to find a slot for the ramp car \( r \) on the main road. When a ramp car \( r \) from the ramp car list \( \text{RampList} \) passes the decision point \( S \), it checks whether there are cars on the main road within its communication range. If there is no car on the main road, \( r \) merges (Line 4.1 – 4.6). Otherwise, it computes the arrival time of the ramp car \( r \) and the group of main road cars (line 4.7 – 4.13). Then a slot is assigned for the ramp car \( r \) in between two main road cars based on the sequence of arrival time (line 4.14 – 4.18).
Algorithm 4: PV – Assign a slot on the main road for the ramp car $r$

**Input**: $\text{RampList}$: list of ramp cars; $\text{MainList}$: list of main road cars before the merging point $O$  
**Output**: $\text{OutList}$: list of cars in the merging sequence

1. **On initialization**: $\text{cond} \leftarrow \text{TRUE} /* \text{cond}: \text{a flag to indicate whether to get } m \text{ from } \text{MainList} */$
2. **while $\text{RampList} \neq \emptyset$ do**
   1. /* There are cars on the ramp */
   2. $r \leftarrow \text{RampList}.\text{pop}() /* \text{Get the first car } r \text{ from the ramp car list} 
   \text{RampList} */$
   3. **if $r.\text{position} > S$ then**
      1. /* $r$ has passed the decision point $S$ */
      2. **if $\text{MainList} = \emptyset$ then**
         1. /* There is no car on the main road so append $\text{RampList}$ to $\text{OutList}$ */
         2. $\text{OutList} \leftarrow \text{OutList}.\text{Append}(\text{RampList}) /* r \text{ can merge} */$
      3. else
         1. **do**
            1. /* Get car $m$ from the main road car list $\text{MainList}$ */
            2. $m \leftarrow \text{MainList}.\text{pop}()$
            3. $t_m \leftarrow m.\text{position}/m.\text{velocity} /* \text{time when } m \text{ arrives at } O */$
            4. $\text{cond} \leftarrow \text{FALSE}$
            5. $t_r \leftarrow r.\text{position}/r.\text{velocity} /* \text{time when } r \text{ arrives at } O */$
            6. /* decide which car gets the right of way: $m$ or $r$ */
            7. **if $t_m > t_r$ then**
               1. /* $r$ arrives at $O$ earlier than $m$ */
               2. $\text{OutList}.\text{Append}(r) /* r \text{ can merge in front of } m */$
            8. else
               1. /* $r$ arrives later or at the same time at $O$ */
               2. $\text{OutList}.\text{Append}(m) /* \text{set } \text{cond} \text{ to true so that } m \text{ is popped in the next iteration} */$
               3. $\text{cond} \leftarrow \text{TRUE} /* \text{Find next possible slot} */$
      4. **while $\text{MainList} \neq \emptyset$ and $\text{cond} = \text{TRUE}$;**

5.4 Multiple Lane Scenario

5.4.1 Lane-Changing Model

Our lane-changing behavior could be seen as car-following in a “wide” lane that consists of the car’s current lane and the adjacent lane to which the car intends to change to. For incentive reasons, a lane change is triggered when the distance is below a certain threshold or a car’s velocity decreases. For safety reasons, there is no neighboring car on the lane that is adjacent to the cars lane. The distance to the car on the adjacent lane in front of the car that wishes to change its lane depends on its own velocity and the velocity difference to its front car on the adjacent lane. A similar rule applies to the back car on the adjacent lane. Therefore, the lane-changing behavior in our model is converted to a car-following model based on IDM.

We use a “virtual car” 1) to ensure no accidents occur due to lane changing and 2) to compute the impact of a lane change on the traffic of that lane. For example, the ramp car assumes a virtual car in front of it based on the gap it detects on the main road. Therefore, the ramp car adjusts its velocity according to two streams: the stream of cars on the main road and the stream on the ramp simultaneously, which guarantees collision-free merging for the ramp car.

5.4.2 Algorithms for the Main Road with Multiple Lanes

When a main road has multiple lanes the cars can change lanes to overtake other cars or to make space for arriving ramp cars. Our experimental results show that PV is the most effective proactive merging algorithm for a single lane. Thus, we extend it to roads with multiple lanes. In addition, we develop a refined version of PV, which we call Cascading PV (referred to as CPV).

CPV assumes two lanes but can be generalized to roads with more lanes. Cars on the outer lane (OL) and the inner lane (IL) of the main road can change lanes prior to the merging area to create a larger gap for the merging ramp cars. The position where the main road cars start to change lanes is denoted as C in Figure 5.2. Other settings
are similar to the single lane scenario. CPV aims to minimize the deceleration (or maximize the acceleration) on both lanes which might result in a lane switch between two cars. In Table 5.2, we summarize CPV merging algorithm.

Table 5.2: Traffic merging algorithm CPV overview

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Condition</th>
<th>Merging Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5.2: Notations for CPV

Figure 5.3: Illustration of lane switch in CPV
Scenario C is the most common and challenging situation, which occurs when the traffic density is high. We explain CPV and the following pseudocode Algorithm 5 using Figure 5.3. The triple of cars on the inner lane \( i_b, i, i_f \), the triple of cars on the outer lane \( o_b, o, o_f \) and the ramp car \( r \) are at positions shown in Figure 5.3. Car \( r \) is attempting to merge in between \( o_b \) and \( o \). CPV computes the gap between \( r \) and \( o \): \( d_{ro} \), and \( r \) and \( i \): \( d_{ri} \). If \( d_{ro} < d_{ri} \) (line 5.1), then there is potentially more space for \( r \) to merge if \( o \) and \( i \) switch lanes. To decide whether \( o \) and \( i \) switch lanes, CPV considers the velocity of \( o, i, o_b, i_b \) and the velocity difference of their front cars since they determine the safety distance (line 5.2 – 5.5). Suppose car \( o \) and \( i \) switch lanes, the four new gaps would be: \( d'_{bi} \), \( d'_{fi} \), \( d'_{bo} \), and \( d'_{fo} \). If each new gap is larger than the corresponding safety distance (line 5.6) and the sum of acceleration is larger than the one without lane changing (line 5.7 – 5.11), which means there is more space to merge, \( o \) and \( i \) will switch lanes (line 5.12). Otherwise, \( o \) and \( i \) will not change lanes (line 5.13 – 5.18). Afterwards, PV (see Algorithm 4) is applied. Therefore, CPV enables more efficient merging while ensuring safety.

5.5 Simulation Results

In this section we compare the performance of distance-based, velocity-based, and platoon velocity-based merging algorithms (referred to as D, V, and PV in the figures) against priority-based algorithm (referred to as R) in a variety of simulation settings.

5.5.1 Results of the Single-Lane Scenario

All the merging strategies use the same simulator configuration but differ in terms of (a) the type of information they use in making merging decisions, and (b) the position at which they make the decision. We compare the algorithms in terms of delay, the total number of cars, flow, and the average velocity.

Every merging strategy can easily cope with a light traffic. However, heavy traffic will show quickly which strategy is efficient and which is not. We evaluate the
Algorithm 5: CPV – Compute whether or not cars on the main road to change lanes

Input: \( r \): the ramp car; \( o_b, o, o_f \): the outer lane main road car list; \( i_b, i, i_f \): the inner lane main road car list

Output: \( o_iLaneChange \): whether \( o \) and \( i \) change lanes

5.1 if \( d_{ro} < d_{ri} \) then

5.2 /* Compute the required safety distance */
5.3 \( s_0 \leftarrow o.calcSafety(i_f) \)
5.4 \( s_i \leftarrow i.calcSafety(o_f) \)
5.5 \( s_{ob} \leftarrow o_b.calcSafety(i) \)
5.6 \( s_{ib} \leftarrow i_b.calcSafety(o) \)

5.7 if \( s_0 > d'_{fo} \) and \( s_i > d'_{if} \) and \( s_{ob} > d'_{bo} \) and \( s_{ib} > d'_{bi} \) then

5.8 /* Compute the acceleration according to the assumed front car */
5.9 \( a'_0 \leftarrow o.acc(i_f) \)
5.10 \( a'_i \leftarrow i.acc(o_f) \)
5.11 \( a'_{ob} \leftarrow o_b.acc(i) \)
5.12 \( a'_{ib} \leftarrow i_b.acc(o) \)

5.13 /* Compare the sum of accelerations */
5.14 if \( a'_0 + a'_i + a'_{ob} + a'_{ib} > a_0 + a_i + a_{ob} + a_{ib} \) then

5.15 \( o_iLaneChange \leftarrow TRUE \) /* \( o \) and \( i \) change lanes */
5.16 else

5.17 \( o_iLaneChange \leftarrow FALSE \)
5.18 else

5.19 \( o_iLaneChange \leftarrow FALSE \)

performance in the closed system without cars leaving the road to investigate the key parameters. We conduct a pre-study to identify these four parameters: the initial density of the main road, the rate of incoming ramp cars, the distance from the decision point to the merging point, and the ramp length. We vary only one parameter in each simulation run and keep the others constant. Furthermore, we identify the initial settings that impact the performance of merging algorithms. Those initial settings are three different settings on the main road, which are light, medium, and heavy traffic and two different settings on the ramp (see Table 5.3). To unify the unit of two traffic streams on the main road and the ramp, we transform the incoming rate of ramp cars (6 and 12 cars per minute, respectively for each setting) to equivalent values shown in
Table 5.3. For simplicity the incoming rate of ramp cars is constant in experiments 1 to 5. The maximal incoming rate is set to be 12 cars per minute because the length of the ramp is limited to 400 meters. In experiment 6 and 7, the rate of incoming ramp cars follows a Poisson distribution to better reflect real traffic situations. The length of the main road is 10 kilometers; the ramp is 400 meters in all experiments except in experiment 5 when we test the impact of ramp length, we reduce it to 200 meters.

Table 5.3: Experiment settings for main road with a single lane

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Main road $(\text{cars/km})$</th>
<th>Ramp $(\text{cars/km})$</th>
<th>Distribution</th>
<th>Merging area length $(m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>7.2</td>
<td>Uniform</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7.2</td>
<td>Uniform</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>7.2</td>
<td>Uniform</td>
<td>400</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>3.6</td>
<td>Uniform</td>
<td>400</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>3.6</td>
<td>Uniform</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>3.6</td>
<td>Poisson</td>
<td>400</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>7.2</td>
<td>Poisson</td>
<td>400</td>
</tr>
</tbody>
</table>

Note that each strategy takes a different time to merge the ramp cars to the main road. This leads to different total simulation times in our experiments.

Impact of the initial density on the main road

We investigate the impact of the initial density of the main road evaluating four different merging algorithms. In each experiment, we start with 50 cars on the main road to simulate light traffic conditions, and then with 100 cars to simulate medium traffic, and then with 150 cars for heavy traffic. We keep the incoming rate of ramp cars constant at a high value (12 cars per minute) as a high incoming rate is a major challenge for any merging algorithm. We expect that the smaller the initial density, the easier it will be for merging algorithms to perform well in terms of delay, throughput, and average velocity. As we expect, Figure 5.4, 5.5, and 5.6 show that for light traffic load (i.e., below 150 cars on the main road), all the four algorithms perform similarly in terms of delay. As the traffic load keeps increasing the delay of priority-based algorithm becomes much higher than the other three algorithms. For example, Figure 5.5
Figure 5.4: Light initial density of the main road
Figure 5.5: Medium initial density of the main road
Figure 5.6: Heavy initial density of the main road
shows that from 220 cars on the main road the delay for priority-based algorithm R is very large (ramp cars stop), which means the load of main road never reaches 220 cars in R. In contrast, at the same point the delay is 10 minutes for PV and 20 minutes for D and V. Therefore, PV outperforms other algorithms dramatically. In Figure 5.4, for 130 cars on the main road, the delay for all algorithms is around 7 minutes. There are 80 ramp cars that merge into the main road (since 50 cars initially). Figure 5.5 shows that merging 80 ramp cars has a delay of 7 minutes for our three algorithms. This means that our algorithms have a better performance because they have the same delay irrespective of the initial number of cars on the main road, whereas for the priority-based algorithm, the delay for merging the same number of ramp cars increases as the initial density on the main road increases.

**Impact of the incoming rate of cars on the ramp**

We test all the algorithms for a low incoming rate of 6 cars per minute on the ramp with medium initial density on the main road. We choose the medium initial density rather than the light initial density because the incoming rate is already low and if the initial density on the main road is also low then it takes a long time to see which strategy fails. Comparing Figure 5.7 and Figure 5.5 we can see that our algorithms perform better for a higher rate of incoming ramp cars in terms of throughput compared with the priority-based algorithm. When the incoming rate of ramp cars decreases, there is less load increase (i.e., the maximum number of cars can merge into the main road) in the priority-based algorithm: decreased from 110 to 100 when the incoming rate of ramp cars decreases half. At first sight, it is surprising that lower capacity in the main road is achieved by a lower incoming rate of ramp cars since we expect that lower incoming rate increases the traffic stability, which leads to a higher capacity. However, when cars are more evenly distributed and with higher velocity, which means larger safety gap, consequently it is more difficult to get in. This limitation is a side effect. It is a consequence of our set-up, in which the main road is a closed loop with no cars leaving the main road.
Figure 5.7: Medium initial density of the main road with low incoming rate of the ramp
Impact of the decision point

In this experiment we analyze the effect of the distance between the decision point and the merging point for the platoon-velocity-based algorithm. Figure 5.8 shows the delay for four different settings of the decision point, ranging from 50 meters to 400 meters away from the merging point. There are 100 cars initially on the main road and 12 ramp cars arrive per minute. We expect that there is an optimal distance between the decision point and the merging point. In general, the benefit will increase as the distance increases and has a maximum value after which it decreases. As we increase the distance between the decision point and the merging point, there is little benefit for making a merging decision at a position far away, since the traffic conditions may vary, especially when the incoming rate of ramp cars is high. In our scenario, for a distance of 400 meters PV performs worse than other shorter distances as shown in Figure 5.8.

Impact of the ramp length

The capacity is influenced by the length of the ramp significantly in the priority-based algorithm R. Only 80 cars can get into the main road when the ramp is half in length (Figure 5.9); 100 cars can get into the main road with the settings specified in Figure 5.7. The decrease is 20%. In contrast, there is little impact of a shorter ramp for
Figure 5.9: The performance of merging algorithms with a short ramp
the proactive merging algorithms. It is mainly because in our proactive merging algorithms cars adjust their velocities before they arrive at the merging section. However, in algorithm R ramp cars adjust their velocities after they arrive at the merging section. Therefore, for the priority-based algorithm a shorter ramp is inadequate for some cars to adjust their velocities properly to merge in the main road. Whereas, unlike the priority-based algorithm, the length of the ramp is less significant for our proactive algorithms. This fact is especially beneficial in highly populated urban areas where we do not have enough space for long ramps.

**Impact of the Poisson distribution**

In this experiment, we set the incoming rate of ramp cars according to a Poisson distribution. This reflects a more realistic traffic situation since we typically find that cars arrive in a bulk fashion with low and high densities, e.g., during rush hours. Figure 5.10 and Figure 5.11 show the delay of the load increase. As we expect, the delay increases for all of the four algorithms since the incoming rate is not constant, compared with Figure 5.7 and Figure 5.5. PV performs better than the other three algorithms, especially when the incoming number of ramp cars is large: 12 cars per minute on average. Meanwhile, the fluctuation of incoming number of cars is also large, which is shown in Figure 5.12.

![Figure 5.10: Incoming number of cars following the Poisson distribution with $\lambda = 6$](image-url)
5.5 Simulation Results

Figure 5.11: Incoming number of cars following the Poisson distribution with $\lambda = 12$

Figure 5.12: Incoming number of cars per minute with $\lambda = 6$ and $\lambda = 12$

5.5.2 Results of the Multi-Lane Scenario

In this section, we compare the performance of proactive merging algorithms for a 2-lane main road against the algorithm R in a variety of simulation settings.
We have two initial settings on the main road and on the ramp to simulate light and heavy traffic (see Table 5.4). Heavy ramp traffic corresponds to 12 cars arriving per minute (i.e., 7.2 cars per km) and light traffic to 9 cars arriving per minute (i.e., 5.4 cars per km). Our previous work shows that heavy traffic is ideal to identify a better performing strategy. The incoming rate of ramp cars is constant in experiments 1 to 3. In experiments 4 to 6*, the ramp cars arrive according to Poisson distribution. The length of the main road is 10 kilometers. The length of the ramp is 500 meters including a 100 meter merging area (i.e., the segment \(OE\) in Figure 5.1, simply referred to as ramp length in this section) in all experiments except in Subsection 5.5.2, where we evaluate the impact of the merging area and double it to 200 meters. Please note, the ramp length is set differently as for the single lane scenario because the strategy R requires a long ramp to perform. Now we focus on comparing proactive merging algorithms which the ramp length does not play an important role as for R. Rather a longer ramp only postpone the traffic breakdown at the ramp. Similar properties are observed compared to a shorter ramp. In experiments 1 to 5 the exit of the main road is closed to saturate the main road quickly and to study the impact of a merging strategy on the overall traffic. Experiment 6* is similar to 5 except for an open exit, which allows randomly selected cars to leave the main road. The aim is to identify which strategy is better at preventing traffic jams in more realistic conditions.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Main road (cars/km)</th>
<th>Ramp (cars/km)</th>
<th>Distribution</th>
<th>Merging area length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>7.2</td>
<td>Uniform</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7.2</td>
<td>Uniform</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>7.2</td>
<td>Uniform</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>7.2</td>
<td>Poisson</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>5.4</td>
<td>Poisson</td>
<td>100</td>
</tr>
<tr>
<td>6*</td>
<td>20</td>
<td>5.4</td>
<td>Poisson</td>
<td>100</td>
</tr>
</tbody>
</table>
Impact of the ramp length

To explore the impact of the ramp length, we test the three algorithms using a short ramp (100 meters) and a long ramp (200 meters). Figure 5.13 and Figure 5.14 show that a longer ramp can mitigate the impact of the less performing strategy R. However, once a high traffic flow on the ramp is established, the strategy R is unable to merge further cars. As expected, the strategy CPV has a smaller overall average velocity as it is the only way to merge more ramp cars. The dashed line in Figure 5.13 and Figure 5.14 illustrates the maximal flow in equilibrium condition and we observe that CPV approaches the theoretical maximum. The length of the ramp significantly impacts the road capacity achieved by algorithm R. Only 100 cars can merge into the main road for a short ramp (Figure 5.13) and 200 cars for a longer ramp with the settings specified in Figure 5.14. The decrease is 100%. For strategy R ramp cars can only adjust their velocity after they arrive at the merging section, and the longer the ramp the easier it is to merge. In contrast, there is no significant impact of a shorter ramp for PV and CPV. In our proactive merging algorithms cars adjust their velocity before they arrive at the merging section, reducing the importance of the ramp length. CPV performs considerably better than PV and R for a long ramp (Figure 5.14). For short ramps CPV benefits from pre-lane-changes over PV (Figure 5.13). This behavior is especially beneficial in highly populated urban areas that do not have enough space for long ramps.

Impact of the initial density on the main road

We investigate the impact of the initial density of the main road. In each experiment, we start with 100 cars on the main road to simulate light traffic conditions, and with 200 cars to simulate heavy traffic. We keep the incoming rate of ramp cars constant at a high value (12 cars per minute) as a high incoming rate is a major challenge for any merging algorithm. We expect that the smaller the initial density, the easier it will be for merging algorithms to perform well in terms of delay, total number of cars, flow, and average velocity. As expected Figure 5.13 and 5.15 show that for light traffic load
Figure 5.13: The performance of merging algorithms with a short ramp
Figure 5.14: The performance of merging algorithms with a long ramp
(i.e., below 150 cars on the main road), all the algorithms perform similarly in terms of delay. As the traffic load keeps increasing the delay of the algorithm R becomes much higher than PV and CPV. For example, Figure 5.15 shows that R is not able to achieve a capacity of 220 cars on the main road (the delay is infinite). In contrast, the total number of cars reaches 360 in PV and 450 in CPV. In summary, CPV outperforms other algorithms by a large margin. In Figure 5.13 for 170 cars on the main road, the delay for all algorithms is around 7 minutes. 70 ramp cars merged into the main road (starting from 100 cars). Figure 5.15 shows that merging 70 ramp cars also has a delay of only 7 minutes for PV and CPV under a high initial main road traffic. This means that PV and CPV perform very well: they incur the same delay independent of the initial traffic load on the main road, whereas R cannot merge the same number of ramp cars if the initial density increases.

Impact of the incoming cars on the ramp

The incoming rate of ramp cars is first constant, then it follows a Poisson distribution. The Poisson distribution reflects more realistic traffic situations, since cars often arrive in a bulk fashion of low and high densities, for example during rush hours or caused by traffic lights. The mean incoming rate is again set to 12 and 9 cars per minute. We compare all algorithms for a high constant incoming rate of 12 cars per minute (shown in Figure 5.15), and in two experiments for a rate that follows the Poisson distribution with means of 12 (shown in Figure 5.16) and 9 ramp cars per minute (shown in Figure 5.17). The maximum arrival rate is 26 cars per minute in the experiment (Figure 5.16, a big challenge for a merging strategy). As we expect, PV and CPV perform better in terms of delay, total number of cars and flow compared to R, although the performance of all the strategies is less efficient for a Poisson distribution than for a constant rate of incoming cars.
Figure 5.15: Initial heavy density of the main road
Figure 5.16: Incoming cars following the Poisson distribution with $\lambda = 12$
5.5 Simulation Results

Figure 5.17: Incoming cars following the Poisson distribution with $\lambda = 9$
Open Exit

From the previous experiments, we learned the achievable road capacity, the delay of a merging strategy, and candidates for good merging algorithms. In this experiment, we open the closed system to investigate more realistic traffic conditions. To identify a good strategy, we start with very heavy load on the main road (200 cars initially) to test which strategy fails first. Each car has a 30% chance to exit the main road. We set this high probability to relax the traffic load. The incoming traffic is not constant, at peak times 15 cars per minute arrive at the ramp. Figure 5.18 shows significant advantages of PV and CPV over R. R fails much earlier than PV, and CPV does not fail at any time. CPV achieves the largest total number of cars, the highest flow and capacity with only a slightly smaller average velocity. Compared to R, the total number of cars increases by 60% and the capacity increases by 35% in CPV, which shows that CPV is the most efficient merging strategy.

In summary, this chapter shows that proactive merging algorithms have the following advantages over the priority-based algorithm R:

- First, the delay in proactive merging algorithms decreases by up to one third before the closed system is saturated.
- Second, the capacity of the main road increases by up to two thirds (Figure 5.15).
- Third, a proactive merging algorithm can increase the traffic flow. Since the traffic flow is the product of the average velocity and the density, and since proactive algorithms merge cars in less time, the larger number of cars can lead to a smaller average velocity to ensure the minimum safety distance.

A merging algorithm must consider several factors, in particular the delay, traffic flow, and average velocity. The traffic flow increases and peaks once the main road is saturated. Then, it decreases as the average velocity decreases due to perturbations resulting from varying velocities. A high traffic flow requires to balance the average velocity and the number of cars that merge onto the main road.
Figure 5.18: Cars exit in 0.3 possibility and income following the Poisson distribution with $\lambda = 9$
Overall, CPV leads to remarkable results in terms of delay and traffic flow. It assigns a slot for a ramp car among a group of main road cars in the segment between the decision point and the merging point, and creates a larger gap for the ramp car as main road cars change lanes prior to the merging area. In summary, CPV optimizes the delay and flow at the expense of the average velocity of the main road traffic. On the other hand, the lower average velocity enables CPV to significantly increase the total number of cars on the main road.

5.6 Conclusion

We have introduced a range of proactive merging algorithms to improve traffic flow. Proactive merging algorithms separate the decision point, where a car computes the optimal merging point, and the actual merging point to adjust the velocity of a merging car in advance. Our proactive merging algorithms compute for a ramp car the optimal slot in a group of main road cars based on their velocities. We compared the algorithms PV and CPV against the reference priority-based merging algorithm R for multiple lanes under a large variety of realistic experimental settings. The experiments show that CPV, where main road cars change lanes prior to the merging area to create a larger gap, significantly outperforms the algorithm R in terms of an increased traffic flow by a factor of two and a decreased delay by 30%.
Chapter 6
Robustness for Merging Algorithms

A generation of sensor-enabled cars will be emerging that can locally exchange sensed information about traffic and use this information to adapt their behavior much earlier than regular cars. Although the traffic merging algorithms discussed in Chapter 5 show remarkable improvement, they are based on the assumption that sensors provide accurate measurements. However, sensor measurements are generally not completely exact. The accuracy level of sensors poses a major challenge for merging algorithms, because inaccuracies can potentially lead to unsafe merging behaviors. In this chapter, we investigate how the accuracy of sensors impacts merging algorithms, and design two novel merging algorithms that are robust to tolerate sensor errors. We adapt the concepts from time geography to achieve a high tolerance of imprecise sensor information to guarantee safe merging, and a smoother merging process to ensure an increased merging capacity and less impact on the subsequent traffic flow. Experimental results show that our time geography-based merging algorithm is able to guarantee safe merging while tolerating two to four times more imprecise positioning information, and can double the road capacity and increase the traffic flow by 25%.

6.1 Introduction

The impact of imprecise information (errors in sensor measurements) has been overlooked by most of the previous work on merging models, or merging algorithms [21, 64, 65, 119, 125]. To date, there is no systematic study for analyzing how the accuracy of sensor measurements impacts the performance of a merging algorithm. Our
work is the first approach that investigates different algorithms under inaccurate sensor measurements in terms of distance and explicitly develops robust algorithms that are tailored to minimize the adverse effect of positioning inaccuracies.

We propose a decentralized robust merging algorithm that combines ideas from time geography [56] with the concept of a safety zone to compute the “smoothest” acceleration and deceleration of cars involved in a merging situation. Our approach uses concepts from time geography, which provides a time-space bound in the form of time-space prisms. It was suggested earlier in 1908 by Minkowski [96] using light cones in the context of physics. We use the prisms/cones to compute all possible future positions of a car based on spatio-temporal constraints. A safety zone defines the minimum safety distance between two subsequent cars, which depends on a car’s velocity and the velocity difference of its front car. Safe merging is only ensured if no violations of a car’s safety zone occurs. The intersection of the space-time bounds and the safety zones of cars on multiple lanes allows us to find those cars that are likely to interact in a merging situation. We also propose a merging algorithm based on local gap optima to restrict merging impact on the subsequent traffic of the main road.

6.2 Time Geography

Time geography, introduced by Hägerstrand [56], is a conceptual framework for understanding individual activities under different constraints in a spatio-temporal context. This concept is related to well-known Minkowski’s visualization of space time in terms of light cones, which define the boundary of all possible past and future positions [96, 120]. There are two fundamental concepts in time geography, namely, space-time path and space-time prism. The space-time path describes the movement of an individual in space and time. Figure 6.1(a) illustrates a space-time path in a two-dimensional space. The slope of the path indicates velocity of an individual. A vertical line represents no movement for a given time interval.

An extension of the space-time path is the space-time prism, which encloses all possible positions a person can reach by taking time constraints into account. Figure 6.1(b)
shows a simple prism. Assume that a person at position $A$ needs to arrive at $B$ during the time interval $(t_1, t_2)$. The prism is determined by the intersection of two cones: a lower cone originates from $(t_1, A)$ orienting forward in time, and an upper cone originates from $(t_2, B)$ and orienting backward in time. Projecting the space-time prism to a geographic space delimits the potential path area. A person can physically interact with another person only if there is an intersection of their prisms.

Time geography has been applied in the area of Location-Based Services (LBS) and Geographic Information Systems (GIS) [147]. In this chapter, we apply time geography for developing a merging algorithm, which computes smooth acceleration
and/or deceleration for all involved cars to minimize the impact of merging on the overall traffic.

### 6.3 Robustness of Merging Algorithms

In our context, “robust” means a merging algorithm can cope well with variations of traffic conditions, driver behaviors, and imprecise sensor measurements. In this section, we first investigate the crucial factors that impact the robustness of a merging algorithm. Then we specify the criteria to develop robust merging algorithms. Finally, we propose new robust merging algorithms.

#### 6.3.1 Factors of Robustness

**Level of accuracy**

In our previous chapter we assumed that the spatial information obtained by sensor-enabled cars was accurate and precise; however, according to [48] automotive sensors typically have a total error of up to 3% over their entire measured range. Therefore, we aim to develop merging algorithms that can cope with imperfect information for sensor-enabled cars, because inaccuracies can potentially lead to unsafe merging behaviors.

There are two well-known approaches to increase the accuracy of sensor information. They are:

- **Sensor fusion.** Li and Leung [88] propose an unscented Kalman filter to fuse different sensors including radar, GPS, inertial navigation system, and camera in platooning of automated vehicles. Rezaei and Sengupta [122] propose an extended Kalman filter to integrate GPS with vehicle sensors to achieve required accuracy for cooperative collision warnings.

- **Communication.** To increase the reliability of sensed data, neighboring cars can share their sensor measurements by communicating with others. For example,
distance measures between subsequent cars could be exchanged for a higher accuracy.

Human factors also influence the robustness of merging algorithms. For example, some drivers do not follow the system’s suggestions and merge at non-optimal slots. In addition, merging algorithms might get influenced if some drivers exceed the speed limit.

In this chapter, we focus on the first factor by exploring what level of accuracy is sufficient for safe merging and developing robust merging algorithms.

### 6.3.2 Criteria of Robustness

Our criteria to examine a robust merging algorithm are:

- **C1 safest merge**: minimize violation of minimum safety distance.
- **C2 most efficient merge**: maximize traffic flow.
- **C3 quickest merge**: minimize merging time.
- **C4 smoothest merge**: minimize acceleration or deceleration.

C1 implies lower velocity and larger gaps; C2 implies higher velocity coupled with tighter gaps; C3 implies higher velocity; and C4 implies an approximate equilibrium system.

### 6.3.3 Design of Robust Merging Algorithms

We describe two robust merging algorithms in this section. Both algorithms aim to increase the tolerance for inaccurate position information.

**Locally gap-optimal algorithm (LO)**

In LO, distances of a group of main road cars are redistributed to accommodate a ramp car to merge. Distances of this group are equal after the ramp car merges. The
impact of inaccuracy level reduces when the distances between cars increase. Another advantage of LO is that the merging impact on the main road is restricted to a certain number of following cars by deciding the size of this group. The size of the group $N_{LO}$ can be 3, 4 or more cars depending on the average distance between two cars.

Another difference between PV, CPV and LO is the criteria to assign a gap to a ramp car. In PV and CPV, the gap is strictly based on the potential arrival time estimated at the time of making decision, assuming that all cars travel at constant velocity. In LO, physical possibility is checked before main road cars make space for a ramp car.

Figure 6.2 gives an example of how LO works. It shows a group of four neighboring cars on the main road ($N_{LO} = 4$). Three gaps among those cars are redistributed into four equal gaps to accommodate the ramp car $r$. The back car $b$ will get position $A$, the front car $f$ will get position $B$, and the ramp car $r$ will get position $C$.

A more detailed procedure of LO is as following:

1. assigns a gap: ramp car $r$ initiates communication and gets a group of cars on the main road within the communication range. Computes the arrival time to the merging point $M$ of $r$ and main road cars based on their current velocity. Based on the arrival sequence of main road cars select the ramp car’s front car ($f$), back car ($b$), the front car of $f$ ($ff$) and the back car of $b$ ($bb$). This procedure is detailed in Algorithm 6, which computes the list of potential main road candidates which will be influenced by merging, i.e., $ff, f, b, bb$ in Figure 6.2. The size of the output list is up to the group size $N_{LO}$. If the output list is empty, then $r$ merges.
Algorithm 6: Compute main road candidates

Input: \( r \): the ramp cars
\( MainList \): the list of main road cars within \( r \)’s communication range and before the merging point \( M \)

Output: \( OutList \): the list of main road candidates

6.1 On initialization: \( cond \leftarrow TRUE \) /* \( cond \): a flag indicates whether to get main road car \( m \) from \( MainList \) */

6.2 if \( MainList == \emptyset \) then
   /* There is no car on the main road so return null */
   \( OutList \leftarrow \emptyset \)
else
   do
      if \( cond == TRUE \) then
         /* Get car \( m \) from the main road car list \( MainList \) */
         \( m \leftarrow MainList.pop() \)
         \( t' \leftarrow m.\text{position}/m.\text{velocity} \) /* The time when \( m \) arrives at \( M \) */
         \( cond \leftarrow FALSE \)
      \( t_r \leftarrow r.\text{position}/r.\text{velocity} \) /* The time when \( r \) arrives at \( M \) */
      if \( t' > t_r \) then
         \( b \leftarrow m \) /* \( r \) arrives at \( M \) earlier than \( m \) */
         \( OutList.\text{Add}(b,'b') \) /* add \( b \) to \( OutList \) identified by \( 'b' \) */
         \( bb \leftarrow b.\text{nextCarOnLane()} \) /* \( bb \) is the car behind \( b \) */
         \( OutList.\text{Add}(bb,'bb') \)
      else
         \( f \leftarrow m \) /* \( r \) arrives later or at the same time at \( M \) */
         \( ff \leftarrow f.\text{prevCarOnLane()} \) /* \( ff \) is the car in front of \( f \) */
         \( OutList.\text{Add}(ff,'ff') \)
         \( cond \leftarrow TRUE \) /* Set \( cond \) to true so that \( m \) is popped in the next iteration */
      end
   while \( MainList \neq \emptyset \) and \( cond == TRUE \);

2. checks the gap length: if the gap between \( b \) and \( f \) is sufficiently large (e.g. bigger than 100m), then \( r \) merges. Otherwise, continues to the next step.

3. checks physical possibility: assume after time \( t' \): (1) the distance between each car is \( optGap \), \( optGap = (|bb,ff| - (N_{LO} - 1) \ast \text{length of car})/N_{LO}; \) and (2) \( b, r \) and \( f \) velocity is \( optV \), which is the maximum safe velocity constrained by \( optGap \). Based on these statements, we have:

\[ v'_r = v'_b = v'_f = optV \leq v_{max} \]

\[ s_r = d_r \]
s_f = d_f + optGap
s_b = d_b - optGap

These equations should be satisfied under the following conditions: (1) acceleration and deceleration of \( b, r, f \) are within the limitation; and (2) the distance between \( bb \) and \( b, f \) and \( ff \) do not violate the required safety distance. If so, \( r \) merges. Otherwise, \( r \) waits until the traffic situation changes (a new car comes into \( r \)'s communication range or \( f \) leaves), goes to step 1.

Figure 6.3 depicts how to compute the acceleration of involved cars on the main road. The dashed lines indicate the bounds of acceleration. The line with arrow is the velocity with a constant acceleration from the initial velocity \( v_i \) to \( optV \) at time \( t' \). The area of the shaded region is the distance traveled during the time period. If a car cannot reach its optimal position with a constant acceleration, then it applies a higher acceleration if more distance required to travel; otherwise, a lower acceleration is applied.

![Figure 6.3: Velocity-time relation in LO](image)

**Time geography-based algorithm (TG)**

The design of our TG-based algorithm relies on the space-time prisms/cones, which we introduced in Section 6.2.
6.3 Robustness of Merging Algorithms

Figure 6.4 shows the basic idea of TG. Since we consider the road as one-dimensional, we use sectors to represent all potential positions for each car both on the ramp and on the main road. $t_L$ is the latest arrival time (at the minimum velocity $v_{\text{min}}$). $t_E$ is the earliest arrival time (at the maximum velocity $v_{\text{max}}$). Since the ramp car’s velocity is between $v_{\text{min}}$ and $v_{\text{max}}$, the ramp car is inside a sector. The curve with arrow indicates the acceleration of the car. The gray region in Figure 6.4(a) is $r$’s safety zone, which means when $r$ arrives at $M$, there is no intersection of main road cars on the target lane. The safety zone is composed of the ramp car’s minimum safety distance to the front car and the back car. The safety distance depends on $r$’s velocity and the velocity difference between two following cars. The shape of the safety zone may not be symmetrical.

When a ramp car arrives at the decision point, it checks whether there will be an intersection of its safety zone with sectors of main road cars at $M$. If there is any intersection, and if the main road consists of multiple lanes, we use multiple sectors for the main road cars and see if they can switch to a different lane. If after a lane switch on the main road intersection still exists, TG computes the acceleration for the ramp car and the involved main road cars.

The major steps of TG are as following:

1. finds potentially interacting cars by checking for overlaps of space-time sectors;
2. checks geometric possibility;
3. assigns a gap;
4. checks physical possibility: acceleration constraints; and
5. computes acceleration for merging.

We show how TG calculates acceleration of cars in Algorithm 7.

After initialization, TG computes the actual arrival time $t'$ of a ramp car $r$ in the time boundary of earliest arrival time $t_E$ and latest arrival time $t_L$ (line 7.1 – 7.3). If the shape of the safety zone could be uniquely determined then we could compute the
optimum time $t'$. However, the safety zone may not be uniquely determined due to the dynamic changes in velocity and velocity difference between two cars. Therefore, we use approximation technique [98] to compute $t'$. During the time period $(t_E, t_L)$, the loop (line 7.4 – 7.14) increases the time variable $t'$ by a fixed timestep $\Delta t$ (line 7.11). The purpose of $\Delta t$ is to find an arrival time of $r$ so that its front car is outside of $r$’s safety zone. Notice that a larger $\Delta t$ makes the algorithm less computational intensive, but a smaller $\Delta t$ makes the acceleration smoother. We can use an adaptive timestep
Algorithm 7: Compute accelerations

**Input:** r: a ramp car
f: r’s front car on the main road based on the output of Algorithm 4
b: r’s back car on the main road

**Output:** ar, af, ab: acceleration of r, f and b

On initialization:

1. \( t_E \leftarrow s_r/v_{\text{max}} \); /* Earliest time of r to arrive at the merging point */
2. \( t_L \leftarrow s_r/v_{\text{min}} \); /* Latest time of r to arrive at the merging point */
3. \( t' \leftarrow t_E \); /* t' is the actual time of r to arrive at the merging point; t' \in (t_E, t_L) */

while \( t' < t_L \) do

5. \( v'_f \leftarrow v_{\text{max}} /\) /* Velocity of f at time t' */
6. \( a_f \leftarrow (v'_f - v_f)/t' /\) /* Acceleration of f */
7. \( s_f \leftarrow v_ft' + \frac{1}{2}a_ft'^2 /\) /* Distance that f travels */
8. \( \text{gap}_f \leftarrow s_f - |FM| /\) /* The gap between f and r at time t' */
9. \( s^*_f \leftarrow s'(v'_r, v'_f) /\) /* Safety distance required */
10. if \( \text{gap}_f < s^*_f \) or \( a_f > a_{\text{max}} \) then

    11. \( t' \leftarrow t' + \Delta t /\) /* Increase t' by \( \Delta t \) */
    12. else

    13. \( a_f \leftarrow (v'_f - v_f)/t' /\) /* Acceleration of f */
    14. \( a_r \leftarrow (v'_r - v_r)/t' /\) /* Acceleration of r */

15. \( s_b \leftarrow v_bl' \)
16. \( \text{gap}_b \leftarrow |BM| - s_b \)
17. \( a_b \leftarrow 0 \)
18. \( s^*_b \leftarrow s'(v_b, v'_b) \)
19. while \( \text{gap}_b < s^*_b \) do

20. /* Violate the safety distance */

21. \( a_b \leftarrow a_b - \Delta a /\) /* Acceleration of b */
22. \( v'_b \leftarrow v_b + a_bl' /\) /* Velocity of b at time t' */
23. \( s_b \leftarrow v_bl' + \frac{1}{2}a_bl'^2 /\) /* Distance that b travels */
24. \( \text{gap}_b \leftarrow |BM| - s_b /\) /* The gap between b and r at time t' */

that is proportional to the density of the traffic. Then TG checks the gap between the ramp car and its back car (line 7.15 – 7.18) and computes acceleration of the back car (line 7.19 – 7.24) to ensure no violation of safety distance.
6.4 Simulation Results

In this section we compare the performance of R, PV, CPV, LO and TG in a variety of simulation settings. We summarize the studied merging algorithms in Table 6.1.

Table 6.1: Traffic merging algorithms overview

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Regular merging algorithm: the main road car has priority</td>
</tr>
<tr>
<td>PV</td>
<td>Platoon velocity-based merging algorithm: the first arriving car has priority</td>
</tr>
<tr>
<td>CPV</td>
<td>Cascading PV: PV and main road cars change lanes to accommodate ramp cars</td>
</tr>
<tr>
<td>LO</td>
<td>Locally gap-optimal merging algorithm: redistribute gaps on the main road</td>
</tr>
<tr>
<td>TG</td>
<td>Time geography-based merging algorithm: compute the smoothest acceleration for involved cars</td>
</tr>
</tbody>
</table>

The experiment settings are given in Table 6.2. The length of the main road is 10 km. The length of the ramp is 500 m including a 100 m merging area in all experiments except when we test strategy R, which requires a longer ramp (set to 400 m) to merge.

Table 6.2: Experiment settings

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Main road (cars/km)</th>
<th>Ramp (cars/km)</th>
<th>Distribution</th>
<th>Sensor Accuracy (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>7.2</td>
<td>Uniform</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>7.2</td>
<td>Uniform</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>7.2</td>
<td>Poisson</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>7.2</td>
<td>Uniform</td>
<td>2; 3; 4</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>3.6</td>
<td>Uniform</td>
<td>2; 3; 4</td>
</tr>
</tbody>
</table>

6.4.1 Impact of Accuracy Level on Safety

Safety is the basic requirement for any traffic merging strategy. We assume that the errors in the position measurements follow a truncated normal distribution, which means errors are within a bounded range [123] (see Section 4.3.2 for more details).
To demonstrate the impact of positioning accuracy on safety, in Figure 6.5 and Figure 6.6 we use two bars for each algorithm. The white bar represents the total number of cars before the first violation of safety distance occurs and the underlying black bar represents the total number of cars on the main road until merging becomes infeasible. When there is no violation, we will only show the white bar. The number next to each bar summarizes the number of violations until merging becomes impossible. Note that this number is not the difference between the white and the black bar.

In experiment 1, we start with a low initial density (10 cars/km) on the main road, a constant incoming rate of ramp cars (12 cars/min) and a high accuracy level (1 m). Figure 6.5 shows that R requires a higher accuracy level than other strategies. 12 violations of safety distance occur for positioning information accuracy within 1 m (Figure 6.5(a)). We increase the initial traffic on the main road in experiment 2 and observe similar results.

In experiment 3, the settings are the same as in experiment 2 except that the incoming ramp cars follow a Poisson distribution with a mean arrival rate \( \lambda = 12 \). In this bursty traffic setting, violations immediately occur in R with 1 m accuracy as shown in Figure 6.5(b).

Then, we vary the range of positioning accuracy from 2–4 m. We omit showing violations in R when the inaccuracies are above 1 m because R can not tolerate inaccuracy more than 1 m. Figure 6.6(b) shows the number of violations in the case of 4 m accuracy. When accuracy is within 3 m, there is no violation for all strategies (thus the figures for 2 m and 3 m are omitted).

There are two main reasons why strategy R tolerates lower imprecise sensor information. First, the main road cars do not cooperate with ramp cars (i.e., create larger gaps). This increases the difficulty for a ramp car to merge. Second, a ramp car starts searching for a gap and adjusting its velocity accordingly later than other proactive algorithms. A ramp car has to slow down if it has not got a chance to merge as it approaches the end of a ramp. Thus, the difference of velocity between a ramp car and main road cars becomes larger. The chance for a safe merging becomes lower and the ramp car stops at the end of a ramp.
In experiment 5, we start with high traffic flow (300 cars) on the main road. Figure 6.6(a) shows the violations in the case of 2 m accuracy. There is one violation in CPV when the total number of cars on the main road is over 400; there are 2 violations in PV when the number of cars is over 420. Thus, the accuracy level required is 1 m for PV and CPV. There is no violation in TG for inaccuracy up to 4 m.

Compared to PV and CPV, LO tolerates higher level of positioning inaccuracy. This
6.4 Simulation Results

![Bar chart](image1)

(a) High initial main road traffic; errors: 2 m

![Bar chart](image2)

(b) Medium initial main road traffic; errors: 4 m

Figure 6.6: Exp. 4–5: impact of errors on safety (We omitted the regular strategy R for more than 1 m inaccuracy)

is because in LO the gaps between main road cars are redistributed to accommodate a car merging from the ramp. The impact of accuracy becomes smaller when the average distance between cars is larger. For example, let us consider an average distance \(d\) of 60 m on the main road. Then after a ramp car merges, its distance from both of the front and back car is about 30 m. In LO this distance is about 45 m assuming four main
Table 6.3: The level of robustness and capacity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Level of robustness</th>
<th>Merge capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>N/A</td>
<td>300</td>
</tr>
<tr>
<td>PV</td>
<td>$\sim 1\ m$</td>
<td>420</td>
</tr>
<tr>
<td>CPV</td>
<td>$\sim 1\ m$</td>
<td>450</td>
</tr>
<tr>
<td>LO</td>
<td>$\sim 2\ m$</td>
<td>550</td>
</tr>
<tr>
<td>TG</td>
<td>$\sim 4\ m$</td>
<td>600</td>
</tr>
</tbody>
</table>

road cars involved in the gap-redistribution (i.e., $0.5d \rightarrow 0.75d$). From Figure 6.5 and Figure 6.6 we also observe that no violation occurs in TG under positioning accuracy ranging $1-4\ m$. TG minimizes the adverse effect of positioning errors because TG computes acceleration for involved cars to avoid violations of safety zone.

We summarize the results of impact on safety in Table 6.3. The level of robustness describes the maximum degree of positional inaccuracy an algorithm can cope with. The merge capacity is the maximum number of cars on the main road an algorithm can merge without collisions.

### 6.4.2 Impact of Accuracy Level on Efficiency

Now we study the impact of accuracy level on traffic flow, merging capability (the number of cars), average velocity and acceleration.

We showed how the theoretical traffic flow is computed in Section 3.3.2 and there is a reduction in practice by disturbances (e.g., lane changing or merging). We plot the theoretical traffic flow in the results for comparison of merging algorithms. Figure 6.7 shows the results of experiment 1. TG can accommodate 600 instead of 300 cars compared to strategy R. TG can handle double of the road demand. This is especially a benefit in busy times such as rush hours. When starting with 200 cars on the main road in experiment 2, the results are very similar.

In experiment 3, the incoming rate of ramp cars follows a Poisson distribution with a mean arrival rate of 12 cars per minute. The maximum arrival rate is 26 cars per minute in the experiment, which is a big challenge for any merging algorithm. As expected, compared to a constant rate of incoming cars, the performance of all
Figure 6.7: Low initial density of the main road, errors within 1 m.
Figure 6.8: Ramp cars following the Poisson distribution with $\lambda = 12$, errors within 1 m.
Figure 6.9: Heavy initial density of the main road, errors within 2 m.
merging algorithms degrades. However, TG and LO degrade least as they exhibit slightly lower traffic flow and less number of merged cars.

When the initial density of the main road is high (30 cars/km), no car is able to merge in strategy R. Thus, we only compare the performance of proactive merging algorithms. Figure 6.9 shows that TG outperforms other strategies especially when there is heavy traffic load on the main road. As more cars merge, the average distance between cars on the main road becomes smaller. To maintain safety distance the average velocity reduces. With the same cost of velocity reduction, TG accommodates more cars merging from the ramp. For example, in Figure 6.9(b) when the average velocity is 17 m/s, there are 512 cars on the main road achieved by TG, whereas only 435 cars by PV. This means TG merges 77 cars more than PV (212 cars merge in TG; 135 cars merge in PV). That is nearly 60% increase in merging capability.

We study the effect on overall braking of main road cars. The results are shown in Figure 6.7–6.9(c) TG minimizes the deceleration: mostly two times (sometimes even three times) better compared to PV or CPV. This is important as there is less impact on the subsequent traffic. R has the least impact of merging on the main road with respect to the deceleration of main road cars. This is simply due to no cooperation between different traffic streams. The adverse effect is that the number of merged cars is significantly less than other proactive merging algorithms. Therefore, TG outperforms other proactive merging algorithms as it achieves more merged cars while maintaining less impact on the main road traffic. We observe that in PV and CPV, deceleration becomes 0 (no need to apply brakes) in most cases when the number of cars on the main road is around 400–450. The reason is that they have reached the limit of merging cars from the ramp.

In summary, TG has the following advantages compared to other merging strategies:

- tolerance of positioning errors (no violation of minimum safety distance): TG: 4 m (Figure 6.6(b)); LO: 3–4 m (Figure 6.6); PV and CPV: 2 m (Figure 6.6(a)); and R: less than 1 m (Figure 6.5).
6.5 Higher Level of Realism

In this chapter and the previous chapter, we have assumed that 1) all the cars have same acceleration and deceleration characteristics; 2) all the cars are sensor-enabled; and 3) human drivers follow traffic rules. However, to achieve a higher level of realism, in this section we relax those assumptions and conduct further experiments. In particular, we first consider the co-existence of sensor equipped and non-equipped cars; vehicle heterogeneity; and some drivers’ unwillingness to follow traffic rules.

6.5.1 Sensor Coverage

It can be anticipated that future road networks will consist of sensor equipped and non-equipped cars. Therefore, we modify our previous assumptions to capture the fact that a certain percentage of cars may not be instrumented in practice. In this section, we investigate the effects of this scenario. The underlying merging logic is the following: a sensor-enabled car makes use of one of our proactive merging algorithms, whereas a non-equipped car has to use algorithm R because it has no access to sensed information. We expect that this hybrid strategy can still achieve significant improvement.

In order to exclude the influence of lane changes on the main road, we consider a main road with a single lane. We vary the percentage of non-equipped cars in
the range 10%–40% and make use of TG for sensor-enabled cars. Figure 6.10 shows the traffic flow and velocity for merging. We find that as the percentage of sensor non-equipped cars increases, the effectiveness of the proactive merging algorithm decreases. However, TG even with 40% non-equipped cars still outperforms R in terms of the traffic flow and the number of cars that can merge. The main reasons are: 1) there is no sensing information in advance for the ramp cars to adjust their velocity; and 2) there is no communication between non-equipped cars and merging is performed based on the human driver’s estimation of surrounding traffic condition. As a consequence, main road cars cannot cooperatively adjust velocity to create larger gaps for ramp cars to merge.

Figure 6.10: Sensor coverage
6.5 Higher Level of Realism

6.5.2 Heterogeneity of Vehicles

We have assumed that all cars have the same acceleration and deceleration characteristics. However, our simulation can be easily adapted to take different types of vehicles into account. For example, trucks take longer time to accelerate and decelerate. In addition, considering the length of a truck, a truck requires more space to merge. The effects of having heterogeneous vehicles on merging algorithms require further investigation. Figure 6.11 shows the impact of the heterogeneous traffic flow (20% trucks) on merging. The trucks are distributed randomly. As expected, the mixed traffic flow of cars and trucks is lower than the traffic flow composed of all cars. As trucks occupy more space than cars, the total number of vehicles is also smaller. The major difference for R and PV is that in spite of trucks, PV achieves more stable traffic flow than R. PV is able to achieve a higher traffic flow even with 20% trucks compared to homogeneous traffic with all cars applying R. The cars are faster than trucks and they queue behind trucks. This automatically leads to a formation of platoons and large gaps in front of trucks. PV exhibits the benefit for reducing the disturbance of the main road traffic by assigning a slot for a ramp car in advance.

6.5.3 Impact of Speeding

In practice some drivers exceed the speed limit in a free flowing condition or to overtake a slower vehicle. We conduct experiments to study the impact of such speeding behavior. Our experiments are for the main road consisting two lanes. We vary the level of speeding in the range of 10%–50% and the percentage of drivers who speed from 10%–50%. Figure 6.12 shows the impact of speeding. From the figure, we observe that speeding has negative impact on overall traffic flow and the total number of cars that can merge. The underlying reason is the existence of large velocity differences of the speeding cars from others, thus leading to unstable traffic flow. Notably, speeding offers an advantage on global average velocity. However, speeding may cause traffic accidents.
Figure 6.11: Heterogeneous traffic flow
6.5 Higher Level of Realism

Figure 6.12: Impact of speeding: 150 km/h
6.6 Conclusion

Different to centralized systems such as traffic light control, we have adopted a decentralized approach, where individual cars negotiate when needed. To the best of our knowledge, we are the first to propose robust merging algorithms for sensor-enabled cars, which work under imprecise sensor measurements to minimize adverse impact of positioning inaccuracies. We have presented two proactive merging algorithms LO and TG to cope with inaccurate information and improve traffic flow. Based on time geography, the algorithm TG computes the “smoothest” acceleration for involved cars and an optimal merging point for a ramp car before it merges into the main road. Based on the local optimum of gap distribution, the algorithm LO increases the tolerance of sensor information by restricting the impact of merging to a small group of main road cars, which results from the redistribution of gaps on the main road. We have compared the algorithms LO and TG against the reference priority-based merging algorithm R and other two proactive merging algorithms, PV and CPV, under a large variety of realistic experimental settings. The experiments show that our proposed merging algorithm TG doubles the road capacity and significantly improves the traffic flow by up to 25%. Most importantly, TG guarantees safe merging while tolerating four times more imprecise information.
Chapter 7

Merging Algorithms for Traffic Diversion at Multiple Ramps

We proposed proactive merging algorithms that improve merging efficiency in Chapter 5 and merging algorithms in terms of robustness to cope with inaccurate sensing measurements in Chapter 6. While the developed algorithms show improved merging efficiency, high traffic demand to merge at a single ramp can cause congestions on the ramp or on the main road. Therefore, in this chapter, we study if and for which merging algorithms the distribution of the total incoming traffic to different ramps could be an effective approach to mitigate congestions. Investigating the behavior of merging at different locations is an essential step to study large scale road networks. Experimental results demonstrate that for proactive traffic merging algorithms, diverting the total incoming traffic to merge into the main road at two ramps achieves significant improvement of traffic flow. Specifically, merging capacity can be doubled, traffic flow can be increased by 28%, and the total travel time can be saved by 16%.

7.1 Introduction

Existing studies on traffic diversion are either based on the information of the travel time [144], the main road velocity [37, 86] or density [61, 108] near a ramp. According to [86, 144], traffic time-based diversion strategies suffer from the information delay because the information of travel time reflects the traffic conditions some time ago. However, the rich spatial information obtained by sensor-enabled cars has not been
exploited in this context.

Merging algorithms can work either individually or in a coordinated manner. Sensor-enabled cars can exchange aggregated traffic information of the merging areas. We investigate a simple traffic network consisting of a main road, two ramps and an exit. We examine whether it is beneficial to divert the incoming traffic from a single ramp to enter the main road via the second ramp. In particular, we address two research questions: (1) the minimum distance between two ramps; and (2) the optimal ratio of splitting the total demand at multiple ramps.

### 7.2 Multiple Ramps

Figure 7.1 illustrates a simple highway network consisting of a main road and two ramps. Two merging points are denoted by $M_1$ and $M_2$. The distance between $M_1$ and $M_2$ is $L_1$. Similar to previous chapters, $S$ denotes the decision point. $D$ denotes the diversion point, where the incoming traffic is split to merge at either $M_1$ or $M_2$.

![Figure 7.1: Notations of traffic diversion algorithms](image)

### 7.3 Simulation Results

In this section, we compare the performance of merging algorithms at two ramps in a variety of simulation settings. We evaluate merging algorithms R, PV, and TG for a single lane main road to eliminate the influence of lane-changing on the main road near the exit area.
Table 7.1 summarizes the experiment settings. The settings for the initial density of the main road are similar to Section 5.5.1: $5 \text{ cars/km}$ for light traffic and $10 \text{ cars/km}$ for medium traffic. For the incoming traffic from the ramp, we only focus on heavy traffic demand: $7.2 \text{ cars/km}$ (i.e., $12 \text{ cars/min}$) in order to identify the impact of diversion. For experiments 1 to 3, we study a closed system, in which the exit is closed. In experiment 4, the exit is open and cars leave the main road after they travel certain required distance. Ratio $\alpha$ is the ratio of arrival traffic between two ramps and $L_1$ is the distance between two ramps.

Table 7.1: Experiment settings

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Main ($\text{cars/km}$)</th>
<th>Ramp ($\text{cars/km}$)</th>
<th>Distribution</th>
<th>Ratio $\alpha$</th>
<th>$L_1$ (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>7.2</td>
<td>Uniform</td>
<td>1:1</td>
<td>$0.5-4$</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7.2</td>
<td>Poisson</td>
<td>1:1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>7.2</td>
<td>Uniform</td>
<td>$3:1-1:2$</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>7.2</td>
<td>Poisson</td>
<td>1:1</td>
<td>2</td>
</tr>
</tbody>
</table>

We first study the impact on the main road traffic of apportioning the total incoming traffic to merge at two ramps. Then, we investigate the impact of the distance between two ramps.

We use R-2, PV-2, and TG-2 to denote the merging algorithms working at two ramps in the figures.

### 7.3.1 Impact of Diverting the Total Incoming Traffic to Two Ramps

In experiment 1 and 2, we start with light traffic on the main road ($50 \text{ cars over 10 km main road}$). Figure. 7.2 and Figure. 7.3 compare merging at the single ramp and two ramps for constant arrival rate. Figure. 7.4 and Figure. 7.5 compare for the arrival rate following the Poisson distribution. In both cases, for two ramps the ratio of splitting the incoming traffic at $D$ is 1:1 and the distance between two ramps is $2 \text{ km}$.

For R, distributing the incoming traffic flow from a single ramp to two ramps does not change the traffic flow. For PV, the performance increased significantly in terms of the number of cars which can merge and the traffic flow. In particular, for the Poisson
Figure 7.2: Incoming traffic merging at a single ramp
Figure 7.3: Incoming traffic apportioned to two-ramp; $L_1 = 2 \text{ km}$, $\alpha = 1 : 1$
Figure 7.4: Incoming traffic following the Poisson distribution merging at a single ramp
Figure 7.5: Incoming traffic following the Poisson distribution; \( L_1 = 2 \text{ km} \), \( \alpha = 1 : 1 \)
arrival, PV-2 achieves merging 205 cars, which is more than double compared to PV (merging 100 cars). The traffic flow increases 28% as well. As expected, the average velocity of PV-2 becomes lower than PV.

R does not improve when giving more opportunity to merge because R highly favors one stream. In the ramp and highway merging case, it gives priority to the main road traffic. This strategy is reasonable for one stream with low traffic but there is disadvantage for merging two streams with similar traffic load, for example, two highways merge.

TG is the most robust strategy under imprecise measurements. It maximizes the number of cars that can merge. The expense is a lower average velocity. To get the traffic flow stable at a high value, we can restrict TG from merging more ramp cars when the average velocity is below a threshold. The benefit for TG is that it reduces the number of ramps compared to PV-2. PV is good for achieving higher traffic flow by splitting the merging to more ramps.

7.3.2 Impact of the Distance between Two Ramps

We investigate what is the optimal distance $L_1$ between two merging points $M_1$ and $M_2$. In each experiment, we double the distance $L_1$ from 0.5 km until 4 km. Our study reveals that, as expected, TG and TG-2 show similar behavior pattern in increasing the number of merged cars. R and R-2 also exhibit similar performance. Therefore, we focus on comparing the performance of PV-2 for different $L_1$. Figure 7.6 shows results for PV-2 under different separation of two ramps. We find that the distance between the ramps should be at least 1 km. The distance of 0.5 km apart also gives some benefit, but much less than that of more than 1 km. This is due to the following reason. The average velocity on the main road near the ramp area decreases as more cars merge from the ramp, which results in smaller gaps between cars. The reduced velocity and small gaps between cars recover after some time if there is no further disturbance. When the second ramp $M_2$ is close to $M_1$, it is difficult for new incoming cars to merge from $M_2$ due to small gaps on the main road. When the distance between $M_1$ and
Figure 7.6: Incoming traffic merging at fixed ratio varying the distance between 2 ramps $L_1 = 0.5-4 \text{ km}, \alpha = 1 : 1$
$M_2$ becomes larger, merging from $M_2$ becomes easier because the gaps between cars become larger.

### 7.3.3 Impact of the Ratio of Distributing the Total Incoming Traffic

We investigate the impact of different ratios of distributing the incoming traffic to two ramps for PV. Figure 7.7 shows the ratio varying from 3:1 to 1:2. To allocate the first ramp more traffic load achieves better results. The main reasons for these results are the following: first, traffic congestions dissipate backwards the traffic direction (i.e., from $M_2$ to $M_1$ in Figure 7.1). Therefore, less traffic load from $M_2$ helps traffic to recover more quickly. Second, a car merges from $M_2$ influences more cars on the main road than a car merges from $M_1$ because front cars can influence the following cars, and not vice versa. This finding is consistent with [150] in a related context on ramp metering rate for multiple ramps. They suggest that the downstream ramp should be restricted more than the upstream ramp.

In Figure 7.7, the traffic flow gets stable at 27 cars/min in the strategy PV-2, compared to PV 23 cars/min, increases 17%. The number of merged cars in the strategy PV-2 is above 180 (230 cars in total) whereas PV is 116. The increase is more than 50%.

### 7.3.4 Open Exit

After examining the performance of a closed system, we now study the performance of merging algorithms for an open system. For this purpose, we define the following terms:

- **trip**: denotes the journey of a car. It starts when a car merges from the ramp to the main road and finishes when a car completes traveling before it leaves the main road.

- **individual travel time**: refers to the time it takes for a car to complete a trip.

- **total travel time (TTT)**: denotes the time to complete trips for a certain number of cars in the system. It depends on traffic flow, total number of cars and velocity.
Figure 7.7: Incoming traffic merging at fixed varying the ratio $L_1 = 2\, km$, $\alpha = 3 : 1; 2 : 1; 1 : 1; 1 : 2$
We measure how long it takes for the system to finish trips for a certain number of cars. To examine this we use TTT as the performance metric. We test which algorithm has the least TTT for a given number of cars to finish traveling. This measurement includes the merging delay, merging capacity and average velocity. The experiment setting is 100 cars initially on the main road. The arrival traffic follows the Poisson distribution.
Figure 7.9: Traffic dynamics for 1-ramp
Figure 7.10: Traffic dynamics for 2-ramp; $L_1 = 2 \text{ km}$, $\alpha = 1 : 1$
distribution with a mean of 12 cars per minute arriving at the ramp. This setting is to reflect traffic at peak hours. We keep track of the number of cars finishing trips and get the TTT for 400 cars.

Figure 7.8 shows the TTT of the merging algorithms for one and two ramps, respectively. Figure 7.8(a) and Figure 7.8(b) respectively show the cars traveling one and two circles before exiting the main road. In most settings, the average flow can reflect TTT. Higher average flow indicates less TTT. One exception is found for TG when the setting of 400 cars finish traveling in 2 circles. In this case, the average flow is higher for merging at a single ramp (20.4 cars/min) but TTT is larger (67 min) compared to merging at two ramps with the average flow of 19 cars/min and TTT of 62 min.

Figure 7.9 and Figure 7.10 respectively show the traffic dynamics in terms of flow, the total number of cars, and the average velocity for one and two ramps. In Figure 7.9, TG reduces TTT by 13% in comparison to R. By comparing to Figure 7.10, we find that PV-2 reduces TTT by 14% than PV, while PV decreases TTT up to 16% compared to R.

7.4 Conclusion

In summary, by varying the distance between two ramps and the separation ratio of the total incoming traffic, we find that:

- distributing incoming traffic from a single ramp to two ramps significantly improves the overall traffic flow. The merging capacity increases up to 100%.
- the two ramps should be separated at least 0.5 km away. When the distance between two ramps is larger than 2 km, the performance converges.
- the overall traffic flow is not impacted by the ratio of incoming traffic between two ramps. However, the impact on the merging capacity is significant. The upstream ramp should be assigned more traffic than the downstream ramp.
The purpose of this thesis is to integrate sensed information with traffic control strategies for sensor-enabled cars to alleviate traffic congestions and bottlenecks. Throughout the thesis, we presented a novel set of proactive traffic merging algorithms and a simulation tool to evaluate the performance of these algorithms. In this chapter, we first summarize the contributions of this thesis. Then we present a list of research issues for future investigations.

8.1 Summary

After analyzing existing traffic control strategies, we found that the major challenges of mitigating congestions are how to improve the merging bottleneck capacity and how to make driving decisions based on the local spatial information. To address these challenges, this thesis focused on optimizing traffic capacity in highway merging scenarios. There is a considerable amount of research addressing merging strategies, in particular applying queueing theory and statistics. However, these approaches neither integrate different information sources available from sensor-enabled cars nor do they compare different merging strategies under more realistic experiments. They typically assume that cars have constant velocity and a certain arrival process. In contrast, our work drops these assumptions as these are not applicable to real traffic conditions.

Our analysis of traffic models revealed that microscopic traffic models are more suitable for investigating merging algorithms. Therefore, we adopted a microscopic traffic model in our work. We detailed the underlying traffic model that was used
Conclusions and Future Directions

as the foundation of our work and provided the methodology to evaluate the performance of traffic merging algorithms. In this context, we presented evaluation measures from both safety and efficiency perspectives.

We developed a microscopic simulator, an extension of Treiber’s simulator [133], to incorporate explicit modeling of the spatial information obtained by sensor-enabled cars and the cooperation among drivers based on the information exchanged. Our simulator exhibits the following advantages: scalability; robustness; easy configuration of the traffic model and simulation parameters; visualization of dynamic traffic properties; and capability to evaluate traffic management algorithms under realistic scenarios.

Different to centralized systems such as traffic light control, we adopted a decentralized approach, where individual cars negotiate when needed. One of our major contributions is to integrate sensed information with traffic control strategies. We proposed a set of proactive traffic merging algorithms to improve traffic flow. Proactive merging algorithms separate the decision point, where a car computes the optimal merging point, and the actual merging point to adjust the velocity of a merging car in advance. Our proactive merging algorithms compute for a ramp car the optimal slot in a group of main road cars based on their velocities. Sensor-enabled cars allow us to decide where and when a car merges before it arrives at the actual merging point. This leads to a significant improvement in traffic flow as velocities can be adjusted appropriately. We compared the proposed platoon velocity-based algorithm (PV) and cascading platoon velocity-based algorithm (CPV) against the reference priority-based merging algorithm R under a large variety of realistic experimental settings. The results showed that CPV, where main road cars change lanes prior to the merging area to create a larger gap, significantly outperforms the algorithm R in terms of an increased traffic flow by a factor of two and a decreased delay by 30%.

We investigated the robustness of merging algorithms in terms of the accuracy level of sensors. To the best of our knowledge, we are the first to propose robust merging algorithms for sensor-enabled cars, which work under imprecise sensor measurements to minimize adverse impact of positioning inaccuracies. While the proposed
proactive merging algorithms showed significant improvement, they are based on the assumption that sensors provide accurate measurements. However, sensor measurements are usually not completely exact. The accuracy level of sensors poses a major challenge for merging algorithms, because inaccuracies can potentially lead to unsafe merging behaviors. We explored how the accuracy of sensors impacts merging algorithms and designed two novel merging algorithms that are robust to tolerate sensor inaccuracies. Based on the local optimum of gap distribution, the locally gap-optimal algorithm (LO) increases the tolerance of sensor information by restricting the impact of merging to a small group of main road cars, which results from the redistribution of gaps on the main road. Adapted from the concept of light cones [96] and space-time prisms [56], the time geography-based merging algorithm (TG) achieves a higher tolerance of imprecise sensor information to guarantee safe merging, and a smoother merging process to ensure an increased merging capacity and less impact on the subsequent traffic flow. Experimental results showed that our proposed merging algorithm TG doubles the road capacity and significantly improves the traffic flow by up to 25%. Most importantly, TG guarantees safe merging while tolerating four times more imprecise information.

Furthermore, we investigated merging algorithms at multiple ramps. It is an essential step to study large scale road networks. Experimental results demonstrate that for proactive traffic merging algorithms, diverting the total incoming traffic to merge into the main road at two ramps achieves significant improvement of traffic flow. Specifically, merging capacity can be doubled, traffic flow can be increased by 28%, and the total travel time can be saved by 16%.

To summarize, this thesis has laid the foundation for alleviating traffic congestions with the use of sensor-enabled cars, coupled with a novel set of proactive merging algorithms. With these contributions, this thesis opens up avenues for future research in relation to traffic control strategies incorporating sensor-enabled cars.
8.2 Future Directions

There are several future research directions that we plan to follow:

8.2.1 Traffic Pattern

In our simulation, we have studied uniform and Poisson distributions. We plan to explore a comprehensive set of traffic conditions, such as obstacles in the way, highly bursty traffic (e.g., burst of traffic in rush hours). We can incorporate with a trace-based simulation, which will allow arbitrary traffic conditions. Furthermore, we plan to simulate abnormal traffic conditions, such as the impact of accidents.

8.2.2 Impact of Global Information on Local Decisions

One promising future work is to investigate the impact of global information on making local decision. For example, traffic awareness control strategies can be combined with on-board route guidance systems for Location-Based Services (LBS). If there is a congestion, or road closure due to a sports event, rather than approaching the traffic jam and waiting, individual merging algorithms will inform drivers earlier and provide alternative routes.

8.2.3 Large Scale Road Networks

Another future direction is how to optimize traffic flow for a large complex road network that involves a number of intersections at urban road networks and multiple on-ramps and exit-ramps at freeway networks. We will investigate how to exploit scheduling algorithms in the context of merging scenarios at the road network.

A generalization of the traffic merging algorithm will be applied in general traffic control. In particular, we are interested in studying the performance of different strategies in road crossings without traffic lights, such as roundabouts. There are existing models in literature. They present models to analyze the different types of
road intersections. For example, Kakooza et al. [71] indicate that under light traffic, roundabout intersections perform better than signalized and unsignalized in terms of easing congestion; under heavy traffic, signalized intersection perform better in terms of easing traffic congestion compared to unsignalized and roundabout intersections. Since sensor-enabled cars can get more information and can communicate among each other, we plan to investigate the performance at roundabouts.

### 8.2.4 Human Factors

It is important to investigate a number of human factors in the future work. A traffic control algorithm needs to accommodate various human factors, such as reaction times, speeding, and be prepared to handle drivers who do not follow the system’s recommendations. It is of great importance for studying traffic safety. Our preliminary work on this issue was briefly covered in Section 6.5.3. We found that speeding violations can lead to dangerous driving situations. Cooperative merging shows a significant improvement, which primarily requires vehicles on the main road to create a larger gap to accommodate a merging vehicle. Changing the behavior of human drivers is complicated. A new set of traffic regulations might be needed for incorporating the emerging sensor-enabled cars.

### 8.2.5 Traffic Prediction Models

Sensor-enabled cars collect and communicate traffic information. In the future, merging algorithms can be coupled with traffic prediction models to provide trip advisory to the drivers in order to make timely and informed travel decisions. These traffic prediction models may exploit the available sensing and merging algorithms covered in this thesis. In our context, such a model can be used to predicate the travel time according to a driver’s response to the on-board system’s merging recommendation. It can be used to provide an incentive for the drivers to follow the system’s recommendation. For example, the prediction model can compute how much travel time
a driver can save in the current traffic condition with following the provided recommendations.
# Appendix A

## Notations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>adaptive cruise control</td>
<td>2</td>
</tr>
<tr>
<td>AHS</td>
<td>automated highway systems</td>
<td>32</td>
</tr>
<tr>
<td>ATMS</td>
<td>advanced traffic management systems</td>
<td>56</td>
</tr>
<tr>
<td>CA</td>
<td>cellular automata</td>
<td>12</td>
</tr>
<tr>
<td>CPV</td>
<td>cascading platoon velocity-based merging</td>
<td>76</td>
</tr>
<tr>
<td>D</td>
<td>distance-based merging</td>
<td>71</td>
</tr>
<tr>
<td>DGPS</td>
<td>differential global positioning system</td>
<td>26</td>
</tr>
<tr>
<td>DSRC</td>
<td>dedicated short range communications</td>
<td>2</td>
</tr>
<tr>
<td>FTL</td>
<td>follow-the-leader model</td>
<td>14</td>
</tr>
<tr>
<td>GIS</td>
<td>geographic information systems</td>
<td>103</td>
</tr>
<tr>
<td>GPS</td>
<td>global positioning system</td>
<td>25</td>
</tr>
<tr>
<td>IDM</td>
<td>intelligent driver model</td>
<td>15</td>
</tr>
<tr>
<td>INS</td>
<td>inertial navigation system</td>
<td>26</td>
</tr>
<tr>
<td>ITS</td>
<td>intelligent transportation system</td>
<td>2</td>
</tr>
<tr>
<td>IVC</td>
<td>inter-vehicle communication</td>
<td>26</td>
</tr>
<tr>
<td>LBS</td>
<td>location-based services</td>
<td>103</td>
</tr>
<tr>
<td>LO</td>
<td>locally gap-optimal merging</td>
<td>105</td>
</tr>
<tr>
<td>MEMS</td>
<td>micro-electro-mechanical systems</td>
<td>24</td>
</tr>
<tr>
<td>MOBIL</td>
<td>minimizing overall braking induced by lane change model</td>
<td>17</td>
</tr>
<tr>
<td>OVM</td>
<td>optimal velocity model</td>
<td>14</td>
</tr>
<tr>
<td>PV</td>
<td>platoon velocity-based merging</td>
<td>72</td>
</tr>
<tr>
<td>R</td>
<td>priority-based merging</td>
<td>68</td>
</tr>
<tr>
<td>RWP</td>
<td>random waypoint model</td>
<td>12</td>
</tr>
<tr>
<td>TG</td>
<td>time geography-based merging</td>
<td>109</td>
</tr>
<tr>
<td>TTC</td>
<td>times-to-collision</td>
<td>50</td>
</tr>
<tr>
<td>V</td>
<td>velocity-based merging</td>
<td>71</td>
</tr>
<tr>
<td>VANET</td>
<td>vehicular ad-hoc networks</td>
<td>2</td>
</tr>
<tr>
<td>WLAN</td>
<td>wireless local area network</td>
<td>29</td>
</tr>
<tr>
<td>WSN</td>
<td>wireless sensor networks</td>
<td>28</td>
</tr>
</tbody>
</table>
Bibliography


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Wang, Ziyuan

Title:
Proactive traffic control strategies for sensor-enabled cars

Date:
2009

Citation:

Persistent Link:
http://hdl.handle.net/11343/35325

File Description:
Proactive traffic control strategies for sensor-enabled cars

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.