A Multi-Sensor Data Fusion Approach for
Real-Time Lane-Based Traffic Estimation

by

Zhuoyang Zhou

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2015 by the
Graduate Supervisory Committee:

Pitu Mirchandani, Chair
Ronald Askin
George Runger
Xuesong Zhou

ARIZONA STATE UNIVERSITY
May 2015
ABSTRACT

Modern intelligent transportation systems (ITS) make driving more efficient, easier, and safer. Knowledge of real-time traffic conditions is a critical input for operating ITS. Real-time freeway traffic state estimation approaches have been used to quantify traffic conditions given limited amount of data collected by traffic sensors. Currently, almost all real-time estimation methods have been developed for estimating laterally aggregated traffic conditions in a roadway segment using link-based models which assume homogeneous conditions across multiple lanes. However, with new advances and applications of ITS, knowledge of lane-based traffic conditions is becoming important, where the traffic condition differences among lanes are recognized. In addition, most of the current real-time freeway traffic estimators consider only data from loop detectors. This dissertation develops a bi-level data fusion approach using heterogeneous multi-sensor measurements to estimate real-time lane-based freeway traffic conditions, which integrates a link-level model-based estimator and a lane-level data-driven estimator.

Macroscopic traffic flow models describe the evolution of aggregated traffic characteristics over time and space, which are required by model-based traffic estimation approaches. Since current first-order Lagrangian macroscopic traffic flow model has some unrealistic implicit assumptions (e.g., infinite acceleration), a second-order Lagrangian macroscopic traffic flow model has been developed by incorporating drivers’ anticipation and reaction delay. A multi-sensor extended Kalman filter (MEKF) algorithm has been developed to combine heterogeneous measurements from multiple sources. A MEKF-based traffic estimator, explicitly using the developed second-order traffic flow model and
measurements from loop detectors as well as GPS trajectories for given fractions of vehicles, has been proposed which gives real-time link-level traffic estimates in the bi-level estimation system.

The lane-level estimation in the bi-level data fusion system uses the link-level estimates as priors and adopts a data-driven approach to obtain lane-based estimates, where now heterogeneous multi-sensor measurements are combined using parallel spatial-temporal filters.

Experimental analysis shows that the second-order model can more realistically reproduce real world traffic flow patterns (e.g., stop-and-go waves). The MEKF-based link-level estimator exhibits more accurate results than the estimator that uses only a single data source. Evaluation of the lane-level estimator demonstrates that the proposed new bi-level multi-sensor data fusion system can provide very good estimates of real-time lane-based traffic conditions.
To my parents,

Mr. Jiayun Zhou and Mrs. Jin Xiao Liu.
ACKNOWLEDGMENTS

I feel deeply in my heart that I would never have been able to finish this dissertation without the guidance from my committee members, help from my friends and support from my family.

I would like to express my deepest gratitude to my advisor and committee chair, Prof. Pitu Mirchandani, who has inspiringly guided and mentored me, patiently corrected my writing, and financially supported my research. He has continually and convincingly conveyed to me what is the right attitude towards research and teaching. His mentoring will benefit me all my life. I would like to thank my other committee members, Prof. Ronald Askin, Prof. George Runger, and Prof. Xuesong Zhou for their insightful comments and guidance. I would also like to thank Prof. Soyoung Ahn, now at the University of Wisconsin, for our discussions on traffic flow theory during a freeway reliability project that we collaborated on.

I would like to acknowledge the interactions with the TU Delft team in the Netherlands: Prof. Hans van Lint, Dr. Femke van Wageningen-Kessels and Dr. Yufei Yuan.

Special thanks to Ning Wang and Qing Li, my dearest colleagues and friends, who have always been there to support me. I also own a great deal of thanks to Lei Liu, Hairong Xie, Suseon Yang, Min Zhang, and many other individuals for their help and encouragement in both life and work. Thanks also to my colleague Kerem Demirtas for the insightful discussions.

My deepest appreciation also goes to my friends and family in China. Thanks to Dr. Yi Sun and many other individuals whose help and encouragement I will always remember.
Thanks very much to my parents, my aunt, my sister, and all my family for their deep love and unwavering support through all these years.

Many thanks to Bing Li, meeting whom was the most beautiful thing that happened to me in Brickyard.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
</tbody>
</table>

## CHAPTER

1 INTRODUCTION .................................................................................................................. 1

1.1 Background .................................................................................................................. 1

1.2 Urban Freeway Monitoring Sensors ............................................................................ 4

1.3 Traffic Flow Modeling ............................................................................................... 7

1.4 Traffic State Estimation ........................................................................................... 10

1.5 Multi-Sensor Data Fusion .......................................................................................... 12

1.6 Problem Objectives and Research Scope ................................................................... 21

1.7 Outline of Dissertation and Summary of Contributions ........................................... 22

2 LITERATURE REVIEW ....................................................................................................... 25

2.1 Introduction .................................................................................................................. 25

2.2 Measurements from Different Sensors ....................................................................... 26

2.3 Data Fusion Algorithms .............................................................................................. 29

2.4 Macroscopic Traffic Flow Modeling .......................................................................... 34

2.5 Research Direction and Motivations .......................................................................... 38

2.6 Summary ....................................................................................................................... 38

3 ARCHITECTURE FOR REAL-TIME LANE-BASED TRAFFIC ESTIMATION .......................... 40

3.1 Introduction .................................................................................................................. 40
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>Bi-Level Data Fusion</td>
</tr>
<tr>
<td>3.3</td>
<td>Summary</td>
</tr>
<tr>
<td>4</td>
<td>A SECOND-ORDER LAGRANGIAN TRAFFIC FLOW MODEL</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Macroscopic Traffic Flow Models</td>
</tr>
<tr>
<td>4.3</td>
<td>A Second-Order Lagrangian Model</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental Analysis: Lane-Drop Scenario</td>
</tr>
<tr>
<td>4.5</td>
<td>Experimental Analysis: US101 Scenario</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
</tr>
<tr>
<td>5</td>
<td>A MULTI-SENSOR DATA FUSION APPROACH FOR REAL-TIME TRAFFIC ESTIMATION: LINK LEVEL</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Model-Based Estimation Algorithms</td>
</tr>
<tr>
<td>5.3</td>
<td>MEKF-Based Lagrangian Link-Level Traffic Estimation</td>
</tr>
<tr>
<td>5.4</td>
<td>Experimental Analysis: US101 Freeway Scenario</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
</tr>
<tr>
<td>6</td>
<td>A MULTI-SENSOR DATA FUSION APPROACH FOR REAL-TIME TRAFFIC ESTIMATION: LANE LEVEL</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>6.2</td>
<td>A Data-Driven Approach For Lane-Level Traffic Estimation</td>
</tr>
<tr>
<td>6.3</td>
<td>Experimental Analysis: Interstate-10 Freeway Scenario</td>
</tr>
<tr>
<td>6.4</td>
<td>Experimental Analysis: US101 Freeway Scenario</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary .................................................................157</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH..............158</td>
</tr>
<tr>
<td>7.1</td>
<td>Research Conclusions and Contributions.........................158</td>
</tr>
<tr>
<td>7.2</td>
<td>Directions for Future Research......................................161</td>
</tr>
<tr>
<td>BIBLIOGRAPHY .................................................................163</td>
<td></td>
</tr>
<tr>
<td>APPENDIX</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>KALMAN FILTER.............................................................171</td>
</tr>
<tr>
<td>B</td>
<td>RMSE COMPARISON FOR BASELINE ANALYSIS IN I-10 FREEWAY LANE-LEVEL ESTIMATION ...........................................173</td>
</tr>
<tr>
<td>C</td>
<td>SPEED MAPS FOR PENETRATION ANALYSIS IN I-10 FREEWAY LANE-LEVEL ESTIMATION ...........................................177</td>
</tr>
<tr>
<td>D</td>
<td>PARAMETER SUMMARY FOR US101 FREEWAY LANE-LEVEL TRAFFIC ESTIMATION.............................................................183</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Parameter Settings for US101 Freeway Macroscopic Simulation</td>
<td>67</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison Settings for US101 Freeway Link-Level Estimation</td>
<td>98</td>
</tr>
<tr>
<td>5.2</td>
<td>RMSE Summary: Multi-Sensor vs. Loop-Detector-Only Estimation</td>
<td>102</td>
</tr>
<tr>
<td>5.3</td>
<td>RMSE Summary: Multi-Sensor vs. Probe-Vehicle-Only Estimation</td>
<td>104</td>
</tr>
<tr>
<td>6.1</td>
<td>Scenario Parameter Settings for I-10 Freeway Lane-Level Estimation</td>
<td>124</td>
</tr>
<tr>
<td>6.2</td>
<td>RMSE for I-10W Freeway Speed Estimation Baseline Analysis</td>
<td>128</td>
</tr>
<tr>
<td>6.3</td>
<td>RMSE for Weight Analysis: $\alpha_{\text{prior}}$ vs. $\alpha_{\text{vehicle}}$</td>
<td>136</td>
</tr>
<tr>
<td>6.4</td>
<td>RMSE for Weight Analysis: $\alpha_{\text{prior}}$ vs. $\alpha_{\text{loop}}$</td>
<td>138</td>
</tr>
<tr>
<td>6.5</td>
<td>RMSE for Weight Analysis: $\alpha_{\text{loop}}$ vs. $\alpha_{\text{vehicle}}$</td>
<td>139</td>
</tr>
<tr>
<td>6.6</td>
<td>RMSE for Weight Analysis: $\alpha_{\text{past}}$</td>
<td>140</td>
</tr>
<tr>
<td>6.7</td>
<td>RMSE for Penetration Rate Analysis: $p_{\text{probe}}$</td>
<td>142</td>
</tr>
<tr>
<td>6.8</td>
<td>RMSE for Loop Detector Configuration Analysis: $\Delta x_{\text{loop}}$</td>
<td>143</td>
</tr>
<tr>
<td>6.9</td>
<td>RMSE for US101 Freeway Lane-Level Speed Estimation</td>
<td>150</td>
</tr>
<tr>
<td>D.1</td>
<td>Parameter Settings for US101 Freeway Lane-Level Estimation</td>
<td>184</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 An Example of Lagrangian Fundamental Diagram</td>
<td>9</td>
</tr>
<tr>
<td>1.2 An Example of the Ramp Metering Control System</td>
<td>11</td>
</tr>
<tr>
<td>1.3 Spatial Alignment</td>
<td>14</td>
</tr>
<tr>
<td>1.4 Temporal Arrival Patterns</td>
<td>16</td>
</tr>
<tr>
<td>1.5 Bayesian Fusion</td>
<td>21</td>
</tr>
<tr>
<td>1.6 Dissertation Layout</td>
<td>23</td>
</tr>
<tr>
<td>3.1 A Bi-Level Architecture for Multi-Lane Traffic State Estimation</td>
<td>42</td>
</tr>
<tr>
<td>3.2 Spatial Discretization (a) low resolution; (b) high resolution</td>
<td>43</td>
</tr>
<tr>
<td>4.1 Network Model Illustrating the Discontinuous Points (Nodes)</td>
<td>54</td>
</tr>
<tr>
<td>4.2 Lane-Drop Node Model</td>
<td>57</td>
</tr>
<tr>
<td>4.3 Lane-Drop Scenario</td>
<td>59</td>
</tr>
<tr>
<td>4.4 Speed Maps using First-Order Model</td>
<td>61</td>
</tr>
<tr>
<td>4.5 Speed Maps using Second-Order Model with Parameter Set 1</td>
<td>62</td>
</tr>
<tr>
<td>4.6 Speed Maps using Second-Order Model with Parameter Set 2</td>
<td>63</td>
</tr>
<tr>
<td>4.7 An Illustration of US101 Freeway Segment</td>
<td>66</td>
</tr>
<tr>
<td>4.8 Choices of the Ground Truth Resolution</td>
<td>69</td>
</tr>
<tr>
<td>4.9 Speed Maps from Simulation in Lagrangian Coordinates</td>
<td>72</td>
</tr>
<tr>
<td>4.10 Speed Maps from Simulation in Eulerian Coordinates</td>
<td>73</td>
</tr>
<tr>
<td>4.11 Density Maps from Simulation in Eulerian Coordinates</td>
<td>74</td>
</tr>
<tr>
<td>4.12 Fixed Location Comparison for the Traffic Flow Models</td>
<td>76</td>
</tr>
<tr>
<td>4.13 Lagrangian Simulation Shown in Speed Map</td>
<td>79</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.1</td>
<td>85</td>
</tr>
<tr>
<td>5.2</td>
<td>87</td>
</tr>
<tr>
<td>5.3</td>
<td>88</td>
</tr>
<tr>
<td>5.4</td>
<td>95</td>
</tr>
<tr>
<td>5.5</td>
<td>95</td>
</tr>
<tr>
<td>5.6</td>
<td>102</td>
</tr>
<tr>
<td>5.7</td>
<td>104</td>
</tr>
<tr>
<td>5.8</td>
<td>105</td>
</tr>
<tr>
<td>5.9</td>
<td>106</td>
</tr>
<tr>
<td>5.10</td>
<td>107</td>
</tr>
<tr>
<td>5.11</td>
<td>108</td>
</tr>
<tr>
<td>6.1</td>
<td>114</td>
</tr>
<tr>
<td>6.2</td>
<td>122</td>
</tr>
<tr>
<td>6.3</td>
<td>128</td>
</tr>
<tr>
<td>6.4</td>
<td>129</td>
</tr>
<tr>
<td>6.5</td>
<td>130</td>
</tr>
<tr>
<td>6.6</td>
<td>131</td>
</tr>
<tr>
<td>6.7</td>
<td>132</td>
</tr>
<tr>
<td>6.8</td>
<td>133</td>
</tr>
<tr>
<td>6.9</td>
<td>134</td>
</tr>
<tr>
<td>6.10</td>
<td>135</td>
</tr>
<tr>
<td>6.11</td>
<td>137</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>6.12</td>
<td>Weight Analysis: $\alpha_{\text{prior}}$ vs. $\alpha_{\text{loop}}$</td>
</tr>
<tr>
<td>6.13</td>
<td>Weight Analysis: $\alpha_{\text{loop}}$ vs. $\alpha_{\text{vehicle}}$</td>
</tr>
<tr>
<td>6.14</td>
<td>Weight Analysis: $\alpha_{\text{past}}$</td>
</tr>
<tr>
<td>6.15</td>
<td>Penetration Rate Analysis: $p_{\text{probe}}$</td>
</tr>
<tr>
<td>6.16</td>
<td>Loop Detector Configuration Analysis: $\Delta x_{\text{loop}}$</td>
</tr>
<tr>
<td>6.17</td>
<td>An Illustration of the US101 Freeway Segment</td>
</tr>
<tr>
<td>6.18</td>
<td>RMSE for US101 Freeway Lane-Level Speed Estimation</td>
</tr>
<tr>
<td>6.19</td>
<td>PoI for US101 Freeway Lane-Level Speed Estimation</td>
</tr>
<tr>
<td>6.20</td>
<td>Speed Map: US101 Freeway Link-Level Estimation</td>
</tr>
<tr>
<td>6.21</td>
<td>Speed Map Comparison for US101 Freeway Lane 1</td>
</tr>
<tr>
<td>6.22</td>
<td>Speed Map Comparison for US101 Freeway Lane 2</td>
</tr>
<tr>
<td>6.23</td>
<td>Speed Map Comparison for US101 Freeway Lane 3</td>
</tr>
<tr>
<td>6.24</td>
<td>Speed Map Comparison for US101 Freeway Lane 4</td>
</tr>
<tr>
<td>6.25</td>
<td>Speed Map Comparison for US101 Freeway Lane 5</td>
</tr>
<tr>
<td>B.1</td>
<td>Comparison of RMSE for Lane 1</td>
</tr>
<tr>
<td>B.2</td>
<td>Comparison of RMSE for Lane 2</td>
</tr>
<tr>
<td>B.3</td>
<td>Comparison of RMSE for Lane 3</td>
</tr>
<tr>
<td>B.4</td>
<td>Comparison of RMSE for Lane 4</td>
</tr>
<tr>
<td>B.5</td>
<td>Comparison of RMSE for Lane 5</td>
</tr>
<tr>
<td>C.1</td>
<td>Speed Maps for Lane 1 under Different Penetration Rates</td>
</tr>
<tr>
<td>C.2</td>
<td>Speed Maps for Lane 2 under Different Penetration Rates</td>
</tr>
<tr>
<td>Figure</td>
<td>Speed Maps for Lane 3 under Different Penetration Rates</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>C.3</td>
<td>Speed Maps for Lane 3 under Different Penetration Rates</td>
</tr>
<tr>
<td>C.4</td>
<td>Speed Maps for Lane 4 under Different Penetration Rates</td>
</tr>
<tr>
<td>C.5</td>
<td>Speed Maps for Lane 5 under Different Penetration Rates</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Background

Traveling is becoming more and more convenient thanks to the modern intelligent transportation system (ITS), whose applications range from complex systems such as the Global Positioning System (GPS) and the Advanced Public Transportation System (APTS) to some convenient and effective traffic management means such as the variable speed-limit signs and the highway emergency notification system. The concept of ITS emerged along with modern technologies and is still a promising area full of innovation opportunities. It integrates advanced communication technologies, including various wireless and wire line communications-based information technologies, to improve safety and mobility and enhance productivity in transportation (US DOT, 2013). ITS technologies have been utilizing increasing data amounts and larger number of available data sources and requiring increasingly accurate and comprehensive traffic condition information. Thus data fusion strategies can be extremely helpful. The concept of data fusion first appeared in the 1960s dealing with data manipulation for military purposes. With over fifty years of development, now it becomes a diverse field encompassing various models and algorithms, with applications not only in the military domain but also for industries and services. Indeed, it is applicable anywhere data, either quantitative or qualitative, plays a role. Narrowly speaking, data refers to the collected figures or measurements that are not necessarily immediately informative but have the potential to provide information. On the other hand, information refers to something informative, organized and helpful for making
decisions. Accordingly, the essential task of data fusion is to transform the system condition from data-rich-information-poor into information-rich and thus leads to meaningful decisions for improving the system performance. Since ITS are usually associated with large amounts of data, such as traffic counts reported by inductive loop detectors and weather information provided by weather service organizations, the data fusion strategies are intrinsically important. Dailey et al. (1996) presented an early review on this topic, introducing how a data fusion framework can be integrated with ITS applications, and which data fusion techniques might be effectively adopted. At that time, neither ITS nor data fusion strategies were well developed comparing to their status today, that review is not up-to-date. Recently, Faouzi et al. (2011) conducted a survey of a similar topic for road traffic problems which summarizes the new development and challenges in ITS data fusion with respect to advanced traveler information systems (ATIS), the automatic incident detection (AID), advanced driver assistance system (ADAS), network control, crash analysis and prevention, traffic demand estimation, traffic forecasting and traffic monitoring, the accurate position estimation (such as GPS), and so forth. They concluded that this emergent field is still somewhat in its infancy.

One of the major challenges regarding ITS data fusion urgently requiring research efforts is to effectively process, integrate and exploit multiple sources of data to yield accurate and timely quantified traffic condition input for lane-level ITS decision systems, that is, the problem of multi-sensor data fusion for real-time lane-based traffic state estimation. One example of its applications is the so-called “HOT” lanes. The high-occupancy-vehicle (HOV) lane is a special freeway lane, usually the innermost one, that only allows high-occupancy vehicles. The HOV-toll (HOT) lane is a relatively new traffic
management system that permits low-occupancy vehicles to utilize the HOV lanes by paying some tolls. The pricing problem of the HOT lane relies on the knowledge of the real-time traffic conditions on the HOV lane and other regular lanes. Another example involves the lane-specific GPS guidance system, which is expected to recommend the travelers which lanes to take given the real-time traffic conditions on different lanes. Although there are more and more advanced ITS involving lane-level decisions, the traffic state estimation approaches are almost exclusively conducted on the link level, meaning that the traffic conditions among different lanes are assumed to be homogeneous. Additionally, the traditional estimators also mostly depend on a single data source. Combining heterogeneous measurements is referred to as multi-sensor data fusion (MSDF) in the data fusion domain. Outside of the transportation research area, multi-sensor data fusion have been widely applied to fields such as monitoring manufacturing processes, medical diagnoses, robotics, and military applications (Hall & McMullen, 2004). Within the traffic estimation field, unfortunately, up to now most research is based on a single sensing source or homogeneous data with essentially the same characteristics.

For example, a framework for link-based freeway traffic state estimation using loop detectors has been proposed by Wang and Papageorgiou (2005) using an extended Kalman filter (EKF) approach. Yuan et al. (2012) presented another well designed estimator that uses either exclusively vehicle data or exclusively loop detector data. The traffic conditions are also aggregated to the link level. On the other hand, several methods for fusing multi-sensor data have been developed but mostly for offline traffic state reconstruction (van Lint and Hoogendoorn, 2010; Treiber et al., 2011).
Difficulties in real-time multi-sensor data fusion for lane-based traffic state estimation include (a) the dynamic modeling of lane-based traffic flows, (b) the noncommensuration of observations compared to the dimension of the traffic state, (c) the computational cost because of the enormous data size, and (d) the heterogeneity of multiple data sources. Those aforementioned traditional methods have their intrinsic limitations to provide real-time estimate on the lane level by incorporating heterogeneous measurements. The research reported in this dissertation attempts to bridge this gap. This research aims to solve the not sufficiently studied real-time lane-based urban freeway traffic state estimation problem, as well as to enhance link-based state estimation, using a multi-sensor data fusion approach.

1.2 Urban freeway monitoring sensors

1.2.1 Stationary sensors

Field traffic data, also referred to as measurements or observations, can come from various sensing sources. The inductive loop detector is probably the most widely applied traffic sensor. It is usually embedded per lane in the pavement and senses vehicles passing over it. The detectors across all lanes at a particular road position constitute a loop detector station. In freeway applications, two such stations could be a mile, half a mile, or 500 meters apart from each other; this distance varies from region to region as is based on jurisdiction’s considerations. Single and double loop detector stations are the most common. In single-loop detection station there is only one detector per lane buried under the freeway surface providing basically lane occupancy (i.e., percentage of the time the detector is occupied) and vehicle counts. In double-loop detection stations, two consecutive
detectors are buried close to each other so that the mean space speed can also be derived given the distance between the loops and the sensed travel time between the two detectors.

Loop detectors have been available for over the last 50 years, which contributes to their popularity. They enhance traffic management by providing relatively stable and persistent traffic data. Since the data are usually laterally aggregated across multiple lanes, they are mostly used to obtain the average traffic conditions. However, as the loop detectors are fixed at preset locations, they can only collect traffic data at specific points on the freeways. This is an inherent drawback of the loop detector system. In addition, the loop detectors generally do not differentiate vehicles from one another due to its sensing mechanism (some advanced loop detector systems classify vehicles based on size and number of axles). Thirdly, loop detectors require constant maintenance which may incur non-negligible costs. Finally, they are prone to sensing noise. The performance of the loop detector is heavily influenced by various factors, such as the weather, road, and installation conditions. Because of the sensing mechanism, the quality of the loop detector data may not always be satisfactory.

Other stationary detectors include video camera detection, microwave detection, magnetic sensors, infrared sensors, and so on. They may be less utilized than inductive loop detectors but mainly provide similar type of fixed-location traffic data, referred to as the Eulerian measurements. The concept of “Eulerian” is discussed in Section 1.3.

1.2.2 Mobile sensors

A contrasting classification to the Eulerian measurements is the Lagrangian measurements, which are intrinsically trajectory-based data collected by mobile sensors. Nowadays, multiple techniques can provide Lagrangian data, such as the radio-frequency
identification (RFID) transponders, smart phones, and Global Positioning System (GPS). RFID transponders uses unique identification readers to match the vehicles at entrances and exits of a freeway segment and thus collects point-to-point travel time data. GPS or smart phone equipped vehicles (referred to as probe vehicles) is becoming another popular traffic monitoring technique because of its fast increasing user population. GPS provides accurate location data and distributed point-based data such as trajectories and vehicle speeds. The mobile sensors can greatly improve the measurement coverage compared to the stationary sensors implying more empirical information. The trajectory-based measurements also provide much additional information that Eulerian measurements do not.

The main concerns on use of the probe-vehicle sensors include (a) the penetration rate and (b) data variance. Firstly, since the penetration rate is based on user participation and is not under the control of the transportation management administration, it cannot be guaranteed to always provide sufficient spatial-temporal coverage of the transportation network, notwithstanding the consideration of privacy issues. Secondly, the variance of vehicle based data could be large, mainly due to the heterogeneity of the drivers. Thus these data may not provide a good representation of the average traffic conditions. It can be understood in the way that, if we view the whole vehicle set as the driving population, then the unaggregated Lagrangian measurements constitute a sample drawn to approximate the population distribution, and the penetration rate indicates the sample size. On the other hand, the Eulerian data represent an estimate of the population mean with a large sample size. The Lagrangian data can also be aggregated to reduce noise; however, the effectiveness of the aggregated measurements highly depends on the penetration rate.
There are also some sensors that are capable of collecting both Eulerian and Lagrangian traffic data, an example of which is the remote sensing using aerial detectors. An aerial sensor can be docked in a surveillance helicopter hovering along the freeway stretch. In such a case, the position of the remote sensing data source is determined by the real-time location of the helicopter. The flight path of the helicopter can be preset or intelligently updated in real time. In either way, the remote sensor takes high resolution snapshots of the ground traffic. Single pictures capture the instantaneous ground traffic conditions, providing traffic characteristics such as traffic density, vehicle counts and vehicle spacings. Two sequential pictures with a small time gap at the same location record the traffic motion during that time period, and thus can derive the traffic flow, the instantaneous individual vehicle speeds, and the mean space speed of the area. A series of consecutive pictures yield, in addition, segments of vehicle trajectories.

Given the above discussion and with all the sources of data at hand, we are inspired that a smart way to take advantages of both the stability and persistence of the Eulerian data and the diversity and spatial-temporal coverage of the Lagrangian data is to combine them using a multi-sensor data fusion approach. Since the loop detector data and the probe-vehicle data are good representatives to Eulerian data and Lagrangian data, respectively, and since traffic data from all other types of sensors may be classified as Eulerian data and/or Lagrangian data, it is assumed in the rest of this dissertation that only the loop detector and the probe-vehicle sensor are available.

1.3 Traffic flow modeling

Traffic flow modeling is to abstract the traffic flow evolution over time and space using mathematical models. There are generally three levels of detail in traffic flow
modeling. The lowest aggregation level depicting detailed individual driving behaviors is referred to as the microscopic traffic flow modeling, and the highest aggregation level focusing on the average motion pattern of the traffic flow while neglecting the heterogeneous driving behaviors is referred to as the macroscopic traffic flow modeling. The mesoscopic traffic model lies between the two levels; it considers more detail than the macroscopic models but still leaves out the individual driving behaviors such as the overtaking. Model-based traffic state estimators mostly use macroscopic models to propagate the aggregated traffic characteristics over time and space on roadway segments.

Macroscopic traffic flow models can be categorized as Eulerian models or Lagrangian models according to which frame of reference is used in modeling. The *Eulerian coordinate system* has its origin fixed in a spatial position, and uses the physical freeway as the frame of reference. An observer standing on the freeway is stationary in this system while an observer riding on a vehicle is moving. In the *Lagrangian coordinate system*, the origin is fixed on the moving vehicles, and thus the frame of reference is the moving vehicles. Any observer stationary on the freeway (stationary in the Eulerian coordinate system) is moving in this Lagrangian coordinate system, and vice versa. The difference of the two systems lies in the change of the observing perspective. Typically, an Eulerian model defines state variables as locally aggregated traffic flow characteristics. For example, traffic density ($\rho$ veh/km), traffic flow ($q$ veh/h), mean space speed ($v$ km/h) of freeway cells/segments are the common measures. On the other hand, a Lagrangian model uses vehicle-group-based characteristics. For example, vehicle group mean speeds ($u$ km/h) and vehicle group mean spacings ($s$ m) are the typical Lagrangian variables. The dynamics of these variables is normally represented by two partial differential equations.
One states the conservation of vehicles in space and time, and the other to depict the physical relationship among the traffic characteristics variables (e.g., $u$ and $s$). The later one is conventionally referred to as the “fundamental diagram”. Figure 1.1 is an example of the Lagrangian fundamental diagram with Smulders’ formulation (Smulders, 1989), which implies that (a) the vehicle speed reduces to zero (complete stop) when the spacing is below a threshold of “jam spacing” ($s_{jam}$); (b) the vehicle speed increases as the spacing increases if the spacing is within a certain range ($s_{jam}$ to $s_{cr}$); and (c) the vehicle speed is bounded by free flow speed ($v_f$) when spacing is larger than $s_{cr}$. Macroscopic traffic flow models can also be classified as first-order or second-order models depending on how many terms are retained in PDE.

![Figure 1.1 An Example of Lagrangian Fundamental Diagram](image)

When applying traffic flow models to traffic state estimation, network discontinuities need to be modeled. Issues such as merges, diverges, and boundary conditions have to be addressed. Inherently, the Eulerian models have the advantage in
modeling the network discontinuities, as its reference frame is in accordance to the physical network.

1.4 Traffic state estimation

State estimation is a typical problem in control theory, where hidden dynamic system state needs to be estimated intelligently. Such intelligence is realized by some estimation algorithms using sensor-collected measurements and system models. The measurements, or the observations (denoted as \( z \)), can be used to determine some hard-to-observe system characteristics, referred to as “system state” (represented as \( \theta \)). Thus, it is expected that the desired hidden system characteristics can be estimated by measuring some observable characteristics.

Traffic state estimation (TSE) specifically deals with estimating the traffic characteristics mentioned in Section 1.3, such as the mean speed and traffic density. These defined traffic variables are referred to as traffic state. TSE can be applied to freeways or controlled local roads, can be specified on the link level or on the lane level, can be conducted in an online manner for real-time estimation or in an offline manner for state reconstruction, and can adopt a model-based approach or data-driven approach. A model-based estimator relies on assuming a dynamic traffic flow model, some observed measurements, and an estimation algorithm, while a data driven estimator infers the system state using traffic measurements and a data-driven algorithm with an underlying statistical model. A model-based traffic state estimator is called an Eulerian estimator if it is based on an Eulerian traffic flow model, or a Lagrangian estimator if a Lagrangian traffic flow model is adopted. The difficulties of TSE lie in (a) its under-determined nature, that is, measurements are sparse compared to the dimension of the desired system state, (b) the
complexity, non-linearity and uncertainty of the dynamic traffic system, and (c) the noise and uncertainty of the traffic measurements.

**FIGURE 1.2 An Example of the Ramp Metering Control System**

Figure 1.2 shows a ramp metering control system to illustrate an example of an ITS application using the real-time traffic state as inputs. In this particular application, the multi-sensor data fusion system assumes the task of traffic state estimation and identifies the system status, and provides the input for the ramp metering control system. Specifically, multiple types of sensors collect noisy data from the freeway, which is an unknown system
in a black box, and constitute the measurement system (possibly together with the archived data set). A model-based estimation algorithm combines the measurements and the traffic model to yield timely estimate of the traffic conditions on the link-level. The estimated traffic state is the input to the ramp metering control algorithm which decides the rate and durations of the green times. This adjustment of these rates and durations has impacts on the traffic inflows to the freeway system, which in turn results in a new set of measurements and new traffic estimates. This recursive procedure continues as the freeway traffic state varies dynamically.

1.5 Multi-sensor data fusion

1.5.1 Concepts, benefits and issues

Multi-sensor data fusion (MSDF) investigates efficient methods to automatically transform (detection, association, correlation, estimation, and combination) imperfect data from different sources into valuable information that facilitates making high-level decisions (White, 1991; Bostrom et al., 2007). It has been widely applied to a broad range of domains such as robotics, real-time tracking and image processing. Advantages of multi-sensor data fusion have been summarized to include robust operational performance, extended temporal and spatial coverage, increased confidence, reduced ambiguity, enhanced spatial resolution, and improved system reliability (Hall & McMullen, 2004). It is obvious that more data means potential of more useful information. Fundamental issues in building a data fusion system for a particular problem involve aspects such as the following (Hall & Llinas, 2001):

a) assumptions and conditions;

b) appropriate architecture to address where the streams of data should be fused;
c) appropriate algorithms for the problem;
d) techniques that can optimize the fusion process dynamically;
e) raw sensing data processing procedure;
f) influence of the sensing techniques; and
g) accuracy of the data fusion system.

These seven issues are typical in designing any data fusion systems. Particularly, in the traffic state estimation problem, the first two issues are about the data fusion architecture design, the next two issues relate to the data fusion algorithm design and may involve system dynamics modeling, e) and f) address how to process multi-sensor measurements, and the last one needs to be answered with respect to the performance evaluation. To sum up, we need to address the aforementioned seven questions to design the architecture, select fusion algorithm (and system dynamics if needed), prepare the measurement, and define system performance goals and evaluation criteria.

Given the growing amount of both Eulerian and Lagrangian sensing data, fusing traffic data from multiple sensors for traffic state estimation would be very beneficial for traffic management. The designed data fusion system should have the advantages gained in multi-sensor data fusion, and is expected to improve the estimation accuracy.

In the rest of this section we present some preliminary knowledge regarding measurement, state space modeling, and algorithm, respectively. Spatial-temporal alignment of measurements is required when multiple sensors report from a broad temporal and spatial range. State space formulation is required in the development of a model-based data fusion system. Bayesian method is the theoretical background for the traffic state estimation.
1.5.2 Spatial and temporal alignments

To realize a computer-based estimation process, both the spatial and temporal dimensions need to be discretized. Data alignment refers to the conversion of all sensor observations into a common spatial-temporal coordinate system that is used by the estimator. This is to make field observations compatible inputs to the developed fusion algorithms.

![Spatial Alignment](image)

(a) Eulerian coordinates  
(b) Lagrangian coordinates

FIGURE 1.3 Spatial Alignment

1.5.2.1 Spatial alignment

For a specific time point, multiple sensors may report observations from different locations along a freeway stretch. The spatial positions of these streams of data need to be transformed to the common spatial coordinate used by the estimation system. In an Eulerian coordinate system, traffic flows are usually modeled on discretized freeway segments, and the spatial alignment task thus becomes mapping the observation positions
to the discrete segments. For example, in Figure 1.3-a, the freeway is discretized into three cells. The loop detector represented as the green square consistently measures the traffic condition of segment 1 at all times. However, the probe-vehicle sensor represented as the blue vehicle measures the traffic condition of segment 2 at time step $k_1$, and then measures the traffic condition of segment 3 at time step $k_2$. Therefore, the probe-vehicle measurements need to be dynamically aligned in terms of space. In a Lagrangian coordinate system, traffic flows are approximated by discrete vehicle groups, and thus observations need to be mapped to indexed vehicle groups. For example, in Figure 1.3-b, the probe-vehicle sensor consistently reports measurements for vehicle group 3 as time goes by (assuming a single lane; however, if over-taking exists it could belong to different groups). However, the loop detector measurements need to be aligned to different vehicle groups at different time steps. To sum up, in the Eulerian coordinates, the Lagrangian data needs to be carefully aligned to different freeway segments, while in the Lagrangian coordinates, the Eulerian data needs to be carefully aligned to different vehicle groups.

1.5.2.2 Temporal alignment

Time alignment is to convert local-time data into a unified time format. In traffic state estimation, this usually means to deal with different reporting rates coming from different traffic sensors, and how they are mapped into the temporal coordinate of the estimation system. Assume only two types of sensors are available, referred to as sensor A and sensor B. Streams of data coming from A and B can be synchronized (Figure 1.4-a) or asynchronous (Figure 1.4-b). For the synchronized case, we can process multi-sensor data as a batch. However, if they are asynchronous, which is common in the multi-sensor case,
they need to be aligned to a universal discrete temporal coordinate, typically the time steps used by the estimation system.

(a) Synchronized

(b) Asynchronized

FIGURE 1.4 Temporal Arrival Patterns

1.5.3 State space modeling

Model-based estimation algorithm requires modeling the dynamic traffic flow system and the measurement system in the state-space format. State space modeling is a technique used in control theory and systems to mathematically model a physical system as a set of first-order differential equations relating input variables to output variables. These variables are expressed as vectors. In traffic estimation, one is interested in the evolution of the system state vector over time (\( \theta \)). To facilitate computation, the continuous time space is first discretized into equal time intervals with length \( \Delta t \), indexed by \( k \). All the time-dependent variables indexed by \( k \) are evaluated at the end of time interval \( k \). Thus, the variable of interest becomes \( \theta_k \), defined as the system state at the end of time...
step $k$. The state vector includes sufficient information to describe the dynamic system. Its evolving behavior is mathematically characterized by the process function $f(\cdot)$ in the difference equation below (Equation 1.1), also referred to as the system dynamics or the system/process model.

$$\theta_{k+1} = f(\theta_k, u_{k+1}) \quad (1.1)$$

Function $f(\cdot)$ can be either linear or nonlinear. It can be adapted from a macroscopic traffic flow model, and provides the a priori estimate of the traffic state at current time step given the traffic state in the immediate previous time step.

The system noise is represented as $u_{k+1}$. Sources of the noise to the dynamic system equation include, but not limited to, (a) the imperfection of the dynamic traffic flow model as a representation of the real world traffic flow system; (b) the random fluctuation of the dynamic traffic state; (c) influential factors such as weather or traffic control methods that are not reflected in the system model.

Equation 1.2 is the measurement equation. In traffic state estimation, function $h(\cdot)$ relates the traffic state ($\theta_{k+1}$), some of which may be hard to observe, to some traffic flow attributes ($z_{k+1}$) that can be directly measured with traffic sensors. It can also be either linear or nonlinear. The measurement noise is represented as $\omega_{k+1}$. Examples of measurement noise sources include sensor failures and the limitations of sensing technologies. Typically, both the system and measurement noise sequences are assumed to be white and mutually independent with known probability density functions.

$$z_{k+1} = h(\theta_{k+1}, \omega_{k+1}) \quad (1.2)$$
Once we have the state-space model and the traffic measurements ready we can apply various model-based estimation approaches. From the Bayesian perspective, the system dynamics and the measurement model are indeed the knowledge of state transitional probability and the likelihood.

1.5.4 Bayesian fusion

The Bayesian fusion approach is a popular method for data fusion. The widely-applied Kalman filter (KF) and the extended Kalman filter (EKF), as well as the particle filter (PF), all fall into this category. Since our link-based real-time traffic state estimator is developed on a variant of EKF, the background theory of Bayesian fusion is introduced below to help the understanding our estimation approach. The fundamental probability law for all Bayesian approaches is the Bayes’ theorem. By adding time-dependent characteristic to the random variables we obtain a theoretical Bayesian fusion approach which is able to recursively obtain optimal estimates of the current time-dependent state variables, in real time, given the measurements up to the current time step. In reality, this theoretical approach may not have an exact practical realization, and thus may require some sub-optimal approximation methods (e.g., EKF, PF). The theoretical background knowledge is summarized below, while practical methods are introduced later when needed. Good references for Bayesian estimation include (Chen, 2003; Ristic et al., 2004).

1.5.4.1 Bayesian Theory and Bayesian Inference

Bayesian inference is a statistical inference method to derive the posterior probability density function (pdf) as a consequence of the prior pdf and the likelihood using the well-known Bayesian theorem. It computes the posterior density as Equation 1.3.
Equation 1.3 states that the posterior pdf \( p(\theta | z) \) is determined by three terms: (a) the prior pdf \( p(\theta) \), (b) the measurement likelihood \( p(z | \theta) \), and (c) the normalization term \( p(z) \). The prior \( p(\theta) \) represents the prior knowledge, even possibly subjective assessment, of the unknown system state. It is independent of the current empirical evidence (measurements). The likelihood \( p(z | \theta) \) assesses the compatibility of the observed evidence with the prior knowledge of the system state. It is the key bond that relates the prior knowledge to the objective information. The model evidence \( p(z) \) provides the normalization constant that results in a probabilistic statement for each possible hypothesis of the system state. The resulting posterior pdf \( p(\theta | z) \), stands for the probability distribution of the system characteristics that we are interested in, which is now dependent of the empirical evidence provided by the measurements.

1.5.4.2 Bayesian fusion: recursive Bayesian estimation

The above discussion assumes a time-independent system. When a dynamic stochastic system is under investigation, the time-independent system state vector becomes a sequence of time-dependent vectors, and the Bayesian inference method is applied recursively. The goal is to achieve the best posterior knowledge given the up-to-now knowledge of measurements at each discrete time step. An important assumption is that the system state can be modeled as a first order Markov process and thus the measurements \( (z_k) \) are fully determined by the corresponding state \( (\theta_k) \). Let \( K = \{k = 1, ..., K\} \) denote the discretized time sequence, where \( t_k = t_0 + \Delta t \cdot k \) for some predefined \( t_0 \) and \( \Delta t \), let
\(\Theta_{k+1} = \{\theta_{k+1} : k \in K\}\) denote the corresponding state sequence, and let \(Z_{k+1} = \{z_{k+1} : k \in K\}\) represent the corresponding measurement sequence. The time-dependent Bayesian updates can be derived by Equation 1.4 (Ristic et al., 2004).

\[
p(\theta_{k+1} | Z_{k+1}) = \frac{p(z_{k+1} | \theta_{k+1}) \cdot p(\theta_{k+1} | Z_k)}{p(z_{k+1} | Z_k)}
\]  

(1.3)

The prior pdf \(p(\theta_{k+1} | Z_k)\) can be derived as the conditional probability of the transitional pdf \(p(\theta_{k+1} | \theta_k)\) given the past posterior \(p(\theta_k | Z_k)\), where \(p(\theta_{k+1} | \theta_k)\) is usually described as the process function \((f(\cdot))\) in a state space model. The likelihood \(p(z_{k+1} | \theta_{k+1})\) is usually described by the measurement function \((h(\cdot))\). The normalization term \(p(z_{k+1} | Z_k)\) can be derived as the conditional expectation of \(p(z_{k+1} | \theta_{k+1})\) given the prior density \(p(\theta_{k+1} | Z_k)\). In state estimation, we are usually interested in the expected value of the posterior, i.e., \(E[\theta_{k+1} | Z_{k+1}]\). In practice, estimation algorithms may be able to either solve for the optimal solution of \(E[\theta_{k+1} | Z_{k+1}]\), or approximate it, depending on the characteristics of the dynamic system of interest (see Section 2.3.1).

The general procedure of the recursive Bayesian estimation is illustrated in Figure 1.5. It typically includes two update stages. The first stage is the time update, where we use the system equation to obtain the a priori estimate of the state expectation, denoted as \(\theta_{k+1}^{-} = E[\theta_{k+1} | Z_k]\), and the second stage is the measurement update where the a priori estimate is updated to obtain the a posteriori estimate, denoted as \(\theta_{k+1}^{+} = E[\theta_{k+1} | Z_{k+1}]\), given the extra information from the measurements.
1.6 Problem objectives and research scope

The objective of this research is to develop a multi-sensor data fusion approach for real-time lane-based freeway traffic state estimation with heterogeneous measurements. Since the developed lane-based traffic estimator adopts a bi-level approach where a model-based link-level estimator is incorporated, research efforts have also been expanded to enhance the performance of the link-based traffic state estimation using the Lagrangian traffic flow model and multi-sensor measurements.

The dissertation research focuses on (a) urban freeway traffic, which means there is no control in the main stream of the traffic; (b) macroscopic traffic state, which means that we are interested in some aggregated characteristics of the traffic flows, such as the mean speed and mean density, without trying to describe the microscopic traffic behaviors such as lane changing or over-taking; (c) multi-lane estimation, which means the current homogeneity assumption of different lanes is relaxed; (d) real-time estimation, which means the algorithm should be computational efficient and should not retrieve a large amount of historical data; and (e) heterogeneous multi-sensor data, which means both Eulerian and Lagrangian measurements are available.
1.7 Outline of dissertation and summary of contributions

The outline of this dissertation can be summarized using the flow chart of Figure 1.6. Chapter 2 presents an up-to-date literature review on the freeway traffic state estimation. The essential body of work addresses the real-time multi-lane multi-sensor traffic state estimation problem using a bi-level data fusion system, which is described in Chapters 3-6. Here the five issues, (a) the data fusion architecture, (b) data fusion algorithms, (c) system dynamics, (d) measurement data, and (e) system performance, as discussed in Section 1.5.1, are all addressed and investigated.

Specifically, Chapter 3 defines the architecture and framework of the proposed multi-sensor data fusion system, which includes a data-driven estimator for lane-level estimation and a model-based estimator for link-level estimation.

Chapter 4 develops a second-order formulation of the macroscopic Lagrangian traffic flow model to incorporate drivers’ anticipation and reaction delays which are not reflected in current first-order macroscopic Lagrangian model. Compared with the first-order model, the second-order model is able to reproduce stop-and-go patterns, and is more effective when the traffic condition goes to extreme.

Chapter 5 presents a multi-sensor extended Kalman filter (MEKF) algorithm to incorporate heterogeneous measurements for the link-level estimation, utilizing a second-order Lagrangian traffic flow model. Performance evaluation shows that the MEKF-based estimator enhances the estimation accuracy by combining multiple-sensor measurements, compared with using only loop-detector measurements or only probe-vehicle measurements.
Chapter 6 presents a detailed design of the lane-based traffic state estimation method. The developed data-driven approach bypasses modeling the dynamics of lane-level traffic flows, and enables the system to discriminate lane state differences by combining lane-level Lagrangian and Eulerian measurements. Performance evaluation
shows that the lane-level estimation using the proposed approach outperforms current link-level estimation.

Chapter 7 summarizes the major conclusions of this research, and identifies some potential future research topics.

Main contributions of this research include:

- Thorough literature review on freeway traffic estimation research. (Chapter 2)
- Development of a bi-level multi-sensor data fusion approach for real-time lane-based traffic state estimation which can effectively utilize heterogeneous measurements. (Chapter 2)
- Development of a second-order Lagrangian traffic flow model. (Chapter 4)
- Evaluation of the second-order Lagrangian traffic flow model using both simulation-based data and real traffic data. (Chapter 4)
- Development of a multi-sensor extended Kalman filter (MEKF) algorithm for utilizing heterogeneous measurements in a model-based approach. (Chapter 5)
- Development of a MEKF-based real-time link-level traffic state estimator which adopts the second-order Lagrangian traffic flow model. (Chapter 5)
- Evaluation of the MEKF-based link-level estimator using real traffic data. (Chapter 5)
- Development of a data-driven algorithm for real-time lane-based traffic state estimation that uses heterogeneous measurements. (Chapter 6)
- Evaluation of the developed lane-based estimator, using both simulation data and real traffic data. (Chapter 6)
- Identified potential directions for future research. (Chapter 7)
CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

Traffic data collection and state estimation approaches have been used to prepare inputs for advanced ITS applications. Examples range from online applications such as ramp metering (Hegyi et al., 2005; Sun et al., 2003; Papageorgiou et al., 1990), travel time prediction (van Lint, 2008; Chien and Kuchipudi, 2003; Ygnace et al., 2000) and incident detection (Karim and Adeli, 2002; Sethi et al., 1995; Westerman et al., 1996), to offline applications such as transportation strategic planning (Faouzi et al., 2011) and performance evaluation (Abdel-Aty et al., 2006; Skabardonis et al., 1995). Noting its significance, many research efforts have been made on developing accurate and reliable traffic state estimators.

Empirical traffic measurements can be categorized as Eulerian measurements and Lagrangian measurements. Traditional traffic state estimation approaches are based merely on aggregated loop detector data (Wang and Papageorgiou, 2005; Muñoz et al., 2003; Sun et al., 2003). As developing sensing technologies provide a wider range of sensors, research attention in the field has been drawn to fusing Lagrangian data (Yuan et al., 2012; Herrera and Bayen, 2010; Work et al., 2008), and multi-sensor data (Treiber et al., 2011; Anand et al., 2011; Cheu et al., 2001). Section 2.2 discusses empirical traffic measurements from multiple data sources.

Data fusion for traffic state estimation can be designed using a model-based approach, where the traffic flow propagation is described by a traffic flow model, or a data-driven approach, which heavily relies on the interpolation of empirical data. Accordingly,
model-based data fusion algorithms such as the extended Kalman filter (Wang and Papageorgiou, 2005; Yuan et al., 2012) or data-driven algorithms such as the adaptive smoothing filter (Treiber and Helbing, 2002; van Lint and Hoogendoorn, 2010), are adopted, respectively. Section 2.3 presents a literature review on data fusion algorithms with respect to model-based and data-driven approaches.

Basic components for a model-based estimator are the state space model (i.e., the system dynamics and the measurement equation), the empirical data, and a model-based data fusion algorithm. As discussed in Section 1.4.3, a macroscopic traffic flow model defines the system dynamics and can be categorized as Eulerian models or Lagrangian models. Macroscopic traffic flow models can also be classified as first-order or second-order models depending on how many terms are retained in developing the partial differential equation (PDE). Literature on macroscopic traffic flow models is elaborated in Section 2.4.

Based on the literature review of the state-of-the-art on real-time traffic state estimation, Section 2.5 sets the direction and challenges for this research. Section 2.6 summarizes the chapter.

2.2 Measurements from different sensors

Eulerian measurements describe the dynamic traffic conditions at fixed spatial locations through aggregation, such as aggregated traffic density (number of vehicles per kilometers), average traffic flow (number of vehicles per minute), and mean space speed. Lagrangian measurements describe the traffic conditions of a dynamic system moving along with the traffic stream. Roughly speaking, using Eulerian measurements in an Eulerian estimator or using Lagrangian measurements in a Lagrangian estimator do not
need complex data transformations (Wang and Papageorgiou, 2005; Yuan et al., 2012). On the other hand, using Lagrangian measurements in an Eulerian sensor or using Eulerian measurements in a Lagrangian estimator involves some efforts on relating observed data to the system state (Yuan et al., 2012; Herrera and Bayen, 2010). A multi-sensor traffic data set can be one of the three situations, including homogeneous Eulerian data, homogeneous Lagrangian data, and heterogeneous Eulerian and Lagrangian data.

Traditionally, traffic state estimation is based on stationary loop detector data. Wang and Papageorgiou (2005) proposed an extended Kalman filter based approach to jointly estimate critical parameters and traffic state using loop detector data. Different configurations of detectors were tested; based on their analysis they concluded that limited loop sensors could still lead to satisfactory estimation accuracy if they were appropriately located. Muñoz et al. (2003), Sun et al. (2003), Mihaylova and Boel (2004), Hegyi et al. (2006), Mihaylova et al. (2007), Tampère and Immers (2007), Wang et al. (2007), and Wang et al. (2008) all used the stationary loop detector data. Validation results all show satisfactory level of accuracy. However, this type of data is limited in coverage due to predefined sensor locations. Additionally, in all of these applications, the loop detector data are aggregated across all lanes to obtain the link-level measurements, as all of them are link-based estimators. Because of lateral aggregation, lane-level information is smoothed out.

Recently, research on incorporating probe-vehicle measurements in traffic state estimation or related topics has drawn some attention, primarily due to the greater spatial coverage and higher accuracy of probe data. Zhou and Mahmassani (2006) proposed a dynamic origin-destination traffic demand estimator using simulated automated vehicle
identification (AVI) data. Work et al. (2008) presented an ensemble Kalman filter based traffic state estimator using synthetic GPS data. They conducted field testing with Nokia N95 mobile device, and concluded that the observations can achieve lane-level position accuracy with a mean velocity error of 3 mph. Sohn and Hwang (2008) used simulated cellular phone data for vehicle passing time estimation using a space-based method. Herrera and Bayen (2010) used GPS data and adopted a Newtonian relaxation approach, and validated the estimator with real traffic data. Yuan et al. (2012) presented a Lagrangian estimator and evaluated it using synthetic probe-vehicle data. Multiple penetration rates have been examined in most of the above studies. The results generally show higher level of effectiveness for higher rate. It also has been shown that a penetration rate as low as 2% may still show improved accuracy (Herrera and Bayen, 2010). Krause et al. (2008) found that optimized selection of observations can significantly reduce the number of queries of sensors needed to achieve a specified level of accuracy. Meanwhile, real-time traffic information providers such as Google and INRIX are also using probe-vehicles to collect traffic information (McClendon 2013; INRIX 2015).

There are also some studies on utilizing multi-sensor measurements. Nanthawichit et al. (2003) studied combing simulated probe vehicle data and loop detector data for estimating traffic state and travel times using EKF. All available measurements are processed in a unified manner, and they show that by utilizing multi-sensor data the estimation accuracy is improved. Chu et al. (2005) designed an adaptive Kalman filter based estimator for travel time estimation where both simulated loop detector measurements and probe-vehicle measurements are first transformed into travel time measurements. Deng et al. (2013) and Zhou et al. (2011) developed a data fusion approach
for travel time estimation and prediction by fusing probe vehicle data such as AVI and GPS, vehicle trajectory data from video images and loop detector data. This approach has been evaluated using simulation-based experiments.

Data driven approaches have also been used to combine multi-sensor data. Treiber et al. (2011) presented an approach purely based on measurements for traffic reconstruction, where the inductive loop detector and the probe-vehicle sensor are assumed to be available. van Lint and Hoogendoorn (2010) developed a multi-sensor spatial-temporal filtering algorithm for traffic state reconstruction, which was validated using simulated loop detector data, probe-vehicle data, and AVI data. Bachmann et al. (2013) presented a simulation-based comparative evaluation of multi-sensor data fusion methods for freeway traffic speed estimation.

In summary, it may be concluded that most multi-sensor data fusion techniques are better than single sensor approaches, and that the data fusion technique, the penetration rate of probe vehicles, and sensor locations all influence the accuracy of the estimations. Therefore, we conclude that (a) these methods are all designed for homogeneous freeway link-level traffic estimation/reconstruction; (b) most of the methods are validated using simulated or synthetic measurements, especially the Lagrangian measurements; and (c) multi-sensor data fusion for traffic state estimation has not been sufficiently studied and point to an important area of research.

2.3 Data fusion algorithms

Data fusion algorithms are used for filtering empirical data to estimate hard-to-observe system state, which can be classified as model-based algorithms or data-driven algorithms. Model-based fusion algorithms incorporate specific traffic flow models, while
data-driven fusion algorithms base their accuracy mainly on the interpolation effectiveness of measurements, using underlying statistical models.

2.3.1 Model-based fusion algorithms

Model-based fusion algorithms have been widely applied in areas such as system control, chemistry, satellite navigation and so forth. They are essentially Bayesian fusion approaches (refer to Section 1.4.5). The assumptions about the model and noise feature have an influential role in determining which type of filtering algorithms can be applied and whether an optimal solution can be achieved. Recall that the goal in Bayesian fusion is to obtain the estimate of $E[\theta_{k+1} | Z_{k+1}]$. If the system is linear with Gaussian white noises, the Kalman filter (KF) (Kalman et al., 1960) approach can be adopted to achieve an optimal estimate of $E[\theta_{k+1} | Z_{k+1}]$, in a sense that no other filter can ever do better in this environment. However, the assumptions are too strong to be applied to many real world scenarios, where systems are commonly nonlinear or the noise is non-Gaussian. Variants of Kalman filter have been developed to approximate $E[\theta_{k+1} | Z_{k+1}]$.

When the system is nonlinear but still with white Gaussian noise, the extended Kalman filter (EKF) (Jazwinsky, 1970) that uses linear approximations can be used to obtain a sub-optimal solution. An additional assumption of the model is that it is continuously differentiable to enable relinearization with a first-order Taylor series expansion. The iterative extended Kalman filter (IEKF) (Jazwinsky, 1970) is based on EKF but reiterates the measurement update stage several times to reduce the estimation discrepancy in EKF (see Section 5.2.2 for details). Other variants of Kalman filters are the unscented Kalman filter (UKF) (Julier and Uhlmann, 1997; Wan and van Der Merwe, 2000;
Ristic et al., 2004), ensemble Kalman filter (EnKF) (Evensen, 1994, 2003), and mixture Kalman filter (Chen and Liu, 2000). Particle filter (PF) (Gordon et al., 1993) is another alternative sub-optimal filter to the above nonlinear filters. Most of these online estimation algorithms use Monte Carlo based computation techniques instead of the relinearization in EKF and thus are capable of dealing with more complicated situations such as non-differentiable state space model, or even non-Gaussian noise.

Specifically, in the traffic state estimation area, Anand et al. (2011) used KF to fuse multi-sensor data. Wang and Papageorgiou (2005) presented an EKF-based framework for a joint estimation of the system state, boundary variables, and critical model parameters at the same time. Traffic density and mean speed of the discretized freeway segments are defined as system state, while traffic flow and mean speed are defined as measurement variables. The second-order METANET traffic flow model (Papageorgiou et al., 1989) was used as the system model. A case study and the adaptive capabilities were investigated for this approach (Wang et al., 2007, 2008). EKF has been widely adopted due to its satisfactory performance and straightforward methodology (Nanthawichit et al., 2003; Tampère and Immers, 2007; Schreiter et al., 2010a; Yuan et al., 2012). Sun et al. (2003, 2004) developed a mixture Kalman filter for traffic state estimation based on a first order traffic model. Hegyi et al. (2006) showed that, based on the METANET model, UKF is slightly better or equal to EKF in terms of estimation performance, and the joint filter is better than the dual filter. Fewer detectors result in larger state estimation errors but have little effect on the parameter estimation error. Work et al. (2008) applied EnKF to highway traffic estimation using GPS enabled mobile devices. Mihaylova et al. (2007) proposed a particle filter application for traffic state estimation. They compared the performance of PF
and UKF, and concluded that PF out-performs UKF. Several similar Eulerian estimators using the PF algorithm have been studied by Mihaylova and Boel (2004), and Mihaylova et al. (2012). In addition to the recursive Bayesian estimation based methods, Herrera and Bayen (2010) developed a Newtonian relaxation based estimation approach to incorporate Lagrangian measurements.

From the above discussions we may conclude that all these model-based traffic state estimators focus on link-based descriptors. The model-based approaches reported have not been applied to lane-based traffic state descriptors because lane-based traffic flow theory has not been advanced sufficiently. The lane changing and over-taking behaviors can be analogous to adding numerous inlets and outlets at dynamic locations in a single lane freeway stretch. These dynamic perturbations can make the flow system hard to model and predict.

2.3.2 Data-driven fusion algorithms

In contrast to the fact that model-based methods are widely adopted to online traffic state estimation, data-driven methods are more common in manipulating data in an offline manner using archived traffic data. Reconstructing traffic state dynamics has seen a successful application. Treiber and Helbing (2002) developed a data-driven method to reconstruct spatial-temporal traffic dynamics from stationary loop detector data. This adaptive smoothing method (ASM) does not specifically adopt any macroscopic traffic flow model but uses weighted smoothing of observations. Two sets of weights need to be computed. The first set is obtained through an adaptive smoothing filter using an exponential kernel function, evaluating the information contribution of nearby observations. This procedure is then repeated for the assumed free flow condition and
congested condition. The second set of weights is calculated through a nonlinear adaptive weight function to evaluate the level of trustworthiness in free flow condition and congested condition. This novel method has led to several improved ASM-based methods. van Lint and Hoogendoorn (2010) argued that for the data that cannot be straightforwardly spatially or temporally aligned, using simpler data fusion methods is a sensible approach. In their research, a generalized ASM filter that is able to fuse multiple data sources was presented using the similar method to that proposed by Treiber and Helbing (2002) but the assumption of fixed sensor location and time interval has been relaxed to combine heterogeneous measurements, especially travel time data from AVI and loop detector data. A third set of weights was used to include the reliability of different sensing techniques. This method was experimentally validated to reconstruct traffic dynamics. It shows satisfactory and robust performance with coarse (noisy) data and with data having up to 50% randomly missing records. Treiber et al. (2011) presented another generalized ASM filter to combine loop detector data and probe-vehicle measurements for the same purpose. Their method is capable of reconstructing traffic flow phenomena such as transitions between free and congested traffic conditions and stop-and-go waves. Such ASM-based data-driven methods for traffic state reconstruction require using measurements from an area of a few kilometers and with 10 minutes or longer data streams (van Lint and Hoogendoorn, 2010) to estimate the traffic state of a particular freeway segment. This requires storing large amounts of time-series data and can result in time-consuming computations, rendering it not suitable for real-time estimation. Besides, unlike offline traffic state reconstruction, future data is obviously not be available for real-time estimation. Some efforts have been made on implementing those reconstructors for online estimation;
Schreiter et al. (2010b) developed two fast implementations of ASM using cross-correlation and fast Fourier transform, both of which show much faster computational speeds.

Other popular data-driven methods for ITS data fusion for freeway traffic state estimation and prediction include artificial neural networks (ANN) (Nelson and Palacharla, 1993; Palacharla and Nelson, 1999; Van Lint et al., 2005), evidence theory (El Faouzi and Lefevre, 2006; Kong et al., 2009), and so forth. It is noted that evidence theory is capable to classify different traffic congestion states instead of to quantify traffic density or speeds.

In summary, the current data-driven methods have the following limitations: (a) these methods do not differentiate heterogeneous traffic conditions among different lanes; (b) most of them are not suitable for online implementation due to intensive computational requirements as well as the need for archiving a large amount of historical data; (c) heavy dependence on empirical data may also make estimators prone to suffer data insufficiency problems; and (d) they do not incorporate any specific traffic flow model and thus cannot capture any enfolding flow phenomenon unless it is already occurred and observed by some sensors (van Lint and Hoogendoorn, 2010).

2.4 Macroscopic traffic flow modeling

Model-based data fusion includes a system dynamics model. It is based on a macroscopic traffic flow model describing the evolution of traffic characteristics over time and space. In traffic estimation applications, it is important to predict traffic conditions before they occur, which is the capability that pure empirical-measurement-based methods do not have. As mentioned in the introduction (Section 1.4.3), macroscopic traffic flow
models can be categorized as Eulerian models and Lagrangian models. This section reviews various macroscopic traffic flow models according to this categorization.

2.4.1 Eulerian macroscopic traffic flow models

Traditionally, macroscopic traffic flows have been modeled in the Eulerian coordinate system, where system state is defined as aggregated fixed-location variables such as the space mean speed and the traffic density. The Lighthill-Whitham and Richards (LWR) model (Lighthill and Whitham, 1955; Richards, 1956) is the most well-known first-order partial differential equation (PDE) model serving as the basis for most macroscopic traffic flow models. It is demonstrated to be capable of reproducing basic physical phenomena in dynamic traffic flows in the real world, such as the conservation of vehicles, the forming and dissolving of congestion at bottlenecks, and the anisotropic propagation of flow disturbances over time and space (Newell, 1993; Daganzo, 1994, 1995b; Lebacque, 1996). Discretized LWR models using the Godunov scheme (Godunov, 1959; Lebacque, 1996) have been used for numerical computations for state prediction (Anand et al., 2011; Herrera and Bayen, 2010).

Daganzo (1994, 1995a) presented the widely recognized Cell Transmission Model (CTM), which is also a discretized LWR model with the assumption that the fundamental relationship between traffic flow and density has a triangular shape. The freeway segment is discretized into a chain of fixed-length cells, assuming that the traffic condition within each cell is homogeneous. At each time step, the traffic state variables are propagated along the spatial cell-chain, given the state of the previous time step. CTM and modified CTM models are widely adopted in traffic state estimation algorithms. Gomes and Horowitz (2006) modified the CTM model using different merging rules, referred to as the
asymmetric CTM (ACTM). Other variants of CTM include the mode-switching type of models (Tampère and Immers, 2007; Muñoz et al., 2003; Sun et al., 2003), the stochastic CTM (Sumalee et al., 2011), and so forth.

Second-order Eulerian traffic flow models extend the first-order models by including an additional term accounting for empirical flow phenomena such as the reaction time delay, drivers’ anticipation, and capacity drop. The second-order term is overlooked in the first-order models and thus contributes to the system noise in the first order system model. In this more comprehensive way of modeling, the depicted traffic flow system is potentially more realistic and accurate than the first order models. Wang and Papageorgiou (2005), Hegyi et al. (2006), and Mihaylova et al. (2012) all chose the METANET model developed by Papageorgiou et al. (1989), which is an important extension of the Payne model (1971). A second-order extension of the CTM model has been proposed by Boel and Mihaylova (2006), which was adopted in Mihaylova et al. (2007) for traffic state estimation. Generally speaking, second-order models are demonstrated to improve the traffic state estimation performance but require more efforts in model calibration, estimation algorithmic procedures, and estimation stability maintenance.

However, there are also arguments questioning the reasonability of the high-order models. Daganzo (1995c) criticizes the possibility of negative flow and speed prediction in some extreme conditions. Others have noticed the underlying assumption that the desired speed distribution is a property of the road but not the drivers (Paveri-Fontana, 1975). Papageorgiou (1998) argues that the second-order models have their merits in remedying the false assumption of the static fundamental diagram, and that the negatively predicted flow/speed values occur only in extreme conditions. Other common blames include that
the second-order models have more parameters to be calibrated, and that the stability requirement becomes stronger in discrete simulation. Fortunately, these difficulties can be successfully overcome with careful model calibration efforts.

2.4.2 Lagrangian macroscopic traffic flow models

Originating from the flow dynamics theory, macroscopic flow movement can also be observed in the perspective of a moving particle in the flow, which defines the Lagrangian coordinate system. Leclercq et al. (2007) presented the formulation of the LWR PDE model in the Lagrangian coordinates. van Wageningen-Kessels et al. (2013) developed a framework for implementing the first-order Lagrangian traffic flow model for mixed-class and multi-class vehicles. In the proposed mixed-class model, the traffic flow is discretized into vehicle groups using the Godunov scheme. The major advantage of this alternative to the traditional Eulerian models is that the traffic characteristics are propagated only in the flow direction since drivers only react to the vehicles in front of them (i.e., the upwind method), which simplifies the discretization procedure. This feature simplifies the numerical computation for freeway links (without discontinuities). Simulation-based analysis shows satisfactory results in reproducing common traffic flow patterns (vehicle conservation, onset and discharge of congestion at bottlenecks, traffic anisotropy, etc.). Yuan et al. (2012) has developed a state estimator based on this first-order Lagrangian traffic model, showing that the Lagrangian model facilitates incorporation of Lagrangian measurements. To the author’s best knowledge, no second-order Lagrangian traffic flow models has been presented yet, and this is a very new area in traffic flow modeling.
2.5 Research direction and motivations

The discussion in previous sections lead to the conclusion that there are several insufficiently studied areas in the freeway traffic state estimation research. Specifically, they are:

- Lane based traffic state estimation using multiple traffic data sources,
- Second-order Lagrangian traffic flow modeling, and
- A general model-based multi-sensor data fusion method for link-level estimation

Current estimation approaches assume that traffic conditions laterally across multiple lanes are homogeneous at specified locations, which is intended to simplify the traffic flow patterns resulting from lane changing and over-taking behaviors. Without this assumption, the traffic flows on a multi-lane freeway is analogous to a highly unstable single lane freeway system with numerous dynamic on-ramps and off-ramps, and unknown ramp flows. In the light of these considerations, it is fair to add such a single lane assumption to a traffic flow model. On the other hand, as noted, data-driven approach has been neglected in dealing with the multi-lane estimation problem. Since the purely data-driven approaches also have their inherent limitations, a unified approach combining a model-based estimator and parallel data-driven estimators for fusing multi-sensor traffic data is promising for achieving accurate multi-lane freeway traffic state estimation. This is the ultimate goal of this research.

2.6 Summary

Multi-sensor data fusion for traffic state estimation is an emerging and promising research area with many important issues left unsolved. With more traffic data sources
available, it is the time to solve the unstudied lane-based freeway traffic state estimation problem. Model-based estimation approaches and data-driven approaches have their respective advantages and disadvantages, so do the Eulerian estimators (Eulerian traffic flow models) and the Lagrangian estimators (Lagrangian traffic models). A unified approach is needed to incorporate heterogeneous traffic data to obtain real-time lane-based traffic state estimation with satisfactory performance. Furthermore, enhancing link-based traffic state estimation from different aspects (i.e., data, model, and algorithm) also motivates this research.
3.1 Introduction

Knowledge of real-time traffic conditions is a critical input for operating modern intelligent transportation systems. Since it is not possible to collect the real-time traffic information from every point on a freeway, state estimation algorithms are adopted to make an intelligent “guess” of the hidden dynamic traffic conditions. A critical but still unsolved problem in the traffic state estimation area is to differentiate traffic state estimates for different lanes. Currently, almost all real-time estimation methods assume homogeneous conditions across multiple lanes where laterally aggregated traffic condition changes are only considered in the flow direction. This homogeneity assumption actually greatly helps in designing and implementing traffic state estimators. However, lane-condition differences can be large and averaging traffic characteristics over all lanes may introduce significant errors for representing traffic conditions in different lanes. This problem has been overlooked for a long while, probably because real-time estimators are mostly model-based, while lane-based traffic flow modeling is quite complex and useful traffic flow models have yet to be developed. The lane changing and over-taking behaviors can be analogous to adding numerous inlets and outlets at dynamic locations in a single lane freeway stretch. These dynamic perturbations can make the flow system highly unstable and hard to predict. In this research, we treat the lane-based traffic state estimation problem as an image processing problem. The link-based traffic state at each time step can be viewed as a low-resolution image while the lane-based estimate can be viewed as a high-
resolution image. The idea is to recursively estimate the high-resolution traffic state given low-resolution estimate and additional lane-based measurements. Although we got the inspiration from the image processing, it turns out that similar data-driven approaches have already been used in the transportation research area for the offline link-based traffic reconstructions (Treiber and Helbing, 2002; van Lint and Hoogendoorn, 2010; Treiber et al., 2011). In contrast, our approach distinguishes itself by providing online lane-based estimation of traffic state.

This chapter introduces the architecture of the data fusion approach for real-time lane-based traffic state estimation using heterogeneous traffic measurements. In the presented bi-level architecture, the link-level estimator incorporates a model-based filter and the lane-level estimator adopts multiple parallel lane-based smoothing filters. This chapter provides an overview of this new data fusion scheme and helps to understand the three chapters that follow.

3.2 Bi-level data fusion

3.2.1 System architecture

A bi-level architecture has been developed to recursively estimate the high-resolution real-time lane-based traffic characteristics. It consists of two sequential filtering processes at each iteration, represented as the link-level process and the lane-level process (Figure 3.1). The link-level process adopts a model-based real time traffic state estimator while the lane-level process encompasses multiple parallel smoothing filters for all lanes. In this chapter, we refer to the link-level estimates as the prior estimates, or the low-resolution representation of the lane-level traffic state; we refer to the lane-level estimates as the posterior estimates, or the high-resolution representation. The former do not take
account lane-specific measurements while the latter do.

As illustrated in Figure 3.1, the link-level estimator produces the aggregated estimate and inputs it as the prior to each of the lane-level estimators. The parallel lane-based estimators then fuse multiple inputs to yield the posterior estimates. The inputs for a lane-based estimator include (a) the heterogeneous lane-level measurements from different type of sensors \( (\mathbf{z}_k^m, m \in \mathbf{M}) \), see Section 6.2.2 for notation explanations), (b) the prior
estimate \( \theta^k \), and (c) the posterior estimate from the past \( \theta^*_{(k-1,l)}, l \in L \). Each lane-based filter yields new posterior estimates for a particular lane. This bi-level process is repeated recursively to estimate real-time traffic characteristics. This architecture intends to combine the model-based approach and the data-driven approach, to keep the computational cost low for online implementation while producing the high-resolution estimates.

3.2.2 Estimation resolution

Figure 3.2 depicts the estimation resolution difference. Figure 3.2-a is a scenario for low resolution estimation, which averages the traffic conditions across all lanes within each discretized freeway segment. The link-level Eulerian estimators adopt this spatial discretization method. Figure 3.2-b gives a corresponding example for high resolution estimation, admitting that the traffic conditions on different lanes could be different, even within the same freeway segment. This spatial discretization method is used in the lane-level estimation. More of the spatial-temporal discretization is discussed in Section 6.2.1.

FIGURE 3.2 Spatial Discretization (a) low resolution; (b) high resolution.
3.2.3 Link level: a model-based fusion method

The link-level estimator could be any valid model-based real-time traffic state estimator for freeway speeds. The link-level estimation results need be transformed into Eulerian format as inputs to the lane-level estimation. Traffic conditions are assumed to be homogeneous within each fixed-length segment. Chapter 5 presents a new link-level estimator for multi-sensor data.

3.2.4 Lane level: a data-driven fusion method

The lane-level estimator adopts parallel spatial-temporal smoothing filters. Detailed description of the methodology is elaborated in Chapter 6.

3.3 Summary

This chapter introduces the architecture of our data fusion approach for real-time lane-based traffic state estimation. A recursive bi-level procedure is presented, which combines a model-based approach and a data-driven approach. At the end of each iteration, we obtain the lane-level traffic state estimates for the current time step. This bi-level architecture defines the structure of this dissertation. Chapter 5 presents the details of the link-based estimator which adopts the second-order Lagrangian traffic flow model developed in Chapter 4. Chapter 6 gives the details of the lane-level estimator. Chapters 3-6 constitute the main contributions of this research.
CHAPTER 4
A SECOND ORDER LAGRANGIAN TRAFFIC FLOW MODEL

4.1 Introduction

A macroscopic traffic flow model describes the evolution of aggregated traffic characteristics over time and space, which is a basic and critical component for various model-based real-time traffic state estimators. Traditional macroscopic traffic flow models have been built in the Eulerian coordinates using Eulerian traffic characteristics, such as the traffic density, flow and speed. Recently, the Lagrangian traffic flow modeling using Lagrangian traffic characteristics, such as spacing and speed, has begun to attract attention. Its main advantages include the accuracy, the upwind method (see Section 4.3.2) which simplifies the computation (van Wageningen-Kessels et al., 2013), and its convenience in incorporating vehicle-based information (Yuan et al., 2012). However, up to now only a first-order Lagrangian model has been developed.

There are a few reasons to motivate the development of a second-order Lagrangian traffic flow model, which include:

a) Drivers’ behavioral reaction delay and anticipation should be reflected in the speed equation;

b) For practical considerations, such a model could be a good candidate to conveniently incorporate vehicle-based spacing and speed information, and to estimate speed and spacing jointly, in model-based traffic state estimation; and

c) For theoretical considerations, there are several existing Eulerian second-order models that serve to complement first-order Eulerian models, while no such a
model exists in the Lagrangian framework.

This chapter develops a second-order Lagrangian macroscopic traffic flow model for freeways. The idea originates from Payne’s second-order Eulerian model which was derived on the basis of car-following considerations (Papageorgiou, 1990). The developed model reflects the fact that a driver usually adjusts the speed based on the traffic condition ahead, and that a time delay exists as a driver reacts to changed traffic conditions. A dynamic speed equation is formulated to represent this behavior.

A lane-drop scenario was simulated to evaluate the performance of the second-order Lagrangian traffic flow model. Simulation-based comparison with the first-order Lagrangian model confirms the effectiveness and advantages of the developed second-order model. In addition, the Lagrangian model was also validated using the NGSIM-US101 real traffic data set. This experimental analysis shows that the second-order terms are effective mainly in extreme traffic conditions.

Previewing the rest of this chapter, we first provide a mathematical overview of the two different types of traffic flow model in Section 4.2. Details of the second-order Lagrangian model are presented in Section 4.3, followed by its performance evaluation in Sections 4.4 and 4.5, using a simulation-based experiment and US101 data, respectively. Finally, some conclusions are made in Section 4.6.

4.2 Macroscopic traffic flow models

A macroscopic traffic flow model generally includes (a) a conservation equation, and (b) a (static or dynamic) speed equation. This section reviews two general type of macroscopic traffic flow modeling approaches: the Eulerian and the Lagrangian approaches.
4.2.1 Eulerian traffic flow models

Traditional macroscopic traffic flow models are built in the Eulerian coordinate system. The Lighthill-Whitham and Richards (LWR) model (Lighthill and Whitham, 1955; Richards, 1956) formulates the traffic flow dynamics as a kinematic wave model described by the partial differential equations (PDE) in Equation 4.1.

\[
\frac{\partial q(x,t)}{\partial x} + \frac{\partial \rho(x,t)}{\partial t} = 0 \tag{4.1-a}
\]

\[
q(x,t) = Q(\rho(x,t)) \tag{4.1-b}
\]

Equation 4.1-a is the conservation equation, stating that the change in flow \(q\) over space \(x\) equals to the change of density \(\rho\) over time \(t\) in the \(x \times t\) space. Equation 4.1-b is the fundamental diagram describing the nonlinear relationship between density and flow. Almost all of the macroscopic models are developed from this LWR model. For example, the widely applied Cell Transmission Model (CTM) (Daganzo 1994, 1995a) is a discretized version of LWR with the assumption that the fundamental relationship between traffic flow and density is triangular. The freeway segment is discretized into a chain of fixed-length cells, assuming that the traffic conditions within each cell are homogeneous. At each time step the traffic state variables are propagated along the spatial “cell-chain”.

With specific formulation of the fundamental diagram and some numerical computation method, the LWR-type models can be implemented in discrete simulations. Much has been conducted in this area to demonstrate the capabilities and drawbacks of the LWR-type models. It has been demonstrated to be capable of reproducing basic dynamic phenomena observed in real-world traffic flows, such as the conservation of vehicles, the
forming and dissolving of congestion at bottlenecks, and the fact that the traffic condition is propagated over time and space differently under different prevailing traffic conditions (e.g., congested vs. free flow). However, one of the main drawbacks of the LWR-type models is that vehicles are assumed to be able to attain their desired speeds (represented by the fundamental diagram) instantaneously, implying infinite acceleration and deceleration. Additionally, the desired speeds are purely determined by the drivers’ current traffic conditions, without looking ahead. These issues and the resulted simulation inaccuracies lead researchers to test higher-order models.

The second-order models assume that before vehicles reach the equilibrium speed described by the fundamental diagram, there is a time delay for acceleration or deceleration. Meanwhile, the desired speeds are determined by the downstream traffic conditions. For example, Payne (1971) formulates the dynamic speed-density relationship as Equation 4.2.

\[ v(x, t + \tau) = V[\rho(x + \Delta x, t)] \]  

Equation 4.2 has the following assumptions:

a) Drivers adjust their speeds according to the downstream traffic conditions, that is, the speed \( v(x, t) \) depends on the downstream density \( \rho(x + \Delta x, t) \);

b) Drivers gradually accelerate or decelerate in reaction to traffic condition changes, that is, the desired (equilibrium) speed would be reached with a time delay (\( \tau \)); and

c) The dynamic speed relationship can be described by the Eulerian fundamental diagram with perturbations \( \tau \) and \( \Delta x \).

Indeed, this model was derived on the basis of car-following considerations. Papageorgiou (1989) extended the model to account for ramp flows.
4.2.2 Lagrangian traffic flow models

Originating from the flow dynamics theory, macroscopic traffic flow movement can also be observed in the perspective of moving vehicles in the flow, which defines the so-called Lagrangian coordinate system. Leclercq et al. (2007) presented the formulation of the LWR PDE model in Lagrangian coordinates:

\[
\frac{\partial s(n,t)}{\partial t} + \frac{\partial u(n,t)}{\partial n} = 0
\]  
(4.3-a)

\[
u(n,t) = U[s(n,t)]
\]  
(4.3-b)

Equation 4.3-a is the Lagrangian conservation equation. It states that the change of spacing \((s)\) over time equals to the change of the mean speed \((u)\) over the cumulative of number of vehicles \((n)\) in the \(n \times t\) space. This Lagrangian conservation equation can be derived from the Eulerian conservation equation by applying \(s = \frac{1}{\rho}\), \(q = \frac{\partial n}{\partial t}\) and \(\rho = -\frac{\partial n}{\partial x}\) (Leclercq et al., 2007). Equation 4.3-b is the Lagrangian fundamental diagram, a function relating the spacing to speed, which can be obtained by applying \(s = \frac{1}{\rho}\) to the Eulerian fundamental diagram. An example of the Lagrangian fundamental diagram is the Smulders’ fundamental diagram (Smulders, 1989) in Lagrangian form (refer to Figure 1.1):

\[
u = U(s) = \begin{cases} 
u_f - s_{cr} \cdot \frac{\nu_f - \nu_{cr}}{s}, & \text{if } s \geq s_{cr} \\ \nu_{cr} - \frac{s - s_{jam}}{s_{cr} - s_{jam}}, & \text{otherwise} \end{cases}
\]  
(4.4)

where \(\nu_f\), \(\nu_{cr}\), \(s_{cr}\), and \(s_{jam}\) denote the free flow speed, the critical speed, the critical density, and the jam density, respectively. Another example is an exponential-form
fundamental diagram (Papageorgiou et al., 1989) in Equation 4.5, where $a$ is a parameter needs calibration.

\[ u = U(s) = u_f \cdot \exp \left[ -\frac{1}{a} \left( \frac{s_{cr}}{s} \right)^a \right] \]  \hspace{1cm} (4.5)

Recently, van Wageningen-Kessels et al. (2013) developed a framework for implementing the first-order Lagrangian traffic flow model for mixed-class and multi-class vehicles. In the mixed-class model, the traffic flow is discretized into vehicle groups using the Godunov scheme. The advantages of this alternative to traditional Eulerian models can be summarized as follows. First, the traffic characteristics are propagated only in the flow direction since drivers only react to vehicles in front of them, resulting the upwind method (see Section 4.3.2) which simplifies the discretization and computation procedures. Second, the discrete simulation results show higher accuracy in many scenarios, compared with the Eulerian model. Third, the Lagrangian formulation helps incorporate vehicle-based measurements. However, the first-order Lagrangian model has the same drawbacks as its Eulerian counterpart, where it assumes that drivers determine their speeds based on the immediate neighborhood traffic conditions instead of looking ahead, and that vehicles can attain the desired speeds instantaneously. Since the Lagrangian modeling is a relatively new area, no second-order traffic flow model has been developed yet.

4.3 A second-order Lagrangian model

Applying the same idea as Payne’s (1971), we start our modeling by modifying Equation 4.3-b to consider the dynamic speed-spacing relationship:

\[ u(n, t + \delta_t) = U[s(n - \delta_n, t)] \]  \hspace{1cm} (4.6)

Equation 4.6 is based on the following three assumptions:
a) Speed is adjusted according to the traffic conditions ahead. As $n$ stands for the accumulated volume of vehicles, it decreases along the flow direction, and $(n - \delta_n)$ corresponds to a position downstream ($\delta_n > 0$);

b) The speed gradually changes based on finite acceleration/deceleration, and the equilibrium speed cannot be reached until $\delta_t$ time later; and

c) The dynamic speed relationship can be described by the spacing-speed fundamental diagram with perturbations $\delta_n$ and $\delta_t$.

4.3.1 Derivation of the dynamic speed PDE

Expanding the right side of Equation 4.6 using Taylor expansion, we obtain Equation 4.7.

$$
\frac{du(n,t)}{dt} + \delta_t \cdot \frac{dU(s(n,t))}{dn} - \delta_n \cdot \frac{dU(s(n,t))}{dn} = U(s(n,t)) - \delta_n \cdot \frac{\partial U(s(n,t))}{\partial s(n,t)} \cdot \frac{\partial s(n,t)}{\partial n} 
$$

(4.7)

We use the following assumptions that are similar to those in the Eulerian models. Assume $\delta_n = \chi \cdot \rho = \chi / s$, where $\chi$ is a constant parameter assumed to be around $\chi = 0.5$.

Assume that $\partial U(s(n,t))/\partial s(n,t)$ is positively constant. Let $\lambda = \chi \cdot \frac{\partial U(s(n,t))}{\partial s(n,t)}$, and the right-hand side of the Equation 4.7 becomes $U(s(n,t)) - \frac{\lambda}{s} \cdot \frac{\partial s(n,t)}{\partial n}$.

The term $\frac{du(n,t)}{dt}$ in Equation 4.7 is the acceleration term which could be written as $\frac{du(n(t),t)}{dt}$, leading to Equation 4.8.
\[
\frac{du(n(t),t)}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial n} \cdot \frac{\partial n}{\partial t}
\]  \hspace{1cm} (4.8)

Applying \( \partial n / \partial t = q = u / s \) and letting \( \tau = \delta \), we finally obtain the dynamic speed PDE given in Equation 4.9.

\[
\frac{\partial u(n,t)}{\partial t} = -\frac{\partial u(n,t)}{\partial n} \cdot \frac{u(n,t)}{s(n,t)} + \frac{1}{\tau} \left[ U[s(n,t)] - u(n,t) - \frac{\lambda}{s} \cdot \frac{\partial s(n,t)}{\partial n} \right]
\]  \hspace{1cm} (4.9)

The conservation equation in Equation 4.3-a, a fundamental diagram in the form of Equation 4.3-b (e.g., Equation 4.4 or 4.5), and the dynamic speed equation (Equation 4.9) constitute the new continuous second-order Lagrangian traffic flow model.

4.3.2 Discrete second-order Lagrangian model

In Lagrangian coordinate system, the traffic flow is discretized into vehicle groups. Let \( j \) be the index for the vehicle group and \( j = 1 \) represents the vehicle group in the most downstream. Since the vehicles only react to the vehicles in front of them, the discretization in Lagrangian formulation uses the upwind method. It means that for calculating the spacing of a vehicle group \( j \), only the traffic conditions of the vehicle groups \( j \) and the front vehicle group \( j - 1 \), are involved. Let \( k \) be the index for the time steps, let \( n \Delta n \) denote the vehicle group size over all lanes, and let \( \eta_j \) be the vehicle group size per lane. Thus \( \eta_j = \Delta n / L_j \) and \( \partial n = -\eta_j \). Note that in the macroscopic level \( n \) is a continuous dimension, and thus \( \Delta n \) is continuous as well. The discrete model consists of the three equations in Equation 4.10. Equation 4.10-a is the discrete Lagrangian conservation equation, Equation 4.10-b is the discrete Lagrangian dynamic speed equation and Equation 4.10-c is the Lagrangian Smulders’ fundamental diagram, which could also be other valid forms.
\begin{align*}
s_j(k+1) & = s_j(k) + \frac{T}{\eta_j} \left[ u_{j-1}(k) - u_j(k) \right] \\
u_j(k+1) & = u_j(k) + \frac{T}{\tau} \left[ U(s_j(k)) - u_j(k) \right] \\
& \quad + \frac{T}{\eta_j} \cdot \frac{u_j(k) [ u_{j-1}(k) - u_j(k) ]}{s_j(k)} + \frac{T \lambda}{\tau \eta_j} \cdot \frac{s_{j-1}(k) - s_j(k)}{s_j(k)} \\
U(s_j(k)) & = \begin{cases} 
    u_j - s_{cr} \cdot \frac{u_j - u_{cr}}{s_j(k)}, & \text{if } s \geq s_{cr} \\
    u_{cr} \cdot \frac{s_j(k) - s_{jam}}{s_{cr} - s_{jam}}, & \text{otherwise}
\end{cases}
\end{align*}

The developed discrete dynamic speed equation (Equation 4.10-b) has four components on the right-hand side. The first component is the current speed, the second component represents the influence of the difference between the desired speed and the current speed, the third component represents the influence of the speed of the vehicle group in front, and the last component represents the influence of the spacing of the vehicle group in front. The last three components together state that the drivers’ desire to reach the equilibrium speed and the influences from the front vehicles result in the finite acceleration/deceleration (i.e., the gradual change of speed) from the current speed. Thus, this dynamic speed equation is more realistic given more representative driver behavior.

The model has two additional critical parameters: \( \tau \) and \( \lambda(\chi) \). \( \tau \) represents the reaction delay time, and \( \chi \) represents how much the current vehicle group is influenced by the vehicle density (spacing) ahead. Thus, with a larger \( \tau \) and a smaller \( \chi \), one may expect less sensitivity of the drivers to downstream conditions, and more oscillations in congestions. Such an expectation is observed in simulation in Section 4.4.
This discrete model can be easily transformed into a state space model, where the state variables are the spacing and speed of the vehicle groups. This formulation facilitates the model-based joint estimation of speed and spacing.

4.3.3 Network modeling to handle discontinuities

The model described by Equation 4.10 is a freeway link model, i.e., it does not take into considerations of the freeway discontinuities. Network modeling is required to account for the freeway discontinuities, such as the inflow boundary, the outflow boundary, the lane-drop and lane-increase scenarios, the merge (e.g., on-ramp flows) and diverge (off-ramp flows), and so forth (Figure 4.1). These discontinuities can be modeled as “nodes” where two links are connected. van Wageningen-Kessels et al. (2013) has developed a complete framework for modeling discontinuities in the Lagrangian coordinate system. However, from simulating these discontinuities, we modified some of constructs for representing the discontinuous flows; the differences are explained in the following subsections.

![Network Model Illustrating the Discontinuous Points (Nodes)](image)

FIGURE 4.1 Network Model Illustrating the Discontinuous Points (Nodes)
4.3.3.1 Boundary conditions

For the inflow boundary the “minimum supply-demand method” is applied (Equation 4.11) (Lebacque, 1996). Let $D_i$ stand for the demand of traffic flow on freeway segment $i$, $S_i$ the supply, and $q_{cr}$ the critical flow. Here the time step index $k$ is omitted for simplicity. The minimum supply-demand method for the inflow boundary is the same as the above Eulerian case except for that the freeway segment $i$ is in fact the available space in the most upstream, that is, the distance between the starting position of the freeway and the position of the last vehicle group that is already in the computation domain. The minimum inflow spacing is a parameter to be calibrated or specified, which influence how many vehicles can be accommodated given the available inflow node space. Segment $i-1$ is a dummy segment where the unsatisfied traffic demand queues. Once the calculated flow $q_{(i-1\rightarrow i)}$ is as large as the vehicle group size $\Delta n$, it is enters the computation domain.

$$q_{(i-1\rightarrow i)} = \min\{D_{i-1}, S_i\}$$

$$D_i = \begin{cases} q_i & \text{if free flow, } s_i > s_{cr} \\ q_{cr} & \text{o.w.} \end{cases} \quad S_i = \begin{cases} q_{cr} & \text{if free flow, } s_i > s_{cr} \\ q_i & \text{o.w.} \end{cases} \quad (4.11)$$

At the outflow boundary, a dummy vehicle group is assumed to represent the out-boundary condition. The first vehicle group within the computation domain follows the dummy vehicle group according to the upwind method until it leaves the computation domain and becomes the new dummy vehicle group.

4.3.3.2 Merge and diverge

In the existing network model (van Wageningen-Kessels, 2013), it has been
suggested that the vehicle group size be kept a constant, and thus the merge (e.g., on-ramps) and diverge (e.g., off-ramps) is conducted by “batching” vehicles. The whole vehicle group is inserted into or deleted from the mainstream traffic flow at once; in other words, if the queue for merging (diverging) is smaller than the predefined vehicle group size, then these vehicles need to keep waiting. However, in reality, the merging and diverging flows are continuous, and thus the aforementioned method may introduce errors. In addition, the numerical computation during discrete simulation of the traffic flow model is highly influenced by system disturbances (discontinuities), and it has been experienced in experimental-based analysis that insertion and deletion of a vehicle group may cause simulation instability. For example, when a vehicle group is inserted, it needs sufficient space in mainstream flow which reduces the available space for the immediate upstream vehicle group (i.e., the following vehicle group). The mean spacing of the immediate upstream vehicle group may suddenly change to a much smaller value which in turn may result in a much lower speed according to the fundamental diagram. In reality the immediate upstream vehicle group does reduce speed to accommodate the incoming vehicles from on-ramps, but the whole process is continuous and persistent. Sudden changes of the traffic conditions may result in stability problems in simulation. Similar reasoning can be applied to the diverging case.

In our network model, the vehicle group size is relaxed to be variable within certain predefined range. When a vehicle group is passing a merging node, its size can be increased if there is a merging queue and if the current vehicle group size is still within the preset limits. In other words, the minimum supply-demand rule is applied to determine how many vehicles are allowed to enter the mainstream. The demand can simply use the average
inflow rate as an estimate, and the supply is based on the difference between the current vehicle group size and the predefined maximal size. The variations of the vehicle group size need to be guaranteed to be within a range to maintain stable simulation; otherwise, the numerical computations may become unstable. For the diverge node, we apply the similar rule but to decrease the size of the vehicle group passing by the diverge node. This method greatly helps simplify the merging and diverging procedure during simulation.

4.3.3.3 Lane drop and lane increase

The lane-drop discontinuities is modeled as done by van Wageningen-Kessels et al. (2013). When a vehicle group approaches the merging area (Figure 4.2), it is propagated ahead in the flow direction using the old fundamental diagram (FD) without reducing the number of lanes. The traffic characteristics (spacing, speed) are then adjusted by assuming that the length of the occupied area is the same while the number of lanes is reduced (e.g., from three to two). The new spacing is calculated as $s_{j,new}^{k} = s_{j,old}^{k} \cdot \left(\frac{L^{new}}{L^{old}}\right)$, where $L^{old}$ and $L^{new}$ represent the number of lanes before the lane-drop and after the lane-drop, respectively. The lane-increase node model can be simulated using similar logic.

![FIGURE 4.2 Lane-Drop Node Model](image)
4.3.4 Discussion

For the discrete simulation of the first-order Lagrangian model, the Courant-Friedrichs-Lewy’s (CFL) condition described in Equation 4.12 ensures numerical computation stability.

\[
\frac{T}{\eta_{\text{min}}} \cdot \max_j \left| \frac{\partial U(s_j(k))}{\partial s_j} \right| \leq 1 
\]  \hspace{1cm} (4.12)

Indeed, it limits the maximum distance that a vehicle group can travel within one time step. The minimum size of vehicle groups per lane (\(\eta_{\text{min}}\)) given time step length \(T\) is obtained by letting CFL=1. However, the second-order model incorporates a dynamic speed equation, and thus the CFL number cannot be approximated using Equation 4.12. The condition is in fact the same that during one time-step, a vehicle group cannot cross two discretization boundaries in the numerical computation, to prevent unbounded accumulation errors. Experience from simulation-based analysis has found that the required minimum vehicle group size for the second-order model is larger than that for the first-order model, but still reasonable for a macroscopic level simulation.

Arguments for and against first-order and second-order Eulerian models have never stopped. As mathematical models are approximations of the physical world, they can hardly be exactly accurate. The same is true for the Lagrangian models. There is no simple conclusion for the comparison, as they both have their own advantages and disadvantages. We believe that the fundamental diagram is only a coarse approximation of the empirical evidence, and thus the presented second-order model is more sensible as it improves the first-order model by dynamically modeling the speed. Particularly, the speed property would be related to vehicle groups instead of a location on the road (see Section 2.4.1).
This makes the Lagrangian second-order model more reasonable than its Eulerian counterpart.

4.4 Experimental analysis: lane-drop scenario

This section presents the experimental analysis of the proposed second-order model using a lane-drop scenario shown in Figure 4.3. This experiment compares the two models to show the potential advantages of the second-order model.

4.4.1 Scenario description

A lane-drop scenario (Figure 4.3) is assumed to emulate the onset and discharge of congestion. The total length of the freeway stretch is 6 km without ramps. At \( x = 3 \) km the number of lanes drops from three to two. The initial condition is set to be low inflow. Later on, a surge occurs in the inflow demand, which forms congestion when it reaches the lane-drop area. The congestion was propagated backward until the demand surge ends and the inflow queue disappears, and then it discharges, leaving a triangular shape of low speed area in the speed map. The lane-drop node model described in Section 4.3.3.3 is adopted to realize it in computer-based simulation.

![FIGURE 4.3 Lane-Drop Scenario](image)

4.4.2 Experiment settings

The Smulders’ fundamental diagram is adopted, and the related parameters \( u_c, u_f \).
\( s_{cr} \) and \( s_{jam} \) are set to be 75 km/h, 120 km/h, 30 m/veh/lane, and 5 m/veh/lane. The simulation duration is 0.8 h to allow a full evolution of congestion. The demand surge occurs between 0.1 h and 0.3 h. The low-demand input flow is 1 veh/s. while the high-demand input flow is 2 veh/s. The time-step duration \( T \) is set to be 1 second, and the vehicle group size \( \Delta n \) is 4.6. For a first order model, the CFL condition requires that \( \Delta n > 2.5 \) with \( T = 1 \text{s} \). Both the first-order model and the second-order model are implemented in MATLAB. All these settings are the same for both models to make the outputs comparable.

The second-order model has two more parameters to calibrate, namely, \( \tau \) and \( \chi \) (or \( \lambda \)). Two nominal sets are tested, where Set-1 uses \( \tau = 1.05 \) and \( \chi = 0.55 \), and Set-2 uses \( \tau = 1.14 \) and \( \chi = 0.40 \). At this point, the parameter sets for the dynamic speed equation are chosen to ensure the simulation stability and yield reasonable (in the sense of traffic flow dynamics) outputs given other settings for both models. They are not optimized towards a real data set. It was observed that given other settings, the parameters \( \tau \) and \( \chi \) can only vary within a small range for simulation to remain stable. Thus one may expect that these parameters would not change greatly from the nominal data sets even when they are calibrated with a real data set.

4.4.3 Results and discussion

Figure 4.4 shows the speed maps for the first-order model, with the full evolution map in Figure 4.4-a and an amplified area shown in Figure 4.4-b; 4.5 and 4.6 show the results for the second-order model with different parameter sets, with the same area amplified in Figure 4.5-b and 4.6-b, respectively. The lane-drop location and the amplified
area are indicated in the figures. Initially, the demand volume is low such that the speed is close to the free flow speed, shown as the deep blue color. Between 360 s and 1080 s, the

(a) Full evolution of congestion

(b) Amplified area

FIGURE 4.4 Speed Maps using First-Order Model
demand volume has increased, resulting in a lower speed shown as light blue color before the lane-drop. As the plotting area is truncated at $x = 1000$ m (Figures 4.4-a, 4.5-a, and
(a) Full evolution of congestion

(b) Amplified area

FIGURE 4.6 Speed Maps using Second-Order Model with Parameter Set 2

4.6-a), this high demand period is slightly shifted to the right in the figures. The high inflow volume creates the congestion as it reaches the land-drop location (the bottleneck), and the
congestion propagates backward. When the inflow demand decreases and the inflow queue dissolves, congestion begins to discharge.

We can see from the figures that all of them are able to reproduce the onset and discharge of the congestion caused by the lane-drop bottleneck. However, these figures have observable differences. Figure 4.5 exhibits similar pattern to that of the first-order model but with some slight oscillations of speed during congestion, while Figure 4.6 shows more stop-and-go waves during the congestion. These stop-and-go waves are amplified towards the bottom of the congestion triangle; similar patterns have been found in real world data (Zielke et al., 2008).

van Wageningen-Kessels et al. (2013) has analyzed the stop-and-go waves that sometimes appear in the first-order Lagrangian model when the discretization resolution is low or CFL=1, and concluded that those waves are artifacts as they are the results of diffusion error, and should not be confused with the real-world stop-and-go waves. However, the second-order model naturally creates the stop-and-go waves in congestion by adjusting how much the speed at current time-step relies on the speed during the last time-step, how fast it can reach the equilibrium speed, and how sensitive it is to the condition of the leading vehicles. When the reaction delay is relatively long, and/or that the vehicles are more concerned about their immediate neighborhood traffic conditions rather than looking far ahead at downstream conditions (e.g., for parameter Set-2 where $\tau = 1.14$ and $\chi = 0.40$), the oscillation becomes more obvious. On the other hand, when the drivers reach their desired speeds faster and/or are paying more attention to look ahead at the downstream conditions (e.g., for parameter Set-1 where $\tau = 1.05$ and $\chi = 0.55$), the
speed map becomes smoother. The resulting differences from the two parameter sets indicate the flexibility of the proposed model.

It is worth noting that the magnitude of the time delay parameter $\tau$ is one order higher than the drivers’ reaction time in microscopic models. In fact, this $\tau$ is two orders higher than the drivers’ reaction time in the Eulerian case. Thus, this parameter $\tau$ should be treated as a macroscopic parameter, rather than microscopic.

4.5 Experimental analysis: US101 scenario

4.5.1 Background

The experimental analysis was carried out in MATLAB on a real traffic data set for a segment of the US Highway 101 (Hollywood Freeway) in Los Angeles, California, United States, from the Federal Highway Administration’s (FHWA) Next Generation Simulation (NGSIM) project. This data set is referred to as US101 data in this dissertation. The time frame starts at 7:50AM and ends at 8:35AM in the morning of June 15, 2005, totally 45 minutes. In the first 15 minutes, the freeway segment experiences a transition from free flow condition to congestion and stays congested for the rest 30 minutes. The initial 30 s period is not used in estimation because of partially missing data. The freeway segment under investigation (Figure 4.7) is approximately 640m (2100 feet), and has five main lanes including one HOV lane. It also has an on-ramp at location 176 m (578 feet) from start and an off-ramp at location 389 m (1276 feet) from start, having an auxiliary lane in between. The measurement data was first collected by video cameras which had full coverage of the freeway segment, and then transcribed to vehicle trajectory data, including information about vehicle identification number, time, position coordinates, lane number, speed, spacing, and so on. The time frequency is $1/10^{th}$ second. (NGSIM)
4.5.2 Experiment setup

4.5.2.1 Parameter selection and model calibration

Since the number of parameters is large and the model is nonlinear, it is hard to obtain the optimal solution; thus parameter selection and model calibration was conducted heuristically according to the RMSE criteria, and the final parameter set is listed in Table 4.1. Specifically, first the parameters are initialized according to expert judgments and related literature. Second, each parameter is calibrated in a sequential manner given other parameters fixed. The calibration for one parameter stops once the RMSE cannot improve any more, and then this parameter is set to be fixed and the calibration of the next parameter starts. The calibration also takes into consideration of simulation stability, which limits the range of parameters. It should be noted that this heuristic calibration method cannot guarantee an optimal parameter set; in addition, the performance criteria RMSE itself may not be a sufficient measure.

The parameters can be categorized into three types. The fundamental traffic flow parameters are critical parameters used in the Lagrangian fundamental diagram, including the free flow speed $u_f$, the critical speed $u_{cr}$, the critical spacing $s_{cr}$, and the jam spacing
\( s_{\text{jam}} \). They are initialized to be the same as that from the PeMS system (Herrera and Bayen, 2010), and then calibrated in simulation.

**TABLE 4.1 Parameter Settings for US101 Freeway Macroscopic Simulation**

<table>
<thead>
<tr>
<th>Fundamental Parameters</th>
<th>Simulation Parameters</th>
<th>Model Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free flow speed ( u_f ) (km/h)</td>
<td>( dN ) (veh)</td>
<td>( \tau )</td>
</tr>
<tr>
<td>Critical speed ( u_{cr} ) (km/h)</td>
<td>109</td>
<td>8.5</td>
</tr>
<tr>
<td>Critical spacing ( s_{cr} ) (m)</td>
<td>( dT ) (s)</td>
<td>1</td>
</tr>
<tr>
<td>Jam spacing ( s_{\text{jam}} ) (m)</td>
<td>On-ramp flow coefficient ( \alpha )</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Off-ramp flow coefficient ( \beta )</td>
<td>8.25</td>
</tr>
</tbody>
</table>

The simulation parameters include the incremental values for the discrete space and time, i.e., \( dN \) and \( dT \) (or \( T \)). As mentioned previously, in the network modeling we use vehicle group size adjustment to account for the on-ramp and off-ramp flows. The adjusting coefficients are determined by the average on-ramp/off-ramp flow rate.

The third type of parameters are associated with the second order traffic flow model, related to the reaction time delay and the drivers’ anticipation, respectively. However, although such parameters are rooted in the microscopic level behaviors, they are macroscopic parameters and thus their values may not be directly interpreted for microscopic behaviors. In addition, they also influence the stability of the simulation model, which in turn limits the range of the parameters.
4.5.2.2 Ground truth representation

Since we have full knowledge of the traffic conditions during data gathering period, the whole data set is treated as ground truth, serving as the benchmark to evaluate the performance of different estimators. Aggregation resolution of the ground truth needs to be determined with careful considerations on the traffic condition homogeneity, as the traffic system is a highly dynamic, nonlinear, and continuous system. If the aggregation resolution is too low, the average condition of a cell blurs out the actual traffic flow patterns and thus is not a good representation of the detailed traffic conditions in the associated area. On the other hand, if the resolution is too high, the average condition of a cell may be highly influenced by a single vehicle in that cell, and thus could be very noisy. Examples of speed maps with different discretization settings are shown in Figure 4.8, where Figure 4.8-a presents a speed map with a low aggregation resolution of $160m \times 60s$, Figure 4.8-b with a resolution of $80m \times 20s$, and Figure 4.8-c with a high resolution of $20m \times 5s$. The blue color indicates faster mean speeds while the red color represents congested traffic conditions or low speeds. The white areas implies lack of actual data. We can see that the low aggregation resolution in Figure 4.8-a cannot satisfactorily capture the flow wave patterns, while the high aggregation resolution Figure 4.8-c presents too much noise where two cells that are very close to each other (e.g., within 40 m) can have very different speeds. The compromised resolution used in Figure 4.8-b appears to be relatively more reasonable. By examining the plots of different spatial-temporal discretization results, a resolution of $80m \times 20s$ was adopted, which is expected to satisfactorily capture the traffic flow patterns while smoothing out most of the noisy fluctuations in the link level.
FIGURE 4.8 Choices of the Ground Truth Resolution
4.5.3 Performance measures

Since the main purpose of the traffic flow modeling is to reproduce the traffic flow patterns observed in the real-world, the experiment results are mainly evaluated by visually comparing the speed maps, density maps and fixed-location curves, with supportive measures of the root-mean-square error (RMSE). RMSE is not an ideal measure for model performance since there is no threshold to determine a single RMSE value to be “good” or “bad”. Denote the ground truth as $\theta$ and the simulated as $\hat{\theta}$, and RMSE is defined as Equation 4.13.

$$RMSE(\hat{\theta}) = \sqrt{E[(\hat{\theta} - \theta)^T(\hat{\theta} - \theta)]}$$  \hspace{1cm} (4.13)
4.5.4 Results and discussions

4.5.4.1 Speed maps and density maps

Speed maps and density maps visually compare the simulated traffic state against the ground truth. Figure 4.9 shows the original simulation results in the Lagrangian format. The red areas represent the congestion waves propagating from downstream to upstream. The traffic conditions at the inflow boundary seem to be very different from the actual immediate downstream. This is because, at the inflow boundary, a vehicle group enters the computation domain based on the supply-demand rule which also determines its spacing (density). The associated speed is determined solely from the fundamental diagram at this node for both the first- and second-order traffic flow models. This transitional area is retained in the results only because the segment length in the experiment is relatively short and it is useful to show the boundary artifacts.

It is also noted that the results are almost the same between the two models, which is also indicated in the Eulerian format speed maps in Figures 4.10 and 4.11. Comparing the simulation results with the ground truth, we obtain the RMSE values as 10.27 km/h and 10.33 km/h for speed and 10.04 and 9.89 veh/km for density, for the first-order model and the second-order model, respectively. These numbers are also very close.

4.5.4.2 Fixed-Location Comparisons

The six figures in Figure 4.12 compare the simulated traffic state against the ground truth at fixed locations. Specifically, it shows the fixed-location comparisons of speed and density estimations for segments of $x = 160$ m to 240 m, $x = 320$ m to 400 m, $x = 480$ m to 540 m, representing an upstream location before the on-ramp, a mid-stream location between the on-ramp and the off-ramp, and a downstream location after the off-
(a) For the first-order model

(b) For the second-order model

FIGURE 4.9 Speed Maps from Simulation in Lagrangian Coordinates
(a) Ground truth

(b) For the first-order model

FIGURE 4.10 Speed Maps from Simulation in Eulerian Coordinates
(c) For the second-order model

CONTINUED FIGURE 4.10  Speed Maps from Simulation in Eulerian Coordinates

(a) Ground truth

FIGURE 4.11  Density Maps from Simulation in Eulerian Coordinates
(b) For the first-order model

(c) For the second-Order Model

CONTINUED FIGURE 4.11  Density Maps from Simulation in Eulerian Coordinates
(a) Speed curves: upstream.

(b) Density curves: upstream.

FIGURE 4.12 Fixed Location Comparison for the Traffic Flow Models
(c) Speed curves: mid-stream.

(d) Density curves: mid-stream.

CONTINUED FIGURE 4.12 Fixed Location Comparison for the Traffic Flow Models
(e) Speed curves: downstream.

(f) Density curves: downstream.

CONTINUED FIGURE 4.12  Fixed Location Comparison for the Traffic Flow Models
ramp locations. It shows that the second-order model goes a little further at extreme conditions than the first-order model, and most of the time this means closer to the ground truth. However, overall the two models produce very similar results. This is due to the fact that the second-order terms become effective mostly when the traffic conditions are extreme and it needs a certain period to accumulate the differences. For the US101 data, the traffic condition fluctuates rapidly and thus there is no stable period for the model to make a significant difference. The observation raises an interesting and important issue in traffic flow modeling: how to make the trade-off between the computational cost and the model performance. The answer depends on the specific application scenario.

FIGURE 4.13 Lagrangian Simulation Shown in Speed Map

4.5.4.3 The upwind method in Lagrangian estimation

Figure 4.13 intends to show the details of the Lagrangian speed map in Figure 4.9-b, as a result of the upwind method. In Lagrangian modeling, the traffic flow is discretized into vehicle groups and each vehicle group follows the front vehicle group during flow
projection. In Figure 4.13, the traffic condition gradually changes from “severe congested” condition (represented as the red color) to “light congested” condition (represented as blue color). The congestion discharges quickly towards the upstream. This upwind method helps in simplifying the traffic flow simulation.

4.6 Summary

In this chapter, we presented a second-order Lagrangian macroscopic traffic flow model. A dynamic speed equation is derived from the realistic modeling of the drivers’ anticipation and reaction delay. In the lane-drop scenario based experimental analysis it has been found that the second-order model is not only able to reproduce the formation and discharge of the congestion as the first-order model does, but also able to naturally create stop-and-go waves which are amplified as the waves propagate backward to the upstream. In the US101 scenario based experimental analysis, we found that the second-order model represents the rapid extreme fluctuations better but the results are similar most of the time. Additionally, the second-order model is computationally more expensive. This raises the question of when to apply the second-order model or the first model: is the additional computational cost worth it? The answer is, it depends. Trade-offs are always in need in real-world applications and one should remember to ask this kind of questions before applying any model. Nevertheless, the developed second-order has its own merits. Based on this model, the next chapter presents a new real-time link-based traffic estimator for multi-sensor measurements.
CHAPTER 5
A MULTI-SENSOR DATA FUSION APPROACH FOR
REAL-TIME TRAFFIC ESTIMATION: LINK LEVEL

5.1 Introduction

In the bi-level architecture presented in Chapter 3, a link-level traffic state estimator is adopted to provide the low resolution estimates. This link-level estimator could be any valid model-based traffic state estimator for the interested traffic flow characteristics. This chapter presents a new link-level estimator to incorporate multi-sensor data.

As mentioned in Chapter 1, the three basic components of a model-based approach include a dynamic traffic flow model, traffic measurements, and a model-based estimation algorithm. In Chapter 4 we developed a second-order macroscopic Lagrangian traffic flow model by incorporating drivers’ anticipation and reaction delay. This model serves as the system model in this chapter. Loop detector data and probe-vehicle data are assumed available, representing the Eulerian and Lagrangian data, respectively. As for the estimation algorithm, a multi-sensor extended Kalman filter (MEKF) have been developed and adopted. EKF is widely applied in various areas for real-time state estimation because of its superiority in computational efficiency and satisfactory performance. Other advanced algorithms like particle filter may be able to enhance the estimation accuracy but typically require much more computation.

In Section 5.2, we provide a short introduction to the KF and EKF basics and some discussions on possible approaches for multi-sensor data fusion using model-based estimation, followed by a description of the new MEKF algorithm. An MEKF-based real-
time traffic state estimator is described in detail afterwards in Section 5.3, which is implemented and tested using US101 real traffic data in Section 5.4. Conclusions and some practical guidelines are given in Section 5.5.

5.2 Model-based estimation algorithms

5.2.1 Kalman filter

Kalman filter (KF) (Kalman, 1960; Welch and Bishop 2001) recursively and optimally solves the real-time state estimation problem for a linear system with Gaussian white noise, which is equivalent to a recursive Bayesian estimation method (Barker et al. 1995). It applies to the linear system of Equation 5.1, obtained by substituting \( f(\cdot) \) and \( g(\cdot) \) with linear functions \( G_{k} \theta_{k} + \nu_{k+1} \) and \( H_{k+1} \theta_{k+1} + \omega_{k+1} \) in Equation 1.1-a and Equation 1.1-b, respectively. \( G \) is an \( N \times N \) matrix relating the system state at time step \( k \) to the state at time step \( k+1 \), in the absence of system noise. \( H \) is an \( M \times N \) matrix relating the measurement to system state. Both \( G \) and \( H \) can vary with time and thus they are indexed by subscript \( k \) denoting time \( k \).

\[
\theta_{k+1} = G_{k} \theta_{k} + \nu_{k+1} \quad \text{(5.1-a)}
\]

\[
z_{k+1} = H_{k+1} \theta_{k+1} + \omega_{k+1} \quad \text{(5.1-b)}
\]

The system noise \( \nu_{k} \) is assumed to be multivariate Gaussian white noise with a known covariance matrix \( Q \), i.e., \( p(\nu) \sim N(0, Q) \). The measurement noise \( \omega_{k} \) is also assumed to be multivariate Gaussian white noise with a known covariance matrix \( R \), i.e., \( p(\omega) \sim N(0, R) \). The noise terms are assumed to be independent of each other, i.e., \( \text{cov}(\nu_{k_1}, \nu_{k_2}) = 0 \) for \( k_1 \neq k_2 \), and \( \text{cov}(\omega_{k_1}, \omega_{k_2}) = 0 \) for \( k_1 \neq k_2 \), and \( \text{cov}(\nu_{k}, \omega_{k_i}) = 0 \) for any \( k \) and...
Given the above assumptions, the conditional pdf \( p(\theta_{k+1} | Z_{k+1}) \) is also Gaussian and thus KF can solve \( E[\theta_{k+1} | Z_{k+1}] \) optimally.

Let \( P_{\theta_k} \) denote the error covariance matrix for the system state \( \theta_k \), where \( P_{\theta_k} = E[(\theta_k - \theta_k^*)(\theta_k - \theta_k^*)^T] \). At each estimation time step, KF can be viewed as a two-stage procedure (Figure 1.4). The first stage is the “time update”, which yields the a priori estimate of the system state \( (\theta_{k+1}^*) \) and its error covariance matrix \( (P_{\theta_{k+1}}^-) \) for the (new) current time step given the a posteriori estimate from the previous time step \( (\theta_k^* \text{ and } P_{\theta_k}^*) \).

The second stage is the “measurement update” during which KF calculates the Kalman gain with new measurements and then update the system state and the error covariance matrix estimates using the Kalman gain. Indeed, the Kalman gain weights the information provided by the a priori estimate and the measurements. This completes the current estimation time step and outputs the a posteriori estimates. The procedure is repeated recursively as the estimation proceeds. The KF algorithm is summarized in Appendix A.

The major limitation of KF is its incapability of dealing with nonlinear systems like the freeway traffic flow system. As we have discussed in Chapter 2, EKF is a widely applied in an ad hoc fashion that approximates \( E[\theta_{k+1} | Z_{k+1}] \) for nonlinear systems.

### 5.2.2 Extended Kalman filter and iterative extended Kalman filter

Equation 5.2 below represents the state-space model of the nonlinear system for estimation.

\[
\theta_{k+1} = f(\theta_k, v_{k+1}) \quad (5.2-a)
\]

\[
z_{k+1} = h(\theta_{k+1}, o_{k+1}) \quad (5.2-b)
\]
Either or both the system function and the measurement function could be nonlinear. In addition, they both need to be differentiable. The system noise $\nu_k$ and measurement noise $\omega_{k+1}$ follow the same assumptions as for Equation 5.1, i.e., $p(\nu) \sim N(0, Q)$, $p(\omega) \sim N(0, R)$, and the noise terms are independent of each other.

Algorithm I: Extended Kalman Filter

Initialization: set $k = 0$, then apply

$$
\theta_0 = E[\theta_0], \\
P_{0_0} = E[P_{0_0}].
$$

Recursive estimations: for each time step, apply

Time update:

$$
\theta_{k+1}^+ = f(\theta_k^+, 0), \\
P_{\theta_{k+1}}^+ = G_{k+1}P_{\theta_k}^+G_{k+1}^T + V_{k+1}QV_{k+1}^T.
$$

Measurement update:

$$
K_{k+1} = P_{\theta_{k+1}}^+H_{k+1}^T[H_{k+1}P_{\theta_{k+1}}^+H_{k+1}^T + W_{k+1}RW_{k+1}^T]^{-1}, \\
\theta_{k+1}^+ = \theta_{k+1}^- + K_{k+1}[z_{k+1} - h(\theta_{k+1}^-, 0)], \\
P_{\theta_{k+1}}^+ = [I - K_{k+1}H_{k+1}]P_{\theta_{k+1}}^-.
$$

$k = k + 1$. Return to the time update step.

EKF (Algorithm I) is an extension of KF that linearizes the nonlinear system about the current state mean and covariance. Specifically, at each estimation time step, the state-space model is relinearized around the previous a posteriori estimate using the current partial derivatives. Then the time update and measurement update can be conducted in a similar manner to that in the KF algorithm with modified equations. Let $G_{k+1}$ and $V_{k+1}$ denote the Jacobian matrix of partial derivatives of $f(\cdot)$ with respect to $\theta_k^-$ and $\nu_{k+1}$. Let
\( \mathbf{H}_{k+1} \) and \( \mathbf{W}_{k+1} \) denote the Jacobian matrix of partial derivatives of \( h(\cdot) \) with respect to \( \theta_{k+1}^- \) and \( \omega_{k+1}^- \). (Welch and Bishop, 2001)

EKF has been widely applied for nonlinear stochastic system estimations; however, the solution may not be optimal or have guaranteed error bound. It is essentially a first-order approximation of \( E[\theta_{k+1} \mid \mathbf{Z}_{k+1}] \). No existing literature has shown convergence results for EKF. (Mendel, 1995)

A critical assumption for EKF to work well is \( \delta \theta_{k+1} = \theta_{k+1}^- - \theta_{k+1}^+ \), which means that the discrepancy between the true state and the estimated state is small enough. To improve the performance of EKF, the iterated EKF (IEKF) (Jazwinski, 1970) was designed to keep \( \delta \theta_{k+1} \) as small as possible. It iterates the measurement update step \( I_{\text{IEKF}} \) times until it converges within a specified tolerance interval, i.e., \( \| \theta_{k+1,i}^+ - \theta_{k+1,i}^- \| \leq \epsilon \) where \( \epsilon \) is a specified small value. The process of IEKF can be depicted in Figure 5.1. It has been shown that, typically, only one additional corrector can improve the estimation performance substantially. (Mendel, 1995)

![Figure 5.1 Iterative Extended Kalman Filter Diagram](image)

The idea of the new multi-sensor estimation algorithm is inspired by IEKF, which also reiterates the measurement update stage multiple times but for different data sets.
5.2.3 Possible solutions for model-based multi-sensor state estimation

There are several possible techniques applicable to combine multi-sensor heterogeneous measurements, which include, but are not limited to, the following three methods. First, we can aggregate the data with the information of noise covariance matrices or associate them using different measurement equations, and then apply the model-based algorithm to the combined aggregated data in each iteration (see Figure 5.2-a). As an alternative approach, we can run parallel filters for each type of sensor and then aggregate the state estimates using the sensor noise information to obtain the final estimate (see Figure 5.2-b). As a third approach, we can run sequential filters where each filter is designed for a particular sensor and its input depends on the output estimate from the previous iteration (Figure 5.2-c). To the author’s best knowledge, there is no guarantee that the results for any of these three different approaches would be the same or very close. In particular, for EKF with nonlinear systems, since the results are approximate solutions instead of optimal solutions, the aggregated results can neither be guaranteed to be optimal or close to optimal. This is a common drawback of EKF and is also true for MEKF.

From these three approaches, we adopt the sequential method (as illustrated in Figure 5.2-c) by modifying the IEKF algorithm. This method is straightforward and flexible for incorporating multi-sensor traffic data. Additionally, it takes advantage of the IEKF method and its possible benefit of improved performance. It is not guaranteed that this method works better than others. However, as demonstrated in the experimental analysis, this method can effectively combine multi-sensor measurements to achieve more accurate estimate compared with the reported EKF-based approach with a single data source.
FIGURE 5.2 Possible Approaches for Multi-Sensor Data Fusion
5.2.4 Multi-sensor extended Kalman filter

Our multi-sensor extended Kalman filter (MEKF) combines the measurements from each type of sensors in a sequential way. As illustrated in Figure 5.3, at each time step, the MEKF algorithm projects the system state one time step ahead, then executes the measurement update step recursively for each type of sensors. The first measurement update procedure takes the input of the \textit{a priori} estimate from the time update procedure; the rest of measurement updates all take the intermediate \textit{a posteriori} estimate from the previous measurement update. The next section presents our MEKF-based estimator for real-time link-level multi-sensor traffic state estimation in detail, including the traffic flow model, the multi-sensor measurements, and the MEKF implementation for the assumed measurement types.

![Multi-Sensor Extended Kalman Filter Diagram](image)

**FIGURE 5.3** Multi-Sensor Extended Kalman Filter Diagram

5.3 MEKF-based Lagrangian link-level traffic estimation

5.3.1 Macroscopic traffic flow model

For the new MEKF-based estimator, we adopt the second-order Lagrangian traffic flow model (SLTFM) developed in Chapter 4. For the sake of clarity, we restate the discrete model as Equation 5.3. Equation 5.3-a is the discrete Lagrangian conservation equation, Equation 5.3-b is the discrete dynamic speed equation, and Equation 5.3-c is the Smulder’s fundamental diagram in Lagrangian form (see Equation 4.4), which would be smoothed at the nondifferentiable point during estimation to meet the requirement of EKF.
\[
s_j(k+1) = s_j(k) + \frac{T}{\eta_j} \left[ u_{j-1}(k) - u_j(k) \right]
\]

(5.3-a)

\[
u_j(k+1) = u_j(k) + \frac{T}{\tau} \left[ U(s_j(k)) - u_j(k) \right] + \frac{T}{\eta_j} \left[ \frac{u_{j-1}(k) - u_j(k)}{s_j(k)} \right] + \frac{T}{\tau} \left[ \frac{s_{j-1}(k) - s_j(k)}{s_j(k)} \right]
\]

(5.3-b)

\[
U(s_j(k)) = \begin{cases} 
  v_f - s_{cr} \frac{v_f - v_{cr}}{s_j(k)}, & \text{if } s \geq s_{cr} \\
  v_{cr} \frac{s_j(k) - s_{jam}}{s_{cr} - s_{jam}}, & \text{otherwise}
\end{cases}
\]

(5.3-c)

The system state is defined as the mean space speeds \( u \) of the Lagrangian vehicle groups and their spacings \( s \) in Equation 5.4.

\[
\mathbf{\theta}_k = [s_k^1, u_k^1, s_k^2, u_k^2, \ldots, s_k^N, u_k^N]^{T}
\]

(5.4)

With the macroscopic traffic flow model and the defined system state, we can derive the state-space model of the dynamic system equation \( \mathbf{\theta}_{k+1} = f(\mathbf{\theta}_k, \mathbf{v}_k) \).

Note that, unlike the Eulerian model, the state space in the Lagrangian model varies over time. In particular, when a new vehicle group enters the computation domain at the inflow boundary, the dimension of the state space increases; on the other hand, when a leading vehicle group leaves the computation domain at the outflow boundary, the dimension of the state space decreases. Thus at each time step, attention should be paid to make sure the state space is mapped from the previous step to the current step to prevent computational errors.

5.3.2 Traffic measurements and spatial-temporal alignment

For loop detectors, we can obtain mean speed for the vehicle groups that have
passed a particular sensor during a given time period. The reporting time interval for loop
detectors is denoted as $T_{loop}$, and thus indicating we have $k_{loop} = 1, 2, ..., K_{loop}$ to index the
reporting time steps of loop detectors. Since $T_{loop}$ could be different from the time step
duration $T$ used in the estimation system (indexed by $k$), we need to map $k_{loop}$ to $k$. The
loop detector measurements also need to be mapped to vehicle groups as loop detector are
installed at fixed locations (see Figure 1.2-b). This can be determined by identifying which
set of vehicle groups have passed the loop detectors during time step $k$. Then the loop
detector data are a representation of the average traffic conditions of the associated vehicles
at time step $k$.

The observation equation for the loop detectors is formulated as Equation 5.5. The
notation ‘*’ is to indicate that the loop detector measurement data are only available to
those vehicle groups that have passed certain locations where the loop detectors are
installed. Otherwise, the loop detector measurement for that vehicle group is unavailable.

$$z_{k}^{loop} = [u_{k}^{1*}, u_{k}^{2*}, ..., u_{k}^{N*}]^{T}$$ (5.5)

The probe-vehicle sensors are assumed to be able to directly measure speed,
spaceing and location for a GPS equipped vehicle. As we have assumed that the speed and
spaceing are homogeneous within each vehicle group, the speed and spacing measurements
for an individual vehicle are also representative of the vehicle group to which it belongs.
The location information is required to map an individual vehicle to a vehicle group in the
discrete flow dimension. Similar to the loop detector data, the probe-vehicle reporting
periods are indexed by $k_{vehicle} = 1, 2, ..., K_{vehicle}$, and $k_{vehicle}$ needs to be mapped to the
estimation system time steps ($k$).
The probe-vehicle measurements include both the speed data and the spacing data, denoted as \( z_k^{veh} = [s_k^{1*}, u_k^{1*}, s_k^{2*}, u_k^{2*}, \ldots, s_k^{N*}, u_k^{N*}]^T \), where "*" also means it only applies when the sensor is available. If multiple vehicles in a group all have sensors, their measurements are averaged. If the spacing data are unavailable or too noisy, as we have experienced in preliminary experimental analysis, the additional spacing measurement may deteriorate the estimation performance. In such a case, we use only the speed measurements, represented as Equation 5.6.

\[
z_k^{veh} = [u_k^{1*}, u_k^{2*}, \ldots, u_k^{N*}]^T \tag{5.6}
\]

With the defined measurement variables and data set, the measurement equation \( z_{k+1} = h(\theta_{k+1}, \omega_{k+1}) \) can be derived. In this case, function \( h(\cdot) \) is linear, i.e.,

\[
\begin{align*}
z_{k+1}^{loop} &= H_{k+1}^{loop} \theta_{k+1} + \omega_{k+1} \\
z_{k+1}^{veh} &= H_{k+1}^{veh} \theta_{k+1} + \omega_{k+1}
\end{align*} \tag{5.7}
\]

where the elements of the time-dependent \( H \) matrix satisfies Equation 5.8 below.

\[
H_{[i,j]} = \begin{cases} 
1 & \text{if } z_{[i]} \text{ is a measurement for } \theta_{[j]} \\
0 & \text{o.w.}
\end{cases} \tag{5.8}
\]

The noise covariance matrices \( Q \) and \( R \) are associated with the specific state and measurement variables, and are inputs to the estimator.

5.3.3 MEKF-based algorithm with two types of sensors

The MEKF presented in the previous section is adopted here. Let \( G_{k+1}, V_{k+1}, H_{k+1} \) and \( W_{k+1} \) still be the Jacobian matrices defined above. The MEKF-based estimator using the loop detector data and the probe-vehicle data for traffic state estimation is summarized as Algorithm II.
Algorithm II: MEKF-Based Estimator with Loop Detector Data and Probe-Vehicle Data

Initialization: set $k = 0$, then apply

\[
\begin{align*}
\theta_0 &= E[\theta_0], \\
P_{0,0} &= E[P_{0,0}].
\end{align*}
\]

Recursive estimation: for each time step, apply

Time update:

\[
\begin{align*}
\theta_{k+1}^- &= f(\theta_k^+, 0), \\
P_{0,i+1}^- &= G_k^i P_i^T + V_{k+1} Q_{k+1}^T.
\end{align*}
\]

Measurement update:

If loop detector measurements are available:

\[
\begin{align*}
K_{k+1}^{\text{loop}} &= P_{0,i+1}^- (H_{k+1}^{\text{loop}})^T [H_{k+1}^{\text{loop}} P_{0,i+1}^- (H_{k+1}^{\text{loop}})^T + W_{k+1}^{\text{loop}} R_{k+1}^{\text{loop}}]^{-1}, \\
\theta_{k+1}^{+\text{loop}} &= \theta_{k+1}^- + K_{k+1}^{\text{loop}} [Z_{k+1}^{\text{loop}} - h(\theta_{k+1}^-, 0)], \\
P_{0,i+1}^{+\text{loop}} &= [I - K_{k+1}^{\text{loop}} H_{k+1}^{\text{loop}}] P_{0,i+1}^-.
\end{align*}
\]

If probe-vehicle measurements are available:

\[
\begin{align*}
K_{k+1}^{\text{veh}} &= P_{0,i+1}^{+\text{loop}} (H_{k+1}^{\text{veh}})^T [H_{k+1}^{\text{veh}} P_{0,i+1}^{+\text{loop}} (H_{k+1}^{\text{veh}})^T + W_{k+1}^{\text{veh}} R_{k+1}^{\text{veh}}]^{-1}, \\
\theta_{k+1}^{+\text{veh}} &= \theta_{k+1}^{+\text{loop}} + K_{k+1}^{\text{veh}} [Z_{k+1}^{\text{veh}} - h(\theta_{k+1}^{+\text{loop}}, 0)], \\
P_{0,i+1}^{+\text{veh}} &= [I - K_{k+1}^{\text{veh}} H_{k+1}^{\text{veh}}] P_{0,i+1}^{+\text{loop}}. \\
\theta_{k+1}^+ &= \theta_{k+1}^{+\text{veh}}, \\
P_{0,i+1}^+ &= P_{0,i+1}^{+\text{veh}}.
\end{align*}
\]

Else if no probe-vehicle measurement is available

\[
\begin{align*}
\theta_{k+1}^+ &= \theta_{k+1}^{+\text{loop}}, \\
P_{0,i+1}^+ &= P_{0,i+1}^{+\text{loop}}.
\end{align*}
\]

Else if no measurement is available:

\[
\begin{align*}
\theta_{k+1}^+ &= \theta_{k+1}^-, \\
P_{0,i+1}^+ &= P_{0,i+1}^- \\
\theta_{k+1}^+ &= \theta_{k+1}^-, \\
P_{0,i+1}^+ &= P_{0,i+1}^-.
\end{align*}
\]

Set $k = k + 1$. Return to the time update step.
At the start of estimation, the system state $\theta_0$ and its covariance matrix $P_{\theta_0}$ are initialized. For each of the new simulation time step $k = k + 1$, the estimator first conducts a time update to obtain the a priori estimate, and then conduct measurement updates if there are any measurement data available, or otherwise continue to the next simulation time step.

In the measurement update stage, the specific steps depend on the availability of different types of measurements. The estimator first updates the a priori estimate using the loop detector data, yielding $\theta_{k+1}^{\text{loop}+}$ and $P_{\theta_{k+1}^{\text{loop}+}}$. If there is no other measurement available, they are treated as the final a posteriori estimate; otherwise if the probe-vehicle measurements are also available, the estimator continues to update $\theta_{k+1}^{\text{loop}+}$ and $P_{\theta_{k+1}^{\text{loop}+}}$ to $\theta_{k+1}^{\text{veh}+}$ and $P_{\theta_{k+1}^{\text{veh}+}}$, and then outputs as a final a posteriori estimate.

5.3.4 Discussion

The following aspects about MEKF should be kept in mind. First, based on the ad hoc EKF approach, MEKF also yields only suboptimal solutions and its performance cannot be guaranteed. Second, as MEKF adopts a sequential procedure for different types of sensors, the sequence also matters; measurement update using the loop detector data then the probe-vehicle data may not provide the same outputs as updating in the reverse sequence. This sequence can be defined according to the application environment. As currently most of the real-time sensors are loop detectors and their availability is more robust than the probe-vehicle measurements, it is reasonable to define the sequence as presented in Algorithm II. Third, the computational cost increases as the number of repeated updates increase. If there are many types of sensors, the sequential procedure may
not be efficient for online implementation. In such cases, one may consider aggregating subsets of measurements and then apply MEKF.

5.4 Experimental analysis: US101 freeway scenario

5.4.1 Background

The experimental analysis was carried out in MATLAB on a real traffic data set for a segment of the US Highway 101 (Hollywood Freeway) in Los Angeles, California, United States, from the Federal Highway Administration’s (FHWA) Next Generation Simulation (NGSIM) project, referred to as US101 data. The time frame starts at 7:50AM and ends at 8:35AM in the morning of June 15, 2005, totally 45 minutes. In the first 15 minutes, the freeway segment experiences a transition from free flow condition to congestion and stays congested for the rest 30 minutes. The initial 30 s period is not used in estimation because of partially missing data. The freeway segment under investigation (Figure 5.4) is approximately 640 m (2100 feet), and has five main lanes including one HOV lane. It also has an on-ramp at location 176 m (578 feet) from start and an off-ramp at location 389 m (1276 feet) from start, and has an auxiliary lane between the on-ramp and off-ramp. The measurement data was first collected by video cameras which had full coverage of the freeway segment, and then transcribed to vehicle trajectory data, including information about vehicle identification number, time, position coordinates, lane number, speed, spacing, and so on. The time frequency is 1/10th second. (NGSIM)

5.4.2 Experiment setup

5.4.2.1 Data processing: ground truth representation and sensor emulation

The ground truth (Figure 5.5) is emulated by processing the whole traffic data set with a resolution of $80\text{m} \times 20\text{s}$ (see Section 4.5.2.1) for link-level evaluation, which is
expected to satisfactorily capture the flow patterns while smoothing out noisy fluctuations.

The loop detector data and the probe-vehicle data are emulated using US101 data set with sampling methods. Specifically, the loop detector data are assumed to be installed at specific locations with certain reporting periods. Thus the mean spacings and harmonic mean speeds of the vehicles passing those locations during each reporting period emulate
the loop detector measurements. For the probe-vehicle detector data, a subset of vehicles are randomly and uniformly sampled out from the vehicle population according to some assumed penetration rate (i.e., giving the percentage of the sensor-equipped vehicles in the population of vehicles). The sampled vehicles are assumed to report their dynamic information periodically, emulating trajectory data from mobile sensors.

The emulated measurements serve as input data to the model-based estimator; no additional artificial noise is added to the measurements. There are generally two types of noise associated with the empirical measurements. The first type of noise is related to the measurement characteristics themselves. For example, a major source of noise for probe-vehicle data is from the individual driver’s behavior; the trajectory data of a particular vehicle may provide the true instantaneous speed measurements, but the driver may drive extremely fast or slow, rendering the associated measurements not representative of the average traffic conditions around. The second type is the result of the sensor hardware. In this experimental evaluation, we do not take into considerations of this second type of noise, i.e., all measurements are emulated without noise. In future research it would be interesting to model potential noise sources and investigate the influence of different noise levels on the performance of MEKF-based estimators.

According to the emulation definitions of the loop detector data and the probe-vehicle data, we would expect that the loop detector data are more stable and representative, while the probe-vehicle data may be noisier. This has also been revealed previously by Yuan et al. (2012) where experiments have shown that traffic state estimation using only the loop detector data yields much better results than using only the probe-vehicle data.
5.4.2.2 Experiment scenarios

The assumptions regarding the loop detector distance, the probe-vehicle penetration rate, and the reporting time intervals need to be realistic to reflect the real world applications. Currently, loop detector data are normally aggregated for every 60 s; however, a much shorter reporting period is also available, like every 20 s. Since the traffic conditions in US101 fluctuates rapidly, a reporting period of 60 s cannot provide sufficient information (Figure 4.11-a). Thus a shorter reporting period of 20 s is also evaluated. As for the distance between detectors, since the total length of the freeway is only 640 m, it is assumed that the detector locations have three possibilities: every 100 m (6 loop detectors), every 200 m (3 loop detectors), and every 500 m (2 loop detectors). Three scenarios for loop detector measurements are (a) every 100 m distance with a reporting period of 20 s, (b) every 200 m distance with a reporting period of 20 s, and (c) every 500 m distance with a reporting period of 60 s. The 100m/20s setting represents a high coverage scenario while the 500m/60s setting represents a low coverage scenario for loop detector data.

Given the characteristics of mobile sensing, it is assumed that the probe-vehicle detectors report every 5 s. Since the original data set has ten frames per second, the instant speed and spacing information are defined as their respective averages during the past ten reporting frames. The penetration rate ($p$) is assumed to be 5%, 10%, or 20%, that is, among all the vehicles traveling through US101 during the investigation period, $p$ portion of vehicles are reporting their trajectory information periodically. A penetration rate of 20% represents a high coverage scenario while a penetration rate of 5% represents a low coverage scenario. For each penetration level, ten random samples were drawn to represent ten random instances for probe-vehicle data.
TABLE 5.1 Comparison Settings for US101 Freeway Link-Level Estimation

(a) Multi-Sensor vs. Loop-Detector-Only Estimation

<table>
<thead>
<tr>
<th>Estimator Components</th>
<th>Baseline Scenarios</th>
<th>Multi-Sensor Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loop</td>
<td>Loop</td>
</tr>
<tr>
<td>Measurement</td>
<td>100m/20s</td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td>200m/20s</td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td>500m/60s</td>
<td>500m/60s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500m/60s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500m/60s</td>
</tr>
<tr>
<td>Model</td>
<td>SLTFM</td>
<td>SLTFM</td>
</tr>
<tr>
<td>Algorithm</td>
<td>EKF</td>
<td>MEKF</td>
</tr>
</tbody>
</table>

(b) Multi-Sensor vs. Probe-Vehicle-Only Estimation

<table>
<thead>
<tr>
<th>Estimator Components</th>
<th>Baseline Scenarios</th>
<th>Multi-Sensor Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Probe-Vehicle</td>
<td>Loop</td>
</tr>
<tr>
<td>Measurement</td>
<td>5%</td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500m/60s</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500m/60s</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>100m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200m/20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500m/60s</td>
</tr>
<tr>
<td>Model</td>
<td>SLTFM</td>
<td>SLTFM</td>
</tr>
<tr>
<td>Algorithm</td>
<td>EKF</td>
<td>MEKF</td>
</tr>
</tbody>
</table>
The experiments were conducted on various combinations of different loop detector placements and probe-vehicle penetrations with a total of fifteen scenarios. In particular, nine scenarios are multi-sensor cases, three are Eulerian sensor only scenarios and three Lagrangian sensor only scenarios. The Eulerian or Lagrangian sensor only scenarios are treated as baseline scenarios setting up benchmarks for performance evaluation. Based on these fifteen scenarios, two sets of comparisons were conducted. The first comparison is between the MEKF-based estimator and the Eulerian measurement only estimator (summarized in Table 5.1-a), while the second comparison is between the MEKF-based estimator and the Lagrangian measurement only estimator (summarized in Table 5.1-b).

The second order Lagrangian macroscopic traffic flow model (SLTFM) developed in Chapter 4 and its associated parameters are adopted for all scenarios to make the results comparable. The boundary conditions and the on-ramp/off-ramp rates are also assumed to be known.

5.4.2.3 Parameter calibration

As the estimator adopts the traffic flow model presented in Chapter 4, all the associated parameters are applied the same (see Section 4.5.2). Additionally, the noise covariance matrices $Q$ and $R$ need to be calibrated. Since they are hard to calibrate, the following ad hoc method is applied in this research. $Q$ and $R$ are assumed to be diagonal and constant, but vary from scenario to scenario. That is, $Q = \text{diag}(\sigma_{\text{spacing}}^2, \sigma_{\text{speed}}^2)$ and $R = \text{diag}(\sigma_{\text{loop}}^2, \sigma_{\text{veh}}^2)$, where the function $\text{diag}(\cdot)$ creates a diagonal matrix according to the input vector in the parentheses. The state covariance matrix is fixed at a reasonable level, and different measurement covariance matrices were tested and adapted to find
appropriate combinations for each given scenario. For example, the measurement noise level for the loop detector data with a setting of 500m/60s is obviously higher than that with a setting of 100m/20s.

5.4.3 Performance measures

Since the probe-vehicle measurements are emulated by random sampling, ten random samples were drawn for each penetration level to obtain statistical sensible results. Thus, for each of the twelve experimental scenarios using probe-vehicle data, ten runs were conducted. The performance of each estimation run was evaluated using the root-mean-square error (RMSE) and the percentage of improvement (PoI) performance measures.

Let \( c = 1, 2, \ldots, C \) index the estimation runs. Denote the ground truth as \( \theta \) and the estimate as \( \hat{\theta} \), and the RMSE is defined by Equation 5.9. Note that for scenarios using only loop detector measurements, there is only one measurement set for each setting, and thus \( C = 1 \).

\[
\text{RMSE}(\hat{\theta}) = \sqrt{E[(\hat{\theta} - \theta)^T(\hat{\theta} - \theta)]} = \left( \frac{1}{C} \sum_{c=1}^{C} (\hat{\theta} - \theta)^T(\hat{\theta} - \theta) \right)^{\frac{1}{2}}
\] (5.9)

The measure PoI is defined by Equation 5.10, which describes the relative magnitude of the improvement in RMSE compared with the RMSE in the baseline scenario.

\[
\text{PoI}(\hat{\theta}) = \frac{\text{RMSE}_{\text{MEKF}} - \text{RMSE}_{\text{baseline}}}{\text{RMSE}_{\text{baseline}}}
\] (5.10)

RMSE is in general the finite-sample approximation of the standard error, thus the smaller the better. However, there is no threshold to determine a single RMSE value to be “good” or “bad”; it may only be used to compare multiple models for the same ground truth. Moreover, even for comparing two alternative models, the RMSE may not reveal all
the differences. Additional measures are required for model performance evaluation. Visualization of speed maps, density maps, and fixed-location comparisons serve as good supplementary methods for comparisons.

5.4.4 Results and discussions

5.4.4.1 MEKF-based estimator vs. Eulerian-sensor only estimator

This comparison evaluates the performance of the new MEKF-based estimator versus the loop detector only estimator. The results are summarized in Table 5.2 and Figure 5.6, which shows that the multi-sensor estimation is superior to the loop detector based estimation in terms of RMSE. The results are generally what we expected. First, the multi-sensor estimation achieves better estimation accuracy. For different loop detector settings, by combining additional vehicle measurements we can reduce the speed RMSE in the range of 5.92% to 63.04%; and the density RMSE reduction in the range of 1.93% to 28.77%. Second, the higher the penetration rate, the better the estimation accuracy. Third, the biggest improvement happens when the loop detector measurement quality is low, while the probe-vehicle measurement quality is high. As we discussed in Section 4.5.2.2, a temporal resolution of 60 s cannot successfully capture the flow pattern of the US101 data. Thus the loop detector setting of 500m/60s represents a low quality measurement. The penetration rate of 20% is assumed to be the high coverage setting for probe-vehicles. The largest improvement appears in this scenario, where PoI is 63.04% for speed and 27.41% for density. Fourth, when the loop detector data has a high resolution setting, the benefit of MEKF-based approach diminishes. In such cases, the loop detector only data can provide effective traffic information, and include most of the information captured by the probe-vehicle sensors. For the loop detector setting of 100m/20s, the improvement of MEKF-
based approach for speed estimation is 13.64%, and for density estimation improvement goes up to 4.29%. Fifth, the percentage improvement is larger for speed estimates than for density estimates. This is mainly due to the fact that the speed-spacing relationship is not

TABLE 5.2 RMSE Summary: Multi-Sensor vs. Loop-Detector-Only Estimation

<table>
<thead>
<tr>
<th>Loop Detector</th>
<th>Speed (km/h)</th>
<th>Density (veh/km)</th>
<th>p</th>
<th>Speed (km/h)</th>
<th>Density (veh/km)</th>
<th>Speed</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop-Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100m/20s</td>
<td>2.12</td>
<td>4.92</td>
<td>5%</td>
<td>2.00</td>
<td>4.83</td>
<td>5.92%</td>
<td>1.93%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10%</td>
<td>1.92</td>
<td>4.77</td>
<td>9.45%</td>
<td>3.12%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20%</td>
<td>1.83</td>
<td>4.71</td>
<td>13.64%</td>
<td>4.29%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5%</td>
<td>2.63</td>
<td>5.06</td>
<td>11.59%</td>
<td>3.09%</td>
</tr>
<tr>
<td>200m/20s</td>
<td>2.97</td>
<td>5.22</td>
<td>10%</td>
<td>2.41</td>
<td>4.91</td>
<td>18.86%</td>
<td>5.92%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20%</td>
<td>2.22</td>
<td>4.89</td>
<td>25.27%</td>
<td>6.40%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5%</td>
<td>4.03</td>
<td>5.99</td>
<td>45.87%</td>
<td>24.52%</td>
</tr>
<tr>
<td>500m/60s</td>
<td>7.44</td>
<td>7.93</td>
<td>10%</td>
<td>3.30</td>
<td>5.65</td>
<td>55.69%</td>
<td>28.77%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20%</td>
<td>2.75</td>
<td>5.76</td>
<td>63.04%</td>
<td>27.41%</td>
</tr>
</tbody>
</table>

FIGURE 5.6 RMSE Plot: Multi-Sensor vs. Loop-Detector-Only Estimation
perfectly modeled, and that only speed measurements were used. Ideally, the measurement model accurately formulates the relationship between the measurement variables and the state variables through a one-to-one mapping; however, in reality, the speed versus density relationship is stochastic. Thus, the measurement correction is not as effective for the density (spacing) estimation as that for the speed estimation.

5.4.4.2 MEKF-based estimator vs. Lagrangian-sensor only estimator

In this experiment we compared the MEKF-based estimator to an estimator using only Lagrangian measurements. Table 5.3 summarizes the experimental results in terms of RMSE and PoI, and Figure 5.7 illustrates the estimation accuracy for different scenarios. First, it shows that the MEKF-based approach generally improves the estimation accuracy. The PoI range for speed estimation is 6.73% - 56.84%, and PoI range for density is -2.77% - 24.07%, with the exception of one negative improvement (-2.77%) exception. Second, when the vehicle coverage is high, additional low coverage loop detector measurement (500m/60s) even deteriorates the density estimation by 2.77%. This is mainly due to the fact that the low coverage loop detector measurements are not an effective representation of the traffic conditions with regard to the temporal resolution of the estimation. Third, when the penetration rate is as high as 20%, the MEKF-based approach can still enhance the speed estimation by up to 37.86% and density estimation by up to 15.98%. In other words, loop detector measurements with a setting of 100m/20s or 200m/20s provide even better traffic information enhanced by probe-vehicle measurements with a penetration rate of 20%. Fourth, even when the penetration rate is as low as 5%, the MEKF-based approach can greatly improve the estimation accuracy.
TABLE 5.3 RMSE Summary: Multi-Sensor vs. Probe-Vehicle-Only Estimation

<table>
<thead>
<tr>
<th>Vehicle Detector</th>
<th>Speed (km/h)</th>
<th>Density (veh/km)</th>
<th>Multi-Sensor</th>
<th>Loop Detector</th>
<th>Speed (km/h)</th>
<th>Density (veh/km)</th>
<th>POI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Speed</td>
<td>Density</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100m/20s</td>
<td>2.00</td>
<td>4.71</td>
<td>56.84%</td>
<td>24.07%</td>
</tr>
<tr>
<td>5%</td>
<td>4.63</td>
<td>6.35</td>
<td>200m/20s</td>
<td>2.63</td>
<td>4.89</td>
<td>43.19%</td>
<td>20.35%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500m/60s</td>
<td>4.03</td>
<td>5.76</td>
<td>12.94%</td>
<td>5.75%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100m/20s</td>
<td>1.92</td>
<td>4.77</td>
<td>46.31%</td>
<td>17.32%</td>
</tr>
<tr>
<td>10%</td>
<td>3.58</td>
<td>5.76</td>
<td>200m/20s</td>
<td>2.41</td>
<td>4.91</td>
<td>32.60%</td>
<td>14.78%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500m/60s</td>
<td>3.30</td>
<td>5.65</td>
<td>7.88%</td>
<td>1.97%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100m/20s</td>
<td>1.83</td>
<td>4.83</td>
<td>37.86%</td>
<td>15.98%</td>
</tr>
<tr>
<td>20%</td>
<td>2.95</td>
<td>5.60</td>
<td>200m/20s</td>
<td>2.22</td>
<td>5.06</td>
<td>24.68%</td>
<td>12.78%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500m/60s</td>
<td>2.75</td>
<td>5.99</td>
<td>6.73%</td>
<td>-2.77%</td>
</tr>
</tbody>
</table>

FIGURE 5.7 RMSE Plot: Multi-Sensor vs. Probe-Vehicle-Only Estimation

5.4.4.3 Visualization with speed maps and density maps

Speed maps and density maps provide visualization to compare the estimates against the ground truth. Figure 5.8 shows the original estimation results using the
Lagrangian model, and Figures 5.9 and 5.10 show the results from the same simulation but in Eulerian representation. The congestion waves and traffic condition fluctuations are clearly shown in the maps. One may notice that the traffic conditions at the inflow boundary are very different from immediate downstream in time and space. This is because the boundary conditions are applied at the end of each time step. As a result, a vehicle group enters the computation domain based on the supply-demand rule (see Section 4.3.3.1) without any measurement update. This computational artifact in the transitional area is retained in these results only because the segment length in the experiment is relatively short and we would like to show the full results. It, in fact, adversely affects the RMSE results as well. In real-world applications, one can avoid this issue by restricting the experimental domain to be shortly downstream to the inflow boundary to more realistically reflect the incoming vehicle flows.

FIGURE 5.8 Link-Level Speed Estimation in Lagrangian Coordinates
FIGURE 5.9 Speed Map Comparison for MEKF-Based Estimator
FIGURE 5.10 Density Map Comparison for MEKF-Based Estimator

(a) Ground truth

(b) Posterior link-level estimate
Figure 5.9 presents the speed maps of the ground truth and the link-level estimation, while Figure 5.10 presents the corresponding color maps for the traffic density. The scenario illustrated has a penetration rate of 10% for probe-vehicles and a reporting resolution of 200m/20s for loop detectors. Note that the white areas in the ground truth is the result of missing data. Generally, the estimated speed map looks very close to the ground truth at the link-level aggregation. For the density map, the estimation is also close to the ground truth, but the ground truth seems “noisier”. This is due to the “smoothing” effect of the estimation procedure.

5.4.4.4 Fixed location comparisons

Performance evaluation can also include comparison of the estimated traffic state versus the ground truth at different locations. Figure 5.11 shows the fixed-location

(a) Speed curves: upstream.

FIGURE 5.11 Fixed Location Comparison for MEKF-Based Estimator
(b) Density curves: upstream.

(c) Speed curves: mid-stream.

CONTINUED FIGURE 5.11 Fixed Location Comparison for MEKF-Based Estimator
(d) Density curves: mid-stream.

(e) Speed curves: downstream.

CONTINUED FIGURE 5.11  Fixed Location Comparison for MEKF-Based Estimator
CONTINUED FIGURE 5.11 Fixed Location Comparison for MEKF-Based Estimator comparisons of speed and density estimations for segments of $x=160 \text{ m}$ to $240 \text{ m}$, $x=320 \text{ m}$ to $400 \text{ m}$, $x=480 \text{ m}$ to $540 \text{ m}$, representing an upstream location before the on-ramp, a mid-stream location between the on-ramp and the off-ramp, and a downstream location after the off-ramp locations. It shows that the estimated results are close to the ground truth but smoother. Compared with Figure 4.12, we can see that the estimation results are much closer to the ground truth than for the simulation results.

5.5 Summary

In this chapter we developed a new link-based real-time traffic state estimator capable of incorporating multi-sensor heterogeneous data. Based on the IEKF algorithm, a multi-sensor data fusion algorithm is developed, and the resulting link-based estimator is presented which adopts the second-order Lagrangian traffic flow model developed in
Chapter 4. The experimental analysis using US101 real traffic data shows that the new estimator improves the estimation accuracy by fusing Eulerian and Lagrangian measurements. The developed estimator is utilized for the link-level estimation in the bi-level data fusion architecture (Figure 3.1). In the next chapter, a lane-based traffic estimation algorithm is presented using this bi-level approach.
CHAPTER 6
A MULTI-SENSOR DATA FUSION APPROACH FOR
REAL-TIME TRAFFIC ESTIMATION: LANE LEVEL

6.1 Introduction

Up to this point we have introduced the bi-level data fusion architecture (Figure 3.1) and a model-based approach for the link-level estimation (Section 5.3). This chapter completes the new data fusion approach by developing a data-driven method for the lane-level estimation, that is, a multi-sensor data fusion approach for real-time lane-based traffic state estimation using heterogeneous traffic measurements.

As in Chapter 3, we again refer to the link-level estimate as the prior estimate, or the low-resolution representation of the lane-level traffic state, and refer to the lane-level estimate as the posterior estimate, or the high-resolution representation of lane-level traffic. The former does not take account of the lane-specific measurements while the latter does.

Detailed description of the lane-level traffic estimation method is presented in Section 6.2. Simulation-based model validation and sensitivity analysis are presented in Section 6.3, followed by another evaluation using real traffic data (US101 data) in Section 6.4. Summary of conclusions is presented in Section 6.5.

6.2 A data-driven approach for lane-level traffic estimation

6.2.1 Spatial-temporal discretization

The lane-discriminative traffic state evolves in a three-dimensional spatial-temporal domain \((k, x, l)\), where \(k\) represents the discretized time dimension (see Section 1.4.4), \(x\) denotes the space dimension in the flow direction, and \(l (l \in L)\) is the discretized
space dimension in the lane direction. The $x$ dimension needs to be discretized to facilitate computer-based estimation. Assuming the Eulerian coordinate system is adopted, then the freeway stretch is discretized into fixed length segments. The start position of segment $i$ in the $x$-direction is denoted as $x_i$, $i \in I$.

![Diagram illustrating spatial discretization](image)

**FIGURE 6.1** Spatial Discretization (a) low resolution; (b) high resolution.

In the $x \times l$ domain, the freeway stretch is discretized into spatial cells and it is assumed that the traffic condition within each cell during each time step is homogeneous. Note that, the discretization resolution is different between the link-based and the lane-based estimators. Figure 6.1 provides an example of the spatial discretization differences. Figure 6.1-a shows the link-based (low resolution) discretization. The freeway is discretized into segments in the flow direction. Figure 6.1-b represents the lane-based (high resolution) discretization, where the additional horizontal purple lines implies that different lanes could have different traffic conditions even when they are in the same segment. The additional vertical light blue lines within each segment are optional and are used to enhance the resolution in the $x$ dimension.
6.2.2 Variable definitions and notation

Let $\theta$ still represent the desired traffic state vector, which is assumed to be the mean space speed ($\theta \equiv v$) in this chapter. Denote the prior estimate for all lanes in freeway segment $i$ during time-step $k$ as $\theta^{\text{p}}_{(k,i)}$ (or $\theta^{\text{p}}_{(k,i,l)} = \theta^{\text{p}}_{(k,i)}$ for all $l \in L$), and let $\theta^{\text{p}}_k \{ \theta^{\text{p}}_{(k,i)}, i \in I \}$ for $k \in K$. Correspondingly, the posterior estimate for lane $l \in L$ is $\theta^{\text{p}}_{(k,l)} \{ \theta^{\text{p}}_{(k,i,l)}, i \in I \}$ for $k \in K$. The location and time information for a freeway cell of interest (referred to as origin to follow the image processing nomenclature) is denoted as $o\text{origin} = o(k, \bar{x}, l)$. Note that, the center location of a cell ($\bar{x} = (x_1 + x_{i+1})/2$) is used as the spatial reference of the origin in the $x$ dimension, which is a reasonable choice for the distance-based information value evaluation (Section 6.2.3).

Empirical measurements are speed observations. Let $z^{m,p}_k$ denote the $p$th sensor of the sensor type $m$ that is available at time $k$, where $p \in \{1, \ldots, P\}$ and $m \in \{1, \ldots, M\}$. A measurement $z^{m,p}_k$ has its associated identification functions $k = k(z^{m,p}_k)$, $x = x(z^{m,p}_k)$, and $l = l(z^{m,p}_k)$, which assumes that its location and time information is available. With such information, the measurement are spatially and temporally aligned (see Section 1.4.2), and denoted as $z^{m,p}_k = z(k, x, l)$.

6.2.3 Distance-based evaluation of information contribution

Since a measurement can be away from the origin, the information value from cell data to a particular origin may vary. Evaluation of the information contribution is based on the distance between the two points through a kernel function. This distance is defined by Equation 6.1.
\[ \delta = (\delta k, \delta x, \delta l) = (k^i - k^o, x^i - x^o, l^i - l^o) \] (6.1)

For the specific lane-based estimation problem, since we are aiming to discriminate traffic conditions among multiple lanes, only those measurements that are from the same lane as the origin are selected. Therefore, this distance is may be written as \( \delta = (\delta k, \delta x) \).

The kernel function is defined as Equation 6.2.

\[ \kappa(\delta k, \delta x) = \lambda(\delta k) \cdot \lambda(\delta x) \] (6.2)

where an exponential choice of \( \lambda(\cdot) \) is adopted, as described by Equation 6.3.

\[ \lambda(\pi) = \exp\left(-\frac{|\pi|}{\sigma_{\pi}}\right) \] (6.3)

\( \sigma_{\pi} \) is the scaling parameter adjusting the level of impact given the magnitude of \( \pi \), which can be understood as a normalizing parameter to make different dimensions comparable. Therefore, the kernel function can be rewritten as Equation 6.4.

\[ \kappa(\delta k, \delta x) = \exp\left(-\frac{|\delta k|}{\sigma_k} - \frac{|\delta x|}{\sigma_x}\right) \] (6.4)

Since the kernel function is decreasing with respect to its arguments, larger deviation \( (\delta = (\delta k, \delta x)) \) yields smaller \( \kappa \) value, and vice versa.

The measurements selected for an origin are limited within a small range in its vicinity, which is defined by the filtering mask. In image processing algorithms, a mask refers to a matrix of influence weights for measurements from spatially (and temporally) neighborhood-cells, which also defines the “vicinity”. The size of a mask thus defines which set of measurements influence the estimation of the state at a particular origin. We need to define the masks small enough to meet online implementation criteria. On the other hand, the data-driven method requires sufficient measurements to proceed an estimation.
When data insufficiency occurs, the link-level estimator can fill the gaps. Therefore, our approach is more robust to data insufficiency than pure data-driven methods. Specific mask size requirements are dependent on other experimental settings (e.g., discretization resolution); the experiments in Sections 6.3 and 6.4 provide some examples.

6.2.4 Data-driven spatial-temporal filtering algorithm

The lane-level estimator adopts parallel spatial-temporal smoothing filters. Each filter utilizes a kernel function to evaluate the potential contributions of each individual input to the posterior estimate (referred to as the “kappa weights”), and a preset weight set \( \{ \alpha_0, \alpha_1, \ldots, \alpha_M, \alpha_{M+1} \} \) (referred to as the “alpha weights”) to appraise the reliability and trustworthiness of the data sources, that is, the prior estimate, each type of sensor, and the past posterior estimate. The spatial-temporal smoothing filter is formulated as Equation 6.5.

\[
\theta^+_{(k,i,l)} = \frac{\alpha_0 \theta^-_{(k,i,l)} + \sum_{m=1}^{M} \alpha_m \sum_{z^p \in mask^m} \kappa(\delta) z^{m-p} + \alpha_{M+1} \sum_{\theta^p, \alpha \in mask^p} \kappa(\delta) \theta^+_{past}}\]

\[
\alpha_0 + \sum_{m=1}^{M} \alpha_m \sum_{z^p \in mask^m} \kappa(\delta) + \alpha_{M+1} \sum_{\theta^p, \alpha \in mask^p} \kappa(\delta)
\]

for \( \alpha_0 = \bar{\alpha}_0, \quad \alpha_{m} = \begin{cases} 0 & \text{if } P_{m} = \emptyset, \text{ and } \alpha_{M+1} = \bar{\alpha}_{M+1} \\ \alpha_{m} & \text{otherwise} \end{cases} \)

As mentioned previously, the inputs to the lane-level filter include three components. First, the prior estimate from the link-level estimation. The main contribution of this input source is to maintain the stability and fill the gaps of the lane-level estimation even when the empirical lane-level measurements are not sufficient. In such cases, the link-level estimator can at least update the traffic conditions using the traffic flow model for prediction. Second, the lane-level measurements; they carry the empirical information about the traffic conditions of each lane. Third and last component, the past high resolution...
estimates; this component also carries the lane-level information from last time interval. The traffic flow evolution over time and space is indeed a continuous system, and thus the traffic conditions at the current time step are highly related to the conditions in the immediately previous time step. When the empirical measurements are not sufficient, this information source can also help fill the gaps. In this way, it also has an underlying assumption that there is no significant change of traffic conditions from the previous time step to the current time step.

The convolution procedure is described as follows. At each time step $k$, first the link-level estimator yields the low-resolution estimate, which serves as the prior lane-based estimate ($\theta_{(k,i,l)}$), feeding to the lane-level estimator. In the lane-level estimation, the following procedure is conducted for each of the spatial cells according to Equation 6.5. Each lane filter first selects measurement sets for each of the input source (e.g., $\theta_{(k,i,l)}$, $\theta_{\text{past}}^+$, and $\zeta_{\text{m,p}}$) according to the predefined masks. Then the contribution of each data input is evaluated by the kernel function defined in Equation 6.4, resulting in the kappa weights. Finally, a weighted average is carried out using the predefined alpha weights and calculated kappa weights to yield the final lane-based posterior estimate for the particular origin. This finishes the lane-based estimation for the current origin and proceeds to the next origin. Parallel lane filters conduct the estimation procedure at the same time for each lane separately. Posterior estimates are therefore obtained for time step $k$. The new estimation cycle starts for time step $k = k + 1$, and the bi-level estimation procedure repeats. This recursive online procedure continues in real time.
6.2.5 Computational efficiency and the filtering mask size

Given a collection of field measurements, the lane-based filter selects a set of inputs from its vicinity for a specific origin. The filtering mask size defines this “vicinity”. In our approach, the mask is defined in both the spatial and temporal dimensions. The mask size is denoted as $mask_{-data\_source}^{k\times x\times l}$, which is specified for each input data source. $k\times x\times l$ represents how many discrete units are used in each dimension. A trade-off has to be made, as the mask size needs to be small enough to ensure online implementation while it has to be large enough to provide sufficient lane-level influence measurements. Examples of masks can be found in the experimental analysis (Sections 6.3.2 and 6.4.2).

Note that the mask size can be extended within a range in each dimension. However, it is not favored for the following reasons. First, the computational cost would increase, since the input data size increases as the mask size increases. Second, since an exponential kernel is adopted, the influence weight of each point in a mask fades out quickly as it moves away from the origin. Moreover, if the mask size is too large, the non-relevant measurements may even deteriorate the performance of the developed filter. Thus, increasing the mask size does not necessarily help in terms of estimation accuracy in our approach. Finally, the traditional adaptive smoothing method (ASM) based data-driven approaches need a large mask size to ensure sufficient measurements; however, as we have discussed in the previous section, the additional inputs from the prior estimate and the past estimate can help fill the gaps due to smaller masks.

6.2.6 Arguments for the proposed data fusion approach

Treib et al. (2002) argued that the conventional smoothing filter could not capture the anisotropic characteristics of the traffic state evolution, and therefore a weighted
averaging process between estimates assuming either free flow condition or congested flow condition was adopted. However, the developed spatial-temporal filter is considered to be appropriate and adequate within the proposed framework of this research for the following reasons. First, for the previous data-driven methods, the capture of traffic flow phenomena relies purely on measurements, while in our approach the lower resolution filter already involves a macroscopic traffic flow model which is expected to reproduce the dynamic traffic flow patterns. Second, the weighted averaging process between estimates assuming either free flow condition or congested flow condition needs identification, which itself may introduce error. Third, when the traffic condition averaging process is not used, the proposed approach reduces the computational load by almost half for online implementation. Fourth, the experimental analysis presented in Sections 6.3 and 6.4 shows that our approach satisfactorily captures the lane-level traffic flow patterns.

This approach has the following advantages. First and the foremost, it can estimate the lane-based traffic state in real-time, which addresses a valuable, but so far not sufficiently studied problem. Second, the data-driven approach does not require a lane-based traffic flow model. It has the advantage that the hard-to-obtain traffic conditions on on-ramps and off-ramps are not essential in this approach, and lane-changing and overtaking behaviors are no longer obstacles for the lane-based traffic estimation. Third, like other data-driven approaches, the lane-level estimator can conveniently combine heterogeneous data, such as the probe-vehicle data and the loop detector data. Fourth, it can effectively utilize lane-based loop detector measurements. Current traffic estimators use the aggregated loop detector data across all lanes as measurements, but loop detectors are in fact installed in each lane, and measure the traffic conditions separately. Hence,
model-based estimators fail to utilize a valuable part of information from the loop detector measurements. Fifth, this approach only adds little additional computational burden in comparison to the link-level real-time estimation, which enables it to be implemented online (see Section 3.3.8 for experiment-based results). Lastly, this approach is flexible in terms of estimation resolution. Different discretization schemes in both spatial and temporal dimensions can be conveniently adopted.

6.3 Experimental analysis: Interstate-10 freeway scenario

6.3.1 Simulation setup

To validate the proposed multi-sensor data fusion approach for real-time lane-based traffic state estimation, simulation-based experiments were conducted and evaluated. A realistic model emulating the traffic behavior of a segment of the US Interstate-10 West (I-10W) (Figure 6.2) in Phoenix, Arizona was built in the VISSIM software (PTV, 2009; Mirchandani and Head, 2011). VISSIM is a widely adopted traffic simulator, which can analyze private and public transport operations for vehicular and pedestrian flows under constraints based on transportation engineering measures of effectiveness. It uses the psycho-physical driver behavior model developed by Wiedemann (1991), which is a microscopic traffic flow simulation model that includes the car following and lane changing models. VISSIM simulates the traffic flow by moving “driver-vehicle-units” through a network, where every driver with his specific behavior characteristics is assigned to a specific vehicle. The described vehicle moving methodology provides a satisfactory approximation of vehicles’ movements. (PTV, 2009) The developed lane-level traffic estimator was implemented in MATLAB.
FIGURE 6.2 An Illustration of the I-10W Freeway Segment

The I-10W segment is the westbound segment between the Elliot Street and the Baseline Street in Phoenix, Arizona. It starts from the end of the on-ramp in the Elliot Entrance, and ends after the Baseline Street off-ramp, with a total length of 3098 m (Figure 6.2). There are totally five lanes including the HOV lane. The traffic load is assumed to approximate the peak period origin-destination (OD) demand during a weekday, lasting for four hours. A severe congestion forms up and discharges during this period because of the overwhelming demand for getting off at the Baseline and Route 60 Exits. Link data and vehicle trajectory data are collected from VISSIM and then processed to emulate the field measurements for each lane, that is, the loop detector measurements and probe-vehicle measurements. Vehicles are assumed to report their speeds and locations every second. Loop detectors are assumed to report the average traffic speed at their locations every minute. Link data are also used for low resolution estimation as well as provide the ground truth for high resolution estimate. The length of each time step is 1 min. The spatial-temporal size of the discretization in the low resolution estimation is set to be 60s×500m×1lane (the last segment is only 98 m long), which is a reasonable setting for a traditional link-based Eulerian estimator. For high resolution discretization, it is set to be
60s×100m×5lane, since the mainstream of the freeway segment has five lanes. Note that the spatial discretization in the flow direction is also enhanced from 500 m to 100 m, which is an optional choice and should be decided based on the overall characteristics of the freeway stretch studied. Thus, a cell in the link-level estimation is divided into 25 cells in the lane-level estimation. Within each high-resolution cell, the traffic conditions are assumed to be homogeneous.

To obtain statistically meaningful results, ten Monte Carlo runs were carried out using ten randomly selected random seeds. A baseline scenario analysis (referred to as Scenario 0) was conducted on each of these ten simulation runs to demonstrate the general effectiveness of the presented approach (Section 6.3.4). Additional experiments were conducted to evaluate different parameter combinations based on one simulation run. Obviously, enumerating all combinations of parameters would result in a large number of scenarios. To limit the total number of experiments, 37 scenarios (referred to as Scenarios 0 to 36) were selected to profile the big picture of the algorithmic behavior of parameter settings. Specifically, different weight combinations (Scenarios 1 to 25, Section 6.3.5), different penetration rates of probe-vehicles (Scenarios 26 to 29, Section 6.3.6) and different loop detector configurations given multiple probe-vehicle penetration rates (Scenarios 30 to 36, Section 6.3.7) were analyzed. Scenario 0 is common in all analysis. Parameters for the kernel functions are tuned in pre-analysis and are assumed to be the same for all scenarios, with $\sigma_x = 300$ m, and $\sigma_k = 30$ s.

Parameter settings for each scenario are presented in Table 6.1. Let $\alpha_{\text{prior}}$, $\alpha_{\text{loop}}$, $\alpha_{\text{vehicle}}$ and $\alpha_{\text{past}}$ stand for the alpha weights for the corresponding components in Equation
6.5. Since only loop detectors and probe-vehicle sensors are available in our experiments, we only need to specify these four alpha weights. Let $p_{probe}$ represent the penetration rate of the probe vehicles. A realistic assumption of $p_{probe}$ needs to take into consideration of both technical and privacy issues. In this experiment it is assumed to range from 1% to 10%. Let $\Delta_{loop}$ stand for the distance between two neighboring loop detectors, taking values of 500 m, 1000 m and 1500 m. Note that for the configuration of $\Delta_{loop} = 1000$ m, two different schemes are investigated: one has sensors located at 0 m, 1000 m, 2000 m, and 3000 m, while the other has sensors located at 500 m, 1500 m, 2500 m.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$\alpha_{prior}$</th>
<th>$\alpha_{vehicle}$</th>
<th>$\alpha_{loop}$</th>
<th>$\alpha_{past}$</th>
<th>$p_{probe}$</th>
<th>$\Delta_{loop}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>0</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.5</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>7</td>
<td>0.6</td>
<td>0.3</td>
<td>0</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>9</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>11</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
</tbody>
</table>
CONTINUED TABLE 6.1 Scenario Parameter Settings for I-10 Freeway Lane-Level Estimation

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$\alpha_{\text{prior}}$</th>
<th>$\alpha_{\text{vehicle}}$</th>
<th>$\alpha_{\text{loop}}$</th>
<th>$\alpha_{\text{past}}$</th>
<th>$p_{\text{probe}}$</th>
<th>$\Delta_{\text{loop}}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0.3</td>
<td>0</td>
<td>0.6</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>14</td>
<td>0.3</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>15</td>
<td>0.3</td>
<td>0.2</td>
<td>0.4</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>16</td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>17</td>
<td>0.3</td>
<td>0.5</td>
<td>0.1</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>18</td>
<td>0.3</td>
<td>0.6</td>
<td>0</td>
<td>0.1</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>19</td>
<td>0.8/3</td>
<td>0.8/3</td>
<td>0.8/3</td>
<td>0.2</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>20</td>
<td>0.7/3</td>
<td>0.7/3</td>
<td>0.7/3</td>
<td>0.3</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>21</td>
<td>0.6/3</td>
<td>0.6/3</td>
<td>0.6/3</td>
<td>0.4</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>22</td>
<td>0.5/3</td>
<td>0.5/3</td>
<td>0.5/3</td>
<td>0.5</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>23</td>
<td>0.4/3</td>
<td>0.4/3</td>
<td>0.4/3</td>
<td>0.6</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>24</td>
<td>0.3/3</td>
<td>0.3/3</td>
<td>0.3/3</td>
<td>0.7</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>25</td>
<td>0.2/3</td>
<td>0.2/3</td>
<td>0.2/3</td>
<td>0.8</td>
<td>10%</td>
<td>500</td>
</tr>
<tr>
<td>26</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>1%</td>
<td>500</td>
</tr>
<tr>
<td>27</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>2%</td>
<td>500</td>
</tr>
<tr>
<td>28</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>5%</td>
<td>500</td>
</tr>
<tr>
<td>29</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>8%</td>
<td>500</td>
</tr>
<tr>
<td>30</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>1000 (a)</td>
</tr>
<tr>
<td>31</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>1000 (b)</td>
</tr>
<tr>
<td>32</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>10%</td>
<td>1500</td>
</tr>
<tr>
<td>33</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>1%</td>
<td>500</td>
</tr>
<tr>
<td>34</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>1%</td>
<td>1000 (a)</td>
</tr>
<tr>
<td>35</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>1%</td>
<td>1000 (b)</td>
</tr>
<tr>
<td>36</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>1%</td>
<td>1500</td>
</tr>
</tbody>
</table>

(a) Sensor locations: 0 m, 1000 m, 2000 m, 3000 m.
(b) Sensor locations: 500 m, 1500 m, 2500 m.
6.3.2 Filtering masks

The mask sizes are predefined based on the resolution settings and the sensor availability. Considering computational efficiency and effectiveness, a mask size of \(mask_{\text{vehicle}} = 1 \times 3 \times 1\) for the probe-vehicle measurements is assumed. It indicates that (a) only new measurements collected in the current time step are taken into consideration, (b) all of probe-vehicle measurements within one and a half times of the high resolution pixel length (100 m) from the center of the origin \((x^o = (x^i + x^{i+1}) / 2)\) contribute to the state estimation for that origin, and (c) only input data in the same lane are used, since our goal is to differentiate traffic state for each lane. Note that as the penetration rate changes, the same mask size may incur different amount of computational requirements. For real-world practices this mask size needs to be adjusted accordingly. Since the loop detectors are installed at fixed locations, we would like to make the mask size large enough to ensure that every pixel is covered by at least one loop detector, if it is still within a reasonable range. Thus, \(mask_{\text{loop}} = 1 \times 5 \times 1\) is used for loop detectors. Similarly, we use \(mask_{\text{past}} = 1 \times 3 \times 1\) and \(mask_{\text{prior}} = 1 \times 1 \times 1\) for \(\theta_{\text{past}}^{+}\) and \(\theta_{(k,i,l)}^{-}\), respectively.

6.3.3 Performance measures

The performance of the real-time lane-based estimation is evaluated by the root-mean-square error (RMSE) and the percentage of improvement (PoI), as well as speed maps. Since data fusion performance evaluation experiment is usually data-dependent, it is common to use Monte Carlo method to run multiple times of simulation and thus to obtain some results in a statistical sense. Let \(c = 1, 2, ..., C\) index the independent runs. The estimation error resulting from one single run is thus independent of the errors coming
from other runs. Denote the ground truth as $\theta$ and the estimate as $\hat{\theta}$, and thus the RMSE is defined in Equation 6.6. Note that for the parameter sensitivity analysis, there is only one simulation run used and thus $C = 1$. 

$$\text{RMSE}(\hat{\theta}) = \sqrt{E\left[\left(\hat{\theta} - \theta\right)^T \left(\hat{\theta} - \theta\right)\right]} = \left(\frac{1}{C} \sum_{c=1}^{C} \left(\hat{\theta} - \theta\right)^T \left(\hat{\theta} - \theta\right)\right)^{1/2}$$  \hspace{1cm} (6.6)

RMSE is in general the finite-sample approximation of the standard error, thus the smaller the better. However, there is no threshold to determine a single RMSE value to be “good” or “bad”; it may only be used to compare multiple models for the same targeting variables. Moreover, even to compare two alternative models, the RMSE may not reveal all possible issues, if any. Additional measures are required for model performance evaluation. Speed maps are a good complementary measure to the RMSE criterion.

The measure PoI is defined in Equation 6.7, which describes the relative magnitude of the improvement in RMSE compared with the RMSE in the baseline scenario.

$$\text{PoI}(\hat{\theta}) = \frac{\text{RMSE}_{\text{high-resolution}} - \text{RMSE}_{\text{low-resolution}}}{\text{RMSE}_{\text{low-resolution}}}$$  \hspace{1cm} (6.7)

6.3.4 Baseline analysis

The parameters in the baseline analysis are set as Scenario 0 in Table 6.1. An initial allocation of alpha weights was applied as $\alpha_{\text{prior}} = 0.3$, $\alpha_{\text{vehicle}} = 0.3$, $\alpha_{\text{loop}} = 0.3$, and $\alpha_{\text{past}} = 0.1$, meaning that the prior estimate, the probe-vehicle measurements and the loop detector measurements are equally trusted while the past estimate is assigned as 0.1. The sum of the alpha weights is set to one to facilitate computation, but not necessary. The probe-vehicle penetration rate was set at 10% and the loop detector distance at 500 m. Table 6.2 presents the RMSE and PoI results averaged over the ten simulation runs.
Compared to using the link-level estimation as a representation to the lane-level traffic conditions, the lane-based estimation greatly improves the estimation accuracy. PoI is at least 40.73%, and the greatest improvement appears in the Lane 5 estimation, which is 82.53%. The same results are illustrated in Figure 6.3 and Figure 6.4 from different perspectives. Appendix B shows the figures to compare the RMSEs for each lane over the ten simulation runs. It can be found that the lane-level estimation improvement is consistent.

TABLE 6.2 RMSE for I-10W Freeway Speed Estimation Baseline Analysis

<table>
<thead>
<tr>
<th>Average RMSE (km/h)</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Level Estimation</td>
<td>14.33</td>
<td>13.39</td>
<td>13.25</td>
<td>13.72</td>
<td>29.88</td>
</tr>
<tr>
<td>Lane Level Estimation</td>
<td>8.50</td>
<td>7.92</td>
<td>7.74</td>
<td>7.24</td>
<td>5.22</td>
</tr>
<tr>
<td>Percentage of Improvement</td>
<td>40.73%</td>
<td>40.92%</td>
<td>41.58%</td>
<td>47.08%</td>
<td>82.53%</td>
</tr>
</tbody>
</table>

FIGURE 6.3 RMSE for I-10W Freeway Speed Estimation Baseline Analysis
The predominant reasons for the estimation differences are discussed below. The traffic congestion originates from the off-ramp at the Baseline Road exit, which disrupts the traffic flow in Lane 1 (Figure 6.6). The triangular shape represents the formation and discharge of the congestion. The congestion then propagates upstream and extends to inner lanes. It results in different shapes of the speed-drop areas and different levels of congestion severity shown in the speed maps in Figures 6.7 to 6.10. Moreover, Lane 5 is not only the innermost lane but also the HOV lane, which allows only high occupancy vehicles. Thus, the impact of Lane 1 on Lane 5 is even less.

Observing the speed maps one can find that the estimated lane-based speeds for Lane 1 to Lane 5 are much closer to the lane-level ground truths (Figures 6.6 - 6.10), compared to the link-level estimates (Figure 6.5). Lane 5 is mostly in smooth flow condition, but the link-level estimate is biased towards the outer congested lanes. In
contrast, the lane-level estimate for Lane 5 (Figure 6.10-b) largely profiles the ground truth speed map of Lane 5 (Figure 6.10-a). The speed maps for Lane 1 to Lane 4 also indicate that the lane-level estimation is capable of providing satisfactory lane-based results, which is in accord with the numerical results in Table 6.2.

MATLAB-based computation for the lane-level traffic estimation takes less than 27 s for all 230 time-steps (240 min minus 10 min warm-up period), indicating that the computation time in the lane-level estimation only adds a small extra time to the link-level estimation, which is reasonable for online implementation. More sophisticated programming tools could make it run even faster.

FIGURE 6.5 Speed Map: I-10W Freeway Link-Level Estimation
FIGURE 6.6 Speed Map Comparison for I-10W Freeway Lane 1

(a) Lane-level estimation

(b) Ground truth
FIGURE 6.7  Speed Map Comparison for I-10W Freeway Lane 2
FIGURE 6.8  Speed Map Comparison for I-10W Freeway Lane 3

(a) Lane-level estimation

(b) Ground truth
FIGURE 6.9 Speed Map Comparison for I-10W Freeway Lane 4

(a) Lane-level estimation

(b) Ground truth
(a) Lane-level estimation

(b) Ground truth

FIGURE 6.10  Speed Map Comparison for I-10W Freeway Lane 5
6.3.5 Weight analysis

Four experiments on alpha-weight analysis were conducted, corresponding to Scenarios 1 to 25 in Table 6.1, along with Scenario 0. Since the weight analysis is based on one simulation run while the baseline analysis is based on ten independent runs, their results for Scenario 0 are different. The first experiment is to evaluate the filter performance with different combinations of $\alpha_{\text{prior}}$ and $\alpha_{\text{vehicle}}$ given the same settings for all other parameters as Scenario 0. Scenario 1 has $\alpha_{\text{prior}} = 0.6$ and $\alpha_{\text{vehicle}} = 0.0$, and Scenario 6 has $\alpha_{\text{prior}} = 0.0$ and $\alpha_{\text{vehicle}} = 0.6$. Scenarios 2 to 5, along with Scenario 0, have values of $\alpha_{\text{prior}}$ and $\alpha_{\text{vehicle}}$ in between, with an increment/decrement of 0.1. All other settings are the same as that in Scenario 0. It can be found in Table 6.3 and Figure 6.11 that best RMSEs are all from Scenario 0, although its differences from Scenario 3 and Scenario 4 are not obvious. Scenario 1 shows the largest RMSEs, which are still smaller than the link-level estimation RMSEs in Table 6.2. This is because of the utilization of the lane-based loop detector data.

<table>
<thead>
<tr>
<th>Scenario Number*</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.45</td>
<td>11.23</td>
<td>10.79</td>
<td>10.66</td>
<td>17.29</td>
</tr>
<tr>
<td>2</td>
<td>8.82</td>
<td>8.27</td>
<td>8.40</td>
<td>7.68</td>
<td>5.99</td>
</tr>
<tr>
<td>3</td>
<td>8.50</td>
<td>7.96</td>
<td>8.06</td>
<td>7.27</td>
<td>5.37</td>
</tr>
<tr>
<td>0</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
<tr>
<td>4</td>
<td>8.56</td>
<td>8.04</td>
<td>8.04</td>
<td>7.27</td>
<td>5.29</td>
</tr>
<tr>
<td>5</td>
<td>8.65</td>
<td>8.14</td>
<td>8.09</td>
<td>7.37</td>
<td>5.34</td>
</tr>
<tr>
<td>6</td>
<td>8.75</td>
<td>8.26</td>
<td>8.17</td>
<td>7.52</td>
<td>5.42</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings
The second weight analysis is to compare the different combinations between $\alpha_{\text{prior}}$ and $\alpha_{\text{loop}}$. Table 6.4 shows the RMSE values for decreasing $\alpha_{\text{prior}}$ from 0.6 to 0 while increasing $\alpha_{\text{loop}}$ from 0 to 0.6, with an increment/decrement of 0.1 (Scenarios 7 to 12, along with Scenario 0). Figure 6.12 depicts the trend of RMSE. All the first three lanes obtain the best estimates in Scenario 12 ($\alpha_{\text{prior}} = 0.0$ and $\alpha_{\text{loop}} = 0.6$). Lane 4 reaches minimum RMSE in Scenario 11 ($\alpha_{\text{prior}} = 0.1$ and $\alpha_{\text{loop}} = 0.5$), and Lane 5 in Scenario 0 ($\alpha_{\text{prior}} = 0.3$ and $\alpha_{\text{loop}} = 0.3$). The RMSE reduction results from the utilization of the lane-level loop detector data. It is emphasized that, although the prior estimate seems less influential than the empirical measurements, it is still needed not only because it still has some contribution to the lane-level estimation, but also because it can fill the gaps when the empirical measurements are not available (see Section 3.2.6).
TABLE 6.4 RMSE for Weight Analysis: $\alpha_{\text{prior}}$ vs. $\alpha_{\text{loop}}$ (km/h)

<table>
<thead>
<tr>
<th>Scenario Number*</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>9.35</td>
<td>8.90</td>
<td>9.14</td>
<td>7.99</td>
<td>5.56</td>
</tr>
<tr>
<td>8</td>
<td>8.98</td>
<td>8.51</td>
<td>8.68</td>
<td>7.65</td>
<td>5.41</td>
</tr>
<tr>
<td>9</td>
<td>8.70</td>
<td>8.20</td>
<td>8.31</td>
<td>7.40</td>
<td>5.32</td>
</tr>
<tr>
<td>0</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
<tr>
<td>10</td>
<td>8.37</td>
<td>7.78</td>
<td>7.79</td>
<td>7.11</td>
<td>5.28</td>
</tr>
<tr>
<td>11</td>
<td>8.31</td>
<td>7.66</td>
<td>7.63</td>
<td>7.06</td>
<td>5.33</td>
</tr>
<tr>
<td>12</td>
<td>8.31</td>
<td>7.59</td>
<td>7.53</td>
<td>7.07</td>
<td>5.42</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings

FIGURE 6.12 Weight Analysis: $\alpha_{\text{prior}}$ vs. $\alpha_{\text{loop}}$ (see Table 6.1 for scenario settings)

The third weight analysis is to investigate the impact of $\alpha_{\text{loop}}$ and $\alpha_{\text{vehicle}}$ on the filter performance. Table 6.5 shows the RMSE values for decreasing $\alpha_{\text{loop}}$ from 0.6 to 0 while increasing $\alpha_{\text{vehicle}}$ from 0 to 0.6, with an increment/decrement of 0.1 (Scenario 13 to
TABLE 6.5 RMSE for Weight Analysis: $\alpha_{\text{loop}}$ vs. $\alpha_{\text{vehicle}}$ (km/h)

<table>
<thead>
<tr>
<th>Scenario Number*</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>12.26</td>
<td>10.78</td>
<td>10.08</td>
<td>9.59</td>
<td>11.39</td>
</tr>
<tr>
<td>14</td>
<td>8.53</td>
<td>7.83</td>
<td>7.87</td>
<td>7.28</td>
<td>5.71</td>
</tr>
<tr>
<td>15</td>
<td>8.35</td>
<td>7.74</td>
<td>7.80</td>
<td>7.11</td>
<td>5.33</td>
</tr>
<tr>
<td>0</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
<tr>
<td>16</td>
<td>8.74</td>
<td>8.25</td>
<td>8.31</td>
<td>7.41</td>
<td>5.30</td>
</tr>
<tr>
<td>17</td>
<td>9.04</td>
<td>8.58</td>
<td>8.66</td>
<td>7.64</td>
<td>5.37</td>
</tr>
<tr>
<td>18</td>
<td>9.46</td>
<td>8.98</td>
<td>9.11</td>
<td>7.95</td>
<td>5.49</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings

FIGURE 6.13 Weight Analysis: $\alpha_{\text{loop}}$ vs. $\alpha_{\text{vehicle}}$ (see Table 6.1 for scenario settings)

Scenario 18, along with Scenario 0), which is also illustrated in Figure 6.13. The trends of RMSE generally show convexity. Scenario 15 ($\alpha_{\text{loop}} = 0.4$ and $\alpha_{\text{vehicle}} = 0.2$) is the best weight combination for Lane 1 to Lane 4, while Scenario 0 ($\alpha_{\text{loop}} = 0.3$ and $\alpha_{\text{vehicle}} = 0.3$)
yields the minimal RMSE for Lane 5. It implies that both loop detector measurements and the probe-vehicle measurements are influential.

The fourth weight analysis is to investigate the impact of the past high-resolution estimate on the current state estimation. The weight $\alpha_{\text{past}}$ is increased from 0.1 to 0.8, and the remaining fraction is equally assigned to $\alpha_{\text{prior}}$, $\alpha_{\text{loop}}$ and $\alpha_{\text{vehicle}}$. Note that the alpha weights are to assess the reliability and trustiness of the input sources, and should not be understood as how much proportion of the information in the final estimate comes from a certain input source. Apparently, an accurate past estimate is more helpful than a poor past estimate, and extreme values of $\alpha_{\text{past}}$ are expected to deteriorate the estimation results.

Therefore, the RMSE trend regarding $\alpha_{\text{past}}$ is expected to be convex, which is verified in Figure 6.14. Scenario 22 ($\alpha_{\text{past}} = 0.5$) produces the best estimate for Lane 1 to Lane 4, while Scenarios 20 and 21 ($\alpha_{\text{past}} = 0.3/ 0.4$) gives the best estimate for Lane 5. Comparing Table 6.6 with Table 6.3, Table 6.4 and Table 6.5, we can see that they are also the smallest

<table>
<thead>
<tr>
<th>Scenario No.</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
<tr>
<td>19</td>
<td>8.09</td>
<td>7.60</td>
<td>7.70</td>
<td>6.95</td>
<td>5.25</td>
</tr>
<tr>
<td>20</td>
<td>7.78</td>
<td>7.33</td>
<td>7.50</td>
<td>6.77</td>
<td>5.24</td>
</tr>
<tr>
<td>21</td>
<td>7.57</td>
<td>7.14</td>
<td>7.38</td>
<td>6.66</td>
<td>5.24</td>
</tr>
<tr>
<td>22</td>
<td>7.44</td>
<td>7.04</td>
<td>7.35</td>
<td>6.63</td>
<td>5.26</td>
</tr>
<tr>
<td>23</td>
<td>7.44</td>
<td>7.05</td>
<td>7.44</td>
<td>6.71</td>
<td>5.32</td>
</tr>
<tr>
<td>24</td>
<td>7.63</td>
<td>7.22</td>
<td>7.70</td>
<td>6.97</td>
<td>5.44</td>
</tr>
<tr>
<td>25</td>
<td>8.19</td>
<td>7.76</td>
<td>8.32</td>
<td>7.61</td>
<td>5.70</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings
From the above experimental analysis we can conclude that (a) all types of inputs contribute to the overall estimation accuracy, and (b) it would be helpful to calibrate the alpha weights for each lane separately.

6.3.6 Penetration rate analysis

The penetration rate analysis investigates the influence of different probe vehicle participation levels \( p_{\text{probe}} \) on the lane-level estimation performance. Penetration rates of 1\%, 2\%, 5\%, 8\% and 10\% are considered along with other baseline default settings.

Table 6.7 summaries the RMSE results. The monotonic trend is clear in Figure 6.15, that is, the larger the penetration rate, the more accurate the estimation. Lane 5 results show a significant deduction of RMSE when the penetration rate increases from 1\% to 5\%, and
its RMSE decreases faster than other lanes. It is also noted that the decreasing rate of RMSE decreases as the penetration rate increases. In Appendix C, speed maps show that with different penetration rates, the resulting speed maps show different accuracy levels. For each of the five lanes, \( p_{\text{probe}} = 1\% \) yields blurred speed maps, due to the fact that the probe vehicle data is not sufficient. \( p_{\text{probe}} = 10\% \) yields speed maps closest to the ground truth.

**TABLE 6.7 RMSE for Penetration Rate Analysis: \( p_{\text{probe}} \) (km/h)**

<table>
<thead>
<tr>
<th>Scenario No.</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>11.66</td>
<td>10.35</td>
<td>9.83</td>
<td>9.33</td>
<td>10.90</td>
</tr>
<tr>
<td>27</td>
<td>11.01</td>
<td>9.73</td>
<td>9.59</td>
<td>8.80</td>
<td>9.72</td>
</tr>
<tr>
<td>28</td>
<td>9.85</td>
<td>8.94</td>
<td>8.94</td>
<td>8.05</td>
<td>6.80</td>
</tr>
<tr>
<td>29</td>
<td>8.93</td>
<td>8.28</td>
<td>8.40</td>
<td>7.45</td>
<td>5.69</td>
</tr>
<tr>
<td>0</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings

**FIGURE 6.15 Penetration Rate Analysis: \( p_{\text{probe}} \) (see Table 6.1 for scenario settings)**
6.3.7 Loop detector configuration analysis

Loop detector configuration analysis is intended to investigate the impact of different spatial distances or locations between neighboring loop detectors. The tested scenarios include the combinations of three different loop detector distances \( \Delta x_{\text{loop}} = 500 \text{ m}, 1000 \text{ m}, \text{and} 1500 \text{ m} \) given two levels of penetration rate \( p_{\text{probe}} = 10\% \) and \( p_{\text{probe}} = 1\% \). Meanwhile, for \( \Delta x_{\text{loop}} = 1000 \text{ m} \), two different configurations were implemented. The first one \( \Delta x_{\text{loop}}^1 = 1000 \text{ m} \) locates loop detectors at \( 0 \text{ m}, 1000 \text{ m}, 2000 \text{ m}, \text{and} 3000 \text{ m} \) of the freeway stretch, while \( \Delta x_{\text{loop}}^2 = 1000 \text{ m} \) has the loop detectors located at \( 500 \text{ m}, 1500 \text{ m}, \text{and} 2500 \text{m} \). Results are summarized in Table 6.8 and illustrated in Figures 6.16. Generally, \( \Delta x_{\text{loop}} = 500 \text{ m} \) performs better than \( \Delta x_{\text{loop}} = 1500 \text{ m} \) for both of the penetration rates. However, the performance with \( \Delta x_{\text{loop}} = 1000 \text{ m} \) varies. We can see that \( \Delta x_{\text{loop}}^2 = 1000 \text{ m} \) yields smaller RMSE than \( \Delta x_{\text{loop}}^1 = 1000 \text{ m} \). A reasonable inference

<table>
<thead>
<tr>
<th>Scenario Number*</th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p = 1% )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>11.26</td>
<td>10.23</td>
<td>9.80</td>
<td>9.29</td>
<td>10.67</td>
</tr>
<tr>
<td>30</td>
<td>12.99</td>
<td>12.39</td>
<td>12.21</td>
<td>11.90</td>
<td>12.58</td>
</tr>
<tr>
<td>31</td>
<td>12.57</td>
<td>10.92</td>
<td>10.37</td>
<td>9.54</td>
<td>11.83</td>
</tr>
<tr>
<td>32</td>
<td>12.73</td>
<td>11.94</td>
<td>11.73</td>
<td>11.40</td>
<td>13.68</td>
</tr>
<tr>
<td>( p = 10% )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>8.50</td>
<td>7.96</td>
<td>8.01</td>
<td>7.22</td>
<td>5.27</td>
</tr>
<tr>
<td>34</td>
<td>9.65</td>
<td>9.06</td>
<td>9.19</td>
<td>7.92</td>
<td>5.36</td>
</tr>
<tr>
<td>35</td>
<td>8.47</td>
<td>7.89</td>
<td>8.01</td>
<td>7.27</td>
<td>5.27</td>
</tr>
<tr>
<td>36</td>
<td>9.36</td>
<td>8.75</td>
<td>8.89</td>
<td>7.69</td>
<td>5.42</td>
</tr>
</tbody>
</table>

*see Table 6.1 for scenario settings
FIGURE 6.16  Loop Detector Configuration Analysis (see Table 6.1 for scenario settings)
Continued Figure 6.16  Loop Detector Configuration Analysis (see Table 6.1 for scenario settings)
is that a better locations of loop detectors could be at the upstream of the bottleneck (2500 m). Although no conclusive trend is shown, they do suggest that both the distance and location of the loop detectors are influential on estimation performance.

6.3.8 Discussions

This simulation-based analysis shows the advantages of the new data fusion approach for real-time lane-based traffic state estimation. The baseline analysis demonstrates the improvement in estimation accuracy in general. Analysis of different alpha-weight combinations show that the weights should be calibrated towards minimizing RMSE values of each individual lane separately. The penetration rate analysis shows that the increasing number of participating vehicles leads to reductions in RMSE. Although no common trends can be found in the loop detector configuration analysis, it indicates that both the location and distance influence the estimation performance.

There are also some aspects that are worth noticing for future practice. In this experimental analysis, the high-resolution estimation only enhances the low resolution estimation in terms of the spatial dimensions. However, similar enhancement could be applied to the temporal dimension. The current time step length is 60s. With the bi-level approach, it can be reduced to fewer seconds such as 30s in the lane-level estimation. In addition, the kernel function parameters have been calibrated in a trial-and-error manner. They are preset to be the same for all scenarios. In practice, they may also be calibrated specifically for each lane. It is worth emphasizing that the total of 37 scenarios cannot enumerate all circumstances; they are only to profile the general behavior of the new data fusion approach.
6.4 Experimental analysis: US101 freeway scenario

This section presents experimental analysis of the developed data fusion approach for the lane-level traffic estimation using real traffic data. The lane-level estimation in this section takes the link-level estimation results in Section 5.3 as inputs; therefore, the rest of this section only focuses on the lane-level estimation.

6.4.1 Background

The experimental analysis was carried out in MATLAB on a real traffic data set for a segment of the US Highway 101 (Hollywood Freeway) in Los Angeles, California, United States, from the Federal Highway Administration’s (FHWA) Next Generation Simulation (NGSIM) project, referred to as US101 data. The time frame starts at 7:50AM and ends at 8:35AM in the morning of June 15, 2005, totally 45 minutes. In the first 15 minutes, the freeway segment experiences a transition from free flow condition to congestion and stays congested for the rest 30 minutes. The initial 30 s period is not used in estimation because of partially missing data. The freeway segment under investigation (Figure 6.17) is approximately 640m (2100 feet), and has five main lanes including one HOV lane. It also has an on-ramp at location 176 m (578 feet) from start and an off-ramp at location 389 m (1276 feet) from start, and has an auxiliary lane between the on-ramp and off-ramp. The measurement data was first collected by video cameras which had full coverage of the freeway segment, and then transcribed to vehicle trajectory data, including information about vehicle identification number, time, position coordinates, lane number, speed, spacing, and so on. The time frequency is 1/10th second. (NGSIM)
6.4.2 Experimental setup

Recall that the data fusion approach for lane-based estimation adopts a bi-level architecture. In this experiment, the link-level estimator is assumed to be the MEKF-based real-time Lagrangian traffic state estimator developed in Chapter 5, given the assumptions that the loop detectors are installed 200m from each other with a reporting period of 20 s, and that the vehicle penetration rate is 10% (see Section 5.3) with a reporting period of 5 s. The time step length of the Lagrangian estimation system is 1 s.

Since the link-level estimator is conducted in the Lagrangian coordinate system, the estimated speeds need to be transformed into the Eulerian format, i.e., fixed-location freeway segments with equal lengths. In the lane-based estimation, the low resolution (link-level) is set to be 5s×40m×1lane, and the high resolution (lane-level) is chosen to be 5s×40m×5lane. This is to analyze how the new estimator performs when the temporal resolution and the spatial resolution in the flow direction remain the same between the two levels of estimation. One may also notice that the link-level spatial-temporal resolution is different from the one used in Chapter 5 (20s×80m). This is because, following the similar procedure presented in Section 4.5.2.2 for choosing a best resolution, it has been found that
the lane-level traffic conditions are even less homogeneous than the link-level conditions, and thus the resolution of $5s \times 40m$ works better for this experiment.

The whole trajectory data set is processed into the resolution of $5s \times 40m \times 5\text{lane}$ to represent the ground truths. Lane-level loop detector data and probe-vehicle measurements are also emulated using the trajectory data (see Section 5.4.1.3). The Lagrangian estimation results from Chapter 5 is transformed into the resolution of $5s \times 40m \times 1\text{lane}$, serving as the comparison basis and the prior estimate input. Other calibrated parameters (e.g., the alpha values, the scaling parameters, and the masks) can be found in Appendix D. Note that here the alpha weights and scaling parameters are tuned for each lane, respectively.

6.4.3 Performance measures

The performance criteria are similar to that in Section 6.3.3, with the only exception that in this experiment we draw ten random vehicle measurement sets as ten random instances of probe-vehicle data to account for sampling randomness of GPS equipped vehicles. For RMSE refer to Equation 6.6, and for PoI refer to Equation 6.7. As RMSE and PoI are not perfect performance evaluation measures, comparison of speed maps, density maps, and fixed-location speed/density plots serve as a good supplementary performance evaluation.

6.4.4 Results and discussions

Table 6.9 presents the summary of RMSE and PoI comparisons. It can be seen that the lane-level estimation can improve the estimation accuracy by at least 25.13% (Lane 3). The largest improvement still appears for Lane 5. Figures 6.18 and 6.19 illustrate these results, from which we can see that the PoI curve presents a convex shape, with Lane 3 showing least improvement. This is because Lane 1 is the outmost lane where traffic flows
are merging and diverging, while Lane 5 is the HOV lane which only allows high-occupancy vehicles to travel through. Both of them have their own distinctive characteristics making them greatly different from the average conditions.

TABLE 6.9 RMSE for US101 Freeway Lane-Level Speed Estimation (km/h)

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>POI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Link-Level</td>
<td>Lane-Level</td>
</tr>
<tr>
<td>Lane 1</td>
<td>8.17</td>
<td>5.80</td>
</tr>
<tr>
<td>Lane 2</td>
<td>6.08</td>
<td>4.46</td>
</tr>
<tr>
<td>Lane 3</td>
<td>5.40</td>
<td>4.00</td>
</tr>
<tr>
<td>Lane 4</td>
<td>5.80</td>
<td>3.92</td>
</tr>
<tr>
<td>Lane 5</td>
<td>8.14</td>
<td>4.72</td>
</tr>
</tbody>
</table>

FIGURE 6.18 RMSE for US101 Freeway Lane-Level Speed Estimation
FIGURE 6.19 PoI for US101 Freeway Lane-Level Speed Estimation

FIGURE 6.20 Speed Map: US101 Freeway Link-Level Estimation
(a) Lane-level estimation

(b) Ground truth

FIGURE 6.21 Speed Map Comparison for US101 Freeway Lane 1
FIGURE 6.22 Speed Map Comparison for US101 Freeway Lane 2
(a) Lane-level estimation

(b) Ground truth

FIGURE 6.23  Speed Map Comparison for US101 Freeway Lane 3
FIGURE 6.24 Speed Map Comparison for US101 Freeway Lane 4
FIGURE 6.25  Speed Map Comparison for US101 Freeway Lane 5

(a) Lane-level estimation

(b) Ground truth
Figures 6.20 to 6.25 present speed maps of the link-level estimation, the lane-level estimation and the lane-level ground truth for all five lanes, respectively. We can see that the traffic condition in Lane 1 fluctuates more frequently than the link-level estimation, and is also more congested, while Lane 5 is more homogeneous and only has the main congestion waves. Other lanes have traffic conditions between these two extremes. Overall, the lane-level estimation is much closer to the ground truth than the link-level estimation.

6.5 Summary

This chapter presents the data-driven approach for the lane-level traffic estimation in the bi-level multi-sensor data fusion approach presented in Chapter 3. A spatial-temporal filter is developed for parallel lane-based estimation. Simulation-based experimental analysis demonstrates its effectiveness and efficiency. Sensitivity analysis indicates the benefits of weight calibration for each lane respectively, and higher probe vehicle penetration rate. A second experimental analysis based on US101 real traffic data further demonstrates the validity of the approach.
CHAPTER 7

CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

7.1 Research conclusions and contributions

This dissertation has developed a new multi-sensor data fusion approach for real-time lane-based traffic state estimation, which provides a practical solution to the problem that cannot be addressed by traditional link-based traffic estimation approaches. The research contributions and conclusions are presented as follows.

First, this research presents a thorough literature review on freeway traffic state estimation. It concludes that, (a) most of the estimation validation experiments have been conducted using a single data source and/or simulation data; (b) almost all the real-time estimators are model-based approaches which are only applicable to freeway links, (c) the data driven approaches have been mainly used in traffic state reconstructions and are rarely applied to real-time estimations; (d) little research has been done on real-time lane-based traffic state estimation; and (e) the Lagrangian macroscopic traffic flow modeling approach is relatively new and little attention has been paid to build higher order Lagrangian models. (Chapter 2)

Second, this research developed a data fusion approach, adopting a recursive bi-level procedure which combines a model-based approach and a data-driven approach, for real-time lane-based traffic state estimation. (Chapter 3)

Third, this research developed a second-order Lagrangian traffic flow model. A dynamic speed equation is derived from the realistic modeling of the drivers’ anticipation and reaction delay. (Chapter 4)
Fourth, this research presents some enhancements for modeling freeway traffic network discontinuities. These enhancements are practical methods that can facilitate computer-based simulation. (Chapter 4)

Fifth, this research conducted a lane-drop simulation based experiment for the proposed second-order traffic flow model, which shows that the model is not only able to reproduce the formation and discharge of congestion as the first-order model does, but also able to naturally create stop-and-go waves which are amplified as the waves propagate backward upstream; (Chapter 4)

Sixth, this research analyzed the US101 real traffic data, and used this data set to represent the ground truth and to emulate the heterogeneous measurements. Different spatial-temporal resolution were analyzed for ground truth representation and the research concluded that because of the rapid fluctuations in US101, the typical setting of 60 s temporal discretization is not satisfactory, but a finer resolution of 20 s provides better representation. (Chapter 4)

Seventh, this research conducted a US101 real traffic data based experiment for the second-order traffic flow model, which showed that the model represents the rapid extreme fluctuations better, but performs very similar to the first-order model at other times. (Chapter 4)

Eighth, this research implemented the traffic discontinuities in computer-based simulation as well. The research also shows that the node network model for US101 can effectively reflect the traffic flow patterns that occurs in discontinuities in the real world traffic system. (Chapter 4)
Ninth, this research identified possible solutions for multi-sensor traffic estimation, and developed a multi-sensor EKF (MEKF) algorithm which can conveniently incorporate heterogeneous data from multiple sources. Currently, little attention has been paid to develop a general approach for multi-sensor measurements in model-based traffic state estimation. (Chapter 5)

Tenth, this research developed a real-time link-level traffic state estimator using the developed second-order Lagrangian traffic flow model and the MEKF algorithm, assuming only loop detectors and probe-vehicle sensors are available. (Chapter 5)

Eleventh, this research conducted a US101 real traffic data based experiment for the developed MEKF-based estimator, which showed that the link-based estimation accuracy generally improves by incorporating heterogeneous measurements, when comparing against the results using only a single sensor type. (Chapter 5)

Twelfth, this research presents a data-driven approach with a spatial-temporal algorithm to fuse multiple information sources and obtain lane-based estimates. This data-driven approach is computationally efficient to satisfy online implementation requirements, and also has the advantage of flexible discretization solutions. (Chapter 6)

Thirteen, this research conducted an Interstate-10 freeway scenario simulation-based experiment for the developed data-driven approach, which thoroughly demonstrates the effectiveness and efficiency of the approach for real-time lane-based traffic estimation. (Chapter 6)

Fourteenth, this research also conducted a US101 real traffic data based experiment for the developed multi-sensor data fusion approach which has also demonstrated the benefits of the approach. (Chapter 6)
Fifteenth, this research presented the potential directions for future research in the traffic estimation area. (Chapter 7)

To sum up, this research developed a bi-level data fusion for real-time lane-based traffic state estimation for freeways. The simulation-based and the real traffic data based experiments demonstrated that the developed approach is effective for real-time freeway traffic state estimation, and contributes substantially to the state of art in traffic estimation.

7.2 Directions for future research

In the lane-level data-driven approach, the high-resolution estimation only enhances the low resolution estimation in terms of the spatial dimensions. The proposed algorithm is flexible in terms of spatial-temporal discretization resolution. It would be worthwhile in future research to extend these results in terms of temporal discretization. Furthermore, this research focused only on the estimation of mean space speeds for lanes. The generalization for estimating other traffic characteristics (such as density) will be worth investigating.

In evaluating the MEFK-based estimator, the emulated measurements are assumed to be the true values, and the measurement noises are only associated with the sensing mechanisms. However, in the real world, the sensors can have different types of noises, such as sensor failure and influences from weather and environment. In other words, the “real traffic data experiment” in this research is not truly real traffic data, but data emulated from observing real traffic. Field experiments are needed to collect traffic measurements using real traffic sensors, and investigation of the impact of different measurement noise levels on the performance of the MEKF-based estimator would be useful. In addition, the developed MEKF processes the multi-sensor measurements sequentially in the
measurement update stage at each time step; however, there are other possible data fusion solutions for incorporating heterogeneous data (Section 5.2.3). Efforts to compare the MEKF-based estimator with other possible multi-sensor solutions for model-based state estimation should be very informative.

In the experimental analysis using real traffic data, the US101 scenario, which is 640 m in total length with one on-ramp and one off-ramp, is a relatively simple case compared to various real world scenarios (e.g., multiple-link merges). With more resources and efforts, future research can apply the proposed approach to complex freeway networks to evaluate the performances of the developed traffic flow model, and both the link-based and lane-based traffic state estimators. Moreover, with a larger real traffic data set, the experiment should be conducted in a way that the model parameters are calibrated using a training data set while the performance evaluation is conducted using a separate testing data set to obtain the generalized estimation error. In the I-10 freeway scenario baseline analysis, ten data sets were generated using ten different random seeds. The lane-level estimation parameters were calibrated using a single simulation run, and then other nine simulation runs utilized the parameters. However, the experiments with US101 data conducted in this research did not differentiate the training data set from the testing data set because of the limited data source. Thus the estimation error is the optimistic error. It would be worthwhile to evaluate the estimation performance using another independent data set.
BIBLIOGRAPHY


Treiber, M. and Helbing, D. (2002). Reconstructing the spatio-temporal traffic dynamics from stationary detector data. *Cooper@ tive Tr@ ns@ tion Dyn@ mics*, 1(3):1-24.


169


APPENDIX A

KALMAN FILTER
Algorithm A.I: Kalman Filter (Kalman, 1960)

Initialization: set $k = 0$, then apply

$$
\theta_0 = E[\theta_0],
$$
$$
P_{0_0} = E[P_{0_0}].
$$

Recursive estimation: for each time step, apply

Time update:

$$
\theta^-_{k+1} = G_k^+ \theta^+_{k},
$$
$$
P^-_{0_k} = G_k^+ P^+_{0_k} G_k^T + Q.
$$

Measurement update:

$$
K_{k+1} = P^-_{0_k} H_k H_k^T P^-_{0_k} + R^{-1},
$$
$$
\theta^+_{k+1} = \theta^-_{k+1} + K_{k+1} [Z_{k+1} - H_{k+1} \theta^-_{k+1}],
$$
$$
P^+_{0_k} = [I - K_{k+1} H_{k+1}] P^-_{0_k}.
$$

$k = k + 1$. Return to the time update step.

See Section 5.2.1 for notations.
APPENDIX B

RMSE COMPARISON FOR BASELINE ANALYSIS

IN I-10 FREEWAY LANE-LEVEL ESTIMATION
FIGURE B.1 Comparison of RMSE for Lane 1

FIGURE B.2 Comparison of RMSE for Lane 2
FIGURE B.3 Comparison of RMSE for Lane 3

FIGURE B.4 Comparison of RMSE for Lane 4
FIGURE B.5 Comparison of RMSE for Lane 5
APPENDIX C

SPEED MAPS FOR PENETRATION ANALYSIS

IN I-10 FREEWAY LANE-LEVEL ESTIMATION
FIGURE C.1  Speed Maps for Lane 1 under Different Penetration Rates
FIGURE C.2 Speed Maps for Lane 2 under Different Penetration Rates
FIGURE C.3  Speed Maps for Lane 3 under Different Penetration Rates
FIGURE C.4 Speed Maps for Lane 4 under Different Penetration Rates
FIGURE C.5 Speed Maps for Lane 5 under Different Penetration Rates
APPENDIX D

PARAMETER SUMMARY FOR

US101 FREEWAY LANE-LEVEL TRAFFIC ESTIMATION

183
TABLE D.1 Parameter Settings for US101 Freeway Lane-Level Estimation

<table>
<thead>
<tr>
<th></th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
<th>Lane 4</th>
<th>Lane 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{\text{prior}}$</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>$\alpha_{\text{vehicle}}$</td>
<td>0.5</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>$\alpha_{\text{loop}}$</td>
<td>0.4</td>
<td>0.5</td>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$\alpha_{\text{past}}$</td>
<td>0.09</td>
<td>0.08</td>
<td>0.07</td>
<td>0.15</td>
<td>0.19</td>
</tr>
<tr>
<td>$p_{\text{probe}}$</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>$\sigma_{x}^{\text{loop}}$</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\sigma_{x}^{\text{veh}}$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>$\sigma_{t}^{\text{past}}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$\text{mask}_{\text{prior _vd_sk}}$</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>1x1x1</td>
</tr>
<tr>
<td>$\text{mask}_{\text{loop _vd_sk}}$</td>
<td>2.5x1x1</td>
<td>2.5x1x1</td>
<td>2.5x1x1</td>
<td>5x1x1</td>
<td>5x1x1</td>
</tr>
<tr>
<td>$\text{mask}_{\text{veh _vd_sk}}$</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>1x1x1</td>
<td>2x1x1</td>
</tr>
<tr>
<td>$\text{mask}_{\text{past _vd_sk}}$</td>
<td>1x3x1</td>
<td>1x3x1</td>
<td>1x3x1</td>
<td>1x3x1</td>
<td>1x3x1</td>
</tr>
</tbody>
</table>