HYBRID STATE ESTIMATION APPLICATIONS FOR JOINT TRAFFIC MONITORING AND INCIDENT DETECTION

BY

REN WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Daniel Work, Chair
Professor Yanfeng Ouyang
Professor Joshua Peschel
Professor Richard Sowers

ABSTRACT

This dissertation is motivated by the practical problems of highway traffic estimation and incident detection using measurements from various sensor types. It proposes a framework to jointly estimate the traffic state and incidents in a hybrid state estimation problem where a continuous variable models the traffic state and a discrete model variable identifies the location and severity of an incident. Clearly, knowledge of an incident can improve post–incident traffic state estimates. Moreover, knowledge of the traffic state can be used to improve detection of incidents, by observing when the predicted traffic state differs significantly from the observed measurements.

Two macroscopic traffic flow models are deployed to describe the evolution of traffic. Both the first order model and the second order model are extended to hybrid models by embedding a model parameter to denote the number of lanes open along the highway. The resulting traffic incident models are capable of describing traffic dynamics under both non–incident and incident scenarios that result in lane blockages.

Next, several nonlinear filters are proposed to solve the joint traffic state estimation and incident detection problem. First, a multiple model particle filter and an interactive multiple model ensemble Kalman filter are proposed, where the particle filter or the ensemble Kalman filter are used to accommodate the nonlinearity of the traffic model, and multiple model methods are deployed to address the switching dynamics of traffic when incidents occur. Next, the multiple model particle filter is extended to a multiple model particle smoother to improve the estimation accuracy when data is limited. Finally, a variant of the multiple model particle filter, called the efficient multiple model particle filter, is developed for field implementations, which requires significantly less computation time compared to the other

filters considered in this thesis.

To validate the framework, the proposed nonlinear filters are implemented on the first order and second order traffic flow models, and tested in the microscopic traffic simulation software CORSIM and on field data collected on I–880 in California, which includes density measurements from inductive loops and speed measurements from GPS equipped vehicles. The results show that with either traffic flow model, the proposed traffic estimation algorithms are capable of jointly estimating the traffic state and detecting incidents when the traffic flow is high (i.e., when an incident results in congestion). The proposed algorithms are also compared with existing algorithms that independently estimate the traffic state or incidents. The results show that jointly estimating the state and incidents in one algorithm may perform better than two dedicated algorithms working independently, especially when loop detectors are sparse and the penetration rate of GPS equipped vehicles is high.

To my parents.

Acknowledgements

I would like to express my deepest gratitude to my PhD advisor Professor Daniel Work, who gave me the opportunity to join his research group and served as an exceptional mentor during my PhD. I will always be indebted to him for everything he has taught me and for every effort he gave to help me become a better version of myself.

I would like to thank Professor Richard Sowers, Professor Yanfeng Ouyang, and Professor Joshua Peschel for serving on my dissertation committee. Professor Sowers advised my research while I was pursuing the Master's degree in Industrial Engineering. His guidance on particle filtering and probability theory benefited me tremendously for my research on traffic estimation. Professor Ouyang introduced me the area of logistic systems and taught me various interesting techniques, such as dynamic programming, which I found very useful for solving a variety of problems in my later studies. Professor Peschel's expertise on unmanned aerial vehicles led me to the area of mobile sensing. During the time participating in the Global Leaders program, Professor Peschel gave me valuable suggestions on how unmanned aerial vehicles can be used to study traffic impacts on the environment.

I would like to express my gratitude to Dr. Shimao Fan, who taught me the second order traffic flow model and shared with me valuable experience on model calibration. The implementation of the second order traffic model on field data could not be possible without his support.

I would like to thank my research group. As the first PhD student in the group, it is pleasant to see how the group grows, and it is great to have a fantastic team to work with. Special thanks goes to my friends, who made my life wonderful at UIUC.

I would also like to acknowledge support from the NEXTRANS University Transportation Center under Grant No. DTRT12-G-UTC05 of the U.S. Department of Transportation,

which provided support for my research presented in this thesis.

Finally, I would like to thank my parents for their endless love and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The objective of traffic estimation is to monitor the traffic state. The traffic state (e.g., traffic density or velocity along the roadway) can be estimated with traffic models and nonlinear filtering techniques, where traffic models are used to predict the traffic state given the initial and boundary conditions, and nonlinear filters are used to adjust the predictions by incorporating information from real time sensor measurements. Within the transportation community, traffic is modeled at different scales. At one end of the spectrum are microscopic traffic models, which describe the dynamics of each individual vehicle and their interactions [2, 3, 4]. At the other end of the spectrum are macroscopic traffic models which describe traffic as a continuum and represent the traffic state in terms of aggregate quantities such as traffic density, flow, and average speed [4, 5, 6, 7]. Because of the reduction of the number of state variables compared to the microscopic model, macroscopic traffic models are commonly used for real time traffic estimation problems [8, 9, 10, 11].

Traffic estimation techniques have advanced rapidly in recent years because of developments in nonlinear filtering techniques, advances in sensing technologies such as GPS data from cellphones, and the availability of cheap computing and communication resources to fuse

the data and models together. Most existing traffic estimation algorithms (e.g., [8, 9, 10, 11]) assume time–invariant parameters in the traffic model and do not account for changes in the dynamics on the highway caused by traffic incidents. While a calibrated traffic estimation model can perform well under normal traffic operating conditions, it can provide poor traffic state estimates when a traffic incident occurs, because the traffic model does not contain any dynamics to describe the traffic flow evolution under incidents. On the other hand, traffic incident detection is also a widely studied problem in the field of transportation engineering [12, 13, 14, 15, 16]. Traffic incident detection algorithms report incidents by observing when the sensor measurements significantly deviate from normal values. While existing incident detection algorithms can detect incidents with reasonable detection rates and false alarm rates, they do not offer a complete picture of the impact of the traffic incident on the traffic conditions or the resulting congestion [16].

This thesis is motivated by the fact that jointly estimating incidents and the traffic state can improve both incident detection capabilities and the traffic state estimates. Clearly, knowledge of an incident can improve post–incident traffic state estimates. Moreover, knowledge of the traffic state can be used to improve detection of incidents by observing when the predicted traffic state differs significantly from the observed measurements. To address the problem of jointly estimating incidents and the traffic state, this thesis poses the problem as a hybrid state estimation problem. The deterministic macroscopic traffic flow models are extended to hybrid state traffic incident models, where a continuous variable denotes the traffic state and a discrete model variable identifies the location and severity of an incident. Then, several nonlinear filtering techniques are proposed to solve the hybrid state estimation problem, using either a first order or a second order traffic flow model in the filters. Finally, the performance of the filters are compared in numerical experiments and using field data.

Figure 1.1 provides an overview of the hybrid state estimation problem for joint traffic state estimation and incident detection. Given the roadway properties (e.g., length of the road, number of lanes available), a traffic flow model can be constructed to describe the

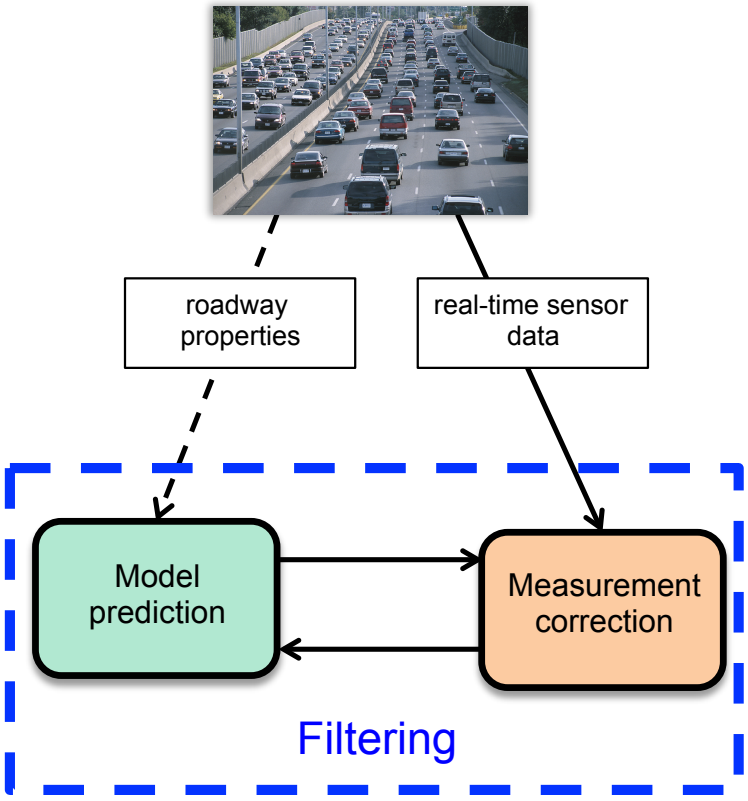


Figure 1.1: An overview of the hybrid state estimation problem for joint traffic state estimation and incident detection.

traffic evolution on the highway. In order to describe traffic dynamics under incidents, the traffic flow model is extended to a hybrid state traffic incident model by embedding a discrete model parameter to encode the possibility of a lane blockage. At each time step, the hybrid traffic model predicts the traffic state for the next time step for possible models involved in the system (i.e., both the normal traffic condition and possible incident scenarios). When the measurements are received, the information can be used to infer the existence of traffic incidents and improve the prediction by the traffic flow model. This step is achieved by using nonlinear filtering techniques. Finally, the estimated traffic state for the current time step will be used to predict the future traffic state for the next time step.

1.2 Problem statement and solution framework

In this thesis, the joint traffic state estimation and incident detection problem is posed as a hybrid state estimation problem. The evolution and observation equations of the hybrid system are given by:

$$\begin{aligned} \gamma^n &= \Pi\left(\gamma^{n-1}\right), \\ x^n &= f\left(x^{n-1}, \gamma^n\right) + \omega^{n-1}, \\ z^n &= h^n\left(x^n, \gamma^n\right) + \nu^n. \end{aligned} \tag{1.1}$$

The variable γ is known as the model variable, which is a time varying vector and denotes the integer number of lanes open along the freeway during the time period $(t^{n-1}, t^n]$. The first equation is known as the model transition equation, which describes the evolution of the model variable γ. The second equation is known as the system evolution equation, which describes the evolution of traffic. Here, the continuous variable x^n denotes the traffic state (e.g., a vector of densities and velocities along the roadway). The traffic evolution equations are constructed from a macroscopic traffic flow model denoted by f, which evolves the traffic state x^{n-1} at discrete time step $n-1$ to time n. When the model variable γ^n is embedded

into the traffic flow model, the traffic model f becomes capable of predicting traffic states under incident conditions. The term z^n denotes the measurements received from various sensor types. For example, it can be a vector of speed or density measurements collected from inductive loops or GPS equipped vehicles. The function h^n is a nonlinear observation operator that relates the system state with the measurements, and is time varying when GPS data in vehicles are used because the sensors move in the traffic stream. The variable ω^n is a random variable representing the noise associated with the traffic model, and ν^n is a random variable that describes the measurement noise. An additive noise model is used for both the evolution and observation equations.

Given the evolution observation system (1.1), the joint traffic state estimation and incident detection problem can be posed as the problem of estimating the traffic state variable x^n and the model variable γ^n given measurements $\{z^1, \cdots, z^n\}$. The hybrid state estimation problem is challenging because of the following. First, the traffic flow model f is nonlinear and there is a switching dynamic associated with the traffic model due to incidents. As a result, standard linear minimal variance techniques such as the Kalman filter are not directly applicable [8, 17]. Second, the estimation of the model parameter γ^n is challenging when data is limited. For example, when two sensors are located far apart and an incident occurs in the middle, the algorithm cannot identify the location and the severity of the incident in real time, because it takes time for the resulting congestion to propagate to the sensors, where derivations from typical conditions can be detected. Third, when traffic is light, a traffic incident may not result in congestion if the remaining lanes have enough capacity to accommodate all the traffic. In this case, the estimation algorithms are not able to detect the model switch because the traffic conditions remain typical at the sensor locations.

In this thesis, several multiple model nonlinear filtering algorithms are proposed to solve the hybrid state estimation problem for system (1.1), where multiple model methods are used to address the switching dynamics of the traffic model, and nonlinear filters, such as the *particle filter* (PF) and the *ensemble Kalman filter* (EnKF), are used to handle the

nonlinearities of the traffic models. When data is limited, the multiple model nonlinear filter is extended to a multiple model nonlinear smoother to improve the estimation accuracy. For field implementation, a variation of the *multiple model particle filter* (MMPF), called *efficient multiple model particle filter* (EMMPF), is proposed to reduce the computational cost for solving the hybrid state estimation problem. The proposed algorithms are implemented on both the first order traffic model and the second order traffic flow model, and tested with incident data from a microscopic simulation software and field data collected on a segment of I–880 in California.

1.3 Contributions of the thesis

The main contributions of this thesis are summarized as follows:

- A hybrid state estimation problem is posed to jointly estimate the traffic state and detect incidents, where the estimate of the continuous variable indicates the traffic state and the estimate of the discrete variable identifies the location, severity, and duration of a traffic incident. The framework is general and can be implemented using a variety of macroscopic traffic flow models and nonlinear filtering techniques, several of which are developed in this thesis.

- A multiple model particle filter and an *interactive multiple model* (IMM) ensemble Kalman filter are proposed to handle the nonlinearity and switching dynamics of the traffic model with incidents. It is shown that the smoother extension to the MMPF, called the *multiple model particle smoother* (MMPS), improves the estimation accuracy when sensor data is limited.

- Another variation of the MMPF, called the efficient multiple model particle filter, is also proposed. The EMMPF requires significantly less computation time compared to the MMPF, the MMPS, or the IMM EnKF and is therefore more suitable for field implementation in real time systems.

- The proposed filtering algorithms are implemented on both first order and second order traffic models, and tested with incident data generated by the microscopic traffic simulation software CORSIM and field data collected on a segment of I–880 in California. It is found the proposed algorithms are capable of jointly estimating the traffic state and detecting incidents. The estimation accuracy of the MMPF is the highest among all algorithms but computationally the most expensive. The first and second order models are shown to have similar performance both in terms of incident detection and traffic state estimation. The proposed algorithms are also compared with both a particle filter and the California algorithm which independently estimate the traffic state and the presence of an incident. The results show that jointly estimating the traffic state and incidents in one algorithm performs at least as good as performing each task independently.

1.4 Organization

This dissertation is organized as follows. In Chapter 2, a literature review is given on traffic estimation, traffic incident detection, and other estimation techniques that are closely related to this thesis.

In Chapter 3, the first order traffic model and the second order traffic flow model used in this work and their numerical discretizations are introduced. Next, a model parameter is embedded into the models so that the models become capable of modeling the traffic evolution under incidents. Since ramps need to be considered in realistic field deployments, the junction solver for the second order traffic model with incidents is introduced. The junction solvers for the first order model can be derived as a simplification of the solvers for the second order traffic model.

In Chapter 4, nonlinear filters are proposed to solve the hybrid state estimation problem associated with system (1.1). In particular, a MMPF and an IMM EnKF are proposed. The

MMPF has also been extended to a MMPS and to an EMMPF.

In Chapter 5, the MMPF, the MMPS, the IMM EnKF, and the EMMPF are tested with incident data generated by the microscopic simulation software CROSIM. The MMPF, MMPS, and IMM EnKF are implemented with the first order model, while the EMMPF is implemented with both the first and second order traffic models. The MMPF and MMPS are tested with different penetration rates of GPS equipped vehicles, and the MMPS is shown to improve the estimation accuracy when the penetration rate of GPS equipped vehicles is low. The MMPF is compared with the IMM EnKF, and with a particle filter and the California algorithm to evaluate traffic state estimation and incident detection performance under various inflows. The performance of the first and second order models are compared to distinguish the relative merits of the traffic flow models for joint traffic state estimation and incident detection.

The field implementation of the estimation framework is described in Chapter 6. The EMMPF is implemented in both the first order traffic model and the second order traffic model on a segment of Interstate 880 in California. The density measurements from inductive loops and speed measurements collected from GPS equipped vehicles in the Mobile Century experiment [18] are used as measurements to the algorithm. The calibration procedures for model parameters are described and the performance of the first and second order traffic flow models are compared. Similar to the CORSIM experiments, the EMMPF is also compared with a particle filter and the California algorithm to compare the traffic state estimation and incident detection performance of the proposed framework and existing works.

In Chapter 7, the main results of the thesis are summarized and future work is discussed, which includes the development of a calibration procedure for model parameters and deployment of parallelization approaches to improve the scalability of the proposed algorithms.

Chapter 2

Related work

Due to the importance from an operational and safety standpoint, there is a considerable amount of literature on both the problems of estimating traffic conditions, and on detecting traffic incidents. In this section, a brief summary is provided on the prior work done in each area, which serves as a basis for the proposed research in the thesis.

2.1 Traffic estimation and nonlinear filtering techniques

2.1.1 Traffic estimation

The main challenge of traffic state estimation is the integration of various types of sensor data (flow, occupancy, speed, etc.) into a nonlinear macroscopic traffic model. The standard state space model for traffic estimation is given as follows:

$$\begin{aligned} x^n &= f\left(x^{n-1}\right) + \omega^{n-1} \\ z^n &= h^n\left(x^n\right) + \nu^n. \end{aligned} \tag{2.1}$$

Note (2.1) is structurally very similar to (1.1), with the notable distinction that the traffic flow model f and the observation equation h do not depend on a model variable. The process of sequential traffic state estimation using experimental data and a flow model

evolution equation began in the 1970's with the early work of Szeto and Gazis [19], who used an *extended Kalman filter* (EKF) to estimate the traffic density in the Lincoln Tunnel in New York City. The EKF algorithm employs linearization of the system (2.1) and models the noise processes as additive, to fit the framework of the Kalman filter. Starting in the early 1980's, a modified version of Payne's macroscopic model was used for a variety of estimation and control problems through the work of Papageorgiou and his collaborators [20, 21, 22, 23].

Recently, the *cell transmission model* (CTM) [24, 25] has been used in state estimation problems through increasingly advanced nonlinear filters, including the *unscented Kalman filter* (UKF) [9, 26] and the *particle filter* (PF) [9, 10, 27]. The particle filter is shown to perform better than the UKF for traffic state reconstruction [9], but has a higher computational cost. Implementation of particle filtering techniques on high dimensional systems (several thousand states or more), is an challenging problem due to inherent scalability challenges for particle filters [28, 29]. The parallelized particle filter [30] has been proposed for large scale implementations. Sun et al. [31] treat the nonlinearity of the (non-differentiable) CTM by recognizing it can be transformed into a linear switching state space model. The density state estimation problem is then solved with a *mixture Kalman filter* for ramp metering or traffic estimation [32]. Other treatments of traffic estimation include ensemble Kalman filtering [33, 34, 35], adjoint–based control and data assimilation [36, 37], distributed local Kalman consensus filtering [38], and direct injection into a Hamilton-Jacobi reformulation of a scalar macroscopic traffic flow model [39, 40].

Among these techniques, the work [35] incorporates multiple models to describe the relationship between traffic density and traffic flow using various data fitting methods. The work [23] uses time varying parameters to jointly estimate densities and second order model parameters, while the others assume a single density–flow relationship for the traffic model. When time–invariant parameters are assumed, the estimation models are not able to account for changes in the dynamics on the highway caused by traffic incidents. In order to jointly estimate the traffic state and detect traffic incidents, a hybrid state estimation problem for

system (1.1) is proposed. By introducing the model variable γ, the estimation model becomes capable of modeling the traffic evolution with an incident.

2.1.2 Nonlinear filtering techniques

A number of techniques in the estimation community have been developed to solve hybrid estimation problems for systems in the form of (1.1). A popular hybrid state estimation problem is the maneuver target tracking problem [41, 42], where a target has several operation modes (e.g., acceleration, deceleration, constant speed) and the objective of the estimation model is to correctly estimate the position of the target and the mode on which it operates.

The MMPF [17] solves the hybrid state estimation problem by allowing the system to have several models. It has a model transition step that describes the switching dynamics of the system mode, and particles are generated for likely system models. The idea of the MMPF is that if the state x^n generated by a model variable γ^n matches well with the measurements, then the MMPF estimates the system is operating in model γ at time n. One central challenge for the MMPF to work in practice is due to its large computational load. When a system has multiple models and some models have very low probability of occurrence (e.g., system fault detection, traffic incident detection), the estimation algorithm requires a large sample size so as to generate enough samples for all possible models of the system. This will lead to a large computational load and possibly prevent the algorithms from being implemented in real time. This problem is addressed by [43], where a model–conditioned PF algorithm is proposed as an modification to the standard MMPF. The computation time can be significantly reduced when a hybrid state system contains rare modes.

Another group of the estimation techniques for solving the hybrid state estimation problem exploits the *multiple model* (MM) approach and the Kalman filter. One of the widely used approaches is the IMM Kalman filter [44, 45, 46]. This method is a model–conditioned Kalman filtering approach. It first computes the weights for all the models of the hybrid system based on the switching probabilities among the models. Then, a Kalman filter is

performed on each model. The choice of the Kalman filter (e.g., extended Kalman filter, ensemble Kalman filter, unscented Kalman filter) is problem dependent. The system state is estimated using the results from each model–conditioned Kalman filter and the weight of each model computed by transition probabilities among the system models. Different from the static MM approach where model switches are not considered, the IMM approach is able to handle dynamical systems with model switches.

These (interactive) multiple model nonlinear filtering techniques will serve as the basis to solve the traffic state estimation and traffic incident detection problem for system (1.1).

2.2 Traffic incident detection algorithms

The main challenge of traffic incident detection is to identify the abnormal traffic patterns caused by traffic incidents and distinguish them from regular traffic congestion. In the past, a number of approaches have been proposed to detect traffic incidents, including the comparative methods [12, 13, 14, 15, 47, 48], time series methods [49, 50], statistical methods [51], probe–based methods [52], and artificial intelligence based methods [53]. A comprehensive review of incident detection studies can be found in the review articles [16, 54, 55].

One group of algorithms are variants of the well known California algorithm [12, 13, 14, 15]. These techniques exploit the idea that an incident will cause a significant increase in the occupancy recorded by an upstream sensor and a decrease in the occupancy recorded by a downstream sensor, where the occupancy is defined as the percent of time that vehicles occupy the sensor detection zone (e.g., is directly above an inductive loop). The California algorithm requires two sensors to collect traffic occupancy values. One sensor must be located upstream from the incident and one must be located downstream. When the measurements are collected, a decision tree structure is used to determine the existence of an incident by comparing the difference and relative difference between the upstream and downstream occupancy values. These values are compared with pre–set thresholds, and if

the values exceed the thresholds, an incident alarm is triggered. Later in the thesis, the performance of the California algorithm will be used as a benchmark and compared with the performance of the proposed algorithms. The California algorithm [14] deployed in this thesis is described in Algorithm 1, where OCC_i and OCC_{i+1} denote the occupancy of the upstream and downstream loop detectors (e.g., collected every two minutes), and T_1, T_2, and T_3 are thresholds which are calibrated from historical data.

Algorithm 1 California algorithm [14]

Collect the upstream sensor occupancy data OCC_i
Collect the downstream sensor occupancy data OCC_{i+1}
if OCC_i-$OCC_{i+1} > T_1$ **then**
  if $(OCC_i\text{-}OCC_{i+1})/OCC_i > T_2$ **then**
    if $(OCC_i\text{-}OCC_{i+1})/OCC_{i+1} > T_3$ **then**
      report incident
    end if
  end if
end if

The double exponential smoothing algorithm [56] uses occupancy, volume, and speed data to detect incident–generated shock waves. The algorithm weights the past and present traffic measurements to predict the short term traffic conditions. The weighting is performed by using a double exponential smoothing function. The errors between the predicted and observed traffic variables are described by an algebraic sum. An incident is reported if the algebraic sum exceeds a preset threshold.

The low pass filter algorithms [57, 58, 59, 60] focus on the measurement processing stage before the data is used to detect an incident. The short-term noise data and inhomogeneities of the measurements are removed by rejecting high frequency fluctuations in the measurements and weighting present and past observations. Then, the detection algorithm traces the spatial occupancy difference between adjacent detectors through time. If the occupancy difference is significant over a short time period, an incident is reported.

The *high occupancy* (HIOCC) and *pattern recognition* (PATREG) algorithms [50] are developed by the *Transport and Road Research Laboratory* (TRRL) for their automatic incident detection system. The idea of the HIOCC algorithm is to identify an incident by analyzing the traffic disturbances caused by an incident. It detects an incident by identifying stationary or slow moving vehicles for several consecutive seconds by using the measurements from individual vehicle detectors. The PATREG algorithm measures the traffic speed between upstream and downstream sensors. It estimates vehicle speeds by measuring the travel time of vehicles between detectors. This method tracks vehicles between sensors and computes the travel time. The speed is estimated by using the travel time and distance between detectors. The speed estimate is compared with the thresholds for a pre-set number of consecutive intervals. If the speed estimate falls below the threshold, an incident is reported. A combination of these two algorithms has been applied for traffic incident detection.

The *autoregressive integrated moving–average* (ARIMA) model [49] assumes the difference of traffic measurements from the current step and the previous step will follow a normal pattern. The method uses datasets from three surveillance systems in Los Angeles, Minneapolis, and Detroit to develop a predictor model for short–term forecasts of traffic data. The autoregressive integrated moving–average model is found to match the datasets well. A confidence interval is established to reflect the normal difference between consecutive measurements. When the difference between two consecutive measurements exceeds the established confidence interval, the model reports an incident occurrence.

The Bayesian algorithm [51] computes the likelihood of an incident occurrence by using Bayesian statistical techniques. This approach develops the frequency distributions of the upstream and the downstream occupancy ratios for both incident and incident–free conditions. Mathematical expressions are developed for the distribution of the ratio from incident and incident–free data. When a measurement is collected, the algorithm computes the likelihood of an incident occurrence by comparing the ratio of the occupancy measurements from the upstream and the downstream with the frequency distributions of the occupancy ratios.

The method has been compared with the California algorithm, where it is found that this algorithm has higher detection rate and lower false alarm rate. However, the mean time to detect is longer.

The catastrophe theory model [61] aims at detecting incidents and distinguishing between incident generated congestion and recurrent congestion. It assumes there will be a sharp change of speed when the traffic switches from free flow to congestion, and the change of traffic flow and occupancy are smooth. The model uses historical data to determine the relationship between flow and occupancy, and the speed variations for congested and uncontested traffic conditions. Then, two tests are applied to determine the existence of traffic incidents. The first test is used to identify whether the traffic is congested. If it is congested, then the second test will evaluate whether the congestion is caused by an incident.

The artificial neural network approach [53] assumes that spatial and temporal traffic patterns can be recognized and classified by an artificial neural network. The artificial neural network is trained with traffic occupancy and volume data from adjacent loop detector stations, including 31 incidents from a typical freeway in the Twin Cities Metropolitan area. The results indicate the neural network is able to learn the main characteristics of a variety of traffic incidents.

The Autoscope incident detection algorithm [62] is based on an image processing technique. First, a video camera is used to monitor the traffic and to collect image data. Then, an image processing program is applied to find stationary or slow–moving vehicles. The program extracts traffic data (e.g., speed, occupancy) from the video and compares them with the pre–set thresholds. When the computed traffic parameters exceed the thresholds, an incident is reported. The video detection system has been tested in Minnesota. The benefits of this approach is that it is possible to check the existence of a traffic incident by reviewing the video, however, more equipment (i.e., video cameras) is required to implement this method.

Another class of approaches are probe–based algorithms. The MIT algorithm [52] exploits

the electronic toll transponders to detect incidents. Three components are proposed to detect incidents: the headway algorithm, the lane switches algorithm, and the lane–monitoring algorithm. The headway algorithm collects travel time and headway data from the electronic toll transponders. The data is compared with pre-set thresholds by applying three different checks. The lane–switching algorithm uses toll transponder data to collect lane–specific, vehicle–specific data from which the number of lane switches can be estimated. If the number of lane switches exceeds a certain threshold, the algorithm reports a traffic incident. The lane–monitoring algorithm monitors the number of vehicles on each lane. The idea is that if more vehicles travel on one lane than another, it indicates there may be an incident on the other lane. The performance of this group of methods depends on the probe penetration rate, the distance between transponder readers, and the availability of lane level traffic data, which is not currently widely available.

The wavelet–based approach [63] uses traffic occupancy and speed data to detect traffic incidents. This approach applies wavelet transform to preprocessing the traffic measurements, and to identify the sharp changes in traffic measurements caused by an incident. Moreover, the algorithm adaptively changes the threshold for incident alarm according to the level of traffic flow. The wavelet–based approach is tested with both simulated and real–world incident data. The algorithm shows to have higher detection rates and lower false alarm rates compared to the California algorithm [13] and the low pass filter algorithm [58].

The probabilistic topic model [64] approach deploys a probabilistic model to describe the state of the traffic. An expectation–maximization algorithm is used to estimate the traffic states parameters under normal traffic conditions. An incident is detected when measurements from vehicles is significantly different from the normal traffic state parameters determined from the expectation–maximization algorithm. The algorithm is tested with data collected in Tokyo and the results show that the proposed approach is able to detect traffic incidents and to distinguish between congestion caused by incident and non–incident scenarios. However, measurements from fast moving vehicles also have the chance to trigger

the incident algorithm, as the behavior of those vehicles is significantly different from normal conditions.

2.3 Macroscopic traffic flow model based incident detection

In contrast to the large literatures on traffic state estimation and on traffic incident detection, few approaches have been proposed to estimate both traffic conditions and the presence of events simultaneously. These methods that rely on macroscopic flow models to describe the traffic dynamics and to detect incidents are described next.

Macroscopic traffic flow model based incident detection approaches have been considered previously. Different from the incident detection algorithms which only uses measurements from the field to infer the existence of an incident, the traffic flow model based approaches detect traffic incident by considering traffic flow dynamics.

A macroscopic traffic model based estimator is introduced in [65] to jointly estimate the traffic state and incidents. The work shows that a traffic incident leads to a drop in the traffic flow. A fault detection algorithm is exploited to detect the incident by comparing the estimated residual between the prediction by the traffic model and the measurements obtained from the field with a defined threshold. This approach is able to detect an incident, however, the incident does not change any properties of the macroscopic model, and the traffic estimates under an incident suffer as a result.

The dynamic model [66] approach uses the second order macroscopic traffic model proposed by Payne [67, 68]. Multiple models are generated by instantiating a new equilibrium fundamental diagram for each incident severity. Then, a multiple model extended Kalman filtering approach is used to select the most likely model (similarly incident severity) and to produce filtered traffic states. The main limitation of [66] is the assumption that sensors are available in every road segment to directly measure the traffic state. While the framework can certainly support a different observation equation, sparse measurements can lead

to poor performance of multiple model filtering for traffic incident detection. Moreover, this approach has only been tested numerically that the true state to be estimated was generated by the same macroscopic traffic model with perturbed parameters, which is a setting known to produce overly optimistic accuracy results [69]. Its performance on an experimental field data set is not investigated.

Other work that are closely related to this thesis are the bi–parameter approach [70] and the adaptive traffic state estimator approach [23], where the traffic state estimation and incident detection problem is posed as a joint state and parameter estimation problem. In [70], the Payne model [67, 68] is used to describe the traffic evolution. Two continuous parameters are introduced to denote the possible capacity drop and speed drop caused by incidents, and a moving horizon parameter estimation scheme is used to estimate the traffic state and the two incident related parameters. In [23], an extended Kalman filter is deployed to jointly estimate the traffic state and key model parameters (i.e., free flow speed, critical density and capacity). Then, the estimated key parameters can be used to infer traffic incidents. Different from [23, 70] where continuous model variables are embedded into the traffic model to denote capacity and speed drop caused by incidents, this thesis uses a discrete model variable to denote incidents, and the estimate of the discrete model variable indicates exactly the location and the severity (i.e., number of lanes blocked) of an traffic incident. Moreover, in this work, the highway is discretized into much smaller segments, as a result, the estimates of the incident location and traffic state have a higher precision compared to [23, 70]. Compared to the extended Kalman filter method, no linearization is needed in this proposed estimation framework since particle filter is capable of handling non–linear models. In additional, the generic second order traffic model [71] is deployed in this dissertation, which has better physical interpretations compared to the Payne model. For example, the Payne model has been criticized for its two major drawbacks [72]. First, the model has the possibility to introduce negative velocities, and the model can propagate information faster than the speed of the fastest vehicle, which means that drivers are influenced by drivers

behind them. The generic second order traffic model [71] deployed in this dissertation does not suffer from the above drawbacks.

Chapter 3

Macroscopic traffic incident model

In this chapter, the first order traffic model and the second order traffic model are reviewed, and the technique to extend both models to incorporate incidents is introduced. Next, the junction solvers for the second order traffic incident model are developed to allow the second order model to be deployed on road networks. The solvers for the first order traffic incident model can be derived by following the solvers for the second order traffic model and fixing the driver property parameter.

3.1 First order traffic model

The *Lighthill-Whitham-Richards Partial Differential Equation* (LWR PDE) [5, 73] is used to describe the evolution of the density $\rho(x,t) \in [0, \rho_{max}]$ at location x and at time t on a roadway. The function v is known as the velocity function, which denotes a constitutive relationship between density and velocity. The LWR PDE expresses the conservation of vehicles on the roadway of length L, and is given by:

$$\begin{aligned} &\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial \left(\rho(x,t)\, v\left(\rho(x,t)\right)\right)}{\partial x} = 0, \\ &(x,t) \in (0,L) \times (0,T) \end{aligned} \tag{3.1}$$

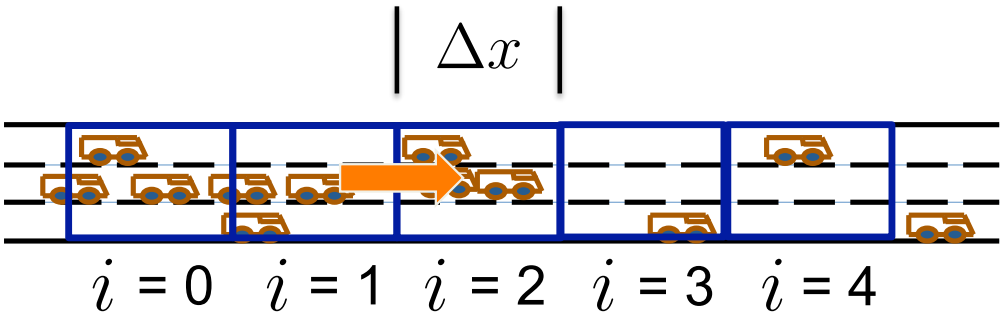


Figure 3.1: Discretization of the highway using the cell transmission model.

with the following initial and boundary conditions:

$$\rho(x,0) = \rho_0(x), \ \rho(0,t) = \rho_l(t), \ \rho(L,t) = \rho_r(t), \tag{3.2}$$

where ρ_0, ρ_l, and ρ_r are the initial, left, and right traffic density boundary conditions.

To close the model, the velocity function v must be specified. The choice of the velocity function v depends on the assumptions on the relationship between density and velocity. A possible relationship is the quadratic–linear function proposed by Smulders [74],

$$v(\rho) = \begin{cases} v_{\max}\left(1 - \frac{\rho}{\beta}\right) & \text{if } \rho \leq \rho_c \\ \frac{v_{\max}\rho_c(\rho_m - \rho)(\beta - \rho_c)}{\rho\beta(\rho_m - \rho_c)} & \text{otherwise.} \end{cases} \tag{3.3}$$

In (3.3), the variable $v_{\max}$ denotes the maximum speed that vehicles can travel on the road. The parameter β determines the shape of the velocity function for the free flow regime. In particular, it determines how the average vehicle speed will change when the traffic density increases from zero to the critical density ρ_c, where the critical density is the traffic density when the highway has the maximum traffic flow. The variable ρ_m denotes the jam density, which corresponds the traffic density when the road is completely congested. The critical density and jam density influence the shape of the velocity function for the congested regime.

For numerical implementation, (3.1) is discretized using a Godunov scheme [75], yielding the *Cell Transmission Model* (CTM) [24, 25, 76]. Specifically, the time and space domains are discretized by introducing a discrete time step ΔT, indexed by $n \in \{0, \cdots, n_{\max}\}$ and a

discrete space step Δx (Figure 3.1), indexed by $i \in \{0, \cdots, i_{\max}\}$. The discretized system is given by:

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta T}{\Delta x} \left(G\left(\rho_{i-1}^n, \rho_i^n\right) - G\left(\rho_i^n, \rho_{i+1}^n\right) \right). \tag{3.4}$$

In (3.4), ρ_i^{n+1} denotes the value of the traffic density at time step $n+1$ and in cell i. According to equation (3.4), the traffic density at a cell in the next time step is determined by the traffic density at the cell in the current time step, plus the traffic flow $G\left(\rho_{i-1}^n, \rho_i^n\right)$ that enters from the upstream cell to the ith cell, and minus the traffic flow $G\left(\rho_i^n, \rho_{i+1}^n\right)$ that exits the ith cell to the downstream cell. The flow (flux) that crosses the cell boundaries is determined by the numerical flux function G, given by:

$$G\left(\rho_i^n, \rho_{i+1}^n\right) = \min\left\{ S\left(\rho_i^n\right), R\left(\rho_{i+1}^n\right) \right\}. \tag{3.5}$$

The flux G is determined by the minimum of the flow that the upstream cell can send and the flow that the downstream cell can receive. The functions S and R are known as the sending and receiving functions, which are given by:

$$S(\rho) = \begin{cases} q(\rho) & \text{if } \rho < \rho_c \\ q(\rho_c) & \text{if } \rho \geq \rho_c \ , \end{cases} \tag{3.6}$$

and

$$R(\rho) = \begin{cases} q(\rho_c) & \text{if } \rho < \rho_c \\ q(\rho) & \text{if } \rho \geq \rho_c \ , \end{cases} \tag{3.7}$$

where the flow $q(\rho) = \rho \times v(\rho)$ is known as the fundamental diagram in the transportation community.

In the sending function (3.6), we can see when the traffic density in the current cell is smaller than the critical density, the cell is able to send whatever flow it has to the next

cell. When the traffic density is greater than the critical density, the cell is able to send the maximum flow of the road. This makes sense because the traffic flow can never go beyond the capacity of the road, even if there are more vehicles available to move to the downstream cell. Similarly, in the receiving function (3.7), if the traffic density in the current cell is smaller than the critical density, it means the cell is currently in free flow, in this case, the cell can receive whatever flow the upstream cell will send, up to the maximum flow. If the traffic density is greater than the critical density, the cell is currently in congestion. The maximum flow the cell can receive is determined by the flow function $q(\rho)$ of the downstream cell. The update equations for (3.4) at the boundary cells are given by:

$$\begin{aligned}\rho_0^{n+1} =& \rho_0^n + \frac{\Delta T}{\Delta x}\left(G\left(\rho_l^n, \rho_0^n\right) - G\left(\rho_0^n, \rho_1^n\right)\right) \\ \rho_{i_{\max}}^{n+1} =& \rho_{i_{\max}}^n + \frac{\Delta T}{\Delta x}\left(G\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n\right) - G\left(\rho_{i_{\max}}^n, \rho_r^n\right)\right),\end{aligned} \tag{3.8}$$

where ρ_l^n and ρ_r^n are the traffic density boundary conditions. To ensure numerical stability, the time and space steps are coupled through the CFL condition [77]: $v_{\max}\frac{\Delta T}{\Delta x} \leq 1$.

3.2 First order traffic incident model

In this section, a model variable γ is embedded into the first order cell transmission model to describe traffic evolution under incidents. The resulting cell transmission traffic incident model is given by:

$$\begin{aligned}\rho_i^{n+1} = \rho_i^n &+ \frac{\Delta T}{\Delta x} G\left(\rho_{i-1}^n, \rho_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) \\ &- \frac{\Delta T}{\Delta x} G\left(\rho_i^n, \rho_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right).\end{aligned} \tag{3.9}$$

In (3.9), γ_i^{n+1} encodes the number of lanes open from time step n to time step $n+1$ in cell i. Similar to equation (3.5), the numerical flux G for the first order traffic incident

model is determined by the minimum of the sending and receiving functions:

$$G\left(\rho_i^n, \rho_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right) = \min\left\{S\left(\rho_i^n, \gamma_i^{n+1}\right), R\left(\rho_{i+1}^n, \gamma_{i+1}^{n+1}\right)\right\}, \tag{3.10}$$

with the modification to allow dependency on the model variables γ_i^{n+1} and γ_{i+1}^{n+1}. The sending and receiving functions S and R are given by:

$$S(\rho,\gamma) = \begin{cases} q(\rho,\gamma) & \text{if } \rho < \rho_c(\gamma) \\ q(\rho_c(\gamma),\gamma) & \text{if } \rho \geq \rho_c(\gamma)\ , \end{cases} \tag{3.11}$$

$$R(\rho,\gamma) = \begin{cases} q(\rho_c(\gamma),\gamma) & \text{if } \rho < \rho_c(\gamma) \\ q(\rho,\gamma) & \text{if } \rho \geq \rho_c(\gamma)\ . \end{cases} \tag{3.12}$$

The flow function is given as $q(\rho,\gamma) = \rho \times v(\rho,\gamma)$. When the model parameter is embedded, the incident velocity function is constructed as follows:

$$v(\rho,\gamma) = \begin{cases} v_{\max}(\gamma)\left(1 - \frac{\rho}{\beta(\gamma)}\right) & \text{if } \rho \leq \rho_c(\gamma) \\ \frac{v_{\max}(\gamma)\rho_c(\gamma)(\rho_m(\gamma)-\rho)(\beta(\gamma)-\rho_c(\gamma))}{\rho\beta(\gamma)(\rho_m(\gamma)-\rho_c(\gamma))} & \text{otherwise.} \end{cases} \tag{3.13}$$

In (3.13), all parameters are a function of γ because when γ refers to an incident model, the number of open lanes will drop, and the parameters in the traffic model will change accordingly (e.g., maximum speed and flow will drop when there is an incident). Figure 3.2 shows the velocity function for a two–lane road. The black curve shows the velocity function for the non–incident case, and the blue curve shows the function when one lane is blocked by an incident. In the presence of an incident, the maximum speed will drop because vehicles slow down near the incident. The critical density changes because the capacity (which is a function of the number of lanes) decreases and the maximum speed drops. The jam density drops proportionally to the lane drop because of the lane closure caused by the incident.

The flow function $q(\rho,\gamma) = \rho \times v(\rho,\gamma)$ is shown in Figure 3.3. Because the fundamental diagram changes in the presence of an incident, the sending and receiving functions will also

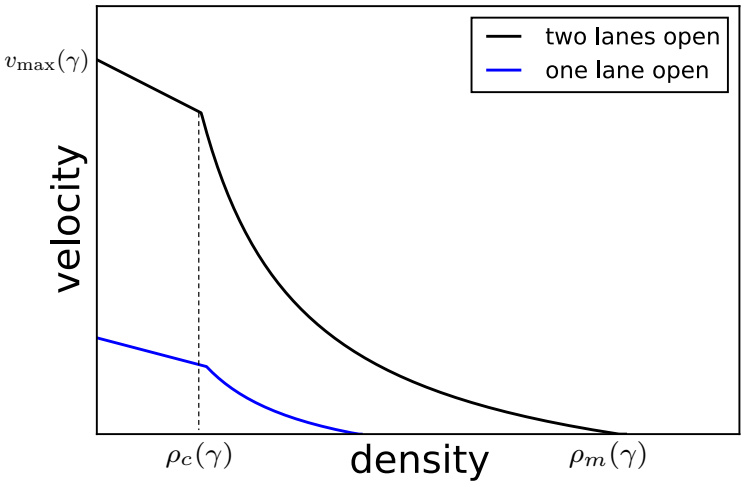


Figure 3.2: The velocity function of the first order traffic incident model for a two–lane road. The black curve corresponds to the non–incident case, and the blue curve shows the velocity function when one lane is blocked on a two–lane road.

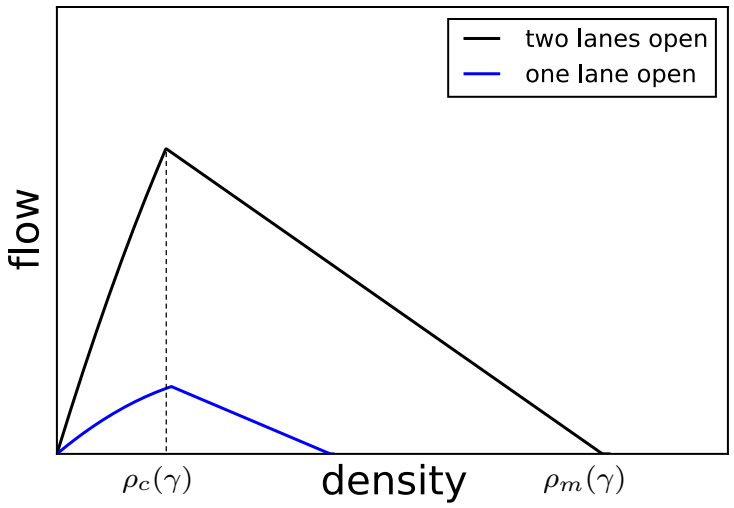


Figure 3.3: The fundamental diagram of the first order traffic incident model for a two–lane road. The black curve corresponds to the non–incident case, the blue curve shows the flow function when one lane is blocked on the two–lane road.

change, and this consequently influences the number of vehicles that can move from one cell to the next. The update equations of (3.9) at the boundary cells are given by:

$$\begin{aligned}
\rho_0^{n+1} &= \rho_0^n + \frac{\Delta T}{\Delta x} G\left(\rho_l^n, \rho_0^n, \gamma_l^{n+1}, \gamma_0^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G\left(\rho_0^n, \rho_1^n, \gamma_0^{n+1}, \gamma_1^{n+1}\right) \\
\rho_{i_{\max}}^{n+1} &= \rho_{i_{\max}}^n + \frac{\Delta T}{\Delta x} G\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n, \gamma_{i_{\max}-1}^{n+1}, \gamma_{i_{\max}}^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G\left(\rho_{i_{\max}}^n, \rho_r^n, \gamma_{i_{\max}}^{n+1}, \gamma_r^{n+1}\right),
\end{aligned} \tag{3.14}$$

where $\gamma_l^{n+1}, \gamma_r^{n+1}$ are the left and right model variable boundary conditions.

3.3 Second order traffic model

The second order traffic incident model is introduced in this section. Compared to the first order traffic model [5, 24, 25, 73]. where the speed is uniquely determined from the density of traffic, the second order traffic model parameterizes the velocity function by a driver property variable, which generates a family of curves to describe the speed–density relationship.

3.3.1 Generic framework of a second order traffic model

The second order traffic flow model that fits into the framework of the *Generic second order model* (GSOM) [71] is deployed to describe the traffic evolution on the highway. The resulting second order traffic model (in non–conservative form) is given by:

$$\begin{aligned}
&\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial\left(\rho(x,t)\,\tilde{v}\left(\rho(x,t), w(x,t)\right)\right)}{\partial x} = 0, \\
&\frac{\partial w(x,t)}{\partial t} + \tilde{v}\left(\rho(x,t), w(x,t)\right)\frac{\partial w(x,t)}{\partial x} = 0, \\
&(x,t) \in (0,L) \times (0,T),
\end{aligned} \tag{3.15}$$

where w represents the *property* of the vehicles or drivers, and $\tilde{v}$ denotes the velocity function for the second order traffic model. The boundary conditions are given by:

$$\begin{aligned} &\rho(x,0) = \rho_0(x),\ \rho(0,t) = \rho_l(t),\ \rho(L,t) = \rho_r(t), \\ &w(x,0) = w_0(x),\ w(0,t) = w_l(t),\ w(L,t) = w_r(t), \end{aligned} \tag{3.16}$$

where the variables w_0, w_l, and w_r are the initial, left, and right boundary conditions for the property variable.

In (3.15), the first equation describes the conservation of vehicles, as does the LWR PDE. The second equation indicates that w is advected with vehicles at the speed of v, and thus the property is conserved along vehicle trajectories. The condition $\partial\tilde{v}/\partial\rho < 0$ is required to guarantee the system (3.15) is hyperbolic for $\rho > 0$ [6]. The conservation form of (3.15) is given as follows:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial\left(\rho\tilde{v}(\rho,w)\right)}{\partial x} = 0, \\ &\frac{\partial y}{\partial t} + \frac{\partial\left(y\tilde{v}(\rho,w)\right)}{\partial x} = 0, \\ &(x,t) \in (0,L)\times(0,T), \end{aligned} \tag{3.17}$$

where the conserved quantity $y = \rho w$. Since w is advected with vehicle flows, ρ is conserved and so is $y = \rho w$.

Note that it is important to give y a clear physical meaning to properly design a discrete cell transmission model [24] equivalent for (3.17). A suitable definition for $y = \rho w$ is to recognize that it is a *total property*, where the property w may have various meanings, such as "aggressivity" [78], "desired spacing" [6], or "perturbations" [79]. Thus, the second conservation equation of (3.17) expresses the conservation of the total property. For example, imaging the property w as the average number of passengers carried by each vehicle, it is clear that the total number of passengers is conserved on a road segment, although this would

not necessarily influence the velocity function. The parameter w can also be interpreted as the different behaviors among multiple classes of vehicles (e.g., trucks and passenger cars) [80].

Note that when all drivers have the same property, the GSOM becomes the LWR PDE [5, 73], where the velocity depends only on the density. Thus, the LWR PDE can be viewed is a special form of the GSOM, with a uniform property $w(x,t) = \bar{w}$ [81] (i.e., $v(\rho) = \tilde{v}(\rho, \bar{w})$).

The velocity function $\tilde{v}$ needs to be specified in order to close the model. Different types of velocity functions have been used based on the assumptions on the property quantity w. The models proposed by *Aw and Rascle* [82] and *Zhang* [83] (ARZ) and the *generalized Aw–Rascle–Zhang model* (GARZ) [78] allow the driver property to influence the velocity function both in the freeflow and congested regimes. However, these models are not appropriate to capture distinct behaviors in the freeflow and congested regimes based on empirical observation by Kerner [84, 85], who observed that the experimental flow data is positively proportional to the density data in freeflow, while the flow–density data exhibits large spread in congestion.

In this work, a quadratic–linear velocity function developed based on the collapsed GSOM is deployed:

$$\tilde{v}(\rho, w) = \begin{cases} v_{\max}\left(1 - \frac{\rho}{\beta}\right) & \text{if } \rho \leq \tilde{\rho}_c(w) \\ \frac{v_{\max}\tilde{\rho}_c(w)(\tilde{\rho}_m(w)-\rho)(\beta-\tilde{\rho}_c(w))}{\rho\beta(\tilde{\rho}_m(w)-\tilde{\rho}_c(w))} & \text{otherwise.} \end{cases} \tag{3.18}$$

In (3.18), the variables $v_{\max}$ and β determine the shape of the velocity function for the free flow regime. In the collapsed model, all vehicles are assumed to drive the same speed in free flow regardless of their property w. The variables $\tilde{\rho}_c$ and $\tilde{\rho}_m$ are the critical density and jam density for the second order traffic model, and determine the shape of the second order velocity function in the congested regime. Since vehicles drive differently based on their properties in congestion, these two variables depend on the vehicle property w. Letting ρ_{c1}, ρ_{c2}, ρ_{m1}, and ρ_{m2} denote the upper bounds and lower bounds for the critical density and

jam density, the critical density and jam density parameterized by w can be written as:

$$\begin{cases} \tilde{\rho}_c(w) = \frac{\rho_{c1}\rho_{c2}}{(1-w)\rho_{c1}+w\rho_{c2}}, \\ \tilde{\rho}_m(w) = \frac{\rho_{m1}\rho_{m2}}{(1-w)\rho_{m1}+w\rho_{m2}}. \end{cases} \tag{3.19}$$

The *phase transition models* (PTM) [79, 86, 87, 88] are also related to the collapsed GARZ model, however in freeflow the PTM uses a first order model of the traffic dynamics, and transitions to a second order model in the congested regime. This allows for more variability in the fundamental diagram in congestion, which matches better with the empirical observation by Kerner [84, 85].

3.3.2 Second order cell transmission model

The numerical discretization of the PDE can be derived by tracking the evolution of the characteristic curves on the Riemann problem, where the Riemann problem is the Cauchy problem equation with a Heaviside initial condition of the PDE (3.1) in an infinite domain [89]. The solution to the Riemann problem for the GSOM (3.15) is more complicated to the Riemann problem for the LWR model and lacks an immediate physical interpretation (e.g., the existence of the *intermediate state* [6] in the construction of the Riemann solver). In [90], a Riemann solver to the GOSM is constructed by analyzing the sending and receiving functions, which is consistent with the original solver derived by analyzing elementary waves [82, 83]). This equivalence makes it possible to derive a *second order cell transmission model* (2CTM). In this work, a 2CTM is constructed based on a Godunov discretization of the

GOSM (3.17). The 2CTM is given as:

$$\begin{aligned}
\rho_i^{n+1} &= \rho_i^n + \frac{\Delta T}{\Delta x} G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right), \\
y_i^{n+1} &= y_i^n + \frac{\Delta T}{\Delta x} G_y \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_y \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right),
\end{aligned} \tag{3.20}$$

which provides evolution equations for both conserved quantities: the traffic density ρ_i^n and total property $y_i^n = \rho_i^n w_i^n$. Here, G_ρ and G_y are numerical fluxes associated with ρ and y, respectively.

To determine G_ρ and G_y, it is important to note that these two kinds of flow are related. Since the property w is always advected with vehicle flow G_ρ, the flow of total property G_y is computed by multiplying the average property w of the upstream cell to the flow of vehicles:

$$G_y \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) = w_{i-1}^n G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right).$$

Thus, the update equations (3.20) simplify to

$$\begin{aligned}
\rho_i^{n+1} &= \rho_i^n + \frac{\Delta T}{\Delta x} G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right), \\
y_i^{n+1} &= y_i^n + \frac{\Delta T}{\Delta x} w_{i-1}^n G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) \\
&\quad - \frac{\Delta T}{\Delta x} w_i^n G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right).
\end{aligned} \tag{3.21}$$

Note that if we assume all vehicles have the same property, i.e., $w = \bar{w}$, the update

equation for y is identical to that of ρ:

$$\begin{aligned}\bar{w}\rho_i^{n+1} = \bar{w}\rho_i^n &+ \frac{\Delta T}{\Delta x}\bar{w}G_\rho\left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n\right) \\ &- \frac{\Delta T}{\Delta x}\bar{w}G_\rho\left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right),\end{aligned}$$

where $\bar{w}$ can be canceled out, and $\bar{w}$ only selects a single curve in the second order fundamental diagram to build the sending and receiving functions. Thus, the 2CTM is consistent with the classical CTM (3.4) when the property quantity is fixed. The update equations at the boundaries of (3.21) are given by:

$$\begin{aligned}\rho_0^{n+1} &= \rho_0^n + \frac{\Delta T}{\Delta x}G_\rho\left(\rho_l^n, \rho_0^n, w_l^n, w_0^n\right) \\ &\quad - \frac{\Delta T}{\Delta x}G_\rho\left(\rho_0^n, \rho_1^n, w_0^n, w_1^n\right), \\ y_0^{n+1} &= y_0^n + \frac{\Delta T}{\Delta x}w_l^n G_\rho\left(\rho_l^n, \rho_0^n, w_l^n, w_0^n\right) \\ &\quad - \frac{\Delta T}{\Delta x}w_0^n G_\rho\left(\rho_0^n, \rho_1^n, w_0^n, w_1^n\right), \\ \rho_{i_{\max}}^{n+1} &= \rho_{i_{\max}}^n + \frac{\Delta T}{\Delta x}G_\rho\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n, w_{i_{\max}-1}^n, w_{i_{\max}}^n\right) \\ &\quad - \frac{\Delta T}{\Delta x}G_\rho\left(\rho_{i_{\max}}^n, \rho_r^n, w_{i_{\max}}^n, w_r^n\right), \\ y_{i_{\max}}^{n+1} &= y_{i_{\max}}^n + \frac{\Delta T}{\Delta x}w_{i_{\max}-1}^n G_\rho\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n, w_{i_{\max}-1}^n, w_{i_{\max}}^n\right) \\ &\quad - \frac{\Delta T}{\Delta x}w_{i_{\max}}^n G_\rho\left(\rho_{i_{\max}}^n, \rho_r^n, w_{i_{\max}}^n, w_r^n\right),\end{aligned} \tag{3.22}$$

where w_l and w_r are the property variable boundary conditions.

The flow function for the second order model is given as $\tilde{q}(\rho, w) = \rho \times \tilde{v}(\rho, w)$. Note that unlike the first order flow model, the property variable w is an input to the velocity function and the flow function. In (3.21) and (3.22), the numerical flux G_ρ is given as:

$$G_\rho\left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n\right) = \min\left\{\tilde{S}\left(\rho_i^n, w_i^n\right), \tilde{R}\left(\rho_{i+1}^n, w_i^n, w_{i+1}^n\right)\right\}.$$

The functions $\tilde{S}$ and $\tilde{R}$ are the sending and receiving functions for the second order traffic model, which are given by:

$$\tilde{S}(\rho, w) = \begin{cases} \rho\tilde{v}(\rho, w) & \text{if } \rho < \tilde{\rho}_c(w) \\ \tilde{\rho}_c(w)\,\tilde{v}(\tilde{\rho}_c(w), w) & \text{if } \rho < \tilde{\rho}_c(w)\ , \end{cases} \tag{3.23}$$

$$\tilde{R}(\rho_M, w^-, w^+) = \begin{cases} \tilde{\rho}_c(w^-)\,\tilde{v}(\tilde{\rho}_c(w^-), w^-) & \text{if } \rho_M < \tilde{\rho}_c(w^-) \\ \rho_M(w^-, w^+)\,\tilde{v}(\rho_M, w^-) & \text{if } \rho \geq \tilde{\rho}_c(w^-), \end{cases} \tag{3.24}$$

where ρ_M and w_M are known as the intermediate traffic state variables. We use w^- (w^+) to denote the property variable in the upstream (downstream) cell, v_+ to denote the velocity in the downstream cell, and v_M to denote the velocity associated with the intermediate traffic state. The intermediate traffic state in (3.24) can be determined by solving the following problem:

$$\begin{aligned} \underset{\rho_M}{\text{minimize:}} \quad & v_+ - v_M \\ \text{subject to:} \quad & w_M = w^-, \\ & v_M \leq v_+, \\ & v_M = \tilde{v}(\rho_M, w_M), \\ & v_+ = \tilde{v}(\rho, w^+). \end{aligned} \tag{3.25}$$

The existence of the intermediate state ρ_M and w_M can be understood as a consequence of the interactions among vehicles with different properties w. Consider two adjacent cells (an upstream cell and a downstream cell) with initial states ρ^-, w^- and ρ^+, w^+, where the variables ρ^- and ρ^+ denote the traffic density at the upstream cell and the downstream cell. The traffic flow that can cross the cell interface and the existence of the intermediate state are shown in Figure 3.4 and interpreted as follows [80]:

1. Downstream vehicles move out of way, which creates space for the upstream vehicles.

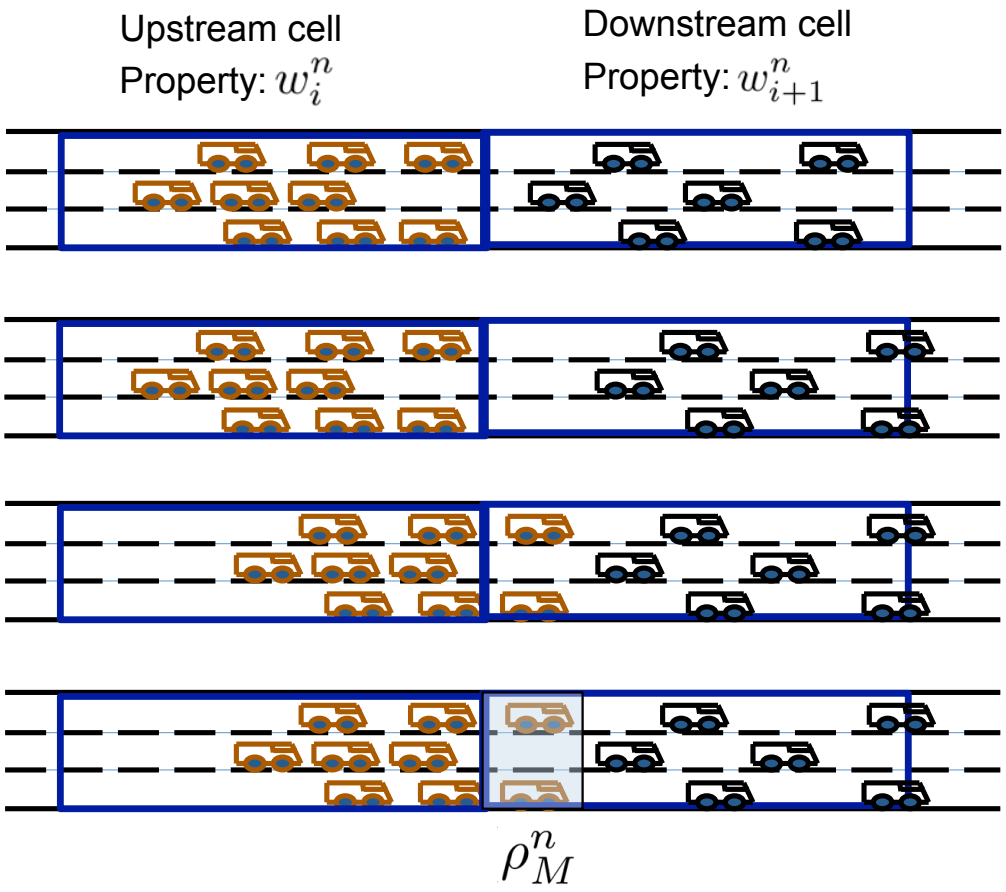


Figure 3.4: Interpretation of the intermediate traffic density. From top to bottom, the first figure shows the initial traffic condition, where the upstream vehicles and downstream vehicles have different properties. The second figure shows the downstream vehicles create space for the upstream vehicles. The third figure shows the upstream vehicles enter the cell. The fourth figure shows the creation of the intermediate traffic state, because the upstream vehicles adjust their spacing (which is the inverse of the traffic density) to reach the downstream velocity while maintaining the upstream cell property.

2. Upstream vehicles maintain their property as they move from the upstream cell to the downstream cell. Thus, the property of vehicles that cross the cell interface is determined by the upstream vehicles (i.e., $w_M = w^-$).

3. When vehicles from the upstream cell enter the downstream cell, they will drive as fast as possible, but not faster than the downstream vehicles. This means v_M is chosen such that the velocity between v_M and v_+ is minimized (i.e., $\min_{\rho_M} \{v_+ - v_M\}$).

4. Vehicles that cross the cell interface with the property w^- adjust their spacing (density) to reach the velocity v_{M}, which creates an intermediate density ρ_{M}, such that $v_{\mathrm{M}} = \tilde{v}\left(\rho_{\mathrm{M}}, w^-\right)$.

Here, we assume the downstream vehicles create space for the upstream vehicles only to calculate the intermediate state. The actual flow across the cell boundary is still determined as the minimum of the sending and receiving functions. The intermediate state is required to compute the number of vehicles the downstream cell can receive. Intuitively, the number of vehicles that can be received by the downstream cell depends not only on the amount of the space that the downstream vehicles can generate, but also on the willingness of the vehicles from the upstream cell to fill the free space (determined by w^-). Therefore, it is not surprising that the receiving function (3.24) is also a function of the property of the upstream vehicles.

3.4 Second order traffic incident model

When the model variable γ is embedded into (3.21), the second order cell transmission incident model is given as:

$$\begin{aligned}
\rho_i^{n+1} &= \rho_i^n + \frac{\Delta T}{\Delta x} G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right), \\
y_i^{n+1} &= y_i^n + \frac{\Delta T}{\Delta x} G_y \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_y \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right).
\end{aligned} \tag{3.26}$$

Following the same arguements from (3.20) to (3.21), the update equations (3.26) for the incident cell transmission model simplify to

$$\begin{aligned}
\rho_i^{n+1} &= \rho_i^n + \frac{\Delta T}{\Delta x} G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right), \\
y_i^{n+1} &= y_i^n + \frac{\Delta T}{\Delta x} w_{i-1}^n G_\rho \left(\rho_{i-1}^n, \rho_i^n, w_{i-1}^n, w_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} w_i^n G_\rho \left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right).
\end{aligned} \tag{3.27}$$

The update equations for the boundary cells of (3.27) are given by:

$$\begin{aligned}
\rho_0^{n+1} &= \rho_0^n + \frac{\Delta T}{\Delta x} G_\rho\left(\rho_l^n, \rho_0^n, w_l^n, w_0^n, \gamma_l^{n+1}, \gamma_0^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho\left(\rho_0^n, \rho_1^n, w_0^n, w_1^n, \gamma_0^{n+1}, \gamma_1^{n+1}\right), \\
y_0^{n+1} &= y_0^n + \frac{\Delta T}{\Delta x} w_l^n G_\rho\left(\rho_l^n, \rho_0^n, w_l^n, w_0^n, \gamma_l^{n+1}, \gamma_0^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} w_0^n G_\rho\left(\rho_0^n, \rho_1^n, w_0^n, w_1^n, \gamma_0^{n+1}, \gamma_1^{n+1}\right), \\
\rho_{i_{\max}}^{n+1} &= \rho_{i_{\max}}^n + \frac{\Delta T}{\Delta x} G_\rho\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n, w_{i_{\max}-1}^n, w_{i_{\max}}^n, \gamma_{i_{\max}-1}^{n+1}, \gamma_{i_{\max}}^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} G_\rho\left(\rho_{i_{\max}}^n, \rho_r^n, w_{i_{\max}}^n, w_r^n, \gamma_{i_{\max}}^{n+1}, \gamma_r^{n+1}\right), \\
y_{i_{\max}}^{n+1} &= y_{i_{\max}}^n + \frac{\Delta T}{\Delta x} w_{i_{\max}-1}^n G_\rho\left(\rho_{i_{\max}-1}^n, \rho_{i_{\max}}^n, w_{i_{\max}-1}^n, w_{i_{\max}}^n, \gamma_{i_{\max}-1}^{n+1}, \gamma_{i_{\max}}^{n+1}\right) \\
&\quad - \frac{\Delta T}{\Delta x} w_{i_{\max}}^n G_\rho\left(\rho_{i_{\max}}^n, \rho_r^n, w_{i_{\max}}^n, w_r^n, \gamma_{i_{\max}}^{n+1}, \gamma_r^{n+1}\right).
\end{aligned} \tag{3.28}$$

The flow function for the second order model is given as $\tilde{q}(\rho, w, \gamma) = \rho \times \tilde{v}(\rho, w, \gamma)$. When the model parameter γ is embedded into (3.21), the velocity function for the second order traffic model is given as follows:

$$\tilde{v}(\rho, w, \gamma) = \begin{cases} v_{\max}(\gamma)\left(1 - \frac{\rho}{\beta(\gamma)}\right) & \text{if } \rho \le \tilde{\rho}_c(w, \gamma) \\ \frac{v_{\max}(\gamma)\tilde{\rho}_c(w,\gamma)(\tilde{\rho}_m(w,\gamma)-\rho)(\beta(\gamma)-\tilde{\rho}_c(w,\gamma))}{\rho\beta(\gamma)(\tilde{\rho}_m(w,\gamma)-\tilde{\rho}_c(w,\gamma))} & \text{otherwise.} \end{cases} \tag{3.29}$$

In (3.29), all parameters are a function of γ because when an incident occurs, the number of open lanes will drop, and the parameters in the traffic model will change accordingly. The velocity is a function of γ and w, where γ determines which family of curves the model should use (i.e., the black curves when there is no incident, or the blue curves when one lane is blocked (see Figure 3.5) and w determines which exact curve is used.

Figure 3.5 shows the velocity function for a two–lane road. The black curves are shown for different w for the non–incident case, and the blue curves show the function when one lane is blocked by an incident. Compared to the first order model, the second order model

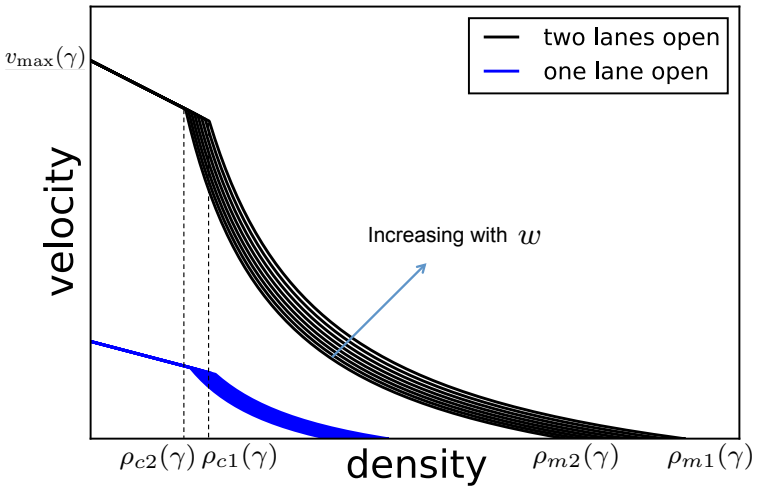


Figure 3.5: The velocity function for the second order traffic incident model for a two–lane road. The black curves correspond to the non–incident case, the blue cures show the velocity function for an incident that blocks a single lane. When the number of lanes open is known, the property variable w determines which exact curve should be deployed.

uses a family of curves to describe the density–velocity relationship in the congestion regime, parameterized by the model parameter w. Again, $\rho_{c1}(\gamma)$, $\rho_{c2}(\gamma)$, $\rho_{m1}(\gamma)$, and $\rho_{m2}(\gamma)$ are used to to denote the upper bounds and lower bounds for the critical density and jam density. The critical density and jam density for the second order traffic incident model can be computed as:

$$\begin{cases} \rho_c(w,\gamma) = \frac{\rho_{c1}(\gamma)\rho_{c2}(\gamma)}{(1-w)\rho_{c1}(\gamma)+w\rho_{c2}(\gamma)}, \\ \rho_m(w,\gamma) = \frac{\rho_{m1}(\gamma)\rho_{m2}(\gamma)}{(1-w)\rho_{m1}(\gamma)+w\rho_{m2}(\gamma)}. \end{cases} \tag{3.30}$$

The fundamental diagram for the second order traffic model is shown in Figure 3.6. In (3.27), the numerical flux G is given as:

$$G_\rho\left(\rho_i^n, \rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right) = \min\left\{\tilde{S}\left(\rho_i^n, w_i^n, \gamma_i^{n+1}\right), \tilde{R}\left(\rho_{i+1}^n, w_i^n, w_{i+1}^n, \gamma_{i+1}^{n+1}\right)\right\}.$$

The functions $\tilde{S}$ and $\tilde{R}$ are the sending and receiving functions for the second order traffic

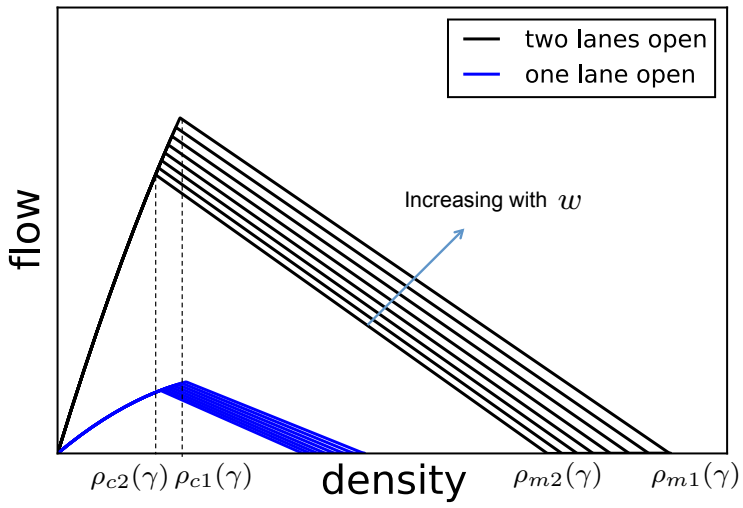


Figure 3.6: The fundamental diagram for the second order traffic incident model for a two–lane road. The black curves correspond to the non–incident case, the blue cures show the fundamental diagram when one lane is blocked on the two–lane road.

incident model, which are given by:

$$\tilde{S}(\rho, w, \gamma) = \begin{cases} \rho\tilde{v}(\rho, w, \gamma) & \text{if } \rho < \tilde{\rho}_c(w, \gamma) \\ \tilde{\rho}_c(w, \gamma)\,\tilde{v}(\tilde{\rho}_c(w, \gamma), w, \gamma) & \text{if } \rho < \tilde{\rho}_c(w, \gamma)\ , \end{cases} \tag{3.31}$$

and

$$\tilde{R}(\rho_M, w^-, w^+, \gamma) = \begin{cases} \tilde{\rho}_c(w^-, \gamma)\,\tilde{v}(\tilde{\rho}_c(w^-, \gamma), w^-, \gamma) & \text{if } \rho_M < \tilde{\rho}_c(w^-, \gamma) \\ \rho_M(w^-, w^+)\,\tilde{v}(\rho_M, w^-, \gamma) & \text{if } \rho \geq \tilde{\rho}_c(w^-, \gamma)\ . \end{cases} \tag{3.32}$$

The intermediate traffic state in (3.32) can be determined by solving the following prob-

lem:

$$\begin{aligned} \underset{\rho_M}{\text{minimize:}} \quad & v_+ - v_M \\ \text{subject to:} \quad & w_M = w^-, \\ & v_M \leq v_+, \\ & v_M = \tilde{v}\left(\rho_M, w_M, \gamma\right), \\ & v_+ = \tilde{v}\left(\rho, w^+, \gamma\right). \end{aligned} \tag{3.33}$$

3.5 Network problem for the second order traffic incident model

To construct a discrete solution to a network problem, one models the road network as a graph composed of links (edges) and junctions (vertices). Consider the discretized 2CTM (3.21), the flux G that can go across between links depends on the solutions to the junction problem, three types of junctions are considered: bottlenecks and lane additions, diverges, and merges. These are the most common freeway junctions and represent lane drops and lane additions, off–ramps, and on–ramps.

The junction solvers for traffic flow models have been widely studied [91, 92, 93, 94]. In this dissertation, we deploy the techniques presented in earlier works [91, 92, 93, 94] and extended the techniques to the second order traffic incident model. As pointed out in [94], the following criteria is deployed in this dissertation to obtain a unique solution for the junction problem: total flux is maximized under admissibility constraints, and a constraint on the splitting of flow or on the allocation of space during merging.

Bottlenecks and lane additions: One Incoming Road and One Outgoing Road

The simplest junction contains two links labeled $m = 1$ (incoming) and $m = 2$ (outgoing) with a different number of lanes. This includes bottlenecks (shown in Figure 3.7), and expansion zones where a lane is added to the roadway. We use $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$ to denote the

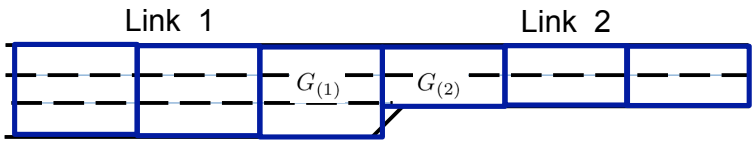


Figure 3.7: Bottleneck highway, junction with one incoming link and one outgoing link.

traffic state in the last cell of the incoming link, and $\rho^+_{(2)}, w^+_{(2)}, \gamma^+_{(2)}$ to denote the traffic state in the first cell of the outgoing link. The flow that can cross the junction can be determined by solving the following optimization problem:

$$\begin{aligned} \underset{G_{(1)},G_{(2)}}{\text{maximize:}} \quad & G_{(1)} \\ \text{subject to:} \quad & 0 \le G_{(1)} \le \tilde{S}_{(1)}\left(\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}\right), \\ & 0 \le G_{(2)} \le \tilde{R}_{(2)}\left(\rho_{(2)M}, w^-_{(1)}, w^+_{(2)}, \gamma^+_{(2)}\right), \\ & G_{(1)} = G_{(2)}, \end{aligned} \tag{3.34}$$

where $G_{(1)}$ and $G_{(2)}$ separately represent the possible flow that leaves link 1 and the flow that enters link 2. $\tilde{S}_{(1)}$ is the sending function of the last cell at link 1, and $\tilde{R}_{(2)}$ is the receiving function of the first cell at link 2. Here, the sending and receiving functions are the same as those defined in equation (3.31) and equation (3.32), just with a subscript to denote each link may have different fundamental diagram parameters. The solution $G^*_{(1)}$ and $G^*_{(2)}$ to the optimization problem (3.34) is:

$$G^*_{(1)} = G^*_{(2)} = \min\{\tilde{S}_{(1)}(\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}),\ \tilde{R}_{(2)}(\rho_{(2)M}, w^-_{(1)}, w^+_{(2)}, \gamma^+_{(2)})\}, \tag{3.35}$$

which is the maximum flow that can go across the junction. Then, the optimal flux $G^*_{(1)}$ and $G^*_{(2)}$ are used as the right and left flow boundary conditions for link 1 and link 2 for traffic prediction, as shown in Algorithm 2.

Algorithm 2 Junction solver: Bottleneck

Initialization:

Link 1 (incoming): $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$

Link 2 (outgoing): $\rho^+_{(2)}, w^+_{(2)}, \gamma^+_{(2)}$

Solve the junction problem (3.34) **according to** (3.35)**:** $G^*_{(1)}$ **and** $G^*_{(2)}$**.**

Prediction:

In the last cell of link 1 (incoming link):

$$\rho^{n+1}_{i_{\max}} = \rho^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - G^*_{(1)}\right)$$
$$y^{n+1}_{i_{\max}} = y^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(w^n_{i_{\max-1}} G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - w^n_{i_{\max}} G^*_{(1)}\right),$$

In the first cell of link 2 (outgoing link):

$$\rho^{n+1}_0 = \rho^n_0 + \frac{\Delta T}{\Delta x}\left(G^*_{(2)} - G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),$$
$$y^{n+1}_0 = y^n_0 + \frac{\Delta T}{\Delta x}\left(w^n_l G^*_{(2)} - w^n_0 G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),$$

In all other cells for link 1 and link 2: apply the 2CTM (see Section 3.4)

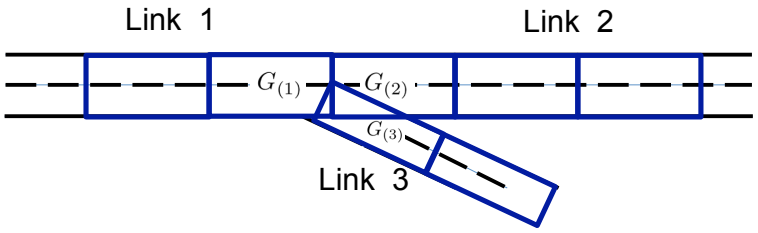


Figure 3.8: Diverge highway, junction with one incoming link and two outgoing links.

Diverge: One Incoming Road and Two Outgoing Roads

A diverge road junction contains three links $m = 1$ (incoming), $m = 2$ (outgoing) and $m = 3$ (outgoing), as shown in Figure 3.8. We use $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$ to denote the traffic state in the last cell of the incoming link, and $\rho^+_{(2)}, w^+_{(2)}, \gamma^+_{(2)}$ and $\rho^+_{(3)}, w^+_{(3)}, \gamma^+_{(3)}$ to denote the traffic state in the first cell of each outgoing link. The flow that crosses the junction, and the flow that enters each outgoing link can be determined by solving the following optimization problem:

$$\begin{aligned}
\underset{G_{(1)},G_{(2)},G_{(3)}}{\text{maximize:}} \quad & G_{(2)} + G_{(3)} \\
\text{subject to:} \quad & 0 \le G_{(2)} \le \tilde{R}_{(2)}(\rho_{(2)M}, w^-_{(1)}, w^+_{(2)}, \gamma^+_{(2)}), \\
& 0 \le G_{(3)} \le \tilde{R}_{(3)}(\rho_{(3)M}, w^-_{(1)}, w^+_{(3)}, \gamma^+_{(3)}), \\
& 0 \le G_{(1)} \le \tilde{S}_{(1)}(\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}), \\
& G_{(1)} = G_{(2)} + G_{(3)}, \\
& \alpha G_{(2)} = (1-\alpha)\, G_{(3)},
\end{aligned} \tag{3.36}$$

where α is known as the split ratio, which determines the fraction of the flow out of link 1 that travels to link 2. In this work, we assume the vehicles that diverge to each outgoing link strictly obey the split ratio specified by α, which is a common assumption of diverge models [94]. Also note that the upstream properties for both downstream links 2 and 3 are the same, since all vehicles entering the outgoing links come from link 1 with property w^-_1. The solution $G^*_{(1)}$, $G^*_{(2)}$, and $G^*_{(3)}$ to the optimization problem (3.36) can be calculated as:

$$\begin{aligned}
G^*_{(2)} &= \min\{\tilde{R}_{(2)}, (1-\alpha)/\alpha \tilde{R}_{(3)}, (1-\alpha)\, \tilde{S}_{(1)}\}, \\
G^*_{(3)} &= \alpha/(1-\alpha)\, G^*_{(2)}, \\
G^*_{(1)} &= G^*_{(2)} + G^*_{(3)}.
\end{aligned} \tag{3.37}$$

Then, the optimal flux $G^*_{(1)}$ is used as the right flow boundary condition for link 1, and the flux $G^*_{(2)}$ and $G^*_{(3)}$ are used as the left flow boundary conditions for link 2 and link 3 for

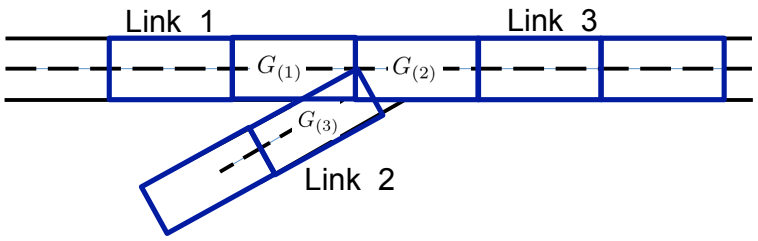


Figure 3.9: Merge highway, junction with two incoming links and one outgoing link.

the one step model prediction, as shown in Algorithm 3.

Merge: Two Incoming Roads and One Outgoing Road

A merge road junction contains three links $m = 1$ (incoming), $m = 2$ (incoming) and $m = 3$ (outgoing), as shown in Figure 3.9. We use $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$, and $\rho^-_{(2)}, w^-_{(2)}, \gamma^-_{(2)}$ to denote the traffic state in the last cells of the incoming links, and $\rho^+_{(3)}, w^+_{(3)}, \gamma^+_{(3)}$ to denote the traffic state in the first cell of the outgoing link. The flow that crosses the junction, and the flow that leaves each incoming link can be determined by solving the following optimization problem:

$$\begin{aligned}
\underset{G_{(1)},G_{(2)},G_{(3)}}{\text{maximize}} \quad & G_{(1)} + G_{(2)} \\
\text{subject to} \quad & 0 \le G_{(1)} \le \tilde{S}_{(1)}(\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}), \\
& 0 \le G_{(2)} \le \tilde{S}_{(2)}(\rho^-_{(2)}, w^-_{(2)}, \gamma^-_{(2)}), \\
& 0 \le G_{(3)} \le \tilde{R}_{(3)}\left(\rho^+_{(3)M}, w^-_M, w^+_{(3)}, \gamma^+_{(3)}\right), \\
& G_{(3)} = G_{(1)} + G_{(2)}, \\
& \alpha G_{(1)} = (1-\alpha)\, G_{(2)},
\end{aligned} \tag{3.38}$$

where w^-_M denotes the property of the vehicles that cross the left boundary of the downstream cell. Importantly, this property is determined by the properties of the flows coming from both link 1 and link 2, so that diverge solver is more complex than the merge or one to one junction. Since we assume the vehicles strictly follow the priority rule specified by α, the

Algorithm 3 Junction solver: Diverge

Initialization:

Link 1 (incoming): $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$

Link 2 (outgoing): $\rho^+_{(2)}, w^+_{(2)}, \gamma^+_{(2)}$

Link 3 (outgoing): $\rho^+_{(3)}, w^+_{(3)}, \gamma^+_{(3)}$

Solve the junction problem (3.36) **according to** (3.37)**:** $G^*_{(1)}$**,** $G^*_{(2)}$**, and** $G^*_{(3)}$**.**

Prediction:

In the last cell of link 1 (incoming link):

$$\rho^{n+1}_{i_{\max}} = \rho^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - G^*_{(1)}\right)$$
$$y^{n+1}_{i_{\max}} = y^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(w^n_{i_{\max-1}} G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - w^n_{i_{\max}} G^*_{(1)}\right),$$

In the first cell of link 2 (outgoing link):

$$\rho^{n+1}_0 = \rho^n_0 + \frac{\Delta T}{\Delta x}\left(G^*_{(2)} - G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right),\right)$$
$$y^{n+1}_0 = y^n_0 + \frac{\Delta T}{\Delta x}\left(w^n_l G^*_{(2)} - w^n_0 G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),$$

In the first cell of link 3 (outgoing link):

$$\rho^{n+1}_0 = \rho^n_0 + \frac{\Delta T}{\Delta x}\left(G^*_{(3)} - G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),$$
$$y^{n+1}_0 = y^n_0 + \frac{\Delta T}{\Delta x}\left(w^n_l G^*_{(3)} - w^n_0 G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),$$

In all other cells for link 1, link 2, and link 3: apply the 2CTM (see Section 3.4)

property variable w_M^- can be computed by applying the methodology introduced in [91]:

$$w_M^- = \alpha w_{(1)}^- + (1-\alpha)\, w_{(2)}^-. \tag{3.39}$$

The solution $G_{(1)}^*$, $G_{(2)}^*$, and $G_{(3)}^*$ to the optimization problem (3.38) can be computed as:

$$\begin{aligned} G_{(1)}^* &= \min\{\tilde{S}_{(1)}, (1-\alpha)/\alpha \tilde{S}_{(2)}, (1-\alpha)\,\tilde{R}_{(3)}\}, \\ G_{(2)}^* &= \alpha/(1-\alpha)\, G_{(1)}^*, \\ G_{(3)}^* &= G_{(1)}^* + G_{(2)}^*. \end{aligned} \tag{3.40}$$

Then, the optimal fluxes $G_{(1)}^*$ and $G_{(2)}^*$ are used as the right boundary conditions for link 1 and link 2, and the flux $G_{(3)}^*$ is used as the left flow boundary condition for link 3 for traffic prediction, as shown in Algorithm 4.

The junction solvers for the first order traffic models can be derived by following the optimization problems and numerical solvers for the second order traffic model, except that the sending and receiving functions in the optimization problems should be replaced with the first order sending and receiving functions.

Algorithm 4 Junction solver: Merge

Initialization:

Link 1 (incoming): $\rho^-_{(1)}, w^-_{(1)}, \gamma^-_{(1)}$

Link 2 (incoming): $\rho^-_{(2)}, w^-_{(2)}, \gamma^-_{(2)}$

Link 3 (outgoing): $\rho^+_{(3)}, w^+_{(3)}, \gamma^+_{(3)}$

Solve the junction problem (3.38) **according to** (3.40)**:** $G^*_{(1)}$**,** $G^*_{(2)}$**, and** $G^*_{(3)}$**.**

Prediction:

In the last cell of link 1 (incoming link):

$$\begin{aligned}
\rho^{n+1}_{i_{\max}} &= \rho^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - G^*_{(1)}\right)\\
y^{n+1}_{i_{\max}} &= y^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(w^n_{i_{\max-1}} G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - w^n_{i_{\max}} G^*_{(1)}\right),
\end{aligned}$$

In the last cell of link 2 (incoming link):

$$\begin{aligned}
\rho^{n+1}_{i_{\max}} &= \rho^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - G^*_{(2)}\right)\\
y^{n+1}_{i_{\max}} &= y^n_{i_{\max}} + \frac{\Delta T}{\Delta x}\left(w^n_{i_{\max-1}} G_\rho\left(\rho^n_{i_{\max-1}}, \rho^n_{i_{\max}}, w^n_{i_{\max-1}}, w^n_{i_{\max}}\right) - w^n_{i_{\max}} G^*_{(2)}\right),
\end{aligned}$$

In the first cell of link 3 (outgoing link):

$$\begin{aligned}
\rho^{n+1}_0 &= \rho^n_0 + \frac{\Delta T}{\Delta x}\left(G^*_{(3)} - G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),\\
y^{n+1}_0 &= y^n_0 + \frac{\Delta T}{\Delta x}\left(w^n_l G^*_{(3)} - w^n_0 G_\rho\left(\rho^n_0, \rho^n_1, w^n_0, w^n_1\right)\right),
\end{aligned}$$

In all other cells for link 1, link 2, and link 3: apply the 2CTM (see Section 3.4)

Chapter 4

Multiple model nonlinear filters

In this Chapter, several nonlinear filters are introduced to solve the hybrid state estimation problem for system (1.1). A multiple model particle filter and an interactive multiple model ensemble Kalman filter are proposed, where the particle filter or the ensemble Kalman filter are used to accommodate the nonlinearity of the traffic incident model, and multiple model methods are deployed to address the switching dynamics of traffic operations. The multiple model particle filter is extended to a multiple model particle smoother to improve the estimation accuracy when data is limited. A variant of the multiple model particle filter, called the efficient multiple model particle filter, is developed for field implementations, and requires significantly less computation time compared to the multiple model particle filter and the interactive multiple model ensemble Kalman filter.

The proposed algorithms all adhere to the following main ideas. First, they are each variants of nonlinear filters which are needed to accommodate the nonlinearity of the traffic models. Second, they are each extended using methods that incorporate multiple models to handle the discrete model variable. Other nonlinear filters such as the unscented Kalman filter could also be considered and extended to solve the problem if desired.

4.1 Incident evolution equations

In the hybrid state estimation problem for (1.1), the model variable γ is used to model incidents through changes in the fundamental diagram. Specifically, the model variable γ is defined as an $i_{\max}+1$ dimensional vector on a single stretch of roadway, where the value in each element denotes the number of lanes open in the corresponding cell. The model variable is modeled as a u-state first-order Markov chain [17] with transition probabilities defined by:

$$\pi_{(k,j)} = p\left\{\gamma^n = j | \gamma^{n-1} = k\right\}, \quad k, j \in \Gamma, \tag{4.1}$$

where the set Γ defines all possible incident conditions ($\gamma \in \Gamma$). For example, consider a stretch of road with two lanes and four cells. Then $\gamma = [2,2,2,2]$ indicates there is no incident, $\gamma = [2,1,2,2]$ indicates there is an incident at cell one and the incident blocks one lane, and $\gamma = [0,2,2,2]$ indicates there is an incident at cell zero and the incident blocks two lanes. If all at most one incident at a time is considered, there are a total of $4\times 2+1$ models (four cells each with two incident severities, plus the no-incident model). The transition probability matrix is defined as $\bar{\Pi} = [\pi_{(k,j)}]$, which is a $u \times u$ matrix satisfying

$$\pi_{(k,j)} \geq 0 \quad \text{and} \quad \sum_{j=1}^{u} \pi_{(k,j)} = 1. \tag{4.2}$$

Equation (4.1) indicates the probability of the transition from one model to another. Here, we use $\Pi(\cdot)$ to denote the function that returns a new model variable $\tilde{\gamma}$ given a model variable γ according to the transition probability matrix $\bar{\Pi}$. In the traffic incident detection problem, it gives a realization of how many lanes will likely be open at the next time step given the incident status at the current time step.

4.2 Observation equation

The numerical simulations and field implementation performed in the dissertation use traffic density measurements from inductive loops and speed measurements from GPS equipped probe vehicles as measurements in the traffic estimation algorithms. The nonlinear operator h^n in (1.1) needs to be defined to link the system state to the measurements. When the first order traffic model is used, the nonlinear operator is given by:

$$h^n\left(\rho^n, \gamma^n\right) = H^n \begin{bmatrix} \rho^n \\ v\left(\rho^n, \gamma^n\right) \end{bmatrix}, \tag{4.3}$$

and when the second order traffic model is used, the nonlinear operator is given by:

$$h^n\left(\rho^n, w^n, \gamma^n\right) = H^n \begin{bmatrix} \rho^n \\ \tilde{v}\left(\rho^n, w^n, \gamma^n\right) \end{bmatrix}, \tag{4.4}$$

where $\rho^n = [\rho_0^n, \cdots, \rho_{i_{\max}}^n]$. The matrix H^n is constructed based on the locations where the measurements are acquired, and it is assumed the measurement vector is arranged with the density measurements stacked on top of the speed measurements. Note, however, that the observation operator h^n is in general nonlinear, due to v. It is time varying because the locations of GPS vehicles are not fixed, and the number of GPS equipped vehicles may change over time.

The observation noise term in (1.1),

$$\nu^n = \begin{bmatrix} \nu_{\text{density}}^n \\ \nu_{\text{speed}}^n \end{bmatrix},$$

is composed of two parts, ν_{density} and ν_{speed}, to emphasize that different error models are assumed for density and speed measurements. The measurement noise is assumed to follow

a normal distribution $\nu^n \sim \mathcal{N}(0, V^n)$, where V^n is the measurement error covariance matrix.

4.3 Particle filter based techniques

4.3.1 Bayesian problem and particle filter

To jointly estimate the continuous state x^n and the model variable γ^n, the augmented system state is defined by a vector $\chi^n = [x^n, \gamma^n]$. When the first order traffic flow model is applied, the system state x^n is defined as the traffic density ρ^n along the roadway. When the second order model is used, the state x^n is composed of both the traffic density and the vehicle property $x^n = [\rho^n, w^n]$. The estimation problem is formulated using the Bayesian approach [69]. This approach estimates the posterior probability density function $p(\chi^n|Z^n)$, where χ^n is the augmented system state and Z^n are the measurements from time step one to time step n, which is defined as $Z^n = \{z^1, \cdots, z^n\}$. The system state χ^n is recursively updated according to:

$$\begin{aligned} p\left(\chi^n|Z^{n-1}\right) &= \int p\left(\chi^n|\chi^{n-1}, \bar{\Pi}\right) p\left(\chi^{n-1}|Z^{n-1}\right) d\chi^{n-1}, \\ p\left(\chi^n|Z^n\right) &= \frac{p\left(z^n|\chi^n\right) p\left(\chi^n|Z^{n-1}\right)}{p\left(z^n|Z^{n-1}\right)}. \end{aligned} \tag{4.5}$$

The first equation is the prediction step and it propagates the posterior distribution of the system state from time step $n-1$ to the prior distribution at n, where $p(\chi^{n-1}|Z^{n-1})$ is the posterior distribution at time $n-1$, and $p\left(\chi^n|\chi^{n-1}, \bar{\Pi}\right)$ can be determined by the system evolution model f. The second equation is the measurement processing step. The new measurements z^n are used to calculate the posterior distribution of the augmented system state χ at time n, where $p(z^n|\chi^n)$ is the likelihood function and $p(z^n|Z^{n-1})$ is a normalizing constant. The likelihood function $p(z^n|\chi^n)$ indicates how well the predicted system state

matches the measurements. The posterior distribution is proportional to

$$p\left(\chi^n|Z^n\right) \propto p\left(z^n|\chi^n\right) p\left(\chi^n|Z^{n-1}\right). \tag{4.6}$$

The particle filter provides an approximate solution to this Bayesian problem by using a sequential Monte Carlo method. The basic idea behind the particle filter is as follows. First, a number of particles are generated to represent a sample approximation of the initial distribution of the system state. Each particle is assigned equal weight. Then, each particle is evolved forward in time according to the system evolution equation to achieve a prior distribution of the system state at the next time step. In the context of filtering, the prior refers to the estimate before measurements are obtained at the current time step. After measurements are obtained, the likelihood of each particle can be computed based on the assumed noise model of the measurements. The particles are then weighted based on the likelihood at this time step and their previous weights. The weights of the particles are normalized so that the sum of the particle weights equals to one.

Note that if we repeat this procedure for a number of time steps, the algorithm may run into the situation that all but one of the importance weights are close to zero. This is known as the *sample degeneracy* problem. To avoid the sample degeneracy problem, resampling is performed to remove low weight particles from the sample set. In the dissertation, the *systematic resampling algorithm* [17] is used, where particles with high weights will be multiplied and particles with low weight will be suppressed from the sample. As a result, particles that remain in the sample match well with the measurements and they will be used as inputs to the system evolution model for the next iteration.

Another common problem associated with particle filter is known as the *sample impoverishment* problem. Because of resampling, particles with large weights are likely to be replicated and particles with low weights are removed. As a result, the diversity of the particles will decrease after the resampling step. When the posterior distribution is represented

by one or a very small number of particles, the estimation accuracy will suffer [17]. In this case, a *regularized particle filter* can be used, where the particles are resampled by using a kernel density function, instead of directly replicating and removing the particles [95].

Particle filtering has been applied to solve traffic estimation problems [9, 10, 96], where no incident dynamics are considered. The particle filtering algorithm for traffic estimation developed in earlier work [10] is shown in Algorithm 5. In Algorithm 5, the notation l is used to index the particles and ζ_l is used to denote the weight of each particle l. The variable M denotes the total number of particles required. In the initial time step, a number of particles are generated based on the initial knowledge of the highway traffic state. These particles are used to predict future traffic state assuming all lanes are open. Next, the measurements from the field (i.e., density measurements and speed measurements) are used to determine the weight of each predicted traffic state, and resampling is performed to avoid the sample degeneracy problem.

Algorithm 5 Particle filter [10]

Initialization ($n = 0$): generate M samples x_l^0 and assign equal weights $\zeta_l^0 = 1/M$, where $l = 1, \cdots, M$

for $n = 1$ to $n_{\max}$ **do**

State prediction: $x_l^n = f\left(x_l^{n-1}\right) + \omega^{n-1}$ for all l

Measurement processing:

calculate the likelihood: $p\left(z^n | x_l^n\right)$ for all l

update weights: $\zeta_l^n = \zeta_l^{n-1} p\left(z^n | x_l^n\right)$ for all l

normalize weights: $\hat{\zeta}_l^n = \zeta_l^n / \sum_{l=1}^{M} \zeta_l^n$ for all l

Resampling: multiply (suppress) samples χ_l^n with high (low) importance weights $\hat{\zeta}_l^n$

Output: posterior distribution of x^n

Reassign weights: $\zeta_l^n = 1/M$ for all l

$n = n + 1$

end for

4.3.2 Multiple model particle filter

In the hybrid state estimation problem, the model involves both continuous variables (associated with the traffic state), and discrete variables (associated with the model variable). Thus, a variant of the particle filter, known as the multiple model particle filter [17], is used to estimate the traffic state and model variables. The main difference between the multiple model particle filter and the standard particle filter is that the multiple model particle filter allows the system to have several models, and particles are randomly generated for possible models given their probability of occurrence. The filter has a model transition step that describes the switching dynamics of the system model. The idea of the multiple model particle filter is that if the state x^n generated by a model variable γ^n is more likely compared to the traffic state generated by other model variables, then the algorithm estimates the system is operating in that model at time n.

Algorithm 6 Multiple model particle filter [17]

Initialization ($n = 0$): generate M samples χ_l^0 and assign equal weights $\zeta_l^0 = 1/M$, where $l = 1, \cdots, M$

for $n = 1$ to $n_{\max}$ **do**

Model transition: $\gamma_l^n = \Pi\left(\gamma_l^{n-1}\right)$ for all l

State prediction: $x_l^n = f\left(x_l^{n-1}, \gamma_l^n\right) + \omega^{n-1}$ for all l

Measurement processing:

calculate the likelihood: $p\left(z^n|\chi_l^n\right)$ for all l

update weights: $\zeta_l^n = \zeta_l^{n-1} p\left(z^n|\chi_l^n\right)$ for all l

normalize weights: $\hat{\zeta}_l^n = \zeta_l^n / \sum_{l=1}^M \zeta_l^n$ for all l

Resampling: multiply/ suppress samples χ_l^n with high/ low importance weights $\hat{\zeta}_l^n$

Output: posterior distribution of x^n and γ^n

Reassign weights: $\zeta_l^n = 1/M$ for all l

$n = n + 1$

end for

The pseudocode of the multiple model particle filter is summarized in Algorithm 6. The multiple model particle filter consists of the following steps.

- *Initialization*: Generate M particles from the initial distribution of χ^0 and assign each particle an equal weight. Like the particle filter, the initial state χ^0, which is composed of x^0 and γ^0, is given by an initial distribution reflecting the knowledge on the initial state.

- *Model transition*: Calculate the model variable prediction for all particles according to the transition matrix (4.1) and (4.2). Then, assign particles to each model. Here, the particles assigned to each model is proportional to the model probability.

- *State prediction*: Calculate the prior distribution of the system state x^n according to the traffic models (e.g., (3.4) or (3.21)).

- *Measurement processing*: Calculate the likelihood of each particle and update the weight of each particle based on the likelihood and its previous weight. Then, normalize the weight for all particles.

- *Resampling*: Resample particles based on their weights. Similar to the standard particle filter, the systematic resampling algorithm [17] is used.

- *Output*: The solution to this problem is a posterior distribution of augmented system state χ^n. If the distribution of the model variable γ^n takes a unique value at all time steps n, it means the algorithm estimates the precise location and severity of the traffic incident (or the lack thereof). If more than one value of γ^n is returned at time n, it means that multiple locations and/or severities of incidents are consistent with the observed data.

The MMPF Algorithm 6 has the potential to perform well when traffic sensors are dense, but the estimation accuracy may decrease if the number of sensors is limited.

On the one hand, when a traffic incident occurs and if there are no sensors nearby, it will take time for the nearest sensor to detect the congestion. Consequently, the correct incident model cannot be identified at the time when the traffic incident occurs, and the particles generated by wrong models will be assigned with high weights. These wrong particles will then be used as inputs to calculate the prior distribution for the next time step. If the true model variable is not identified for several consecutive time steps, more particles in the sample will become incorrect. Eventually, when the incident information propagates to the sensor, the measurements may not match with any particles in the sample and the filter will collapse.

On the another hand, when there are no incidents and the sensors are sparse, the particles generated by incident models may match well with the measurements. If there are no sensors near the predicted incident location, the filter cannot verify that there is actually no congestion. Consequently, particles generated by incident models will remain in the sample set, and the estimation accuracy of the algorithm will decrease if too many incorrect samples are retained in the sample. To address these problems, we apply the idea of fixed-lag smoothing [95, 97] and combine it with the multiple model particle filter.

4.3.3 Multiple model particle smoother

A smoothing algorithm estimates the posterior distribution of the system state at time n given measurements up to some later time T ($T > n$). If the estimate of the system state is not required instantly, measurements at a later time can help to provide a better estimation of the current system state. The fixed–lag approximation [95, 97] is described by:

$$p\left(\Psi^n|Z^T\right) \approx p\left(\Psi^n|Z^{\min\{n+\Delta S,T\}}\right), \tag{4.7}$$

where $\Psi^n = \{\chi^0, \cdots, \chi^n\}$, and ΔS is a fixed time lag. In general, $n + \Delta S$ is smaller than T. The assumption for this approximation is that measurements after time $n + \Delta S$ bring no

additional information about the state Ψ^n.

In our problem, the objective is to jointly estimate the traffic state x^n and the model variable γ^n at each time step. By applying smoothing, the model variable γ^n is identified by its performance at ΔS time steps in the future beyond time n. In other words, an additional ΔS time steps are allowed to let the traffic information propagate to the nearest sensor (mobile or fixed), where incorrect models can be rejected.

The fixed–lag smoothing algorithm is combined with the multiple model particle filter. The resulting MMPS is shown in Algorithm 7. The main difference between the multiple model particle smoother and the multiple model particle filter is the measurement processing stage. In Algorithm 7, the variable τ is the time index of the smoothing window. Here, $x_l^n(\tau)$ denotes the traffic state of particle l for time step $n+\tau-1$. Similarly, $\gamma_l^n(\tau)$ denotes the model variable γ and $\zeta_l^n(\tau)$ denotes the weight of each particle during the smoothing window from time step n to time steps $n+\tau-1$. During the smoothing step, the weight of each particle is determined by its previous weight and its likelihood calculated at the current time step. In the multiple model particle smoother, the weight of each particle is determined by its previous weight and the likelihood calculated during the time period $\Delta S+1$ [98]:

$$\zeta^n \propto p\left(z^{n+\Delta S}|Z^{n+\Delta S-1}, \Gamma^{n-1}\right), \tag{4.8}$$

where $\Gamma^{n-1} = \{\gamma^0, \cdots, \gamma^{n-1}\}$. Accordingly, resampling is performed using the weight calculated from the measurements up to $n+\Delta S$. During smoothing, each particle is evolved forward in time. Thus, the state x^n generated by a model variable γ^n is evaluated for an additional ΔS time steps. The choice of ΔS is up to the algorithm designer, but practically it should be set as a function of the number of sensors available. If sensors are dense, the value of ΔS can be small. If sensors are located far apart, it takes more time for the information to propagate to sensors, and a larger value for ΔS is needed to see any significant improvement in performance. Obviously, there is a price for accuracy improvement. Instead

Algorithm 7 Fixed–lag multiple model particle smoother

Initialization ($n = 0$): generate M samples χ_l^0 and assign equal weights $\zeta_l^0 = 1/M$, where $l = 1, \cdots, M$

for $n = 1$ to $n_{\max}$ **do**

Model transition: $\gamma_l^n = \Pi\left(\gamma_l^{n-1}\right)$ for all l

State prediction: $x_l^n = f\left(x_l^{n-1}, \gamma_l^n\right) + \omega^{n-1}$ for all l

Measurement processing and smoothing:

for $\tau = 1$ to $\Delta S + 1$ **do**

initialization:

$x_l^n(1) = x_l^n$, $\gamma_l^n(1) = \gamma_l^n$, and $\zeta_l^n(0) = 1/M$ for all l

calculate the likelihood:

$p\left(z^{n+\tau-1} | x_l^n(\tau), \gamma_l^n(\tau)\right)$

update weights:

$\zeta_l^n(\tau) = \zeta_l^n(\tau - 1)\, p\left(z^{n+\tau-1} | x_l^n(\tau), \gamma_l^n(\tau)\right)$ for all l

normalize weights:

$\hat{\zeta}_l^n(\tau) = \zeta_l^n(\tau) / \sum_{l=1}^M \zeta_l^n(\tau)$ for all l

$\zeta_l^n(\tau) = \hat{\zeta}_l^n(\tau)$ for all l

if $\tau \neq \Delta S + 1$ **then**

$\gamma_l^n(\tau + 1) = \Pi\left(\gamma_l^n(\tau)\right)$ for all l

$x_l^n(\tau + 1) = f\left(x_l^n(\tau), \gamma_l^n(\tau + 1)\right) + \omega^{n+\tau-1}$

end if

$\tau = \tau + 1$

end for

Resampling: multiply/ suppress samples χ_l^n with high/ low importance weights $w_l^n(\Delta S + 1)$

Output: posterior distribution of x^n and γ^n

$n = n + 1$

end for

of real time estimation, the fixed–lag multiple model particle smoother practically estimates traffic with a lag of $\Delta S \times \Delta T$.

Another way to perform smoothing is to calculate the weight and resample at each time step during the smoothing period. However, frequent resampling can result in a loss of diversity of the particles which leads to the sample impoverishment problem [17].

4.3.4 Efficient multiple model particle filter

The computation time for solving the hybrid state estimation problem for system (1.1) is proportional to the number of models involved in the system, and the number of models involved is a function of the number of cells and lanes of the highway. For field implementation, the total number of models of the hybrid system can be large because the road network can contain many miles of highway and may include segments with many lanes. Moreover, the road needs to be discretized into smaller cells when ramps are considered, so that the ramps are located at (or close to) the boundary between cells while maintaining a fixed spatial discretization throughout the network. In this case, we show later that the proposed MMPF (Algorithm 6) may not run in real time for even moderate network sizes.

The MMPF is computationally costly because the algorithm runs a particle filter on likely models of the system, where particles are assigned to each model proportional to the model transition probability matrix $\bar{\Pi}$. Since traffic incidents are rare events and the transition probability to an incident model is small, a very large number of particles must be drawn in order to generate samples that cover many of the incident states.

In this section, an EMMPF is proposed to approximately solve the hybrid state estimation problem (1.1), shown in Algorithm 8, which requires significantly less computation time compared to the algorithms in works [99, 100]. The proposed EMMPF is essentially a multiple model approach applied to the particle filter, where the interactive multiple model approach is used to estimate the correct model, and the particle filter is performed on the estimated model for traffic estimation. Different from the MMPF where the estimate is

determined using samples from all models generated during the model transition step, the EMMPF only uses a single sample in each model to infer the correct model. Then, all particles are evolved forward in time using the most likely model determined in the model selection step.

Algorithm 8 Efficient multiple model particle filter

Initialization: generate M samples x_l^0 and assign equal weights $\zeta_l = 1/M$, where $l = 1, \cdots, M$. The most likely sample of x_l^0 is denoted by $\bar{x}^0$.

for $n = 1$ to $n_{\max}$ **do**

1. Model probability update (for all $\gamma \in \Gamma$):

Compute model probability: $\mu_{(\gamma)}^n = \pi_{(\gamma,\gamma^{n-1})}$

Predicted state: $x_{(\gamma)}^n = f\left(x^{n-1}, \gamma\right) + \omega^{n-1}$

Calculate the likelihood: $p_{(\gamma)}^n = p\left(z^n | x_{(\gamma)}^n, \gamma\right)$

Update the model probability: $\mu_{(\gamma)}^n = \mu_{(\gamma)}^n p_{(\gamma)}^n$

2. Model selection: $\gamma^n = \text{argmax}_\gamma \left(\mu_{(\gamma)}^n\right)$

3. Model–conditioned particle filter:

Prediction: $x_l^n = f\left(x_l^{n-1}, \gamma^n\right) + \omega^{n-1}$ for all l

Calculate the likelihood: $p\left(z^n | x_l^n\right)$ for all l

Update weights: $\zeta_l^n = \zeta_l^{n-1} p\left(z^n | x_l^n\right)$ for all l

Normalize weights: $\hat{\zeta}_l^n = \zeta_l^n / \sum_{l=1}^M \zeta_l^n$ for all l

Resampling: multiply/ suppress samples x_l^n with high/ low importance weights $\hat{\zeta}_l^n$

Output: selected model γ^n and posterior distribution of x^n

Reassign weights: $\zeta_l^n = 1/M$ for all l

$n = n + 1$

end for

In Algorithm 8, step one and step two describe the effective multiple model approach, where the model γ^n is selected. The term $\mu_{(\gamma)}^n$ is the probability associated with model γ at time step n. In step one, the model probability is reinitialized based on the model γ at the previous time step and the transition probability. Next, the algorithm predicts the traffic state by assuming γ is correct, and determines the likelihood of γ by evaluating how well

the predicted system state $x^n_{(\gamma)}$ matches with the measurements z^n. Here, the system state $\bar{x}^{n-1}$ is the most likely particle of the posterior distribution x^{n-1}_l from the last time step. The model probability $\mu^n_{(\gamma)}$ is updated by multiplying the initial model probability and the model likelihood. This procedure is repeated for all models defined in the system. In step two, the algorithm selects the γ with the highest model probability as the correct model in the current time step.

In step 3, a standard particle filter is performed on the selected model. Here, the prior distribution of x^{n-1}_l is the posterior distribution of the system state from the previous time step. During the prediction step, we predict the future traffic state for all particles. Then, we compute the likelihood of each particle and update its weights. Next, the weights of the particles are normalized so that the total weight of all particles sums to one. Finally, the systematic resampling algorithm [17] is performed to resample the distribution based on the weights of the particles. The selected model γ^n and the resulting posterior distribution of x^n are outputted and used as inputs to the algorithm for the next time step.

Compared to the MMPF, where particles are assigned according to the transition probability and the particle filter is performed using samples from all of the realized models during the model transition step, the proposed EMMPF runs a single sample for each model to infer the correct model, and performs the filter only on the model that is estimated to be true. As a result, significant amount of computation time can be saved when the number of models is large.

4.4 Kalman filter based techniques

4.4.1 Ensemble Kalman filter

A second type of algorithm to solve the hybrid state estimation problem is constructed based on the seminal Kalman filter [101], which is a *best linear unbiased estimator* (BLUE). Compared to the particle filter, where the full statistics are represented by a distribution

of the system state that propagates though the model and is corrected in a fully Bayesian update step, the Kalman filter performs a linear correction that minimizes the posterior error covariance.

The classical Kalman filter is only applicable for linear systems (e.g., with a linear evolution equation and a linear observation equation, both with additive Gaussian noises). To apply the ideas of Kalman filtering to nonlinear systems, several approaches have been developed, the most common being the extended Kalman filter [102], the unscented Kalman filter [26], and the ensemble Kalman filter [103].

The extended Kalman filter requires linearizing the evolution and observation equations of the system around the best estimate at each time step, and then performs the standard Kalman update equations on the linearized system. For highly nonlinear systems, the extended Kalman filter can suffer poor performance which has in turn motivated the development of alternative methods. The unscented Kalman filter is notable because it performs a deterministic sampling of the posterior state error covariance matrix at the previous time step, then evolves the deterministic samples through the nonlinear model to generate the correct prior error covariance after propagation through the nonlinear model. This error covariance can then be used in the minimal variance linear update step of the Kalman filter. The ensemble Kalman filter takes a slightly different approach, and performs a Monte Carlo integration of the posterior state error distribution through the nonlinear evolution equation before the linear minimal variance correction step is performed when measurements are made available. When the number of samples (ensembles) is large, the EnKF converges to the Kalman filter for linear systems [103].

In this dissertation, the EnKF is used for traffic state estimation due to its computational efficiency and ease of implementation compared to other extensions of the Kalman filter for nonlinear systems. The standard EnKF is described in Algorithm 9. Similar to the particle filter, the EnKF first draws an initial distribution $x_l^{0|0}$ of the traffic state based on the initial traffic state distribution, where $l = 1, \cdots, M$ denotes the samples (or ensembles). Each

sample is evolved forward in time using the traffic model to predict the future traffic state and the covariance of the predicted traffic state is computed. Next, the minimal variance linear correction (Kalman gain) K is computed by using the predicted covariance matrix and the measurement noise. The posterior estimate of the traffic state is calculated using the Kalman gain correction and the measurements collected from the field. Here, the variable ν_l^n denotes a realization of the model noise ν^n for sample l. The index $n|n-1$ denotes the prior of a variable (before the measurements at time n are obtained) and the subindex $n|n$ denotes the posterior of a variable (after the measurements at time n are obtained). Finally, the posterior covariance matrix is updated. Note that the form of the EnKF shown in Algorithm 9 highlights its structural similarities with the Kalman filter, but in numerical implementations, a variety of speedup techniques can be implemented to avoid the explicit construction of the state error covariance matrix to compute the Kalman gain [104].

The EnKF can be used for estimation when a system is nonlinear. However, it cannot directly be used to solve hybrid state estimation problems due to the discrete variables involved in the state. In the next section, we introduce how the EnKF can be extended in order to solve the joint traffic state estimation and incident detection problem.

4.4.2 Interactive multiple model Ensemble Kalman filter

In this section, an interactive multiple model ensemble Kalman filter is proposed to solve the hybrid state estimation problem for system (1.1). This IMM EnKF is developed based on the IMM method [46]. The EnKF is incorporated into the IMM framework, and the resulting IMM EnKF algorithm is summarized in Algorithm 10. In the algorithm, l denotes the ensemble index, M is the total number of ensembles for each model, $\mu_{(\gamma)}$ is the probability of model γ, $x_{(\gamma,l)}$ is the state generated by model γ and ensemble l, $\Sigma_{(\gamma)}$ is the predicted covariance matrix, $K_{(\gamma)}^n$ is the Kalman gain of model γ at time n, and $L_{(\gamma)}$ is the likelihood of each model calculated from the EnKF.

At each time step, the algorithm first determines the probability of each model γ^n based

Algorithm 9 Ensemble Kalman filter [8]

Initialization: generate M samples $x_l^{0|0}$, where $l = 1, \cdots, M$.

for $n = 1$ to $n_{\max}$ **do**

Predicted state: $x_l^{n|n-1} = f\left(x_l^{n-1|n-1}\right) + \omega^{n-1}$ for all l

Predicted covariance: $\Sigma^{n|n-1} = \frac{1}{M-1}\sum_{l=1}^{M}\left(x_l^{n|n-1} - \frac{1}{M}\sum_{l=1}^{M} x_l^{n|n-1}\right)(\cdots)^T$

Kalman Gain: $K^n = \Sigma^{n|n-1}\left(H^n\right)^T\left(H^n\Sigma^{n|n-1}H^{nT} + V^n\right)^{-1}$

Updated state: $x_l^{n|n} = x_l^{n|n-1} + K^n\left(z^n + \nu_l^n - H^n x_l^{n|n-1}\right) \quad l = 1, \cdots, M$

Updated covariance: $\Sigma^{n|n} = \Sigma^{n|n-1} - K^n H^n \Sigma^{n|n-1}$

Note: when the observation equation is nonlinear, the calculation of the covariance in the Kalman gain follows [105]:

$\Sigma^{n|n-1}\left(H^n\right)^T = \frac{1}{M-1}\left(x_l^{n|n-1} - \frac{1}{M}\sum_{l=1}^{M} x_l^{n|n-1}\right)\left(h^n\left(x_l^{n|n-1}\right) - \frac{1}{M}\sum_{l=1}^{M} h^n\left(x_l^{n|n-1}\right)\right)^T$

$\left(H^n\right)\Sigma^{n|n-1}\left(H^n\right)^T = \frac{1}{M-1}\left(h^n\left(x_l^{n|n-1}\right) - \frac{1}{M}\sum_{l=1}^{M} h^n\left(x_l^{n|n-1}\right)\right)(\cdots)^T$

end for

on the previous system model γ^{n-1} and the transition matrix defined in (4.1). Then, for each model, an EnKF is performed to estimate the state. The EnKF algorithm first computes the one step predicted state and covariance through the traffic model. After the measurements are received, it updates the predicted state and covariance using the measurements at the current time step and the computed Kalman gain. The probability of each model γ^n is updated by considering the model probability $\mu_{(\gamma)}^{n|n-1}$ and the model likelihood $L_{(\gamma)}^n$, which indicates how well the state generated by γ^n matches with the measurements. It is calculated by using the mean of the estimated state $\left(\Sigma_{l=1}^{M} x_{(\gamma,l)}^{n|n}\right)/M$, the measurements z^n, and the noise model ν^n. The model γ^n with the highest probability is used to estimate the true state. Again, the calculation of the covariance in the Kalman gain should follow the equations at the end of the algorithm when a nonlinear observation operator is needed (e.g., if velocity measurements are received).

In terms of computational cost, the IMM EnKF requires less computation time compared to the MMPF for traffic estimation and incident detection. Recall that the number of samples assigned to each model by the MMPF is proportional to the model probability specified in

Algorithm 10 Interactive multiple model ensemble Kalman filter

Initialization: generate M samples $x_l^{0|0}$, where $l = 1, \cdots, M$.

for $n = 1$ to $n_{\max}$ **do**

1. Model–conditioned reinitialization (for all $\gamma \in \Gamma$):

Predicted model probability: $\mu_{(\gamma)}^{n|n-1} = \pi_{\left(\gamma,\gamma^{n-1|n-1}\right)}$

2. Model–conditioned EnKF (for all $\gamma \in \Gamma$):

Predicted state: $x_{(\gamma,l)}^{n|n-1} = f\left(x_{(l)}^{n-1|n-1}, \gamma\right) + \omega^{n-1}$ for all l

Predicted covariance: $\Sigma_{(\gamma)}^{n|n-1} = \frac{1}{M-1}\sum_{l=1}^{M}\left(x_{(\gamma,l)}^{n|n-1} - \frac{1}{M}\sum_{l=1}^{M} x_{(\gamma,l)}^{n|n-1}\right)(\cdots)^T$

Kalman gain: $K_{(\gamma)}^n = \Sigma_{(\gamma)}^{n|n-1} H^{nT}\left(H^n \Sigma_{(\gamma)}^{n|n-1} H^{nT} + V^n\right)^{-1}$

Updated state: $x_{(\gamma,l)}^{n|n} = x_{(\gamma,l)}^{n|n-1} + K_{(\gamma)}^n\left(z^n + \nu_l^n - H^n x_{(\gamma,l)}^{n|n-1}\right) \quad l = 1, \cdots, M$

Updated covariance: $\Sigma_{(\gamma)}^{n|n} = \Sigma_{(\gamma)}^{n|n-1} - K_{(\gamma)}^n H^n \Sigma_{(\gamma)}^{n|n-1}$

3. Model probability update (for all $\gamma \in \Gamma$):

Model likelihood: $L_{(\gamma)}^n = p(z^n | x_{(\gamma,l)}^{n|n}) \quad l = 1, \cdots, M$

Model probability: $\mu_{(\gamma)}^{n|n} = \frac{\mu_{(\gamma)}^{n|n-1} L_{(\gamma)}^n}{\sum_{\gamma' \in \Gamma} \mu_{(\gamma')}^{n|n-1} L_{(\gamma')}^n}$

4. Model inference (for all $\gamma \in \Gamma$):

Model selection: $\gamma^{n|n} = \text{argmax}_\gamma\left(\mu_{(\gamma)}^{n|n}\right)$

State selection: $x_{(l)}^{n|n} = x_{(\gamma_{n|n},l)}^{n|n}$

Note: When the observation equation is nonlinear, the calculation of the covariance in the Kalman gain follows [105]:

$\Sigma_{(\gamma)}^{n|n-1}(H^n)^T =$

$\frac{1}{M-1}\left(x_{(\gamma,l)}^{n|n-1} - \frac{1}{M}\sum_{l=1}^{M} x_{(\gamma,l)}^{n|n-1}\right)\left(h^n\left(x_{(\gamma,l)}^{n|n-1}, \gamma^n\right) - \frac{1}{M}\sum_{l=1}^{M} h^n\left(x_{(\gamma,l)}^{n|n-1}, \gamma^n\right)\right)^T$

$(H^n)\Sigma_{(\gamma)}^{n|n-1}(H^n)^T = \frac{1}{M-1}\left(h^n\left(x_{(\gamma,l)}^{n|n-1}, \gamma^n\right) - \frac{1}{M}\sum_{l=1}^{M} h^n\left(x_{(\gamma,l)}^{n|n-1}, \gamma^n\right)\right)(\cdots)^T$

end for

the transition probability matrix. As a result, a very large sample size is required in order to generate sufficient samples in all of the possible models since traffic incidents are rare events, and moreover, the number of severities and locations is large. In contrast, the IMM EnKF approach specifies the model probability for all models based on the transition probability, then runs an EnKF on each model with a fixed number of samples. The estimation results from the filter are combined with the model probability determined from the transition probability to infer the correct model. As a result, the total number of samples needed for the IMM EnKF is much less than that of MMPF. Compared to the EMMPF, the IMM EnKF requires more computation time because the IMM EnKF runs EnKF on all possible models and then determines the correct model after the measurements are received and the state is corrected, while the EMMPF runs a single sample on all models to select the correct model, and only runs the particle filter on the selected model.

4.5 Summary of proposed algorithms

In this chapter, four algorithms were presented for solving the joint traffic state and incident detection problem: *i*) the multiple model particle filter, *ii*) the multiple model particle smoother, *iii*) the efficient multiple model particle filter, and *iv*) the interactive multiple model ensemble Kalman filter. The multiple model particle smoother is an extension of the multiple model particle filter, which is able to improve the estimation accuracy by allowing a time lag during the estimation so that measurements received after time n can be used to determine the correct model at time n. The efficient multiple model particle filter is developed to reduce the computation time compared to the MMPF. Instead of running particles through on all of the possible models and calculating a posterior on the model variable, the algorithm runs a single sample on each model to determine the most likely system model, and performs traffic estimation only on the estimated model. This efficient multiple model particle filter can significantly reduce the computation time compared to

the other algorithms. Later in the field implementation, we will show the efficient multiple model particle filter runs in real time for moderate network sizes while the other algorithms do not. The reduced run time comes at the cost of a lower accuracy on estimating the model variable, since only a single sample is used to select the model.

It should be mentioned that the efficient multiple model particle filter can be extended to an efficient multiple model particle smoother by applying the same methodology presented in Section 4.3.3. Similarly, the IMM EnKF can be extended to an IMM ensemble Kalman smoother by following the techniques introduced in [106], and an efficient multiple model EnKF can also be developed by replacing the particle filter with an EnKF in Algorithm 8, after the model is selected. The proposed methods developed in this chapter are designed to give some insights of the potential implications of the various tradeoffs between accuracy and run time by exploring two different filtering methodologies (e.g., particle filtering and ensemble Kalman filtering), by exploring tradeoffs between filtering and smoothing, and by exploring tradeoffs in the different multiple model formulations to handle the discrete variables introduced when modeling incidents.

Chapter 5

Numerical experiments in microsimulation with CORSIM

To test whether the proposed algorithms have potential to work in practice, the CORSIM microscopic simulation software is used to simulate traffic and congestion caused by incidents. The simulation results from CORSIM are used as the source of the traffic measurements, and also as the definition of the *true* state, to be estimated by the proposed algorithms. The main idea is that if the algorithms are able to detect traffic incidents using the data generated by CORSIM, which is an entirely different modeling framework from the macroscopic models used in the estimator, it has a higher potential to perform well in the field.

The microscopic simulation software CORSIM is developed by the *Federal Highway Administration* (FHWA). It models individual vehicle movements based on car following and lane-changing theories on a second by second basis. The model also includes random processes to model different driver, vehicle, and traffic system behaviors. This is in contrast to the macroscopic traffic models used in the estimator, which model only conservation of vehicles (and possibly conservation of total property). An important feature of this simulation environment is that the assumed true model CORSIM has dynamics that are not directly incorporated in the macroscopic models used in the estimator. This also occurs when one

applies macroscopic estimators to estimate traffic in experimental field deployments.

For illustration purposes, the CORSIM simulations are performed on a four mile long, three–lane freeway segment with a speed limit of 65 mph. The simulation is performed for one hour (180 time steps when ΔT is chosen as 20 seconds). One incident is created in cell four, which is 1.36 miles from the starting point of the freeway segment. The incident occurs between time steps 60 and 120, and it blocks one lane. In CORSIM, a *rubberneck factor* of 50% is given for the remaining lanes at the incident location to model the phenomenon that drivers slow down when passing an incident on the road, even if no congestion occurs in front of the drivers.

In this chapter, we first test the proposed MMPF and the MMPS algorithms on the first order traffic incident model with incident data generated by CORSIM while varying the penetration rate of GPS equipped vehicles to illustrate the potential benefits of smoothing. Next, the MMPF and the IMM EnKF algorithms are compared with a standard macroscopic traffic estimator based on particle filtering applied to a scalar traffic model, and the California incident detection algorithm. This allows for comparisons between standard approaches for traffic incident detection and traffic estimation, and also allows comparisons between the different families of estimators (e.g., particle and Kalman filters). The traffic volume is varied to highlight the fact that incident detection depends critically on the flow rate. Then, the EMMPF is implemented on both the first order model and the second order model, and tested with the incident data generated by CORSIM, again while varying the flow rate. Finally, a sensitivity analysis is performed on the EMMPF with the first order traffic model, to study how the estimation accuracy is affected by the model parameters.

There are several important results of the simulations. First, they show that the MMPF, MMPS, IMM EnKF, and EMMPF are able to jointly estimate traffic state and detect incidents. Second, compared to the MMPF, the MMPS can improve estimation accuracy when data is limited due to the extra smoothing window that allows more measurements to be used to determine the correct model variable. Third, the MMPF and the IMM EnKF can

provide better traffic state and incident estimates than the particle filter and the California algorithm working independently, which highlights the benefit of estimating both incidents and the traffic state in a same algorithm compared to two separate algorithms. Fourth, it is verified that the EMMPF requires significantly less computation time compared to the MMPF and the IMM EnKF without a significant deterioration in the accuracy of the estimates. Finally, the comparisons between the first and second order models illustrate that similar traffic state and incident estimates are obtained when either model is used for the traffic dynamics.

5.1 Fundamental diagram calibration for CORSIM traffic

To test the proposed algorithms in CORSIM, the fundamental diagram in the macroscopic traffic models needs to be calibrated. In particular, the shape of the fundamental diagram for the traffic evolution of CORSIM needs to be determined. To calibrate the model, we build a three–lane freeway in CORSIM and conduct simulations with various inflow values and downstream speed limits (to generate congestion). All other CORSIM parameters are set as the default values. For each simulation, we collect the traffic occupancy data and traffic flow data over a two minute interval by processing the vehicle trajectory data exported from CORSIM passing over a simulated inductive loop detector. Then, the traffic occupancy data is converted to traffic density data to construct the density–flow relationship, which is analogous to the processing that occurs in traffic monitoring systems relying on inductive loops. The red dots in Figure 5.1 show the resulting density–flow relationship obtained from the CORSIM simulations, from which the fundamental diagram can be calibrated. In an experimental deployment, the density–flow data would be obtained by using historical data from the loop detectors collected over many days or weeks to calibrate the parameters of the fundamental diagram.

In the CORSIM simulations, a piece-wise quadratic function is used to fit the data. The

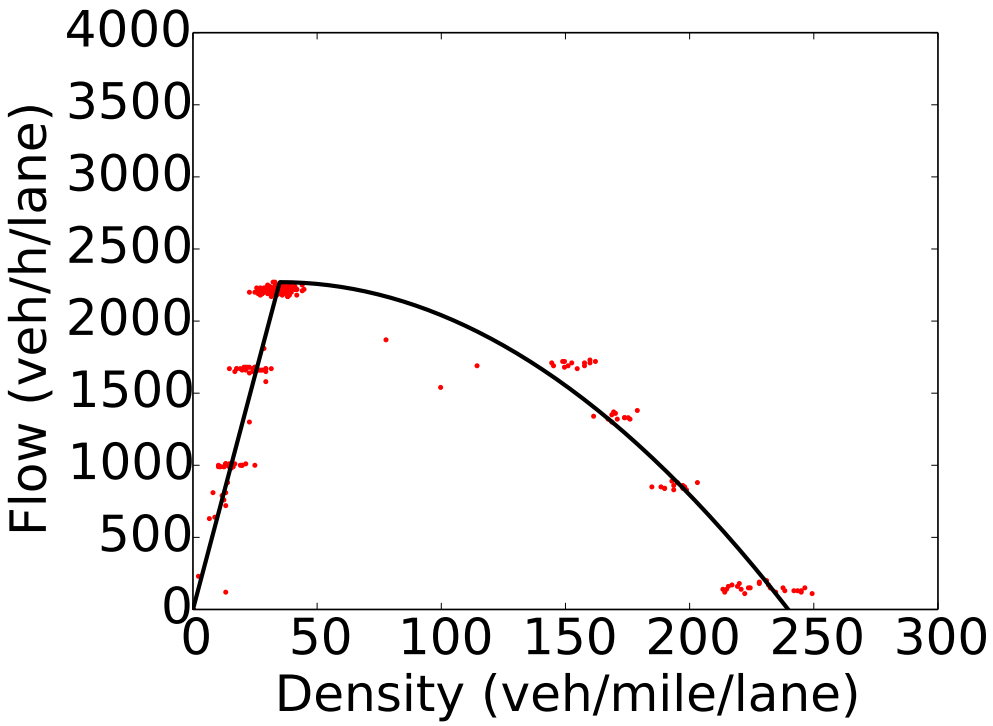


Figure 5.1: Density–flow relationship for the first order traffic model (equivalent to the second order traffic model with $w = 1$). The red dots are measurements obtained from CORSIM and the solid black line is the calibrated density–flow model.

flow functions for the first order traffic model and the second order traffic model are given by:

$$q(\rho,\gamma)=\begin{cases}\rho v_{\max}(\gamma)\left(1-\frac{\rho}{\beta(\gamma)}\right) & \text{if } \rho\le\rho_c(\gamma)\\ a(\gamma)\rho^2+b(\gamma)\rho+c(\gamma) & \text{if } \rho\ge\rho_c(\gamma),\end{cases} \tag{5.1}$$

and

$$\tilde{q}(\rho,w,\gamma)=\begin{cases}\rho v_{\max}(\gamma)\left(1-\frac{\rho}{\beta(\gamma)}\right) & \text{if } \rho\le\rho_c(w,\gamma)\\ a(w,\gamma)\rho^2+b(w,\gamma)\rho+c(w,\gamma) & \text{if } \rho\ge\rho_c(w,\gamma),\end{cases} \tag{5.2}$$

where the function $q(\rho,\gamma)$ denotes the flow function for the first order model and the function $\tilde{q}(\rho,w,\gamma)$ is the flow function for the second order model.

For the first order traffic model, the parameters a, b, and c determine the density–flow relationship for the congested regime, and can be computed by traffic model parameters $\rho_c(\gamma)$, $\rho_m(\gamma)$, $v_{\max}(\gamma)$, and $\beta(\gamma)$:

$$\begin{cases}a\left(\rho_m(\gamma)\right)^2+b\left(\rho_m(\gamma)\right)+c=0\\ -\frac{b}{2a}=\rho_c(\gamma)\\ \frac{4ac-b^2}{4a}=\rho_c(\gamma)v_{\max}(\gamma)\left(1-\frac{\rho_c(\gamma)}{\beta(\gamma)}\right).\end{cases} \tag{5.3}$$

Similarly, for the second order traffic model, the parameters a, b, and c can be determined by the traffic model parameters $\tilde{\rho}_c(w,\gamma)$, $\tilde{\rho}_m(w,\gamma)$, $v_{\max}(\gamma)$, and $\beta(\gamma)$:

$$\begin{cases}a\left(\tilde{\rho}_m(w,\gamma)\right)^2+b\tilde{\rho}_m(w,\gamma)+c=0\\ -\frac{b}{2a}=\tilde{\rho}_c(w,\gamma)\\ \frac{4ac-b^2}{4a}=\tilde{\rho}_c(w,\gamma)v_{\max}(\gamma)\left(1-\frac{\tilde{\rho}_c(w,\gamma)}{\beta(\gamma)}\right).\end{cases} \tag{5.4}$$

Next, the calibration procedures for the model parameters are described. The parameter calibration is performed for the first order traffic model. The model parameters for the second traffic model are determined by perturbing the first order model parameters (i.e., ρ_c, ρ_m). The maximum flow is calibrated as 2,210 veh/hour/lane according to the calibration

Parameters	three lanes	two lanes	one lane	unit
$v_{max}(\gamma)$	65	18	18	vpm
$q_{max}(\gamma)$	2210	1624	1127	veh/hour/lane
$\rho_c(\gamma)$	34	90.2	62.6	veh/mile/lane
$\rho_m(\gamma)$	239	239	239	veh/mile/lane

Table 5.1: Traffic model parameters

procedure described in [107], which is the highest flow value observed from the data. The maximum speed $v_{\max}$ is set as 65 mph, which is also the free flow speed specified in CORSIM. The jam density is calibrated as 239 veh/mile/lane by using a least squares fit. The variable β controls the curve of the free flow regime. From the data shown in Figure 5.1, a linear relationship can be observed between traffic density and traffic flow. The variable β is set as 10,000 veh/mile/lane so that the density flow relationship is almost linear while still maintaining a strictly decreasing velocity function. The critical density is computed as 34 veh/mile/lane. These parameters (i.e., $v_{\max}$, ρ_c, ρ_m, and $q_{\max}$) are subject to change when an incident occurs. The values from the *Highway capacity manual* (HCM) and [108] are used to determine the parameters used in this work. The resulting traffic model parameters for the first order traffic incident model are summarized in Table 5.1. When the second order model is deployed, the upper bounds of the critical density and jam density (i.e., ρ_{c1} and ρ_{m1}) are set as the critical density and jam density calibrated for the first order traffic model (i.e., $w = 1$). The lower bounds of the critical density and jam density are set as 80 percent of the upper bounds (i.e., $w = 0$).

Other parameters used for the discretized traffic model and noise models within the estimation algorithms are summarized in Table 5.2. In this numerical implementation, all of the noise models are specified by a Gaussian distribution, however, other types of distributions are applicable since the particle filter is able to handle non–Gaussian noise.

Link length	4 miles
Number of cells	11
ΔT	20 seconds
Δx	0.36 miles
ω	$\mathcal{N}(\mathbf{0}, 5.0^2 \times I)$
ν_{density}	$\mathcal{N}(\mathbf{0}, 13.5^2 \times I)$
ν_{speed}	$\mathcal{N}(-4.0 \times \mathbf{1}, 4.8^2 \times I)$

Table 5.2: Setup for the macroscopic model and noise model used in the estimator, where $\mathbf{0}$ and $\mathbf{1}$ are vectors with all elements of zero and one respectively, and I is the identity matrix of the appropriate dimensions.

5.2 Assumptions for the model variable evolution

We make several assumptions on the evolution of the model variable. First, we assume there is a one percent probability for the occurrence of a traffic incident at next time step, provided the freeway does not have any incidents at the current time. If an incident occurs, it has an equal probability to occur anywhere between the two inductive loop detectors with three possible severities: one, two, or all lanes blocked. Second, if there is an incident on the freeway at the current time step, there is a 99% probability for the incident to remain in the next time step, and a 1% probability for the incident to be cleared. With these assumptions, the transition matrix $\bar{\Pi}$ can be constructed. Note a relatively high probability for the occurrence of a traffic incident is assumed. This is because in the multiple model particle filtering algorithm, the number of particles in each model is proportional to the transition probability for each model. Consequently, a relatively high transition probability into an incident is needed in order to get particles in each model if the sample sizes are to remain tractable. If we assume a lower probability of an incident, a much larger sample size may be needed.

5.3 Simulation description and error metrics

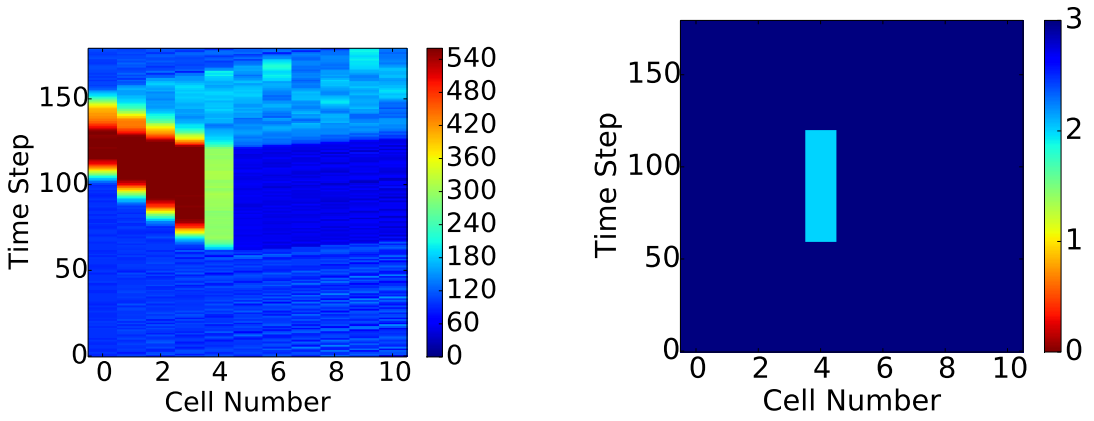
Two inductive loop detectors are assumed to be available in order to calibrate the fundamental diagram and they are located in cells one and nine. In the numerical simulations, the proposed algorithms are tested by assuming different penetration rates of GPS vehicles, and different boundary conditions. For the proposed multiple model particle filtering and smoothing algorithms, the sample size is set as $M = 2,500$. For the proposed IMM EnKF and EMMPF, the sample size is set as 100.

The initial condition in all cells are assumed to follow a normal distribution, where the mean is the average of the density measurements from the inductive loop detectors located near both ends of the freeway, and the standard deviation is five percent of the mean. In CORSIM, the simulation starts after a warm–up period, so the initial density values are nonzero.

The estimation accuracy of the state vector x^n and the model variable γ^n is quantitatively evaluated by computing the average error as follows:

$$\begin{aligned} e_x &= \frac{1}{(i_{\max}+1)(n_{\max}+1)} \sum_{i=0}^{i_{\max}} \sum_{n=1}^{n_{\max}} |\hat{\rho}_i^n - \bar{\rho}_i^n|, \\ e_\gamma &= \frac{1}{(i_{\max}+1)(n_{\max}+1)} \sum_{i=0}^{i_{\max}} \sum_{n=1}^{n_{\max}} |\hat{\gamma}_i^n - \bar{\gamma}_i^n|, \end{aligned} \tag{5.5}$$

where $\hat{\rho}_i^n$ is the estimated density (mean of the posterior distribution), $\bar{\rho}_i^n$ is the true density, $\hat{\gamma}_i^n$ is the estimated model variable (maximum a posterior estimate), and $\bar{\gamma}_i^n$ is the true model variable at each time n and location i.



(a) Density (veh/mile)

(b) Model variable (lanes open)

Figure 5.2: True evolution of the traffic density and the model variable.

5.4 MMPF and MMPS estimation results with different GPS penetration rates

In the first set of simulations the influence of various penetration rates of GPS data on the estimation accuracy are explored while the inflow in CORSIM is specified as 6,000 veh/hour to maintain a high inflow and large congestion resulting from an incident. The density and the model variable in the time and space domain for the true traffic conditions obtained from CORSIM are shown in Figures 5.2a and 5.2b. After the incident occurs, a congestion wave is generated and the congestion propagates to the boundary shortly after time step 100. After the incident is removed, the congestion begins to dissipate before finally clearing just before time step 150.

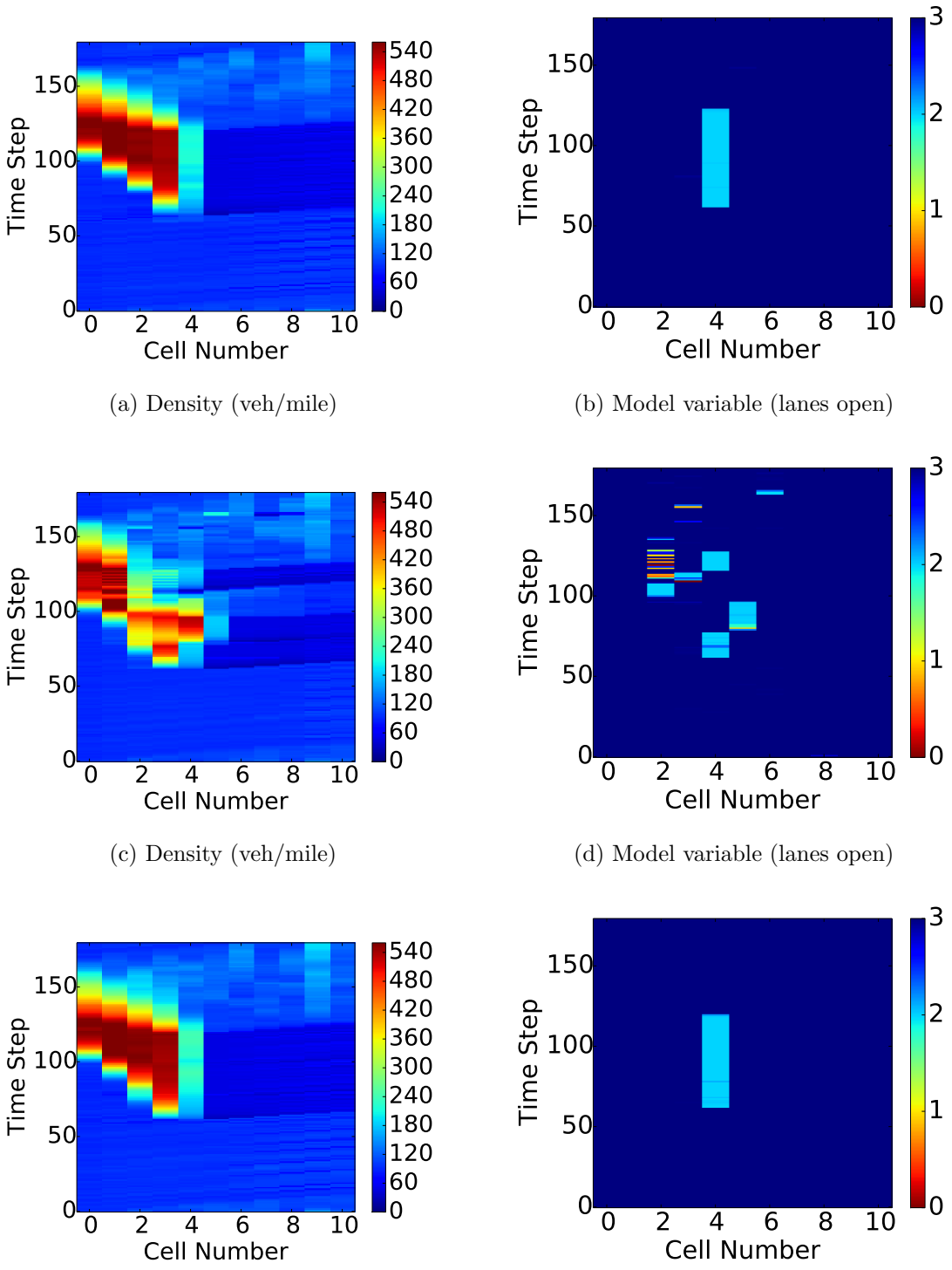
The first order traffic flow model (3.9) is assumed in the estimator using the calibrated fundamental diagram described in Section 5.1. The left boundary condition assumed in the estimator is $5,900 + \mathcal{N}(0, 150^2)$. The right boundary condition is in free flow and therefore does not influence the one step model prediction. The MMPF algorithm is first tested by assuming penetration rates of four percent and one percent, and the estimation results without smoothing are shown in Figures 5.3a, 5.3b, 5.3c, and 5.3d. As the result shows, when

the penetration rate is four percent, the algorithm is able to correctly estimate the model variable. When the penetration rate is decreased to one percent, the estimation accuracy for both the traffic state and the model variable decreases, due to the reasons discussed in Section 4.3.2.

Next, the multiple model particle smoother is tested when the GPS penetration rate is one percent and the results are shown in Figures 5.3e and 5.3f. The ΔS for this simulation is set as three. Compared to Figures 5.3c and 5.3d, which have the same penetration rate (without smoothing), the accuracy of both the traffic state and model variable estimates improves significantly with smoothing. Thus, when the penetration rate of probe vehicles is low, smoothing might be a meaningful way to improve estimation accuracy, without the need for additional probe data. The increased accuracy comes at the cost of a lag in the estimate. In this experiment, the three time step lag generates a one minute delay in producing the state estimate.

To provide a more comprehensive analysis, the MMPF and MMPS are tested by assuming four penetration rates. For each penetration rate, five tests are conducted with the MMPF and the MMPS with the smoothing window ΔS set as three. The results are summarized in Figure 5.4, where the reported error is the average over the five tests. The results show the error of both the state and model variable estimates becomes large when the penetration rate of GPS vehicles decreases, and smoothing is able to improve the estimation accuracy.

To give a more detailed view of the performance of the multiple model particle filter, Figure 5.5 shows the number of distinct particles in the posterior distribution of the multiple model particle filter for each time step. The figure corresponds to an experiment when the inflow is 6,000 veh/hour and the penetration rate is four percent. From the figure, we can see that there are approximately 1,500 number of particles that have relatively high weights from time step zero to time step 60. Then, the number of distinct particles drops significantly at the moment when the incident occurs. This occurs because the relatively low transition probability from the non-incident model to an incident model (i.e., one percent).



(a) Density (veh/mile)

(b) Model variable (lanes open)

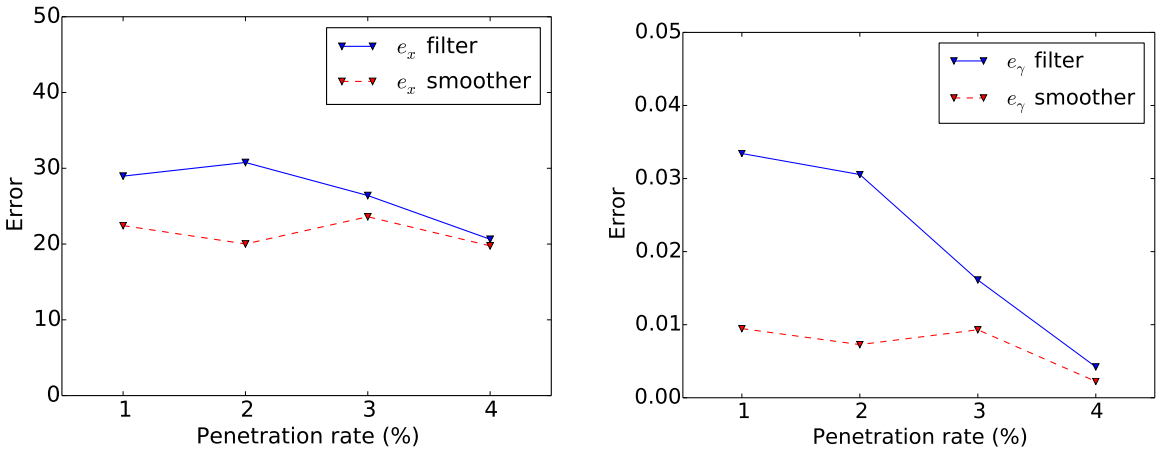
(c) Density (veh/mile)

(d) Model variable (lanes open)

(e) Density (veh/mile)

(f) Model variable (lanes open)

Figure 5.3: Estimate of the multiple model particle filter, penetration rate of four percent (first row) and one percent (second row). Estimate of the multiple model particle smoother, penetration rate of one percent and $\Delta S = 3$ (third row). The values of the traffic state (left) and model variable (right) estimate at each time and space domain are described by the color bar. The value shown is the mean of the posterior distribution.



(a) State error (veh/mile)

(b) Model error (lanes/cell)

Figure 5.4: Average error (five tests) for density (left) and model variable (right) estimates under different penetration rates. The smoothing window ΔS for these simulations is set as three.

Moreover, the one percent of incident samples have an equal probability to occur at any of the cell locations and for any severity. As a consequence, a very small number of particles are actually generated with the correct model variable at the moment the incident occurs, but those particles generated by the correct incident model are found to be consistent with the measurements and so it receives much higher weight (and eventually more particles in the resampling step). After the traffic model transitions to the incident model, the number of particles that are consistent with measurement will increase due to the high probability of staying in an incident mode, which can be observed from the figure (i.e., time steps 60 to 120).

It is also observed from Figure 5.5 that the number of distinct particles decreases again from time steps 120 to 170. This is because when the traffic incident clears, the traffic evolution dynamics in CORSIM are not consistent with any models defined in the traffic model with incidents. As shown in the true traffic density evolution (i.e., Figure 5.2a), the true traffic density can reach 240 veh/mile downstream of the incident location after the incident is cleared. However, from the macroscopic traffic flow theory, vehicles should be in free flow and therefore the one step ahead prediction with the model used in the estimator

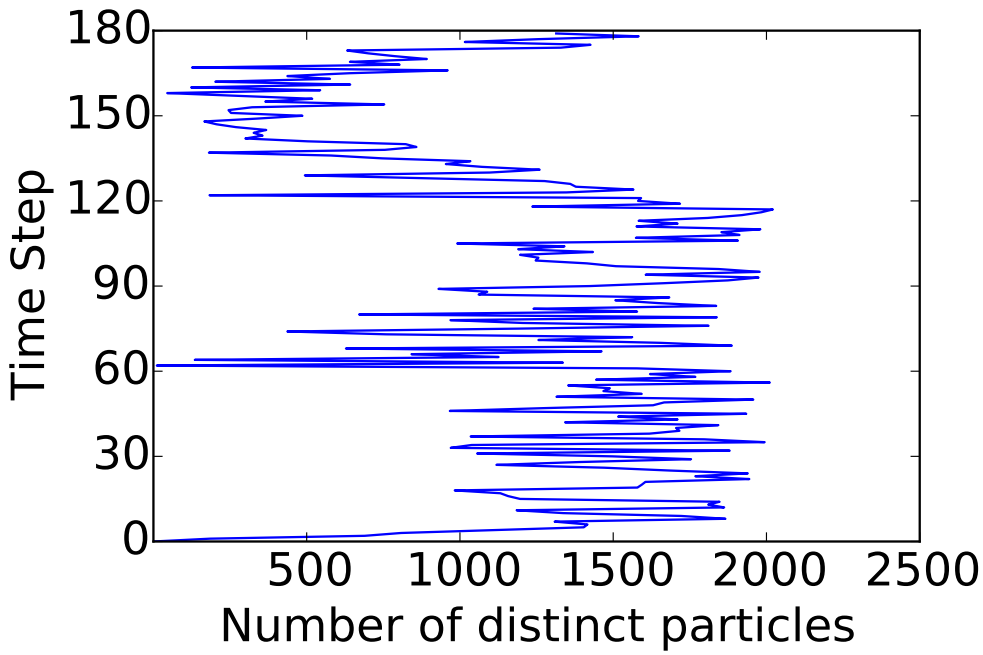
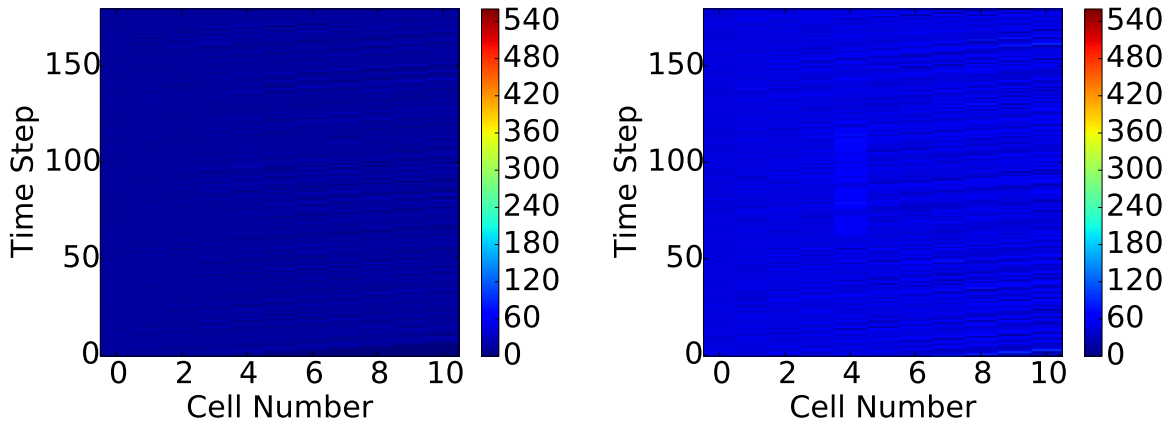


Figure 5.5: Number of different particles in the posterior distribution of the multiple model particle filter for each time step. The inflow is 6,000 veh/hour and the penetration rate is four percent.

has a larger model error. Regardless, it is shown in Figure 5.3a that the proposed MMPF is able to estimate the traffic state even though the number of matched samples decreases during this period, and the MMPF does not provide any false positive incident predictions. This lack of false positives occurs because even the predictions by the (correct) non–incident traffic model does not match well with the measurements, it has higher agreement compared to the predictions generated by incident models and therefore these samples retain larger weights.

Inflow (veh/h)	State estimation error (veh/mile)				Incident detection			
	CA	PF	IMM EnKF	MMPF	CA	PF	IMM EnKF	MMPF
1000	-	3.5	4.8	3.3	not detected	-	not detected	not detected
2000	-	4.7	6.0	4.2	not detected	-	not detected	not detected
3000	-	6.4	9.4	8.3	not detected	-	not detected	not detected
4000	-	19.8	14.6	11.3	not detected	-	detected 3.3 minutes late	detected 1.6 minutes late
5000	-	51.2	24.4	17.9	detected 26 minutes late	-	detected 3.0 minutes late	detected 3.0 minutes late
6000	-	55.1	19.5	19.7	detected 18 minutes late	-	detected 1.6 minutes late	detected 1.6 minutes late

Table 5.3: Comparison among CA, PF, IMM EnKF, and MMPF



(a) Density (veh/mile)

(b) Density (veh/mile)

Figure 5.6: True evolution of traffic density. (a) Inflow 1,000 veh/hour. (b) Inflow 3,000 veh/hour.

5.5 Comparison with the particle filter, the California algorithm and the IMM EnKF for different inflows

In this section, the performance of the MMPF and the IMM EnKF are tested with different boundary conditions and compared with a particle filter shown in Algorithm 5 applied to the scalar traffic flow model (3.9) and the California algorithm [14], which estimate the traffic state and incidents independently. The penetration rate is four percent for all experiments. The California algorithm shown in Algorithm 1 is implemented and the occupancy measurements from the inductive loop detectors at cell one and cell nine are used. The thresholds of the California algorithm T_1, T_2, and T_3 are calibrated and set to 0.33, 2.55, and 0.0003 respectively to maximize the accuracy of the algorithm.

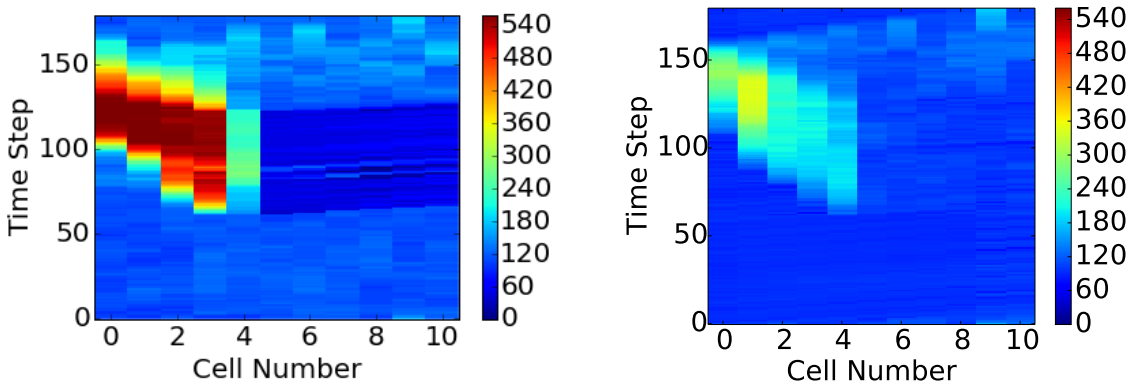
Table 5.3 shows the comparison between the algorithms for traffic estimation and incident detection when the inflow ranges from 1,000 veh/hour to 6,000 veh/hour. For the MMPF, an incident is reported if the most likely model in the model variable posterior distribution is an incident model for consecutive three time steps. For the IMM EnKF, an incident is reported if an incident model is selected for consecutive three time steps.

When one lane is blocked, the remaining capacity for the three–lane road is approximately 3,300 veh/hour in CORSIM. When the remaining lanes have enough capacity to accommodate all of the traffic (i.e., for the inflows between 1,000 to 3,300 veh/hour), the traffic incident does not generate significant congestion that can be detected by any of the algorithms. Figure 5.6 shows the true traffic evolution when the inflows are small and highlights the lack of congestion, even though in both simulations, there is an incident at cell four between time step 60 and 120.

As a result of the lack of congestion, the predicted evolution of traffic is accurate when the correct incident model is used, and also if the model incorrectly assumes all lanes are open. As a consequence, all traffic estimation algorithms (the PF, the IMM EnKF, and MMPF) have very low state estimation error as reported in Table 5.3. At the same time, all of the incident detection algorithms (the California algorithm, the IMM EnKF, and the MMPF) fail to detect the incident.

When the inflow exceeds the remaining capacity of the road (e.g., inflows of 4,000 veh/hour, 5,000 veh/hour, or 6,000 veh/hour), significant congestion will form after the occurrence of an incident. The MMPF and the IMM EnKF are able to detect the incident from the sensor data and switch to the incident model, while the PF continues to estimate the traffic assuming all lanes are open because the scalar traffic model used in the filter does not contain any incident dynamics. Consequently, the PF collapses and provides poor state estimates in comparison to the IMM EnKF and in comparison to the MMPF, which has the highest traffic state accuracy. We conclude the MMPF and IMM EnKF perform better than a particle filter in terms of traffic estimation under incidents resulting in congestion.

To understand the performance of the algorithms, Figure 5.7 shows the density estimation results by the IMM EnKF and the PF when the inflow is 6,000 veh/hour and the penetration rate is four percent. Compared to the estimates of the MMPF shown in Figure 5.3a, we see both the MMPF and the IMM EnKF are able to estimate the resulting congestion caused by the incident. In contrast, the particle filter cannot provide a good traffic state estimate



(a) Density (veh/mile)

(b) Density (veh/mile)

Figure 5.7: Density estimates of the IMM EnKF and PF, inflow = 6,000 veh/hour, penetration rate four percent.

in the presence of a traffic incident.

Figure 5.8 shows the number of distinct particles in the posterior distribution for each time step of the particle filter (Algorithm 5) for the same simulation. We can see that the number of distinct particles drops significantly after time step 60 when the traffic incident occurs. Since the particle filter always estimates the traffic state assuming all lanes are open even when there is an incident, none of the particles predict the congestion caused by the incident. When compared to Figure 5.5 which shows the number of distinct particles in the multiple model particle filter for the same simulation, it is clear that embedding the traffic model with incident dynamics is critical to prevent the filter from collapsing.

Note that the estimation result of the particle filter in Figure 5.7b shows that the particle filter still identifies some congestion in the upstream area of the incident even though the severity of the congestion is not accurate. The particle filter is able to partially track the congestion is due to the random model nose added to the traffic flow model at each time step. When the model noise adds congestion in the appropriate cells, it will match the measurements better than the particles with congestion in the wrong cells or no congestion at all. As a result, the particles will receive higher weights (and will be replicated in the resampling step).

As indicated, the state estimation accuracy of the MMPF is generally higher compared to the IMM EnKF. This is because the IMM EnKF proposed in [99] is a model conditioned filter and the posterior distribution contains samples propagated only by the most likely model. When the traffic model does not perfectly represent the true traffic dynamics, a posterior distribution generated by multiple models may approximate the true traffic state better. However, compared to the MMPF, the computation time for the IMM EnKF is shorter. It takes the IMM EnKF eight minutes to estimate an hour of traffic for the roadway modeled in CORSIM, while the computation time for the MMPF is 14 minutes.

The results for estimating the presence of incidents is analyzed next. The California algorithm is not able to detect the incident when the inflow is 4,000 veh/hour or lower, even though the IMM EnKF and the MMPF detect the incidents when the flow is as low as 4,000 veh/hour. At higher flows (i.e., 5,000 and 6,000 veh/hour), the California algorithm detects the presence of an incident at time steps 138 and 114. Figure 5.9 shows the true evolution of traffic density when the inflows are 4,000 and 5,000 veh/hour. From the true density evolution shown in Figure 5.9a, it is clear the California algorithm cannot detect the incident when the inflow is 4,000 veh/hour since the incident generated congestion does not propagate to the upstream sensor at cell one, where the anomalous congestion can be used to detect the presence of an incident. When the inflow increased to 5,000 veh/hour, the California algorithm reports the incident after the congestion propagates to the sensor at cell one. Note that the detection time by the California algorithm is faster at higher flows, because the resulting shockwave generated by the incident travels more quickly to the boundary when the inflow rate is higher.

In comparison, both the MMPF and the IMM EnKF are able to detect the traffic incident nearly as soon as it occurs. The algorithms can detect the incident because they are able to leverage the mobile measurements from GPS equipped vehicles. This also illustrates one of the benefits of mobile sensing, namely that the sensors can move towards the congestion to detect it earlier than if the sensor is stationary and must wait for the congestion wave to

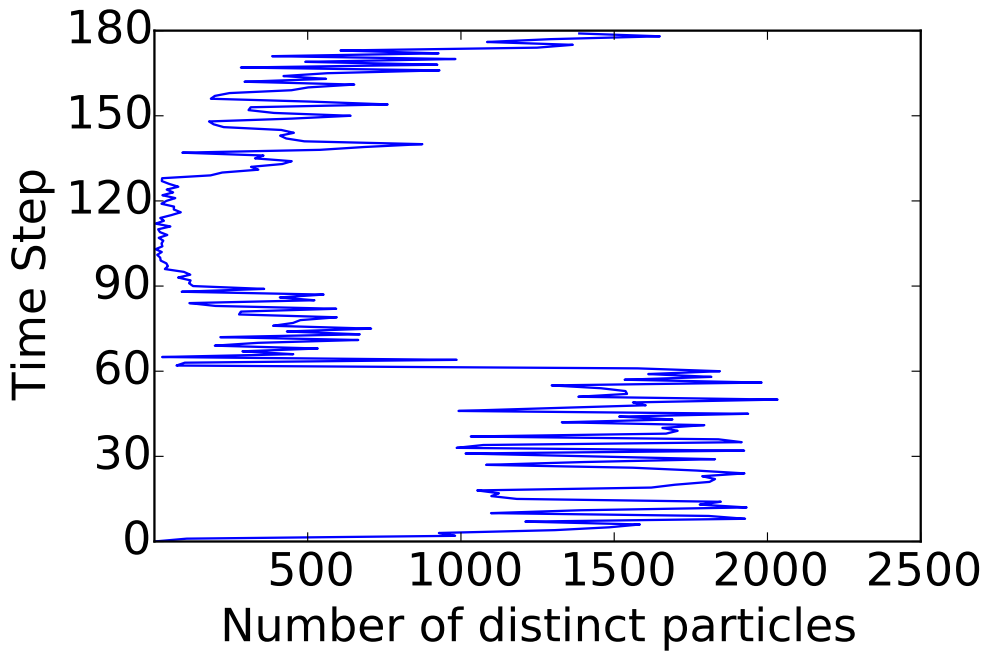
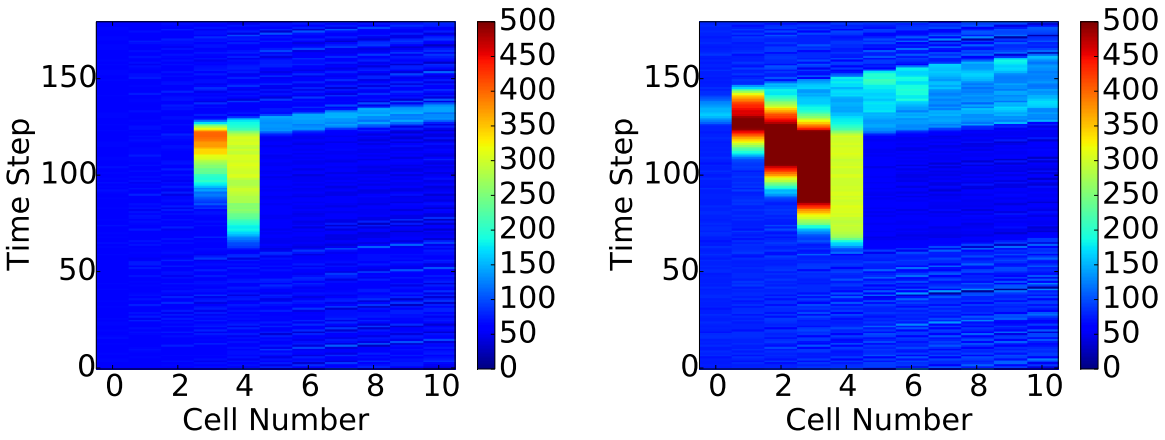


Figure 5.8: This figure shows the number of distinct particles in the posterior distribution of the particle filter applied to the CTM for each time step after resampling. The inflow is 6,000 veh/hour and the penetration rate is four percent.

arrive. If the fixed sensors are densely placed on the road network, the proposed algorithms may have similar performance with the California algorithm, but these dense fixed sensor networks are expensive to install and maintain and consequently are currently available only in dense urban areas in the US.

5.6 EMMPF estimation results on the first and second order traffic flow models

In this set of experiments, the EMMPF is implemented on both the first order traffic model and the second order traffic model to jointly estimate the traffic state and detect incidents. The algorithm is tested with the incident data from CORSIM for various inflows ranging from 1,000 veh/hour to 6,000 veh/hour with a fixed penetration rate of GPS equipped probe vehicles of four percent. The estimation results are shown in Table 5.4.



(a) Density (veh/mile)

(b) Density (veh/mile)

Figure 5.9: True evolution of traffic density. (a) Inflow 4,000 veh/hour. (b) Inflow 5,000 veh/hour.

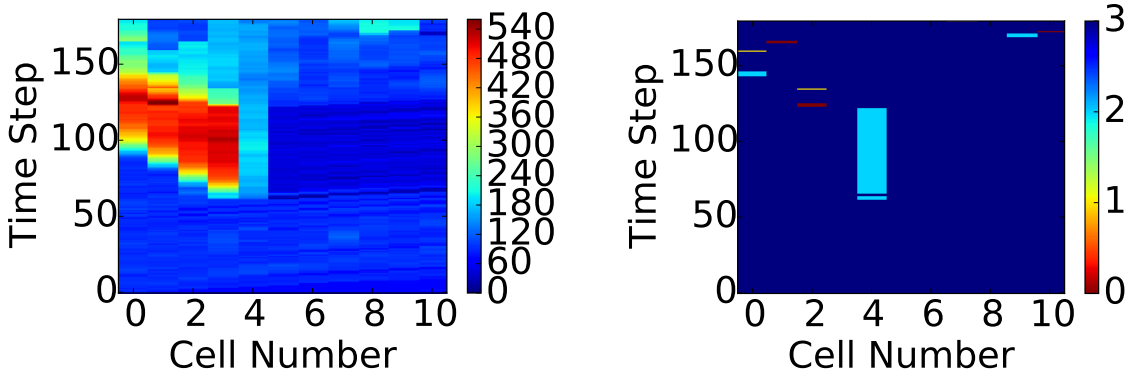
Inflow (veh/h)	State estimation error (veh/mile)		Incident detection	
	1st order model	2nd order model	1st order model	2nd order model
1000	4.4	4.2	not detected	not detected
2000	6.0	6.1	not detected	not detected
3000	8.6	8.4	not detected	not detected
4000	54.8	56.2	detected 3.0 minutes late	detected 3.0 minutes late
5000	27.8	33.1	detected 1.6 minutes late	detected 1.6 minutes late
6000	33.6	32.9	detected 1.6 minutes late	detected 1.6 minutes late

Table 5.4: Estimation results of the EMMPF on the first order model and the second order model.

Compared to the MMPF and the IMM EnKF (see Table 5.3), the EMMPF has a higher state estimation error. As shown in Figure 5.10, the EMMPF estimates the incident is cleared for a few time steps after the incident occurs, and it has several false incident predictions after the incident is cleared. Moreover, when the inflow is 4,000 veh/hour, the EMMPF estimates an incorrect incident severity when implemented on both the first and second order traffic models. As a result, the reduction of computation time is at a cost of estimation accuracy, even though the main features of the incident and resulting congestion were captured by the filter.

From Table 5.4, it is also noted that the EMMPF has similar performance when it is implemented with the first order traffic model and the second order traffic model, which indicates there may not be a significant benefit to implement the second order traffic flow model for incident detection problems when only the density estimate is desired. The high agreement between the two modeling approaches is partially explained by the fact that both the left and right boundary conditions are in free flow. In this case, the properties of vehicles cannot be identified from the flow, density, and speed data, and consequently the boundary conditions for the second order model may have larger noise than the density inputs. Second, there is no direct mechanism to control the corresponding macroscopic property variable in the microscopic simulation environment. Thus, it may be the case that the underlying traffic dynamics assumed in CORSIM for the true state in this experiment do not exhibit the dynamics that are able to be captured by the higher order traffic flow model. In this case, the best second order model may be the one with a constant property of $w = 1$, in which case the model collapses back to the first order model.

The main advantage of the efficient multiple model particle filter is clear when examining the runtime of each of the filtering algorithms. All models and estimation algorithms are implemented in Python and run on a 3.0GHz Intel Core i7 Macbook Pro. Each one hour numerical experiment can be run in about 14 minutes when the MMPF is applied. When the filter is extended to a smoother, and the smoothing window is set to three steps, the



(a) Density (veh/mile)

(b) Density (veh/mile)

Figure 5.10: Density estimates of the EMMPF on the second order traffic model, inflow = 6,000 veh/hour, penetration rate of four percent.

runtime increases to approximately 36 minutes, which highlights another cost of the smoother beyond the lag in the estimate. When the IMM EnKF is used for estimation, the experiment completes in about 8 minutes. Finally, when the EMMPF is used, the runtime drops to approximately 25 seconds to complete the simulation, which is significantly faster compared to the MMPF, MMPS, or the IMM EnKF. On the experiments on larger networks using field data, the cost of all the algorithms except the EMMPF become computationally too expensive to run in real time.

5.7 Sensitivity analysis on the calibrated model parameters

In this section, a sensitivity analysis is performed to study how the estimation accuracy is affected by the model parameters. The EMMPF is implemented with the first order traffic models for traffic state estimation and incident detection. The simulations are performed with inflow of 6000 vehicles per hour. We investigate how the critical density, jam density, model noise, and incident transition probability impact the estimation accuracy. For the critical density, jam density and model noise, the calibrated values are perturbed by plus and minus 20 percent, and five values equally spaced in the plus and minus 20 percent interval

are used for traffic estimation. For the incident transition probability, the parameter range is specified as 0.95 to 0.9999. Similarly, five values equally spaced in this range are used. During the simulation, the parameters are perturbed one at a time, as a result, a total of 20 simulations (i.e., four parameters, five values for each parameter) are performed for the sensitivity analysis.

To evaluate the estimation accuracy, a *receiver operating characteristic* (ROC) curve is used to show the fraction of true positive and false positive for incident detection. The results are shown in Figure 5.11 and Figure 5.12. As Figure 5.11 shows, overall, the points are located at the upper left of the figure, which indicates the algorithm has a good performance (i.e., high true positive and low false positive). Figure 5.12 provides a closer view of the ROC results, we can see the estimation accuracy is less sensitive to the critical density and model noise. Actually, better estimation results can be achieved with some perturbation of these parameters. The estimation algorithm is more sensitive to the transition probability and jam density. This makes sense because the transition probability directly determines the probability of each model, and consequently impacts when a model switch will occur. The jam density controls the fundamental diagram curve in the congested regime. In the presence of an incident, the traffic is in congestion, and it makes sense that the traffic dynamic is sensitive to the jam density variable.

5.8 Summary of the main findings from experiments in CORSIM

In this chapter, we implemented the proposed MMPF, MMPS, IMM EnKF, and EMMPF with traffic and incident data generated by CORSIM. The main findings are summarized as follows.

The MMPF, MMPS, IMM EnKF, and EMMPF are capable of jointly estimating traffic state and detecting incidents. Compared to the MMPF, the MMPS can improve the estimation accuracy when data limited, although this improvement comes at the cost of a lag in the

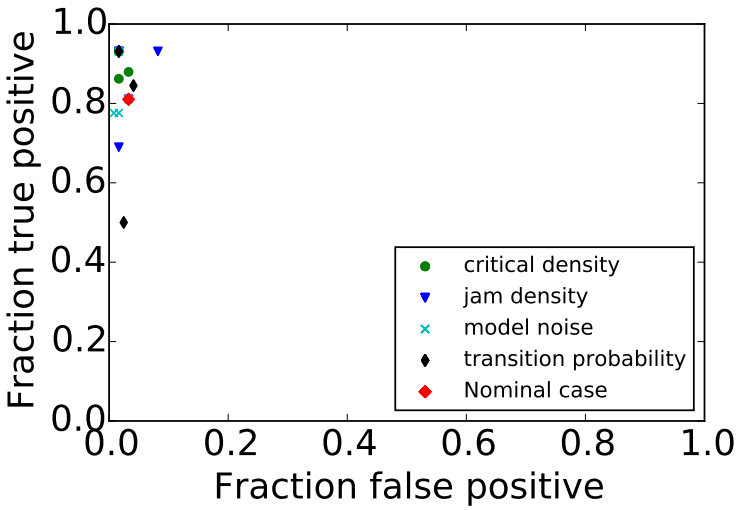
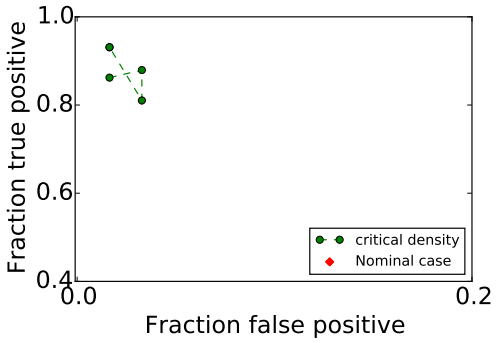


Figure 5.11: Different markers represent the results for different types of parameters. Some of the markers overlap.

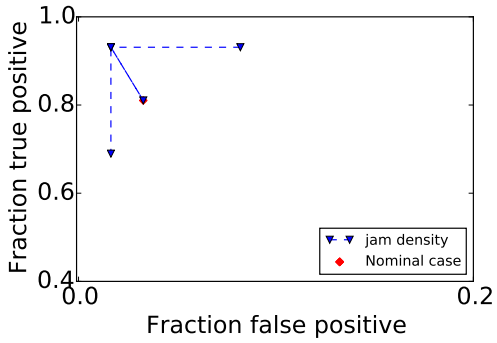
estimates equal to the smoothing window length, and an increase in the runtime that also grows with the smoothing window. The estimation accuracy of the MMPF is the highest when compared to the IMM EnKF and EMMPF, but it is also the most computationally costly filtering algorithm.

On the other end of the runtime spectrum, the EMMPF requires significantly less computation time compared to the other algorithms. This is achieved by using a single particle evolved through each model to determine the correct model at the model selection step of the algorithm. This results in a slightly higher false positive incident prediction rate which consequently also reduces the estimation accuracy. However, the EMMPF is more suitable for field implementation when the network sizes grow larger.

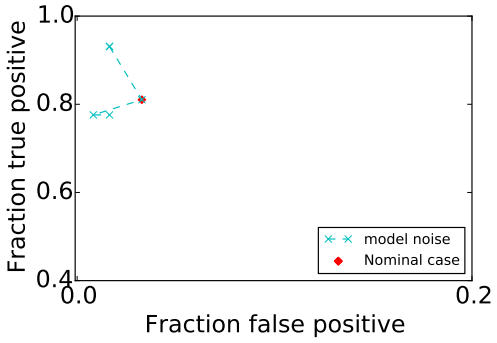
The proposed MMPF and IMM EnKF are also compared with a particle filter and the California incident detection algorithm. It is found jointly estimating traffic state and incidents in one algorithm outperforms both a particle filter for estimating the traffic state and the California algorithm for detecting incidents, each of which work independently from the other algorithm.



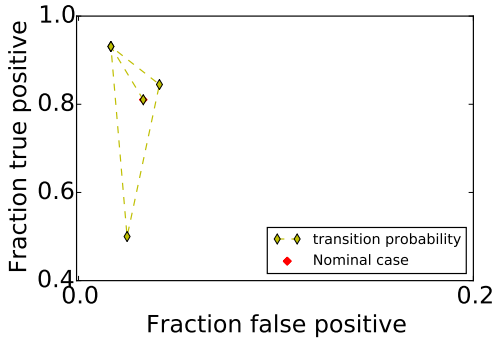
(a) critical density



(b) jam density



(c) model noise



(d) transition probability

Figure 5.12: The figures above show the ROC curves for the sensitivity analysis of the four model parameters.

Chapter 6

Experiments using field data obtained from the Mobile Century experiment

In this section, the proposed EMMPF is tested with field data collected on I–880 in California during the Mobile Century experiment [18]. Both first and second order traffic flow models for networks are used as the model predictors. A brief overview of the network setup and the experimental data is given. Then, the model calibration procedure and the experiment setup are described. Finally, the implementation results of the IMMPF on both first and second order models are presented and discussed.

Note that the MMPF and IMM EnKF were also implemented with the field data to further compare the influence of the algorithm on the estimation results, however both algorithms are far too slow to complete even a single experiment for analysis. The EMMPF runs in real time and it is able to jointly estimate the traffic state and detect an incident that occurred when the field data was collected.

6.1 Implementation overview

The proposed EMMPF algorithm is tested on a segment of I–880 in California. Density measurements from inductive loops and speed measurements from GPS equipped vehicles

are used as measurements in the traffic estimation algorithm, where density measurements are obtained from the *Caltrans Performance Measurement System* (PeMS) and the speed data is collected from the GPS devices deployed during the Mobile Century experiment [18].

PeMS is a highway monitoring system and it collects and records 30 second loop detector data for all of California. Users can get access to the historical and real time loop detector data by visiting their website [109]. The data provides a comprehensive view of the highway performance, and enables the development of traffic control, traffic estimation, and policy strategies to improve traffic safety and operations [110]. In the field implementation of the traffic estimation and incident detection algorithms developed in this dissertation, the 30 second interval loop detector data from PeMS is used to provide density measurements for the estimation algorithm.

The speed measurements used in the field implementation are collected through the Mobile Century experiment [18]. The Mobile Century experiment collected speed data from approximately 77 GPS equipped vehicles which run continuously on the segment of highway for 6 hours (10am to 4pm) on February 8, 2008. The highway segment used in the experiment is shown in Figure 6.1 [18]. While data is available on both the north and southbound directions, the experiments performed in this dissertation focus on the traffic in the northbound direction due to the more interesting traffic dynamics, including the presence of an incident. The penetration rate of GPS equipped vehicles is approximately two percent of the total traffic flow over the course of the data collection.

The geometry of the highway segment used in this dissertation is shown in Figure 6.2. The segment of highway is six miles long, from postmile 21.3 to postmile 27.3. The algorithm is tested with six hours of data from 10 am to 4 pm. A unique feature of the dataset is that it contains an incident that occurred around postmile 26.4 from 10:27 am to 11:00 am, which serves as the benchmark incident in the experiment to be estimated by the algorithms. The incident was also recorded in the California Highway Patrol traffic incident feed which is also archived on PeMS and can be verified through their website [109].

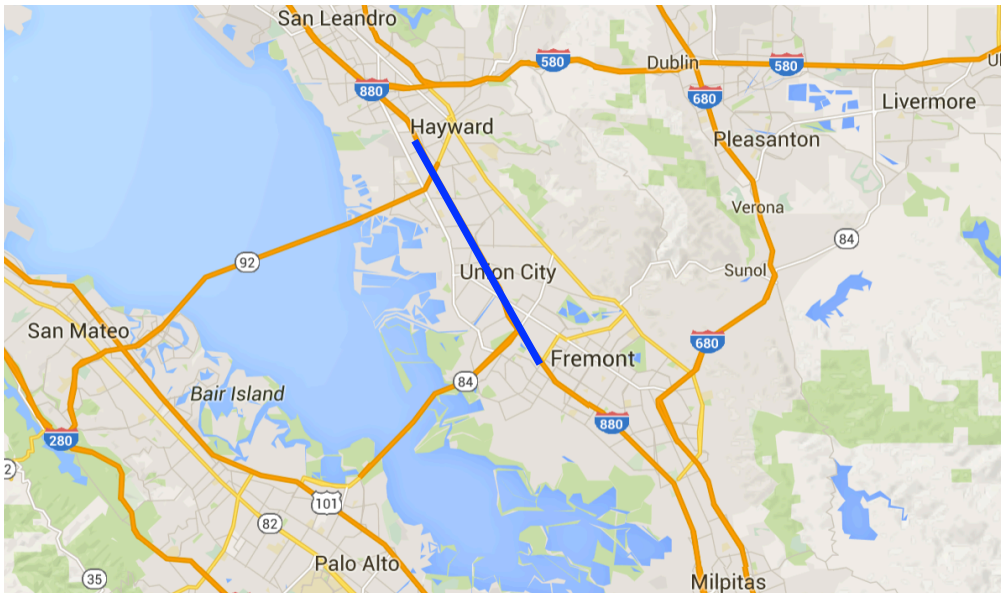


Figure 6.1: Stretch of highway I–880 in Califronia, used in the Mobile Century experiment. The blue line shows the segment of I–880 where the experiment is performed.

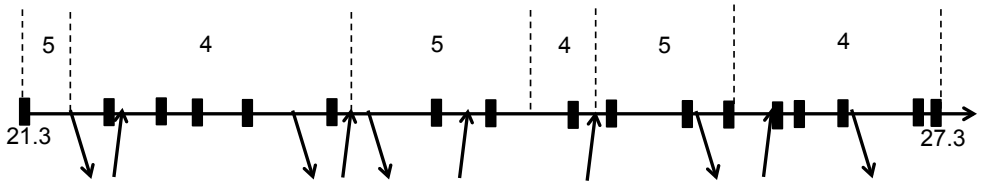


Figure 6.2: Geometry of the segment of I–880 in California used for the field implementation. The black squares in the figure indicate loop detectors, and the arrows denote on–ramps and off–ramps. The numbers above each segment indicate the number of lanes of the highway. The two numbers at the left and right denote the starting and ending post mile of the highway segment. The traffic moves from the left to the right as indicated by the arrow.

Link length	6 miles
Simulation length	6 hours
ΔT	5 seconds
Δx	0.1 miles
Number of cells	60
Number of time steps	4,320

Table 6.1: Setup for model discretization

6.2 Model calibration

For the field implementation, there are five groups of parameters that need to be determined: the discretization of the traffic model, the parameters (e.g., maximum flow, maximum speed, jam density) that determine the shape of the fundamental diagram, the transition probability between incident models and non–incident models, the model noise and measurement noise, and the boundary conditions for the entrance and exit ramps. The calibration is more complicated than the CORSIM implementation since the underlying traffic is more complex and contains merging and diverging traffic, heavy congestion, and real driver behaviors. Moreover, determination of the ramp flows is significantly more challenging because there are no sensors at the ramps to create good historical estimates of the ramp flows. In this section, we describe how these model parameters are determined.

The discretization of the traffic flow models is summarized in Table 6.1. The highway is discretized into 60 cells, and each cell is 0.1 mile in length, so that the ramps are located close to the boundaries of cells. The total experiment simulation length is six hours. The time step ΔT is set as five seconds so that the CFL condition [77] $v_{\max}\frac{\Delta T}{\Delta x} \leq 1$ is satisfied. As a result, there are total 60 cells and 4,320 time steps in the discretization.

The velocity functions (3.13) and (3.29) described in Chapter 3 are used for the first

order and second order traffic incident model. The fundamental diagram related parameters for the non–incident and incident scenarios are determined as follows. For the non–incident scenario, the density–flow data collected from inductive loops are used to calibrate the parameters. In this section, the parameter calibration is performed on the first order traffic model for the non–incident scenario. On the highway, the maximum speed is calibrated using a least squares fit using the density–flow data collected from inductive loops under free flow conditions. The critical density ρ_c (veh/mile/lane) and the parameter β (veh/mile/lane) is jointly determined so that the maximum flow is close to the observed maximum flow from the data [107]. The jam density ρ_m (veh/mile/lane) as well as the fundamental diagram related parameters for ramps are manually calibrated. The calibration is performed for the first order traffic model (equivalently when $w = 1$ in the second order model).

For the second order traffic model, the upper bounds of the critical density and jam density (i.e., $\rho_{c1}(\gamma)$ and $\rho_{m1}(\gamma)$) are set as the critical density and jam density calibrated for the first order traffic model. The lower bounds of the critical density and jam density are set as 80 percent of the upper bounds, and the resulting parameters are summarized in Table 6.2.

For the incident scenarios, the calibration of either model is difficult since incidents are rare events and density–flow data for different incident severities is limited. In this work, the parameters for incident scenarios are determined using the HCM and a previous study [108], where the HCM values for the capacity loss are used to model the fundamental diagrams on the multiple lane highways when different numbers of lanes are blocked (shown in Figure 6.3), and the work [108] is used to model the free flow speed under incidents [107]. It is likely that the parameters from the references do not exactly correspond to the incident dynamics for this specific segment of highway, however, later we show our proposed EMMPF is able to estimate traffic state and detect incident even when the incident related parameters are not calibrated from the field.

The assumptions associated with the transition probability matrix are described as fol-

	First order traffic model				Second order traffic model					
	$v_{\max}$	ρ_c	ρ_m	β	$v_{\max}$	ρ_{c1}	ρ_{c2}	ρ_{m1}	ρ_{m2}	β
	mph	vpmpl	vpmpl	vpmpl	mph	vpmpl	vpmpl	vpmpl	vpmpl	vpmpl
Highway	70	24	130	10,000	70	24	19	130	104	10,000
Ramp	40	40	110	10,000	40	40	32	110	88	10,000

Table 6.2: Fundamental diagram related parameters

lows. We allow four incident severities: one lane blocked, two lanes blocked, three lanes blocked, and four lanes blocked. We assume there is at most one incident at a time, which results in a total 241 possible system models including the non-incident model. We also assume that if there is no incident at the current time step, there is 0.1×10^{-4} probability to have an incident at the next time step, and the probability of each possible incident model to occur is equal. If there is an incident at the current time step, then there is a 0.1×10^{-4} probability for the incident to be cleared for the next time step, otherwise, the incident will remain. In the field implementation, we give a smaller but more realistic incident probability compared to the value (one percent probability to have an incident) for the numerical implementation with CORSIM. This is possible because the required number of samples in the filter and consequently the computation time of the EMMPF is not related to the transition probability. In contrast, the MMPF assigns particles to each model proportionally to incident probabilities. When the incident model probability is very small, the MMPF will require a very large sample size in order to generate particles for incident models, and may not be able to run in real time.

As was mentioned previously, there are two types of sensors that are available to be used in the estimation algorithm, namely the inductive loops which collect traffic occupancy and traffic flow data, and the GPS equipped vehicles that provide vehicle speed measurements along their trajectories. The inductive loops collect data every 30 seconds and as a result, the

algorithm has density measurements (converted from the occupancy measurements) every six time steps. Both the model noise and the measurement noises are assumed to be additive and follow a normal distribution. The noise models are calibrated and summarized as follows: the model noise ω follows $\mathcal{N}(0, 20^2 \times I)$, the density measurement noise follows $\mathcal{N}(0, 40^2 \times I)$ and the speed measurement noise follows $\mathcal{N}(0, 20^2 \times I)$, where I is the appropriately sized identity matrix. Here, the unit of the density measurement noise is in vehicles per mile.

The boundary conditions for the main freeway can be estimated from historical data from the sensors located near both ends of the highway segment. When the first order traffic flow is used, the boundary condition is the traffic density, and when the second order traffic model is deployed, knowledge of the traffic density and the driver property are required on the boundaries. The traffic density and driver property w can be calculated from traffic density and flow data. Since no sensors are located on the ramps, it is harder to generate reasonable estimates of the boundary flows. Instead the traffic density ramp boundary conditions are manually calibrated. When the second order traffic model is used, the diver property parameter w is set as one. In this case, the traffic model on the ramps reduces to a first order traffic model.

Ideally, the model parameters should be calibrated following some standard calibration procedure. However, there is no existing calibration procedure that has been developed for the second order traffic model (3.27) and the fundamental diagram deployed in this dissertation. The calibration procedures for transition probabilities, incident parameters and boundary conditions for second order models also do not exist. As a result, we manually calibrate a subset of these parameters as a proof of concept to show the proposed algorithms have the potential to work well with field data, and leave the development of automatic calibration procedures for future work.

Number of Freeway Lanes by Direction	Shoulder Disablement	Shoulder Accident	One Lane Blocked	Two Lanes Blocked	Three Lanes Blocked
2	0.95	0.81	0.35	0.00	N/A
3	0.99	0.83	0.49	0.17	0.00
4	0.99	0.85	0.58	0.25	0.13
5	0.99	0.87	0.65	0.40	0.20
6	0.99	0.89	0.71	0.50	0.26
7	0.99	0.91	0.75	0.57	0.36
8	0.99	0.93	0.78	0.63	0.41

Table 6.3: Proportion of freeway segment capacity available under incident conditions. Source: Highway Capacity Manual 2000 [1]

6.3 Experiment description

The proposed EMMPF is tested with both the first order traffic flow model and the second order traffic flow model for both traffic estimation and incident detection. At the initial time step, the prior distribution of the traffic density is assumed to follow a normal distribution, where the mean is set as 90 veh/mile and the standard deviation is five percent of the mean. When the second order traffic flow model is used, the prior distribution of property values w is assumed to follow a uniform distribution $w \sim \mathcal{U}(0.0, 1.0)$.

Because the ground truth of the traffic evolution is not known, the measurements from the inductive loops and GPS equipped vehicles are used to provide a noisy and incomplete view of the traffic evolution, shown in Figure 6.3a and Figure 6.3b. The red area early in the day is the congestion caused by the incident, while the high density and slow speed in other regions correspond to non–incident related congestion.

To evaluate the performance of the proposed EMMPF algorithm, we select three loop detectors that do not send measurements into the traffic estimation algorithm. Later the estimated traffic state at these three locations are compared with the measurements from the sensors to evaluate the performance of the estimation algorithms. We choose one sensor in the upstream, one sensor in the downstream, and one sensor in the middle of the domain. In particular, the second sensor, the twelfth sensor, and the sixteenth sensor (from left to right in Figure 6.2) are removed during the traffic estimation, and are subsequently used to evaluate the performance of the proposed algorithms. During the field implementation, the number of particles M is set as 100 particles.

6.4 EMMPF estimation results

The proposed EMMPF is implemented with both the first and second order traffic flow models and the estimation results are shown in Figure 6.4. The results show that the

Algorithm	Traffic flow model	Density error (veh/mile)
EMMPF	1st order model	32.1
	2nd order model	31.7
PF	1st order model	33.3
	2nd order model	31.8

Table 6.4: Density error comparison between the first and second order traffic models and between different estimation algorithms.

proposed EMMPF is capable of detecting the incident and providing a good traffic state estimate. The traffic density error is computed following the equation (5.5). Here, the errors are computed and averaged only at the locations of the three sensors and for the time steps for which measurements are available, and the true state is taken to be the value recorded by the sensor. The results are summarized in Table 6.4.

It is found the performance of the first order traffic model and the second order traffic model are very close for traffic estimation and incident detection, which is similar to the findings in the CORSIM experiments. Causes of the high agreement may be due to the lack of knowledge of the property w at the boundaries which as a consequence need to be calibrated. In this experiment all vehicles from ramps are assumed to have the same property and behave the same as the first order traffic model, although later this is perturbed through the additive noise model on the evolution equation for the total property. Moreover, if the model noise is large (for example, if one models large uncertainties of the boundary value of the property variable), then the model prediction will be more heavily influenced by the noise model and less by the dynamical system used to propagate the state forward.

In this chapter, the algorithms are implemented in Python and run on a 3.0 GHz Intel Core i7 Macbook Pro. The six hour experiment can be run in about four hours and 20 minutes. Thus, the proposed EMMPF is suitable for real implementation. In comparison, the MMPF [100] requires at least 2,410,000 particles in order to expect at least one particle

for all possible models. To update all 2,410,000 particles to the next time step alone takes approximately two hours to complete. When the IMM EnKF [99] is used, it requires 24,100 samples to run the EnKF with 100 samples in each model. It takes about 70 seconds for the IMM EnKF to complete one time step (5 seconds) prediction. While this is a large improvement over the MMPF, the IMM EnKF is still an order of magnitude too slow to be implemented in real time.

6.5 Comparison with the particle filter

The particle filter presented in Algorithm 5 is deployed to estimate the traffic state on the same segment of highway. The simulation results are shown in Figure 6.5 and the state errors are shown in Table 6.4. From the results, we can see with either the first order traffic model or the second order traffic model, the standard PF is able to estimate the congestion in the incident scenario. This contradicts with the results in CORSIM simulation, where the particle filter is not able to track the traffic state in the presence of an incident.

The particle filter has good performance in the field implementation because of the following. First, compared to the numerical implementation in CORSIM, we give a much higher model noise to the traffic model in the field implementation. This is due in part to the fact that the boundary conditions at the ramps are unknown and therefore increases the model prediction error. As a result of the larger model noises, the particle filter can quickly track the congestion even through it always estimates traffic state assuming all lanes are open, and the distinction between the one step ahead predictions of the first order traffic flow model and the second order model are less apparent.

6.6 Comparison with the California algorithm

With either the first order model or the second order model, the EMMPF reports the incident at 10:29 am around postmile 26.4. The EMMPF initially estimates three lanes are blocked

by the incident. It should be noted that the EMMPF estimates the incident is cleared after a few minutes, but reports the incident again at the same location after some time. The EMMPF does not have any false positive incident reports during the simulation. One possible explanation of the brief incident clearance (false negative) predicted by the EMMPF is that the incident severity is initially overestimated. Later the algorithm may switch to the none–incident model to reduce the congestion, but switch back to a less severe incident model after some time steps. Since the parameters of the incident scenarios are not calibrated from the field, it is possible to estimate a wrong incident severity during estimation.

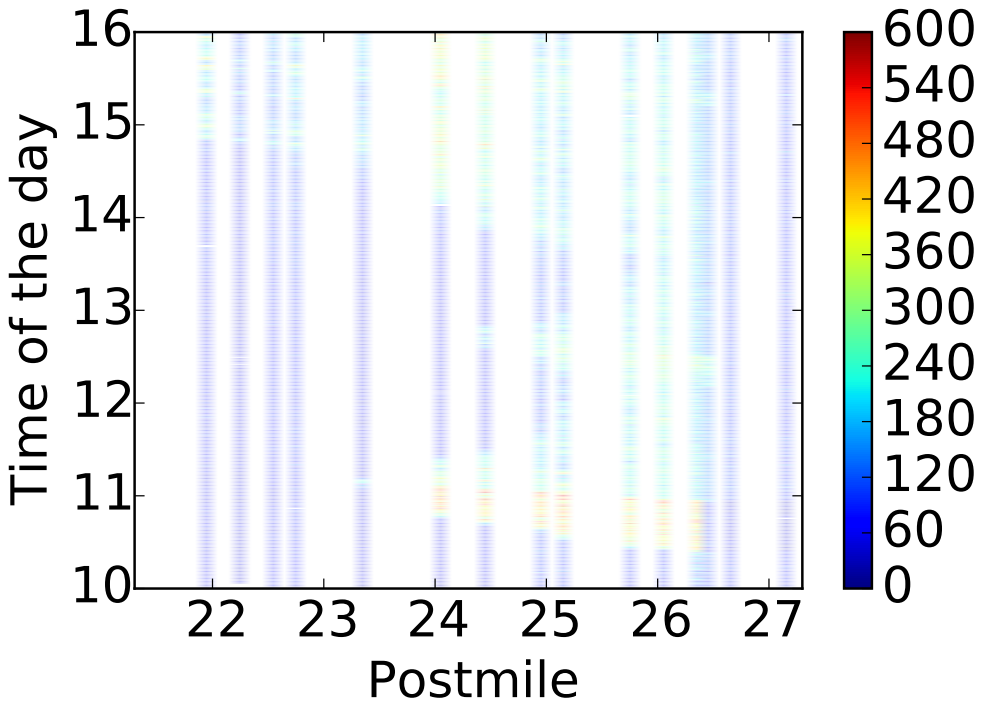
Next, the California algorithm [14] presented in Algorithm 1 is deployed for traffic incident detection. The California algorithm is performed between every two consecutive loop detectors and runs every two minutes. The thresholds T_1, T_2 and T_3 are calibrated as 0.33, 2.55 and 0.0003. The California algorithm reports the incident at 10:30 am around postmile 26.4, and it does not provide any false positive incident report during the six hour simulation. The California algorithm is able to detect the incident close to real time because there happens to be two inductive loops that are near the incident location, one in the upstream and one in the downstream (i.e., the thirteenth sensor and the fourteenth sensor). Moreover, the distance between the sensors and the incident location is less than 0.2 miles (i.e., two cells).

When the inductive loops are sparse, it will take the California algorithm long to detect an incident since it takes time for the congestion to propagate to the sensors, as shown in the CORSIM simulation and Table 5.3. As a result, we conclude that for the purpose of traffic incident detection, the proposed hybrid state estimation techniques may require less time to detect an incident compared to the California algorithm if the sensors are sparse and the penetration rate of GPS equipped vehicles is high. When the inductive loops are dense, the takes similar time for the proposed traffic estimation algorithm and the California algorithm to detect an incident.

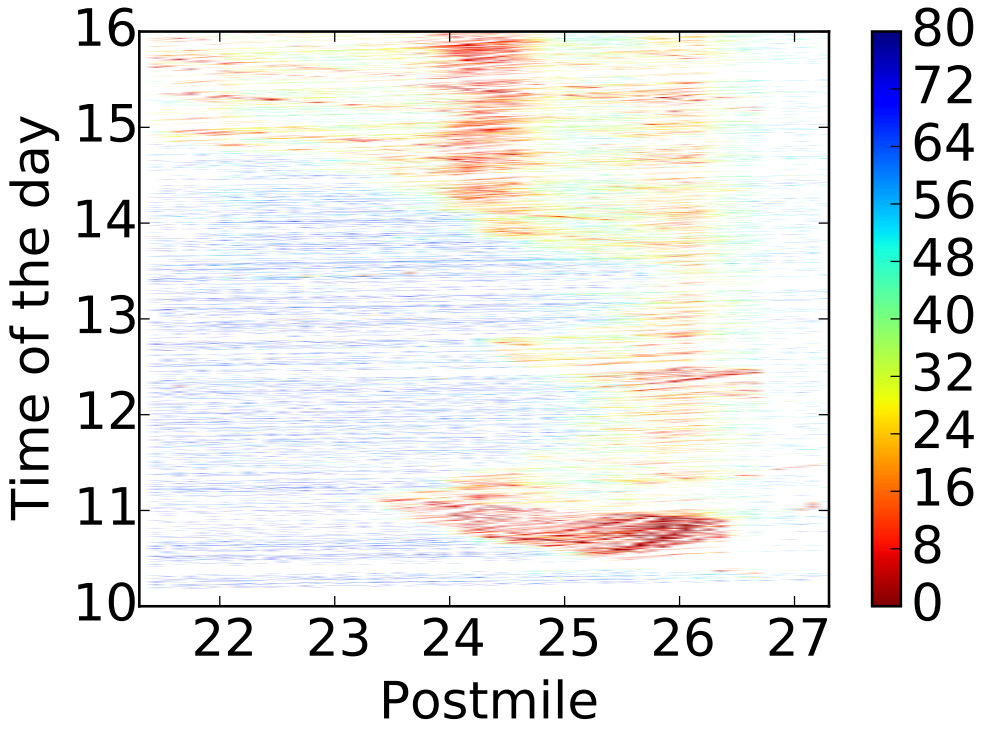
6.7 Summary of the main results of the field implementation

In this chapter, the EMMPF is implemented on the first and second order traffic models with field data on a segment of I–880 in California. The main findings are summarized as follows.

The EMMPF runs in real time and is able to jointly estimate traffic state and detect the incident when implemented on either the first order traffic model or the second order traffic model. The performance of the EMMPF is similar when implemented with either traffic model. The model noise is higher in field implementation compared to the numerical simulation in CORSIM. In this case, the particle filter is able to provide similar traffic state estimation accuracy compared to the EMMPF, even in the presence of incidents. Regardless of the larger model noise, the EMMPF algorithm is still able to correctly identify the correct location of the incident.

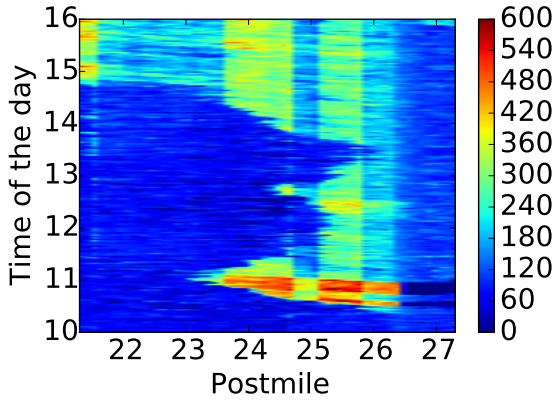


(a) Density measurements

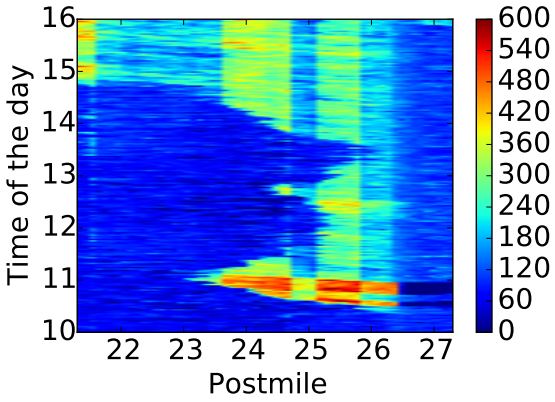


(b) Speed measurements

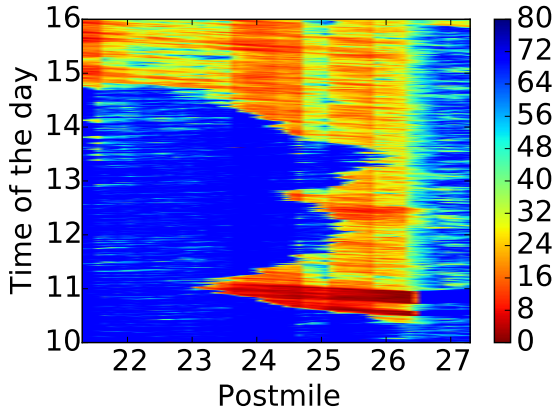
Figure 6.3: Density measurements (top) and speed measurements (bottom). Missing values appear in white.



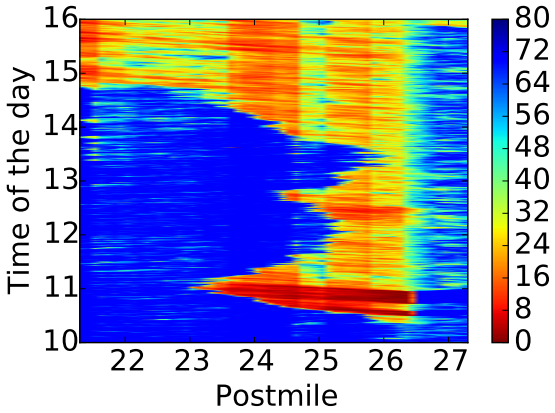
(a) Density (veh/mile)



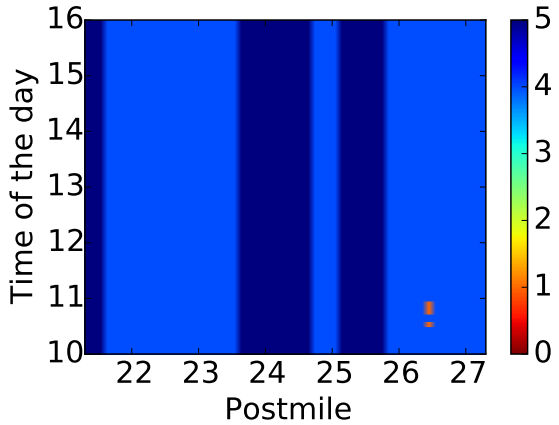
(b) Density (veh/mile)



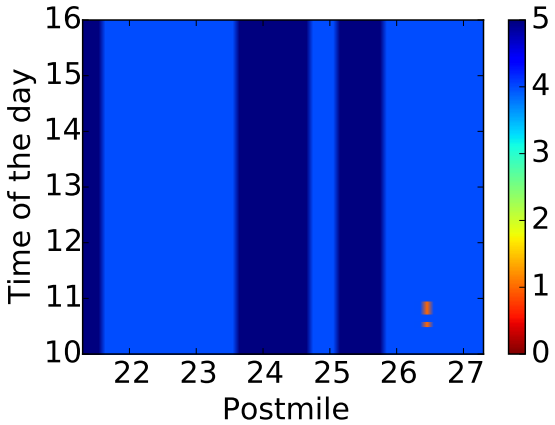
(c) Speed (mile/hour)



(d) Speed (mile/hour)

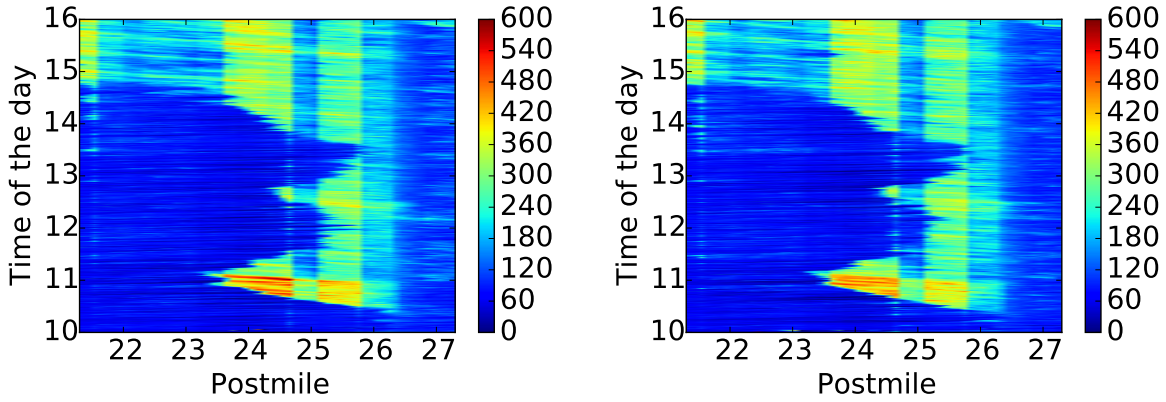


(e) Model variable (lanes open)



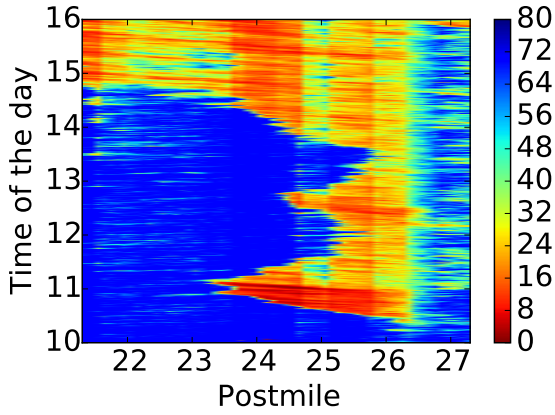
(f) Model variable (lanes open)

Figure 6.4: Estimation results of the EMMPF for traffic density (first row), traffic speed (second row) and the model variable (third row). The first column shows the results of the first order traffic model, and the second column shows the results for the second order traffic model.

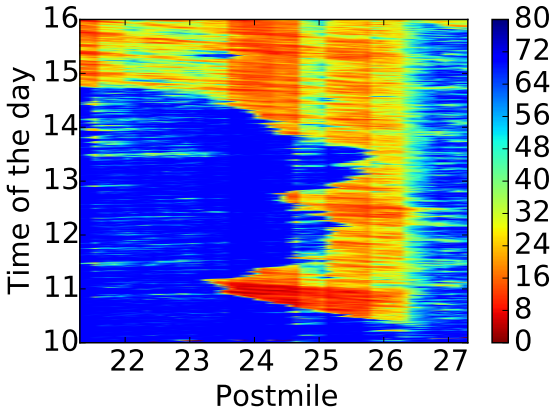


(a) Density (veh/mile)

(b) Density (veh/mile)



(c) Speed (mile/hour)



(d) Speed (mile/hour)

Figure 6.5: Estimation results of the particle filter for traffic density (first row) and traffic speed (second row). The first column shows the results of the first order traffic model, and the second column shows the results for the second order traffic model.

Chapter 7

Conclusions and future work

7.1 Conclusions

This dissertation posed the traffic state estimation and incident detection problem as a hybrid state estimation problem, where a continuous variable was used to denote the traffic state and a discrete variable was used to denote the severity and location of incidents.

There are several benefits to the proposed hybrid state estimation framework for jointly estimating incidents and the traffic state. First, since it is posed as a hybrid state estimation problem, standard state estimation algorithms such as the particle filter and variants of the Kalman filter can be modified to solve the estimation problem. In this dissertation, several of these extensions were developed and analyzed in terms of accuracy and run time, including a multiple model particle filter, a multiple model particle smoother, an interactive multiple model ensemble Kalman filter, and an efficient multiple model particle filter. More algorithms can be easily developed by incorporating other ideas in multiple model filtering to select the appropriate model, extending the filters into smoothers, considering other nonlinear filtering frameworks, or by leveraging alternative particle filtering algorithms with different resampling techniques.

Second, the framework allows for the identification of both the location and severity of

the incident, which is an improvement over other incident detection methods. For example, the California algorithm can only localize incidents to the nearest sensor pair, which may cover a large region (a mile or more) outside of the densest urban areas. Moreover, most incident detection algorithms do not identify the severity of the incident, which might be useful for control algorithms to mitigate the congestion by directing an appropriate amount of vehicles to seek alternate routes.

Third, the framework is flexible with respect to the data types that can be incorporated into the algorithm. If fixed sensor data such as inductive loops or radar sensors, or GPS data from navigation devices, smart phones, or connected vehicles are available, one only needs to change the observation equation to incorporate the measurements into the algorithm and use them for traffic state estimates. This is in sharp contrast to many of the customized incident detection algorithms that are designed specifically for probe data or specifically for loop data but cannot easily accommodate new data types, or fail when the primary source of data is not available.

Fourth, it was shown that the joint estimation of traffic states and incidents using models and algorithms that implement the framework performs at least as well as algorithms performing each task independently. In the numerical experiments in CORSIM, the performance improvement was large, both in terms of the accuracy of the traffic state, and in terms of the speed of detection of the incident. In the field experiment, similar performance was observed between the algorithms that jointly estimate incidents and traffic states, and those that estimate one without knowledge of the other. It is not a surprising finding because the density of fixed sensors on the experiment site was high (17 sensors on a six mile stretch of freeway), in which case all algorithms were able to detect the incident quickly. The traffic state estimates are also similar from all algorithms due to the larger model uncertainty required in the evolution equations associated with the traffic flow models. The larger model noise is required in part due to the large number of entrance and exit ramps on the stretch of highway and the lack of knowledge of the ramp boundary conditions. Even with the larger

model noise, both the first and second order traffic flow models are able to correctly localize and identify the severity of the incident while simultaneously providing accurate traffic state estimates.

Finally, several important challenges were overcome to implement the algorithms in the framework for field deployments. Both the first and second order traffic flow models were modified to incorporate the effects of an incident on the traffic dynamics. To apply the models to a network of freeways, the models were further extended to show how to couple the shared boundary conditions of the roadway segments under the influence of incidents on either segment. When larger networks are considered, the large number of possible models combined with the low probability of incident occurrence does not allow algorithms such as the MMPF, the MMPS, or the IMM EnKF to run in real time, even for moderate network sizes such as the network considered for the field experiment on the Mobile Century dataset. To overcome this challenge, an efficient multiple model particle filter was proposed, and experiments both on field data and on microsimulation data show that the algorithm can run in real time with a modest deterioration in the accuracy of the resulting state and incident estimates.

7.2 Future work

While this dissertation offered new insights into the challenges and potential benefits of jointly estimating incidents and events in a single estimation framework, several areas are open for further exploration. As indicated throughout the dissertation, computational scalability of the algorithms remains a major concern. Even for the most efficient algorithm (the efficient multiple model particle filter), a forward prediction of a particle for each model in the system is required. For large networks with thousands or tens of thousands of locations, each with multiple severities, even a single forward prediction might become too costly to implement in real time. Fortunately these predictions can be done in parallel and therefore

might benefit from high performance or distributed computing architectures which were not explored in this dissertation. Alternatively, the large networks might be partitioned into smaller ones, each of which can be solved on commodity hardware. Further algorithmic efficiencies could also be explored.

Like many model based traffic estimation problems, model calibration remains an important but cumbersome task. If poor model parameters are selected to model the fundamental diagram, the accuracy of both the traffic state estimates and incident estimates will suffer. Moreover, improved methods are needed to estimate boundary flows, especially for second order models which have twice as many boundary conditions as the first order models due to the extra state variable. Methods to reliably estimate the incident transition matrix from field data could also reduce the effort required to deploy the algorithms developed in this dissertation. While it is possible to simultaneously estimate the model parameters while jointly estimating the incidents and events, the large number of parameters in the model coupled with the nonlinearity and switching dynamics of the models implies that a serious research effort will be required to design algorithms that solve this problem.

Finally, it is possible to further enhance the accuracy of the incident detection capabilities by improving the determination of when an incident has occurred. For example, a classifier could be trained to link the posterior distribution or the best estimate of x^n and γ^n from the filter to the identification of an incident. Moreover, even when the algorithm switches to an incident model, additional verification is desired to infer the existence of an incident. For example, the estimation results from the dissertation could be combined with social media to infer the existence of an incident. Such investigations would be essential in order to get good practical performance in the field, without producing too many false positives or too many missed incidents.

Bibliography

[1] W Reilly. Highway capacity manual 2000. *TR News*, 1997.

[2] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981.

[3] D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4):199–329, 2000.

[4] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of modern physics*, 73(4):1067, 2001.

[5] M. J. Lighthill and G. B. Whitham. On kinematic waves II. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 229, pages 317–345, 1955.

[6] P. Zhang, S. C. Wong, and S. Q. Dai. A conserved higher-order anisotropic traffic flow model: description of equilibrium and non-equilibrium flows. *Transportation Research Part B: Methodological*, 43(5):562–574, 2009.

[7] S. Fan, B. Piccoli, and B. Seibold. A collapsed generalized Aw-Rascle-Zhang model. *SIAM Journal on Applied Mathematics*, In preparation.

[8] S. Blandin, A. Couque, A. Bayen, and D. Work. On sequential data assimilation for scalar macroscopic traffic flow models. *Physica D: Nonlinear Phenomena*, 241(17):1421–1440, 2012.

[9] L. Mihaylova, R. Boel, and A. Hegyi. Freeway traffic estimation within particle filtering framework. *Automatica*, 43(2):290–300, 2007.

[10] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2106–2111, 2004.

[11] J. Sau, N. El Faouzi, A. B. Assa, and O. De Mouzon. Particle filter-based real-time estimation and prediction of traffic conditions. *Applied Stochastic Models and Data Analysis*, 12, 2007.

[12] S. C. Tignor and H. J. Payne. Improved freeway incident detection algorithms. *Public Roads*, 41(1):32–40, 1977.

[13] H. J. Payne and S. C. Tignor. Freeway incident–detection algorithms based on decision trees with states. *Transportation Research Record*, (682), 1978.

[14] H. J. Payne, E. D. Helfenbein, and H. C. Knobel. Development and testing of incident detection algorithms, volume 2: Research methodology and detailed results. Technical Report FHWA–RD–76–20, Technology Service Corporation, 1976.

[15] M. Levin, G. M. Krause, and J. A. Budrick. Incident–detection algorithms. part 2. on–line evaluation. *Transportation Research Record*, (722):58–64, 1979.

[16] E. Parkany and C. Xie. A complete review of incident detection algorithms & their deployment: what works and what doesn't. Technical Report NETCR37, University of Massachusetts Transportation Center, 2005.

[17] B. Ristic, S. Arulampalm, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[18] J. C. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field

experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, 2010.

[19] M. Szeto and D. Gazis. Application of Kalman filtering to the surveillance and control of traffic systems. *Transportation Science*, 6(4):419–439, 1972.

[20] M. Cremer and M. Papageorgiou. Parameter identification for a traffic flow model. *Automatica*, 17(6):837–843, 1981.

[21] M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*. Springer-Verlag New York, 1983.

[22] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2):141–167, 2005.

[23] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo. An adaptive freeway traffic state estimator. *Automatica*, 45(1):10–24, 2009.

[24] C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

[25] C. F. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

[26] S. J. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[27] N. Polson and V. Sokolov. Bayesian analysis of traffic flow on interstate I-55: The LWR model. *arXiv preprint arXiv:1409.6034*, 2014.

[28] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):629–4640, 2008.

[29] T. Bengtsson, P. Bickel, and B. Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

[30] L. Mihaylova, A. Hegyi, A. Gning, and R. K. Boel. Parallelized particle and gaussian sum particle filters for large-scale freeway traffic systems. *IEEE Transactions on Intelligent Transportation Systems*, 2012.

[31] X. Sun, L. Muñoz, and R. Horowitz. Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application. In *Proceedings of the American Control Conference*, volume 3, pages 2098–2103, 2004.

[32] J.-C. Herrera and A. Bayen. Incorporation of Lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological*, 44(4):460–481, 2010.

[33] D. Work, S. Blandin, O. P. Tossavainen, B. Piccoli, and A. M. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1):1–35, 2010.

[34] D. Work, O. P. Tossavainen, S. Blandin, A. Bayen, T. Lwuchukwu, and K. Tracton. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5062–5068, 2008.

[35] L. Li, X. Chen, and L. Zhang. Multimodel ensemble for freeway traffic state estimations. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1323–1336, 2014.

[36] D. Jacquet, C. Canudas de Wit, and D. Koenig. Traffic control and monitoring with a macroscopic model in the presence of strong congestion waves. In *Proceedings of the*

44th IEEE Conference on Decision and Control, and European Control Conference, pages 2164–2169, 2005.

[37] D. Jacquet, M. Krstic, and C. Canudas de Wit. Optimal control of scalar one-dimensional conservation laws. In *Proceedings of the 25th American Control Conference*, pages 5213–5218, 2006.

[38] Y. Sun and D. Work. A distributed local kalman consensus filter for traffic estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6484–6491, 2014.

[39] C. G. Claudel and A. M. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part I: theory. *IEEE Transactions on Automatic Control*, 55(5):1142–1157, 2010.

[40] C. G. Claudel and A. M. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part II: Computational methods. *IEEE Transactions on Automatic Control*, 55(5):1158–1174, 2010.

[41] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part I. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.

[42] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part V. multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[43] S. Tafazoli and X. Sun. Hybrid system state tracking and fault detection using particle filters. *IEEE Transactions on Control Systems Technology*, 14(6):1078–1087, 2006.

[44] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model

methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

[45] H. AP. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.

[46] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part v. multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[47] A. L. Kurkjian. *The Estimation of Traffic Variables and Detection of Incidents Using Presence Detector Data*. PhD thesis, Massachusetts Institute of Technology, 1978.

[48] Nasser E Nahi. Freeway-traffic data processing. *Proceedings of the IEEE*, 61(5):537–541, 1973.

[49] M. S. Ahmed and A. R. Cook. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transportation Research Record*, (722):1–9, 1979.

[50] J. F. Collins, C. M. Hopkins, and J. A. Martin. Automatic incident detection–TRRL algorithm HIOCC and PATREG. Technical Report 526, Transport and Road Research Laboratory, 1979.

[51] M. Levin and G. M. Krause. Incident detection: a Bayesian approach. *Transportation Research Record*, (682):52–58, 1978.

[52] E. Parkany and D. Bernstein. Design of incident detection algorithms using vehicle–to–roadside communication sensors. *Transportation Research Record*, (1494):67–74, 1995.

[53] Y. J. Stephanedes and X. Liu. Artificial neural networks for freeway incident detection. *Transportation Research Record*, (1494):91–97, 1995.

[54] H. S. Mahmassani, C. Haas, S. Zhou, and J. Peterman. Evaluation of incident detection methodologies. Technical Report FHWA/TX–00/1795–1, Center for Transportation Research at the University of Texas at Austin, 1998.

[55] K. N. Balke. An evaluation of existing incident detection algorithms. Technical Report FHWA/TX–93/1232–20, Texas A&M University, Texas Transportation Institute, 1993.

[56] A. R. Cook and D. E. Cleveland. Detection of freeway capacity–reducing incidents by traffic–stream measurements. *Transportation Research Record*, (495):1–11, 1974.

[57] Y. J. Stephanedes and A. P. Chassiakos. Freeway incident detection through filtering. *Transportation Research Part C: Emerging Technologies*, 1(3):219–233, 1993.

[58] Y. J. Stephanedes and A. P. Chassiakos. Application of filtering techniques for incident detection. *Journal of Transportation Engineering*, 119(1):13–26, 1993.

[59] Y. J. Stephanedes, A. P. Chassiakos, and P. G. Michalopoulos. Comparative performance evaluation of incident detection algorithms. *Transportation Research Record*, (1360):50–57, 1992.

[60] A. P. Chassiakos and Y. J. Stephanedes. Smoothing algorithms for incident detection. *Transportation Research Record*, 1394:8–16, 1993.

[61] A. I. Gall and F. L. Hall. Distinguishing between incident congestion and recurrent congestion: a proposed logic. *Transportation Research Record*, pages 1–8, 1989.

[62] P. G. Michalopoulos, R. D. Jacobson, C. A. Anderson, and T. B. DeBruycker. Automatic incident detection through video image processing. *Traffic Engineering and Control*, 34(2):66–75, 1993.

[63] Y. Jeong, M. Castro-Neto, M. K. Jeong, and L. D. Han. A wavelet-based freeway incident detection algorithm with adapting threshold parameters. *Transportation Research Part C: Emerging Technologies*, 19(1):1–19, 2011.

[64] Akira Kinoshita, Atsuhiro Takasu, and Jun Adachi. Real-time traffic incident detection using a probabilistic topic model. *Information Systems*, In press, 2015.

[65] A. Lemarchand, D. Koenig, Martinez M., and John J. Incident detection for an uncertain traffic model. In *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, volume 8, pages 648–653, 2012.

[66] A. Willsky, E. Chow, S. Gershwin, C. Greene, P. Houpt, and A. Kurkjian. Dynamic model–based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control*, 25(3):347–360, 1980.

[67] L. Isaksen and H. J. Payne. Freeway traffic surveillance and control. *Proceedings of the IEEE*, 61(5):526–536, 1973.

[68] H. J. Payne. Models of freeway traffic and control. *Mathematical models of public systems*, 1(1), 1971.

[69] J. P. Kaipio and E. Somersalo. *Statistical and computational inverse problems*. Springer, 2005.

[70] A. Dabiri and B. Kulcsár. Freeway traffic incident reconstruction. a bi-parameter approach. *Transportation Research Part C: Emerging Technologies*, In Press, 2015.

[71] J. P. Lebacque, S. Mammar, and H. Haj-Salem. Generic second order traffic flow modelling. In *Transportation and Traffic Theory*, pages 755–776, 2007.

[72] C. F. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B: Methodological*, 29(4):277–286, 1995.

[73] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

[74] S. Smulders. Control of freeway traffic flow by variable speed signs. *Transportation Research Part B: Methodological*, 24(2):111–132, 1990.

[75] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[76] J. P. Lebacque. Les modeles macroscopiques de trafic. In *Annales des Ponts et chaussées*, number 67, pages 24–45, 1993.

[77] R. J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser, 1992.

[78] S. Fan, M. Herty, and B. Seibold. Comparative model accuracy of a data-fitted generalized Aw-Rascle-Zhang model. *Networks and Heterogeneous Media*, Vol. 9, No. 2:239–268, 2014.

[79] S. Blandin, D. Work, P. Goatin, B. Piccoli, and A. Bayen. A general phase transition model for vehicular traffic. *SIAM Journal on Applied Mathematics*, 71(1):107–127, 2011.

[80] S. Fan and D. Work. A heterogeneous multiclass traffic flow model with creeping. *SIAM Journal on Applied Mathematics*, 75(2):813–835, 2015.

[81] S. Fan and B. Seibold. A comparison of data-fitted first order traffic models and their second order generalizations via trajectory and sensor data. *arXiv preprint arXiv:1208.0382*, 2013.

[82] A. Aw and M. Rascle. Resurrection of second order models of traffic flow. *SIAM Journal on Applied Mathematics*, 60:916–944, 2000.

[83] H. M. Zhang. A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B: Methodological*, 36:275–290, 2002.

[84] B. S. Kerner. Experimental features of the emergence of moving jams in free traffic flow. *Journal of Physics A*, 33:221–228, 2000.

[85] B. S. Kerner. *Phase transitions in traffic flow*. Traffic and Granular Flow. Springer, 2000.

[86] S. Blandin, J. Argote, A. Bayen, and D. Work. Phase transition model of non stationary traffic: Definition, properties and solution method. *Transportation Research Part B: Methodological*, 52:31–55, 2013.

[87] R. M. Colombo. Hyperbolic phase transitions in traffic flow. *SIAM Journal on Applied Mathematics*, 63(2):708–721, 2003.

[88] R. M. Colombo, P. Goatin, and F. S. Priuli. Global well posedness of traffic flow models with phase transitions. *Nonlinear Analysis: Theory, Methods & Applications*, 66(11):2413–2426, 2007.

[89] C. M. Dafermos. Polygonal approximations of solutions of the initial value problem for a conservation law. *Journal of Mathematical Analysis and Applications*, 38(1):33–41, 1972.

[90] J. P. Lebacque, H Haj-Salem, and S. Mammar. Second order traffic flow modeling: supply-demand analysis of the inhomogeneous Riemann problem and of boundary conditions. In *Proceedings of the 10th Euro Working Group on Transportation (EWGT)*, number 108–115, 2005.

[91] M. Herty and M. Rascle. Coupling conditions for a class of second order models for traffic flow. *SIAM Journal on Mathematical Analysis*, 38(2):595–616, 2006.

[92] G. Bretti, R. Natalini, and B. Piccoli. A fluid-dynamic traffic model on road networks. *Archives of Computational Methods in Engineering*, 14(2):139–172, 2007.

[93] G. M. Coclite, M. Garavello, and B. Piccoli. Traffic flow on a road network. *SIAM Journal on Mathematical Analysis*, 36(6):1862–1886, 2005.

[94] M. Garavello and B. Piccoli. Traffic flow on a road network using the aw–rascle model. *Communications in Partial Differential Equations*, 31(2):243–275, 2006.

[95] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[96] H. Chen, H. A. Rakha, and S. Sadek. Real–time freeway traffic state prediction: A particle filter approach. *Proceedings of the IEEE conference Intelligent Transportation Systems*, pages 626–631, 2011.

[97] A. Doucet, De F. N., and N. Gordon. *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[98] A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[99] R. Wang and D. B. Work. Interactive multiple model ensemble Kalman filter for traffic estimation and incident detection. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 804–809, 2014.

[100] R. Wang, D. Work, and R. Sowers. Multiple model particle filters for traffic estimation and incident detection. *Submitted to IEEE Transactions on Intelligent Transportation Systems*, 2014.

[101] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 1960.

[102] B. Anderson and J. Moore. *Optimal filtering*. Dover Publications, 1979.

[103] G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[104] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[105] A. C. Lorenc. The potential of the ensemble Kalman filter for NWP–a comparison with 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3183–3203, 2003.

[106] G. Evensen and P. J. Van Leeuwen. An ensemble kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

[107] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya. Automatic calibration of the fundamental diagram and empirical observations on capacity. In *Transportation Research Board 88th Annual Meeting*, 2009.

[108] B. Pan, U. Demiryurek, C. Shahabi, and C. Gupta. Forecasting spatiotemporal impact of traffic incidents on road networks. In *Proceeding of the IEEE Conference on Data Mining*, pages 587–596, 2013.

[109] http://pems.dot.ca.gov/.

[110] T. Choe, A. Skabardonis, and P. Varaiya. Freeway performance measurement system: operational analysis tool. *Transportation Research Record: Journal of the Transportation Research Board*, (1811):67–75, 2002.