USDOT Region V Regional University Transportation Center Final Report

NEXTRANS Project No 097IY04.

Joint parameter and state estimation algorithms for real-time traffic monitoring

By

Ren Wang

PhD Candidate

University of Illinois at Urbana Champaign

renwang2@illinois.edu

And

Daniel B. Work (PI)

Assistant Professor

University of Illinois at Urbana Champaign

dbwork@Illinois.edu

Report Submission Date: December 31, 2013

DISCLAIMER

Funding for this research was provided by the NEXTRANS Center, Purdue University under Grant No. DTRT12-G-UTC05 of the U.S. Department of Transportation, Research and Innovative Technology Administration (RITA), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

USDOT Region V Regional University Transportation Center Final Report

TECHNICAL SUMMARY

NEXTRANS Project 097IY04 Final Report, December 2013

Joint parameter and state estimation algorithms for real-time traffic monitoring

Introduction

A common approach to traffic monitoring is to combine a macroscopic traffic flow model with traffic sensor data in a process called state estimation, data fusion, or data assimilation. The main challenge of traffic state estimation is the integration of various types of sensor data (e.g. speed, flow, travel time, etc.) into the flow model due to the nonlinearities of the traffic model. When parameters are also estimated, the nonlinearity of the estimation problem increases, motivating the development of advanced estimation algorithms to handle the additional nonlinearity. To improve performance of traffic state estimation algorithms this work investigates the problem of simultaneously or jointly estimating both the traffic state and the parameters of the traffic model. It uses two new traffic parameter and state estimation algorithms based on *multiple model particle filtering*, and *multiple model particle smoothing*. Because incidents on freeways can be modeled through parameter changes in the traffic model, this work applies both algorithms to the problem of incident detection.

Findings

The main contributions of this work are as follows. First, it is shown that the problem of detecting incidents can be posed as a traffic flow model parameter estimation problem. A family of parameters known as regime variables are used to model the location and severity of the incident. The regime variable is used to indicate the number of open (unobstructed) lanes everywhere along the roadway. A reduction of the number of open lanes, as indicated by the regime variable, indicates that an incident has occurred.

Second, two new algorithms are developed to jointly estimate the traffic state, and the regime variable parameters. Both algorithms are able to estimate the traffic conditions based on speed data obtained from GPS equipped vehicles, and/or flow and density data from inductive loop detectors. Both algorithms are online, sequential algorithms, and the primary distinction between the filter and the smoother is that the smoother runs with a slight lag compared to the filter. In other words, the smoother uses all data up to the current time to estimate traffic conditions in the past (e.g. two minutes ago), while the filter uses the same data to estimate the traffic at the current time.

Third, we test both algorithms in two numerical environments. In the first set of experiments, synthetic data is generated from a similar macroscopic model to the one assumed in both estimation algorithms. These experiments represent an upper bound on the best-case performance of the algorithms. These experiments are used to verify that incidents can only be detected if the traffic volumes are sufficiently large. The traffic flow must be large enough to activate the bottleneck created by the incident.

An additional numerical setup is developed in CORSIM, which is a microscopic traffic simulation software. CORSIM is used to simulate an incident and produce measurements to integrate into the multiple model particle filtering and smoothing algorithms. This experiment provides a better understanding of the practical performance of the proposed joint parameter and state estimation algorithms if deployed in the field on experimental data. The CORSIM experiments show that incidents can be detected when the headway between GPS vehicles is 20 seconds (representing a penetration rate of about 2%). Moreover, if the headway between GPS vehicles is increased to 60 seconds, the multiple model particle smoother can produce similar performance, but with a delay of 40 seconds.

The smoothing algorithm is able to improve the accuracy of the state and parameter estimate. However, it comes at the cost of a time lag. The time slag should be a function of the sensor density.

Recommendations

This work develops two new joint parameter and state estimation algorithms, and demonstrated good performance can be achieved on estimating both the traffic state and the parameters representing incidents when the traffic flows are sufficiently large and the GPS penetration rate exceeds 2% of the flow. Several topics are open for additional study.

The output of the multiple model particle filter and smoother is a posterior distribution on the regime variable everywhere along the roadway; it does not directly predict an incident. Future work should explore the sensitivity to various thresholds to determine how high the probability of an incident should be in order to declare that indeed an incident has occurred. The optimal threshold should minimize false positives without missing the ability to detect incidents quickly.

The algorithms should be tested and validated with field data. This is challenging because it requires a sensor network with an incident feed that contains accurate information about time, location, and severity of the incident. The scalability of the algorithms could also be explored as the size of the freeway network increases.

Other possible extensions include field comparisons with existing incident detection algorithms relying on statistical approaches.

Contacts

For more information:

Prof. Daniel B. Work
University of Illinois at Urbana Champaign
1203 Newmark Civil Engineering Lab
205 N Mathews Ave.
Urbana, IL, 61801

Email Address: dbwork@illinois.edu

NEXTRANS Center
Purdue University - Discovery Park
2700 Kent B-100
West Lafayette, IN 47906

nextrans@purdue.edu
(765) 496-9729
(765) 807-3123 Fax

www.purdue.edu/dp/nextrans

NEXTRANS Project No. 097IY04

Joint parameter and state estimation algorithms for real-time traffic monitoring

By

Ren Wang
PhD Candidate
University of Illinois at Urbana Champaign
renwang2@illinois.edu

And

Daniel B. Work (PI)
Assistant Professor
University of Illinois at Urbana Champaign
dbwork@Illinois.edu

Report Submission Date: December 31, 2013

Contents

CHAPTER 1. INTRODUCTION

1.1 Motivation for joint parameter and state estimation algorithms

Traffic congestion is a major problem. As the US population continues to grow and move towards urban areas, the impact of traffic congestion on human mobility, the economy, and the environment is ever increasing. In many cases, solutions to traffic congestion will ultimately depend on improved management of existing infrastructure through the use of innovative, integrated solutions in technology and policy.

Currently, real-time traffic monitoring is undergoing a major revolution do to (*i*) the rapidly increasing availability of data from mobile devices such as smartphones, and (*ii*) cheap, scalable computing on the cloud. A common approach to traffic monitoring in the research literature is to combine a macroscopic traffic flow model, such as the *Lighthill Whitham Richards* (LWR) *partial differential equation* (PDE) [1,2] or its discrete counterpart known as the *Cell Transmission Model* (CTM) [3,4], with traffic sensor data in a process called state estimation (also known as data fusion or data assimilation).

The main challenge of traffic state estimation is the integration of various types of sensor data (e.g. speed, flow, travel time, etc.) into the flow model due to the nonlinearities in traffic flow, which is also captured by the LWR PDE. The existence of these nonlinearities have led to the application of sophisticated nonlinear data fusion techniques such as *extended Kalman filtering* (EKF) [5], *ensemble Kalman filtering* (EnKF) [6,7], *unscented Kalman filtering* (UKF) [8], and *particle filtering* (PF) [9] for traffic monitoring in the transportation literature. These algorithms have seen limited application in industry to date due to their high computational cost, but they should see more widespread use in the near future due to computational advances such as cloud computing. Notably, the *Mobile Millennium* project [7, 10] used the ensemble Kalman filtering technique to monitor traffic on Northern California freeways as part of a large-scale field experiment.

One important limitation to the ensemble Kalman filtering technique deployed in Mobile Millennium is that it is a minimum variance filter. Due to the nonlinearities of the traffic flow model, it is possible to generate multi-modal distributions on the estimate of the state. When this occurs, the minimal variance correction in the update step can result in a poor estimate of the true state. Moreover, when parameters are estimated, the nonlinearity is increased.

In the specific problem of incident detection, filters such as the EnKF, EKF, and UKF are not directly applicable because the parameter that indicates the incident is an integer variable. It parameterizes the fundamental diagram and scales it according to the number of lanes that are blocked due to an incident. In this work, we propose two new estimation algorithms to jointly estimate the parameters and the state variables to overcome this limitation. Both algorithms use the discretized LWR equation as the evolution equation, and a Bayesian update step.

1.2 Application to the problem of detecting incidents

The new joint parameter and state estimation algorithms are applied to the problem of detecting incidents on freeways. The problem of identifying incidents is a specific application domain that could immediately benefit from improvements in traffic estimation algorithms, because traffic incidents are a major cause of safety, delay and congestion on the freeway.

About 60 percent of all freeway delays are caused by incidents [11]. Significant delays occur when the incident results in a lane blockage, effectively reducing the capacity of the freeway. The resulting capacity drop can effectively introduce a new bottleneck, and if the traffic flow exceeds the bottleneck capacity, congestion will result.

In 2007, more than 41,000 people were killed in crashes [12]. Moreover, traffic incidents reduce the safety of other motorists, so real-time incident detection is an effective way to improve traffic operations and safety. Early actions could be taken to save lives, clear the incident, and recover the normal traffic operations.

The main challenges for real-time traffic incident detection are related to sensing and interpretation of the sensor data. First, due to the traditionally high cost of monitoring infrastructure, the number of sensors on the freeways is limited. If fixed sensors are located far from each other and a traffic incident occurs on the roadway between the sensors, it may take a long time for the effect of the incident to propagate to a sensor, where it can be registered as an anomaly. Second, it is hard to distinguish between traffic incidents from similar traffic pattern, which results in a high false alarm rate. Consequently, accurately locating the exact location and severity of a traffic incident is a difficult problem.

By posing the incident detection problem as a joint parameter and state estimation problem, both the location and severity can be estimated in real-time using density measurements from inductive loop detectors and speed measurements from GPS vehicles. A multiple model particle filter and a multiple model particle smoothing algorithm is developed to jointly estimate the state and parameters. The proposed method is tested by using synthetic data generated from a macroscopic traffic model and data generated by a microscopic traffic simulation software (CORSIM).

1.3 Outline and contributions

The report is organized as follows. In chapter 2, the existing methods for joint parameter and state estimation using a macroscopic flow model are reviewed. A summary of alternative approaches for incident detection is also presented. Most of the existing methods use data from fixed sensors to detect incidents. The popularity of probe based methods has increased in recent years, and their performance depends on the penetration of probe equipped vehicles.

In chapter 3, the macroscopic traffic flow model used in the joint parameter and state estimation algorithms is introduced. In the proposed traffic model, traffic density is used to describe the evolution of traffic in discrete time and space. In particular, we specify a parameter that denotes the number of lanes open at each time everywhere along the roadway.

This parameter is then embedded in the traffic model, and the evolution of this parameter is modeled as a Markov Chain. It indicates the probability of having a traffic incident at the next timestep, given the current traffic condition.

In chapter 4, the traffic incident detection problem is posed as a joint parameter and state estimation problem, and discussion is provided on when the parameters are difficult to identify. To provide a solution to the joint state and parameter estimation problem, a multiple model particle filter is introduced. When sensors are located far apart, the multiple model particle filter performance can suffer. This is because it takes time for the information to propagate to the nearest sensor, and consequently, the algorithm is not able to keep track with the correct state and parameter at each timestep. To further improve the accuracy of the joint state and parameter estimates, the multiple model particle smoothing algorithm is introduced. The multiple model particle smoother allows the information to propagate to the nearest sensor by estimating the traffic state with a slight lag. This effectively allows the estimation algorithm use more measurements to reject poor particles, and increases the accuracy of both the state and parameter estimates.

In chapter 5, the proposed algorithms are tested using synthetic incident data that is generated by the same macroscopic traffic model assumed as an evolution equation for the estimators. Because the model error is low, the resulting estimates represent a best case performance of the estimation algorithms. The accuracy as a function of the headway of GPS equipped vehicles is explored. The simulation results show that the multiple model particle filtering is able to correctly detect the location and severity of a traffic incident when the penetration rate of GPS vehicles is high. As the penetration rate decreases, the multiple model particle filtering is not able to correctly estimate the state and the parameter indicating an incident. When the multiple model particle smoother is used, there is a significant improvement for both the state and parameter estimates.

In chapter 6, the proposed algorithms are tested using traffic incident data simulated by CORSIM. If the proposed algorithm is able to detect the traffic incident generated by the microscopic CORSIM simulation, it has a high potential to work in practice, since the simulation environment is completely decoupled from the traffic model assumed in the estimators, except through the measurements. A new fundamental diagram is constructed to represent the density-flow relationship observed in CORSIM, and it is used in the macroscopic evolution equation used in the traffic estimation algorithms. The simulation results show the proposed algorithms are able to detect the location and severity of the traffic incident with the traffic incident from CORSIM when the data rates or the smoothing window are sufficiently high.

Conclusions and directions for future work are presented in chapter 7.

2.1 Overview of macroscopic model based traffic estimation

A common framework for problems that combine model dynamics with measurements can be investigated through a general *evolution-observation* model of the form:

$$\begin{cases} x^{n+1} = f(x^n, \theta, \omega^n) \\ z^n = h(x^n, \nu^n) \end{cases} \tag{2.1}$$

The state evolution equation $f(\cdot,\cdot,\cdot)$ describes the system dynamics (which is nonlinear and possibly non-differentiable for traffic flows), while $h(\cdot,\cdot)$ describes the relationship between the measurement process and the state. Through appropriate discretization of a macroscopic traffic model, x^n represents the traffic flow property such as density, speed, or flow, while z^n denotes the vector of measurements at time n. The vector θ represents the parameters of the model. The state noise ω^n accounts for errors introduced through the chosen model abstraction as well as uncertainties on model parameters, while the observation noise ν^n accommodates measurement and sampling errors.

The aim of the *state estimation problem* (also known as *filtering* or *data assimilation*, or *data fusion* when multiple measurement types are considered) is to recover information about the distribution of X^n given measurements $z^1, z^2, \dots, z^n$ of the observation process up to and including time n. When the parameters θ are also treated as random variables to be estimated, the resulting problem is known as an *inverse problem* (also *parameter estimation* or *system identification*). In the context of real-time traffic monitoring, a wide class of *online* algorithms are based on nonlinear extensions of the seminal *Kalman filter* (KF), which is an algorithm for sequentially computing the *Best Linear Unbiased Estimate* (BLUE) [13] of the state given a stream of measurements.

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 [14], 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 evolution observation model (2.1) and models the noise processes as additive, to fit the framework of the KF. Starting in the early 1980's, a modified version of Payne's model was used for a variety of estimation and control problems through the work of Papageorgiou and his collaborators [5, 15-17]. Sun et al. [18] 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 [19,20].

Recently, the cell transmission model has been used in state estimation problems through increasingly advanced nonlinear filters, including unscented Kalman filtering [8, 21] and particle filtering [8, 9, 22]. The particle filter is shown to perform better than UKF for traffic state reconstruction [8], but has a higher computational cost. Implementation of particle filtering techniques on high dimensional systems (several thousand states or more), remains an open problem due to inherent scalability challenges for particle filters [23]. Other treatments of traffic estimation include ensemble Kalman filtering [6, 7], adjoint-based control and data assimilation [24, 25], and direct injection into a Hamilton-Jacobi reformulation of the LWR PDE [26, 27].

2.2 Methods for incident detection

Traffic incident detection problems have been widely studied in the past several decades. A comprehensive overview of incident detection studies can be found in the review articles [28-30]. Here, we summarize a subset of the existing methods following the work [28].

One group of algorithms is based on the well known *California algorithm* [31-34]. 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. The California algorithm requires two sensors to collect traffic occupancy values. One sensor must be located upstream from the incident, while the other must be 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 threshold, and if the values exceed the threshold, an incident alarm is triggered. Consequently, the method is referred to as a comparative algorithm.

The *double exponential smoothing algorithm* [35] 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, and the weighing is performed with a double exponential smoothing function. The errors between predicted and observed traffic variables are described by an algebraic sum, which is used as a tracking signal for detecting the traffic incident. The idea of this method is that the tracking signal should be close to zero if there is no traffic incident, while if it exceeds some threshold, an incident is reported.

The *low pass filter algorithms* [36-39] focus on pre-processing the occupancy data from inductive loop detectors to improve incident detection. The short-term noise data and inhomogeneities of the measurements are removed by rejecting high frequency fluctuations in the measurements and weighing 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 are two algorithms that work in combination to detect traffic incidents [40]. 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 estimates the traffic speed between upstream and downstream sensors, by measuring the travel time of vehicles 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.

The *autoregressive integrated moving average* (ARIMA) model [41] assumes the difference between 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 data sets 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* [42] computes the likelihood of an incident occurrence by using Bayesian statistical techniques. This approach develops the frequency distributions of the upstream and downstream occupancy ratios for both incident and incident-free conditions. Mathematical expressions are developed for the distribution of the ratios from incident and incident-free data. When a measurement is collected, the algorithm computes the likelihood of an incident occurrence by comparing the radio of the occupancy measurements from upstream and downstream and the frequency distributions of the occupancy ratios. The method has been compared with the California algorithm, where it was found to have a higher detection rate and lower false alarm rate. However, the mean-time-to-detect is higher.

The *catastrophe theory model* [43] aims at detecting incidents and distinguishing between incident congestion and recurrent congestion. It assumes there will be a sharp change of speed when the traffic becomes congested from free flow, and the change of traffic flow and occupancy will be smooth. The model uses historical data to determine the relationship between flow and occupancy. The different speed variations for congested and uncongested traffic states are also determined. Then, two tests are applied to determine the existence of a traffic incident. 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* [44] approach assumes that spatial and temporal traffic patterns can be recognized and classified by an artificial neural network. The artificial neural network was 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 detection rate and false alarm rates outperforms other algorithms that have been tested with the same data sets.

The *Autoscope incident detection algorithm* [45] is based on image processing techniques. First, a video camera is used to monitor traffic and collect image data. Then, a 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 to compare with a preset threshold. When the computed traffic parameters exceed the threshold, an incident is reported. The video detection system has been tested in Minnesota. Another benefit of this approach is that it is possible to check the existence of a traffic incident by reviewing the video, however, more equipment is required to implement this method.

The technique most closely related to the current work is the *dynamic model* [46] approach, which uses a second order macroscopic traffic model to describe the evolution of traffic. The nonlinear differential equation of the model relates speed-density and flow-density by using the fundamental diagram. In this approach, two hypothesis testing techniques are used to detect traffic incidents. This method performs better in terms of distinguishing shock waves from traffic incidents. However, it assumes traffic sensors are available at each link along the freeway.

Another class of algorithms is based on probe data. The *MIT algorithm* [47] exploits electronic toll transponders to detect incidents using a headway algorithm, a lane switching algorithm, and a 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. The lane-switching algorithm uses toll transponder to collect lane-specific and vehicle-specific data to obtain the number of lane switches. It is premised on the idea that if the number of lane switches exceeds a certain threshold, it is an indication of a traffic incident. The lane-monitoring algorithm monitors the lane in which each vehicle travels. The idea is that if more vehicles travel on one lane than compared to another, it indicates there may be an incident in one of the lanes. The performance this this group of methods depends on the probe penetration rate and the distance between transponder readers.

CHAPTER 3. TRAFFIC EVOLUTION EQUATIONS

In this chapter, the traffic evolution equation used for joint parameter and state estimation is described. The scalar traffic model is parameterized with a regime variable that identifies the location and severity of incidents. An evolution of the regime variable is also provided.

3.1 Traffic evolution equation

The *Lighthill--Whitham--Richards* (LWR) *partial differential equation* (PDE) [1, 2] is used to describe the evolution of the density $\rho(x,t)$ at location x and at time t on a roadway. This model expresses the conservation of vehicles on the roadway, and is given by:

$$\frac{\partial\rho(x,t)}{\partial t}+\frac{\partial\rho(x,t)v_\gamma(\rho)}{\partial x}=0, \qquad (x,t)\in(0,L)\times(0,T) \tag{3.1}$$

with the following initial and boundary conditions:

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

and where ρ_0, ρ_l, and ρ_r are the initial, left, and right boundary conditions. To close the model, a constitutive relationship between density and velocity, denoted $v_\gamma(\cdot)$, must be specified. One common assumption is the triangular model [3, 48], where the velocity function v is described by

$$v_\gamma(\rho)=\begin{cases} v_{max} & if\ \rho\le\gamma\rho_c\\ -w_f(1-(\rho_{max}/\rho)) & otherwise\end{cases} \tag{3.3}$$

where v_{max} and w_f are the maximum velocity and the maximum backward propagating wave speed. The parameter ρ_c and ρ_{max} are the critical density at which the flow transitions from free-flow to congested and the maximum density. The regime variable $\gamma(x,t)$ denotes the number of lanes open at location x and time t.

For numerical implementation, (3.1) is discretized by using a Godunov scheme [49], yielding the cell transmission model [3, 4]. Specifically, the time and space domains are discretized by introducing a discrete timestep ΔT, indexed by $n \in \{0, \dots, n_{max}\}$ and a discrete space step Δx, indexed by $i \in \{0, \dots, i_{max}\}$. The discretized system is given by:

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

where the numerical flux $G(\rho_i^n, \rho_{i+1}^n) = \min\{\mathrm{S}(\rho_i^n), \mathrm{R}(\rho_{i+1}^n)\}$. The functions S and R are known as the sending and receiving functions, which are given by

$$S(\rho) = \begin{cases} q(\rho) & if\ \rho \leq \gamma\rho_c \\ q(\rho_c) & otherwise \end{cases} \tag{3.5}$$

$$R(\rho) = \begin{cases} q(\rho_c) & if\ \rho \leq \gamma\rho_c \\ q(\rho) & otherwise. \end{cases} \tag{3.6}$$

Here, the function $q = \rho \times v_\gamma(\rho)$. In order to ensure numerical stability, the time and space steps are coupled by the CFL [50] condition: $\alpha_{max}\frac{\Delta t}{\Delta x} \leq 1$, where α_{max} denotes the maximal characteristic speed.

In (3.4), ρ_i^n denotes the value of the computed traffic density at timestep n and space step i. The boundary condition of (3.4) is given by:

$$\rho_0^{n+1} = \rho_0^n + \frac{\Delta T}{\Delta x}\left(G(\rho_l^n, \rho_0^n) - G(\rho_0^n, \rho_1^n)\right) \tag{3.7}$$

$$\rho_{i_{max}}^{n+1} = \rho_{i_{max}}^n + \frac{\Delta T}{\Delta x}\left(G\left(\rho_{i_{max}-1}^n, \rho_{i_{max}}^n\right) - G\left(\rho_{i_{max}}^n, \rho_r^n\right)\right)$$

The traffic model defined by (3.4) and (3.7) defines the deterministic component of the evolution operator f in (2.1).

3.2 Parameterization of the fundamental diagram to incorporate incidents

Figure 3.1 shows the fundamental diagrams under traffic incident scenarios for a three-lane freeway. We assume the maximum speed v_{max} and backward propagating wave speed w_f are fixed. When a traffic incident occurs, the maximum flow will drop depending on how many lanes are blocked. Accordingly, the critical density and jam density values change. Thus, if we could identify which fundamental diagram is used in each cell at each timestep, we could know the number of lanes blocked at these positions.

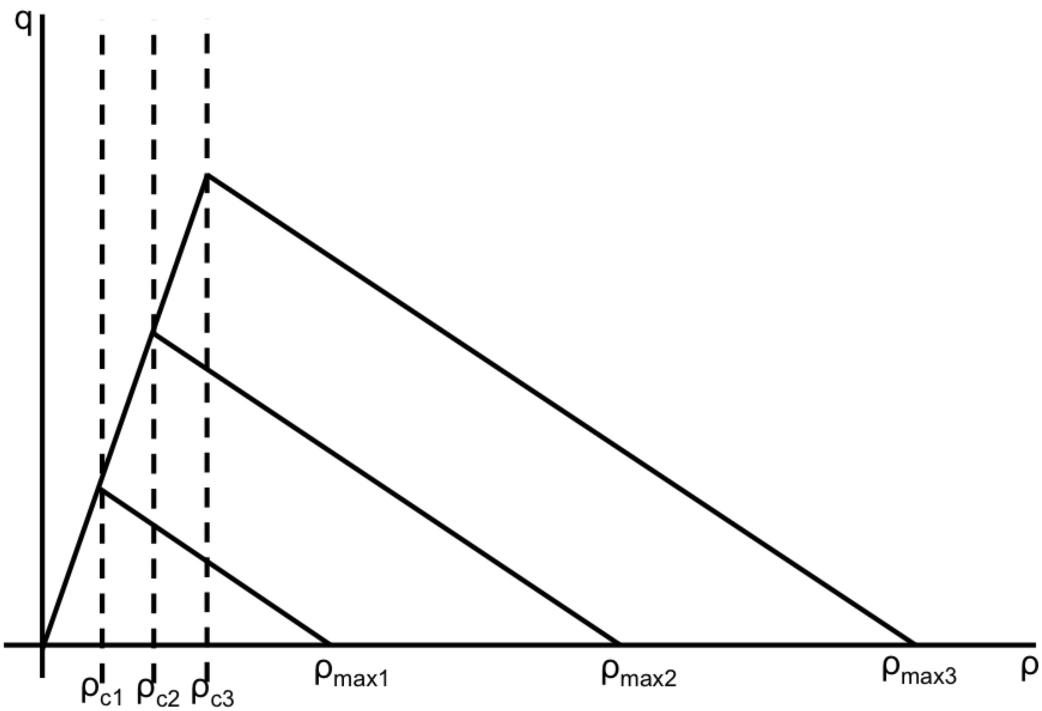


Figure 3.1: Relationships between traffic density and flow under traffic incident for a 3-lane freeway

3.3 Parameter evolution equations to model incidents

The regime variable γ is the parameter used to model incidents in the fundamental diagram. Specifically, the regime variable γ is defined as a $i_{max} + 1$ dimensional vector, where the value in each dimension denotes the number of lanes open for the corresponding cell. The regime variable is modeled as a s-state first-order Markov chain [51] with transitional probabilities defined by:

$$\pi_{ij} = p\{\rho^n = j | \rho^{n-1} = k\} \; (k, j \in S), \tag{3.8}$$

where $S = \{1,2, \dots, s\}$. The set S defines the possible regimes under which the system can operate. In this work, it is the number of lanes open along the freeway under all possible incident conditions, so the total number of regimes is the number of cells times the number of lanes in each cell.

The transitional probability matrix is defined as $\Pi = [\pi_{kj}]$, which an $s \times s$ matrix satisfying

$$\pi_{kj} \geq 0 \; and \; \sum_{j=1}^{s} \pi_{kj} = 1. \tag{3.9}$$

The regime probabilities are defined as:

$$\mu_k = p\{\gamma_1 = k\}, \tag{3.10}$$

for $k \in S$, such that

$$\mu_k \geq 0 \; and \; \sum_{k=1}^{s} \mu_k = 1. \tag{3.11}$$

This multiple switching dynamic model indicates the probability of the transition from one regime to another. In the traffic incident detection problem, it specifies how likely the number of lanes will open at each discretized cell at the next timestep given the number of lanes open at the current time.

3.4 *Sensitivity of the traffic estimate on the regime parameter*

The real-time estimation of the traffic density and the regime parameter is a challenging problem. In fact, it is not always possible to correctly and uniquely estimate the regime parameter. This can occur because there is insufficient data, and the particle filter estimate can diverge due to an insufficient number of samples and an increased parameterization of the model.

On the other hand, even if sufficient data is available, it is possible the regime variable cannot be uniquely determined due to the nature of the problem. For example, the traffic incident cannot be detected when it has no influence on the traffic state, and consequently on the measurements. In other words, if the remaining lanes have enough capacity to accommodate all the traffic, the traffic incident will not influence the measurements collected from the sensors.

We consider a specific example on a three lane freeway in more detail. Suppose we want to know if there is a traffic incident in cell i, (3.4) indicates the density is determined by the sending function of cell i-1, the sending and receiving functions of cell i, and the receiving function of cell i+1. The sending and receiving functions associated with these three cells are shown in figure 3.2. Assuming that there are no incidents in cells i-1 and i+1, a change of the regime variable in cell i does not influence the density evolution, and so the regime variable cannot be uniquely determined. In other words, if we use the fundamental diagram corresponding to the cases with one lane open, two lanes open, or all lanes open, the evolution equations will give the exact same solution for the predicted traffic density. Because the densities are the same under three different regime variables, we are not able to identify the correct regime variable γ at this timestep.

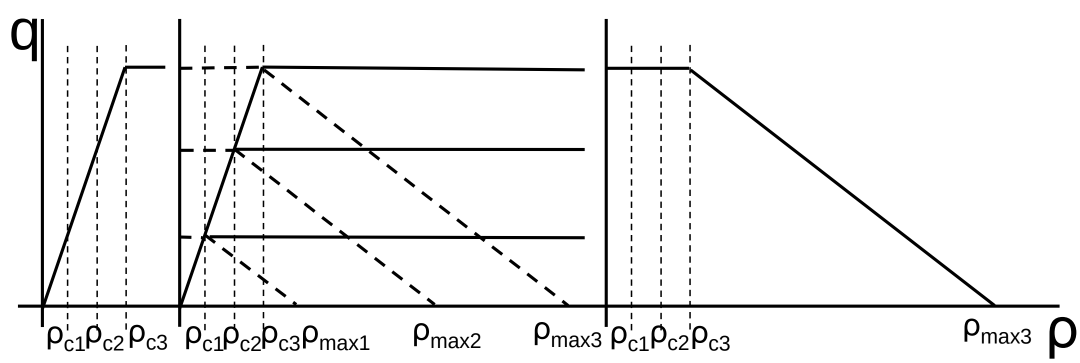


Figure 3.2: Fundamental diagrams: sending function of three lanes open at cell $i-1$ (left), sending and receiving functions of one, two, or three lanes open at cell i. Dashed lines are receiving functions and solid lines are sending functions (center), receiving function of three lanes open at cell $i+1$ (right)

The above scenario will occur whenever $\rho_{i-1} \leq \rho_{c1}$ and $\rho_i \leq \rho_{c1}$. This scenario describes the condition when the freeway is in free flow. When the traffic can be accommodated with one lane, even if one or two lanes have been blocked, there will be no influence on the measurements collected by traffic sensors. Similarly, if the density is larger than the critical density for one lane but less than the critical density for two lanes, the incident cannot be detected if only one lane is blocked. It is possible however distinguish if two or three lanes are blocked.

CHAPTER 4. JOINT STATE AND PARAMETER ESTIMATION

In this chapter, we first pose the traffic incident detection problem as a joint state and parameter estimation problem. Then, a multiple model particle filter is introduced to solve the estimation problem. The multiple model particle filter is further extended to a multiple model particle smoothing algorithm in order to improve the accuracy of the estimate.

4.1 Incident detection as a parameter estimation problem

The joint parameter and state estimation approach is applied to the problem of estimating the location and severity of traffic incidents in real time on a freeway segment using density measurements from inductive loop detectors and speed measurements from GPS equipped vehicles. The number of lanes blocked specifies the severity of the traffic incident. The traffic evolution equations are encoded in a macroscopic traffic flow model denoted by f, which evolves the traffic state x^{n-1} (i.e. a vector of densities along the roadway) at discrete time $n-1$ to x^n. Because we use an additive noise model for both the evolution and observation equation, we can rewrite (2.1) as:

$$\begin{aligned} x^n &= f(x^{n-1}, \gamma^n) + \omega^{n-1} \\ z^n &= h^n(x^n, \gamma^n) + \nu^n \end{aligned} \tag{4.1}$$

For our specific problem, z^n is the vector of speed and/or density measurements, h^n is a possibly nonlinear observation operator parameterized by the time varying regime variable γ^n. The observation noise term

$$\nu^n = \begin{bmatrix} \nu^n_{density} \\ \nu^n_{speed} \end{bmatrix}$$

is composed of two parts, $\nu_{density}$ and ν_{speed}, which allow for different errors to be placed on density and speed measurements.

Since traffic density measurements from inductive loops and speed measurements from GPS equipped probe vehicles are assumed to be available, the nonlinear operator h in (4.1) needs to be defined to match the system state to the measurements. The system state at time n is defined by vector $x^n = \left[\rho_0^n, \dots, \rho_{i_{max}}^n\right]^T$. The observation operator h is given by:

$$h^n(x^n, \gamma^n) = H^n \cdot \begin{bmatrix} x^n \\ v_{\gamma^n}(x^n) \end{bmatrix}. \tag{4.2}$$

The predicted density values and velocity values are given by the terms x^n and $v_{\gamma^n}(x^n)$. The matrix H^n is a linear operator that matches these predicted values to the measurement z^n. Note, however, that the observation operator h^n is nonlinear, due to v_{γ^n}. The matrix H^n is constructed based on the locations of where the measurements are acquired. It is time varying because the locations of GPS vehicles are not fixed, and the number of equipped vehicles may change over time.

Given the evolution observation system (4.1), the incident detection problem can be posed as the inverse problem of estimating the regime variable γ^n given measurements z up to time n. If γ^n can be uniquely determined, the location and severity of traffic incident is known. However, one of the central difficulties in incident detection is that it may not have a unique solution, and thus the traffic incident can not be detected. Thus, we formulate the problem in a Bayesian setting [52], where the solution of the problem is the posterior probability distribution of system state x^n and regime variable γ^n.

Because of the switching linear dynamics of the traffic model, we propose a multiple model particle filter to estimate the system state. In order to solve the inverse problem and improve the estimate of the regime variable γ^n, a modified version of fixed-lag smoothing algorithm is combined with the multiple model particle filter to improve the accuracy of the parameter estimate.

4.2 Particle filtering

The traffic state and parameter estimation problem is posed for the system described in (4.1). The augmented system state is defined by the vector:

$$y^n = \begin{bmatrix} x^n \\ \gamma^n \end{bmatrix}.$$

The estimation problem is solved using a Bayesian approach [52]. We compute the posterior probability $p(y^n|Z^n)$, where y^n is the augmented system state and Z^n is the measurement vector from the initial time to time n, which is defined as $Z^n = \{z^0, \dots, z^n\}$. The key quantities of interest are:

$$p(y^n|Z^{n-1}) = \int p(y^n|y^{n-1}, \Pi) p(y^{n-1}|Z^{n-1})\, dy^{n-1}, \tag{4.3}$$

$$p(y^n|Z^n) = \frac{p(z^n|y^n) p(y^n|Z^{n-1})}{p(z^n|Z^{n-1})}.$$

The first equation is the prediction step and it propagates the posterior distribution of the system state from timestep $n-1$ to n, where $p(y^{n-1}|Z^{n-1})$ is the posterior distribution at time $n-1$, and $p(y^n|y^{n-1}, \Pi)$ can be determined by the system evolution model. The parameter Π is the switching dynamic of the regime variable γ.

The second equation is the measurement processing step. The new measurement z^n is used to calculate the posterior distribution of the augmented system state y at time n, where $p(z^n|y^n)$ is the likelihood function and $p(z^n|Z^{n-1})$ is a normalizing constant. The likelihood function $p(z^n|y^n)$ indicates how well the predicted system state matches the measurement. The posterior probability distribution is proportional to

$$p(y^n|Z^n) \propto p(z^n|y^n)\, p(y^n|Z^{n-1}). \tag{4.4}$$

An analytical solution to this problem is often hard to derive, and the particle filter provides an approximate solution to this problem by using sequential Monte Carlo method. A particle filter is an appropriate choice for this problem since the traffic model is nonlinear.

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. 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 timestep. In the context of filtering, the prior refers to the estimate before measurements are obtained. After measurements of the system state 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 timestep and their previous weights. Particles with high weight 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 measurement and they will be used as input to the system evolution model for the next iteration.

The particle filter has been applied to traffic estimation problems [8,9,53], where time invariant traffic parameters are assumed. In our problem, the regime variable is time varying. Additionally, our problem involves both continuous variables (associated with the traffic state), and discrete variables (associated with the regime variable). Thus, a variant of the particle filter, known as the multiple model particle filter, is used to estimate the state and parameters.

4.3 Multiple model particle filter

4.3.1 Multiple model particle filter algorithm

The multiple model particle filter [51] is proposed as an extension to the previous particle filter approaches [8,9,53]. The main difference between the multiple model particle filter and the standard particle filter algorithm is that the multiple model particle filter allows the system to have several modes, and particles are generated for each system mode. It has a regime transition step that describes the switching dynamics of the system mode. The regime transition of parameter γ is specified by the transition matrix Π. In the incident detection problem, the modes of the system are the regime variables γ. The idea of the multiple model particle filter is that if the system state x^n generated by a regime variable γ matches well with the measurements, then we believe the system is operating on mode γ at time n.

Algorithm 1 Multiple model particle filter

Initialization (n=0): generate M samples y_l^0 and assign equal weights $w_l^0 = 1/M$, where $l = 1, \ldots, M$

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

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

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

Measurement processing:

calculate the likelihood: $p(z^n | y_l^n)$ for all l

update weight: w_l^n=$w_l^{n-1} p(z^n | y_l^n)$ for all l

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

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

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

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

$n = n + 1$

end for

The pseudo code of multiple model particle filter for incident detection is summarized in algorithm 1. The initial state y^0, which is composed of x^0 and γ^0, is given by an initial distribution reflecting our knowledge on the initial state. Measurements from inductive loop detectors and GPS vehicles are assumed to be available at each timestep n.

- Initialization: Generate M particles from the initial distribution of y^0 and assign each particle with equal weight. The notation l is used to denote particles.
- Regime transition: Calculate the regime variable for all particles according to the parameter evolution equations (3.8) and (3.9).
- Prediction: Calculate the prior distribution of the system state x^n according to the traffic model f.
- 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. Resampling is applied to avoid the degeneracy problem in the particle filter. A detailed algorithm and introduction for resampling can be found in [51].
- Output: The solution to this problem is a posterior distribution of augmented system state y^n, which is composed of the posterior distribution of x^n and γ^n. If the distribution of regime variable γ^n takes a unique value at all timesteps 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 the distribution reflects the fact that multiple locations and/or severities of incidents are consistent with the observed data.

4.3.2 Discussion on multiple model particle filter

This multiple model particle filter method will work well when traffic sensors are dense. However, this approach will fail if the number of sensors is limited. When a traffic incident occurs and there is no traffic sensor nearby, it will take time for the information to propagate to the nearest sensor. Consequently, the correct regime variable can not be identified at the time when the traffic incident occurs, and the particles generated by wrong regime variables will be assigned with high weights. These wrong particles will then be used as input to calculate the prior distribution for the next timestep. If the correct regime variable could not be identified for a few timesteps, more and more particles remaining in the sample will become incorrect. Eventually, all the predicted particles will not match the measurements. To address this problem, we apply the idea of fixed-lag smoothing and combine it with this multiple model particle filter algorithm.

4.4 Multiple model particle smoothing

4.4.1 Fixed-lag smoothing algorithm

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 the later time will help to provide a better estimation of the current system state. The fixed-lag approximation [54, 55] is described by:

$$p(Y^n|Z^T) \approx p\left(Y^n\middle|Z^{\min(n+\Delta S,T)}\right), \tag{4.5}$$

where $Y^n = \{y^0, \dots, y^n\}$. In general, $n + \Delta S$ is smaller than T. The assumption for this approximation is that the measurement after time $n + \Delta S$ brings no additional information about the state Y^n.

In the traffic incident detection problem, the objective is to identify the correct regime variable γ^n at each timestep. We applied the idea of fixed-lag smoothing as follows: if the traffic state generated by a regime variable γ^n has a high weight for timesteps $1 + \Delta S$, then we believe the regime variable γ^n is a consistent regime at time n. By applying this idea, the regime variable γ^n is identified by its performance at ΔS timesteps in the future beyond time n. Another way to look at this is that additional ΔS timesteps are allowed to let the traffic information propagate to the nearest sensor.

4.4.2 Multiple model particle smoothing

The fixed-lag smoothing algorithm is combined with the multiple model particle filter. The resulting multiple model particle smoothing algorithm is described in algorithm 2.

The main difference between the multiple model particle smoothing algorithm and the multiple model particle filter algorithm is the measurement processing stage. In the multiple model particle filtering algorithm, the weight of each particle is determined by its previous weight and its likelihood calculated at the current timestep. In the multiple model particle smoothing algorithm, the weight of each particle is determined by its previous weight and the likelihood values calculated during $1 + \Delta S$ time periods. During smoothing, each particle is evolved forward in time, and it is assumed there is no regime transition. Thus, the system state x^n generated by a regime variable γ^n is evaluated for $1 + \Delta S$ timesteps. Again the benefit of these additional ΔS steps is that it effectively allows the traffic information (specifically anomalies that indicate an incident has occurred) to propagate to the nearest sensor, so the change can be detected.

Algorithm 2 Multiple model particle smoothing algorithm for traffic incident detection

Initialization (n=0): generate M samples y_l^0 and assign equal weights $w_l^0 = 1/M$, where $l = 1,\ldots,M$

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

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

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

Measurement processing and smoothing:

trigger=0

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

if trigger=0: $\bar{w}_l^n = w_l^n(t-1)$ for all l, with $w_l^n(0) = 1/M$

calculate the likelihood:

$p(z^{n+t-1}|x_l^n(t), r_l^n)$, with $x_l^n(1) = x_l^n$ for all l

update weight:

$w_l^n(t)$=$w_l^n(t-1)p(z^{n+t-1}|x_l^n(t), \gamma_l^n)$ for all l

normalize weight:

$\hat{w}_l^n(t) = w_l^n(t)/\sum_{l=1}^{M} w_l^n(t)$ for all l

$w_l^n(t) = \hat{w}_l^n(t)$ for all l

if $t \neq \Delta S + 1$: $x_l^n(t+1) = f(x_l^n(t), \gamma_l^n, \omega^n)$

if $\Sigma w_l^n(t) \leq \varepsilon$: trigger=1

if trigger=1: $w_l^n(\Delta S + 1) = \bar{w}_l^n$ for all l

end for

Resampling: multiply/ suppress samples y_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

One more issue must be addressed to make the multiple model particle smoother work in practice. When the true regime of the freeway changes between the time period n to $n + \Delta S$, particles generated by the regime variable γ^n will not match well with the measurements for the entire smoothing period ΔS. This is because we assume a constant γ during smoothing, however, the true regime of the freeway must necessarily change during gone of the smoothing periods when an incident occurs. To address this issue, we evaluate the magnitude of the weight of each particle during time period ΔS. If the sum of the weight for all particles is less than a predefined threshold ϵ, we believe there is a change of the true regime on the freeway. Then, we update the weight of each particle by using its weight from previous timestep that has an acceptable magnitude and the algorithm advances. This is practically achieved through the *trigger* variable in algorithm 2.

4.4.3 Influence of the smoothing window

The multiple model particle smoothing algorithm estimates the regime variable γ^n by using measurement from ΔS more timesteps in the future. 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 bigger value for ΔS is needed. Obviously, there is a price for the improvement of estimate accuracy. Instead of a real-time traffic incident detection, the multiple model particle smoothing algorithm practically estimates traffic with a lag of $\Delta S \cdot \Delta T$.

CHAPTER 5. MACROSCOPIC NUMERICAL IMPLEMENTATION

In this chapter, the proposed algorithms are tested by using synthetic incident data generated by the discretized traffic model with different GPS penetration rates and two inductive loop detectors. The same macroscopic traffic flow model is used to generate the synthetic measurements, and to evolve the traffic state in the filtering and smoothing algorithms. Filtering and smoothing are performed to show how smoothing can improve the joint state and parameter estimate.

5.1 Problem description

In this simulation, we study a four lane freeway segment that is 10 miles long. The freeway is discretized into 25 cells, and the length of each cell is 0.4 miles. Inductive loop detectors are assumed available at cell one and cell 23, as shown in figure 5.1. GPS equipped vehicles are assumed to traverse the freeway and periodically send speed measurements. The penetration rate of GPS vehicles is specified indirectly by the headway between vehicles entering the stretch of roadway.

In this study, we compare the accuracy of the estimates when the headway between GPS vehicles is 140 seconds and 180 seconds. We assume the traffic incident may occur anywhere between cell two to cell 22 with three severities: one lane blocked, two lanes blocked, and three lanes blocked. The objective is to accurately estimate the density and the location and severity of the traffic incident in real-time.

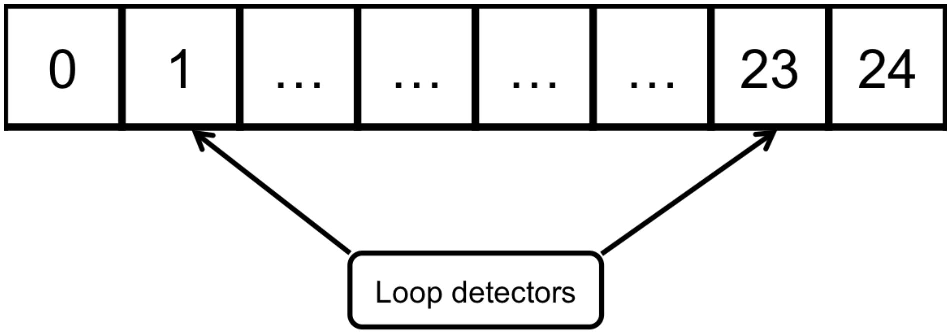


Figure 5.1: Freeway discretization and sensor positions

5.1.1 Parameters of the macroscopic traffic model

The parameters used for the discretized LWR model are summarized in table 5.1. We set the timestep ΔT at 20 seconds, which also defines the time interval that we inspect the occurrence of an incident. Measurements are also assumed to be available at every timestep. The traffic parameters v_{max} and w_f are assumed fixed throughout the simulation, and the parameter q_{max}, representing the maximum flow for a single lane, is also constant per lane.

Link length	10 miles
Number of cells	25
ΔT	20 seconds
Δx	0.4 miles
CFL	0.97
v_{max}	70 mile/hour
w_f	30 mile/hour
q_{max}	2000 veh/hour/lane
ρ^0	$\mathcal{N}(114.28, 2.28^2)$ veh/mile
ρ_r	0 veh/mile

Table 5.1: Parameters used in the macroscopic traffic simulation

The main parameter to be jointly estimated with the state is the regime variable, indicating the number of open lanes. We assume the total capacity of the four lane freeway is 8000 veh/hour. When an incident occurs, the capacity may drop to 6000, 4000, or 2000 veh/hour depending on the severity of the incident. The values of critical density and jam density for the segment of freeway (veh/mile) can be calculated based on parameters v_{max}, w_f, q_{max} and the number of lanes open as indicated through the regime variable. The initial traffic density for all cells are assumed constant and equal to the critical density, and in the estimator it is specified similarly but with a normal distribution to reflect uncertainty on the estimate.

5.1.2 Descriptions of the noise models

The model noise ω on the discretized traffic model is assumed to be an additive noise. The noise model for the measurements are specified by two categories. One noise model is given for density measurements, and a second noise model is given for speed measurements. In this numerical implementation, all of the noise models are specified by a Gaussian distribution, however, other types of distributions are applicable since particle filtering is able to handle non-Gaussian noise. The noise models used in this numerical implementation are summarized in table 5.2.

Noise for traffic density prediction $\boldsymbol{\omega}$	$\mathcal{N}(0, 0.61^2)$
Noise for measurement v_{density}	$\mathcal{N}(0, 1.5^2)$
Noise for measurement v_{speed}	$\mathcal{N}(0, 2.69^2)$

Table 5.2: Noise models in macroscopic simulation

5.1.3 Assumptions for parameter evolution equations

We make several assumptions on the evolution of the regime variable. First, we assume there is at most one traffic incident on the freeway at the same time. Second, there is a ten percent probability for the occurrence of a traffic incident at next timestep, 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 cell two to cell 23, with three possible severities. Third, if there is an incident on the freeway at the current timestep, there is a 90 percent probability for the incident to remain into the next timestep. Finally, there is a 10 percent probability for the incident to switch to the other two severities, or to be cleared (3.3% each). With these assumptions, the transitional matrix Π can be constructed.

In this work, we assume a relatively high probability for the occurrence of a traffic incident. This is because in the multiple model particle filtering algorithm, the number of particles in each mode is proportional to the transition probability for each regime. Consequently, we need to assume a relatively high transition probability to incident in order to get enough particles in each mode. If we assume a lower probability of an incident, a larger sample size may be needed.

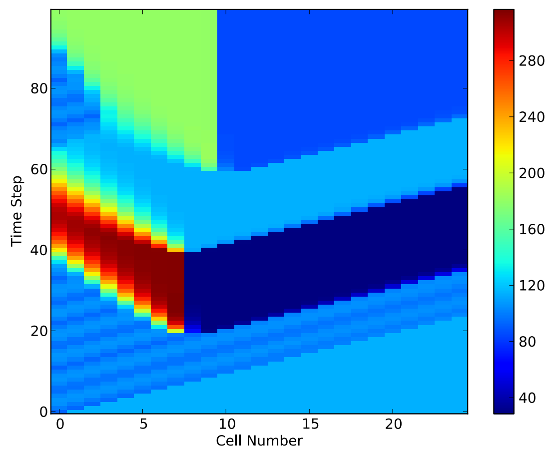
5.1.4 Generation of synthetic GPS data

In this work, the penetration rate of GPS vehicles is specified by the headway between vehicles. To obtain measurements from GPS vehicles, the trajectory of the j^{th} GPS vehicles χ^{jth} is modeled according to:

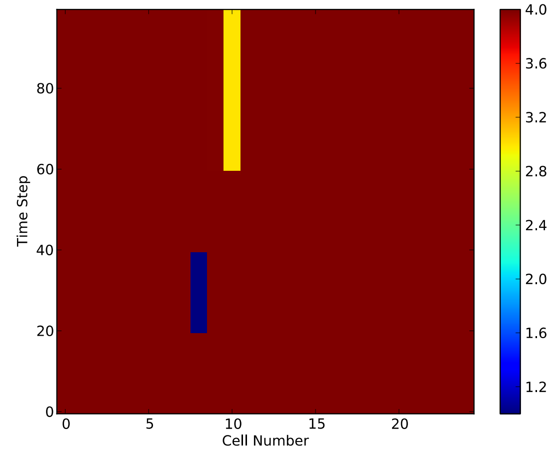
$$\dot{\chi_J}\,(t) = v_\gamma\left(\rho\big(\chi_j(t), t\big)\right), \tag{5.1}$$

where $\chi_j(t)$ denotes the location of the j^{th} vehicle at time t. We solve this ordinary differential equation by integrating both sides over ΔT. Following the discretization by the Godunov scheme [49], the velocity and density ρ is constant within each cell, so the integration is trivial to compute.

For each timestep ΔT, the GPS vehicle may travel in at most two consecutive cells. This occurs when the vehicle is located near the end of a cell at the start of the timestep. When this happens, the solution can be obtained by integrating two piecewise constant velocity functions over the corresponding time periods that the GPS vehicle stays in each cell.



(a) Density



(b) Number of lanes open

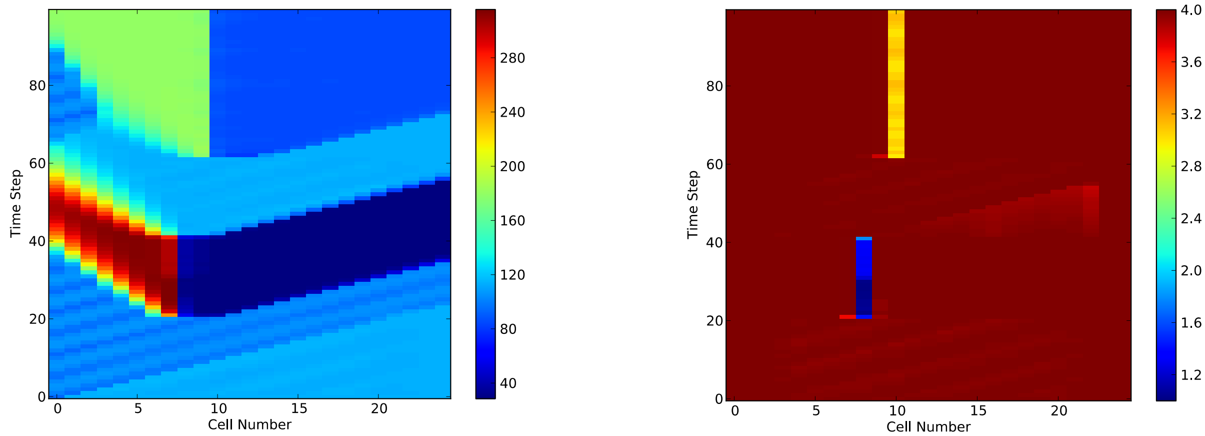
Figure 5.2: True evolution of the traffic density and the regime parameter

5.1.5 Simulation description

The sample size for this simulation is 5000 particles and the time duration is 100 timesteps. Since each timestep is 20 seconds, about 30 minutes of traffic is simulated. The left boundary condition is specified by an inflow, and it is generated by a sine function. This is to allow some variations for the inflow traffic. Specifically, the inflow q_{in}^n is specified as:

$$q_{in}^n = 6500 + 1000 \cdot |\sin(10n)|.$$

We create two incidents in the time domain during the simulation. The true evolution of the density ρ and number of lanes open throughout time and space are shown in figure 5.2. When the inflow is high and an incident occurs, a shock wave that propagates backwards can be observed. The density upstream from the incident location depends on the number of lanes the incident blocks.



(a) Density

(b) Number of lanes open

Figure 5.3: Estimate of the multiple model particle filter, probe vehicle headway 140s

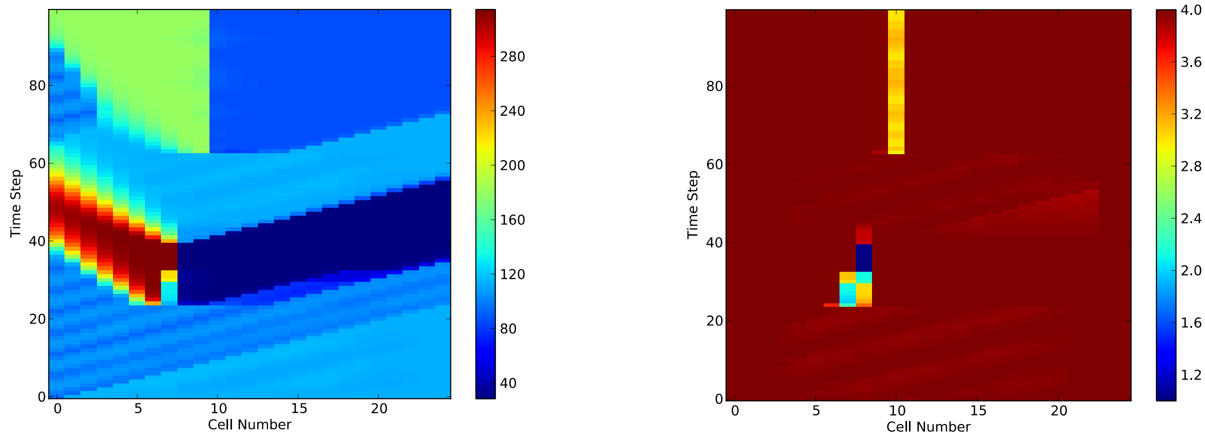
5.2 Estimation results

5.2.1 Estimation without smoothing

The estimation results for the multiple model particle filter (without smoothing) are shown in Figure 5.3 and 5.4.

Figure 5.3 shows the results when the headway of GPS vehicle is 140 seconds. The plots 5.4a and 5.4b describe the density and number of lanes open along the time and space domain respectively. The mean of the posterior distribution is plotted.

If we compare the results with the true solution shown in figure 5.2, we conclude the multiple model particle filter algorithm performs well in estimating both the traffic density evolution and the regime variable. However the regime variable is not perfectly recovered for all samples. The light red area at around cell 22 at time 45 is due to the fact that some samples of the regime variable show less than four lanes are open. Similarly, the light blue area during the first simulated incident denotes that some samples do not completely capture the incident.



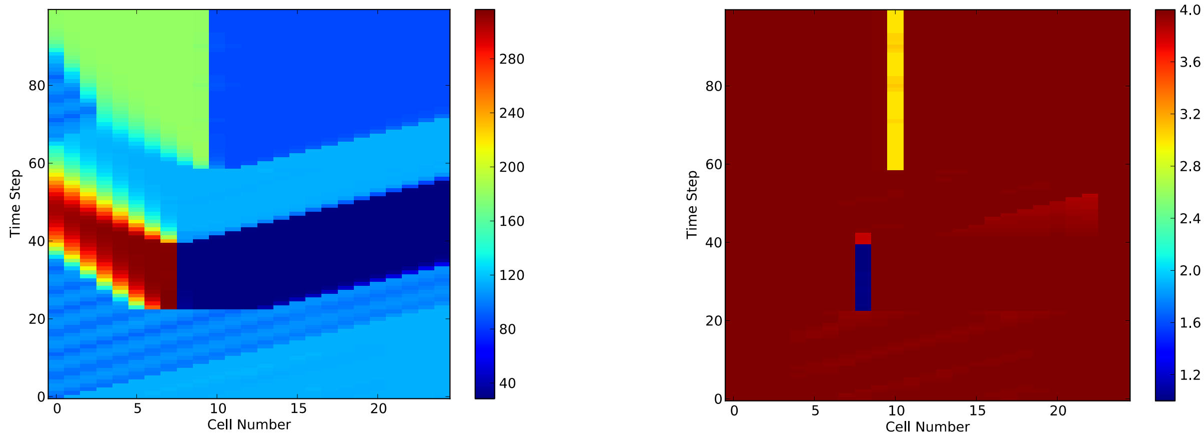
(a) Density

(b) Number of lanes open

Figure 5.4: Estimate of the multiple model particle filter, probe vehicle headway 180s, no smoothing

Figure 5.4 shows the results when the headway of the probe data is increased to 180 seconds. It can be seen that both the density and the regime variable estimate begins to deteriorate, because of the information propagation issue discussed in section 4.3.2.

If the penetration rate is further reduced, the multiple model particle filter may completely fail. This will happen because at each timestep the algorithm use all possible regimes to predict traffic state for the next timestep, since the measurements are not able to help the algorithm to correctly identify the true regime at the current timestep. Thus, the filter keeps a number of samples with the a regime variable that is different from the true model. If this occurs for too many consecutive timesteps most of the samples will end up in the wrong regime. Then, when the sensors begin to detect an incident, there might not be many (or any) samples with the correct regime, and all of the particles with the wrong regime will be discarded. If the penetration rate continues to decrease, the algorithm may completely fail to detect the incident or produce correct traffic estimates.



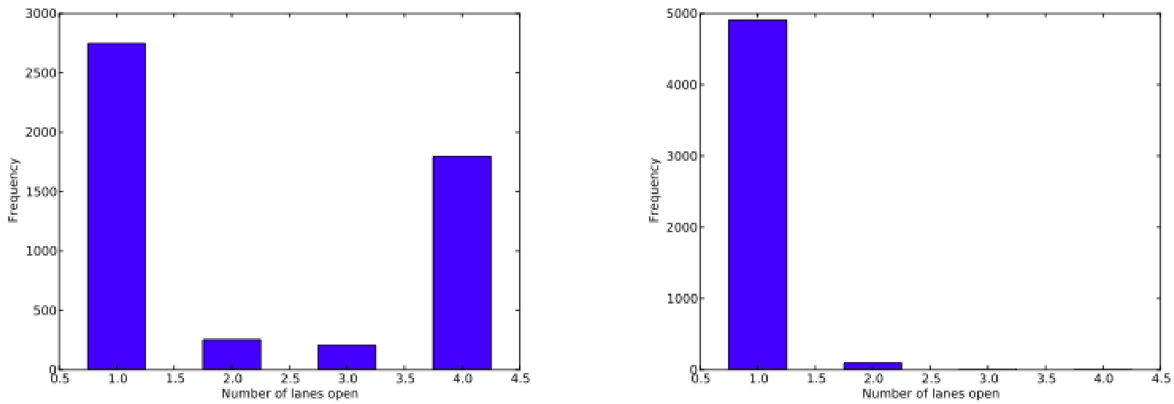
(a) Density

(b) Number of lanes open

Figure 5.5: Estimate of the multiple model particle smoother, probe vehicle headway 180s, with smoothing $\Delta S = 2$

5.2.2 Estimation with smoothing

When the headway of GPS vehicles is 180 seconds, the simulation results for multiple model particle smoothing are shown in figure 5.5. The ΔS is set as 2. Compared to figure 5.4, smoothing shows a significant improvement for the accuracy of the parameter estimate. In figure 5.6, the posterior distribution of γ is compared at a specific point in time and space, to highlight the significant improvement for the accuracy of the parameter estimates when smoothing is used. Thus, smoothing is able to improve the accuracy of the parameter estimate by using the measurements both behind and in front of the timestep at which the estimate is produced. However, an extra waiting time of ΔS timesteps is required to obtain the future measurements.



(a) Distribution of γ at cell 8, timestep 30, no smoothing

(b) Distribution of γ at cell 8, timestep 30, with smoothing

Figure 5.6: Comparison of the distributions of γ without and with smoothing

CHAPTER 6. MICROSCOPIC NUMERICAL SIMULATION

To test whether the proposed algorithm has a potential to work in practice, a microscopic simulation software CORSIM is used to simulate a traffic incident on a two-lane freeway segment. The simulation results from CORSIM will be 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 claim is that if the algorithms are able to detect the traffic incident by using the data generated by CORSIM, which is an entirely different modeling framework from the model used in the estimator, it has a higher potential to perform well in the field.

6.1 Fundamental diagram for CORSIM

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 behaviors, vehicle, and traffic system behaviors. This is in contrast to the discretized LWR model used in the estimator, which models only conservation of the vehicles and a relationship between average speed and density.

To test the proposed algorithm with CORSIM, the fundamental diagram in the macroscopic LWR model needs to be calibrated to represent the traffic evolution of CORSIM. In particular, the shape of the fundamental diagram needs to be determined. To calibrate the model, we run a number of simulations and plot the speed-density relationship observed on the roadway. Similarly, the density-flow relationship can be constructed. The blue dots in Figure 6.1 show the resulting fundamental diagram obtained from our CORSIM simulation.

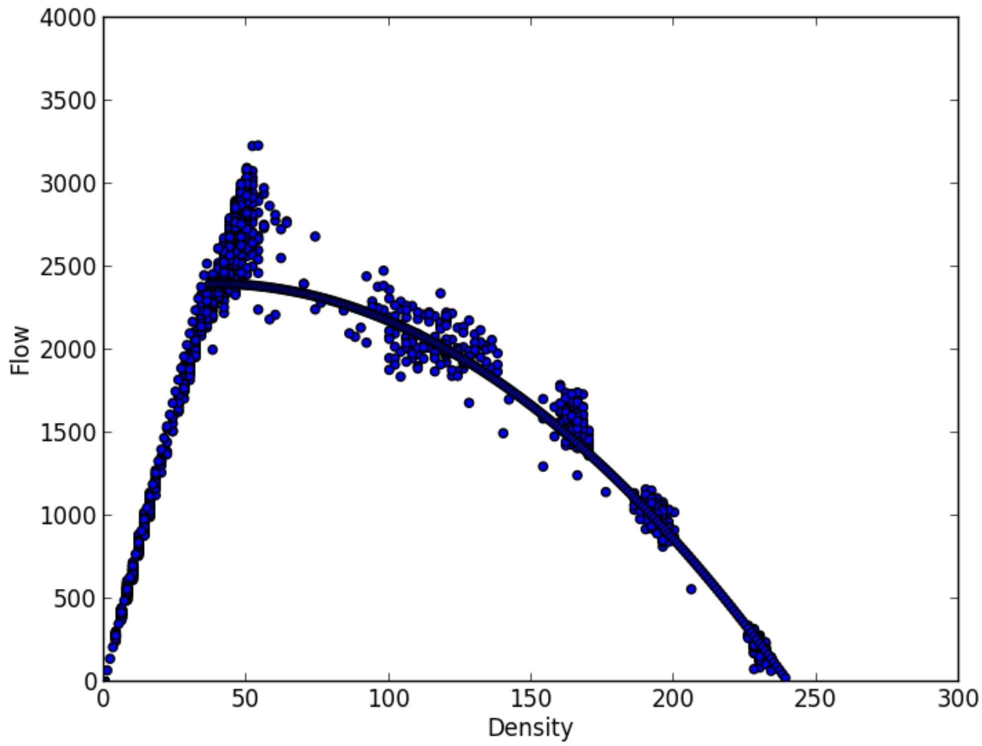


Figure 6.1: Fundamental diagram. Measurements obtained from CORSIM (blue dots) and calibrated fundamental diagram (solid black line)

To derive the mathematical expression of the density-flow relationship, a linear function is used to fit the data when the density is smaller than the critical density, and a quadratic function is used when the density is greater than the critical density. This is given by:

$$q(\rho) = \begin{cases} v_{max}\rho & if\ \rho \le \rho_c \\ a\,\rho^2 + b\rho + c & otherwise \end{cases} \tag{6.1}$$

The parameters a, b and c can be calculated based on the traffic parameters in table 6.1, under the following assumptions. When there are two lanes, the maximum speed is the same as one lane, but the maximum flow and jam density are doubled. The parameters associated with the quadratic function are calculated and summarized in table 6.2.

v_{max}	65 mph
ρ_J	240 veh/mile/lane
q_{max}	2400 veh/h/lane

Table 6.1: Summary of parameters for the CORSIM fundamental diagram

Number of lanes	a	b	c
1 lane	-0.0588	4.47	2315.07
2 lanes	-0.0291	4.31	4640.53

Table 6.2: Resulting parameters used in (6.1)

6.1.1 Parameters of the macroscopic traffic model

The parameters used for the cell transmission model within the estimation algorithms are summarized in table 6.3. The density value in the first cell is used as the left boundary condition of the model, and it is obtained from the density measurements CORSIM. The initial density condition in all cells are also known from the initial density condition From CORSIM. In CORSIM, the simulation starts after a warm-up period, so the initial density values are not zero.

Link length	10 miles
Number of cells	11
ΔT	20 seconds
Δx	0.36 miles
CFL	0.99
v_{max}	65 mile/hour
q_{max}	2400 veh/hour/lane
ρ_J	240 veh/mile/lane
ρ_r	0 veh/mile

Table 6.3: Setup for the macroscopic model

6.1.2 Selections of noise model

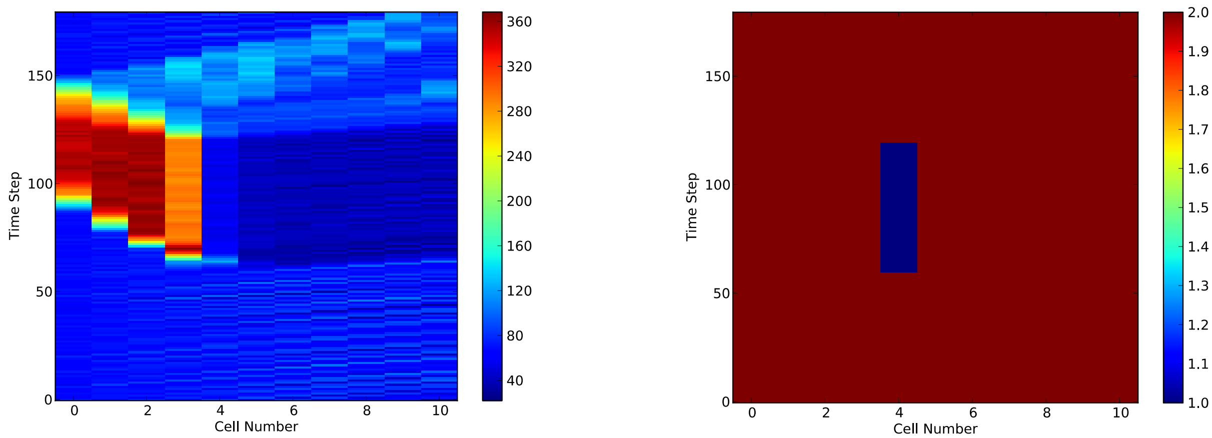
The types of distributions of the noise models are the same as those in the macroscopic experiments in Chapter 5. In this microscopic experiment, only speed measurements are used to infer the incident, so only the velocity noise model is specified. The noise models are calibrated and summarized in table 6.4.

Noise for traffic density prediction ω	$\mathcal{N}(0, 1.17^2)$
Noise for measurement v_{speed}	$\mathcal{N}(0, 4.90^2)$

Table 6.4: Noise models in macroscopic simulation

6.1.3 Simulation description

The CORSIM simulation is performed on a two-lane freeway segment. The inflow for CORSIM is set as 7000 veh/hour. A total of 180 timesteps are simulated, and since each timestep is 20 seconds, the total simulation length is one hour. In CORSIM, one incident is created at cell four between timesteps 60 and 120. The density ρ and number of lanes open in the time and space generated by CORSIM is shown in figure 6.2. The assumptions for the parameter evolution equations are the same as the macroscopic simulation from Chapter 5. The only difference is that this simulation is performed on a two-lane freeway instead of a four-lane freeway. For the proposed multiple model particle filtering and smoothing algorithms, the sample size is set to 1000.

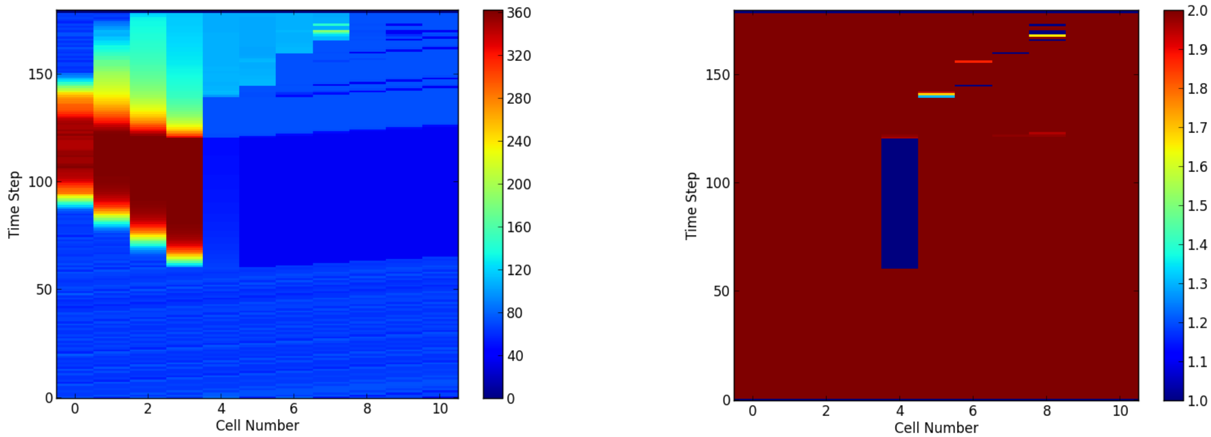


(a) Density

(b) Number of lanes open

Figure 6.2: True evolution of the traffic density and the regime parameter

The true macroscopic density field is computed from CORSIM and plotted in Figure 6.2. If we qualitatively compare figure 6.2 and figure 5.2, it is observed there is much more variation in the traffic density in the CORSIM simulation, while the simulation generated by the from the cell transmission model is more homogeneous. Consequently, the result from figure 5.2 is closer to a theoretical scenario while the result from figure 6.2 is probably more representative of true traffic, since the density is calculated from the individual vehicle trajectories, not modeled directly. This increases the error on the model assumed in the estimator, and explains the claim that if the proposed algorithm is able to detect the incident generated by CORSIM simulation, it has a higher potential to work in practice compared to if it only works in a macroscopic experiment.



(a) Density

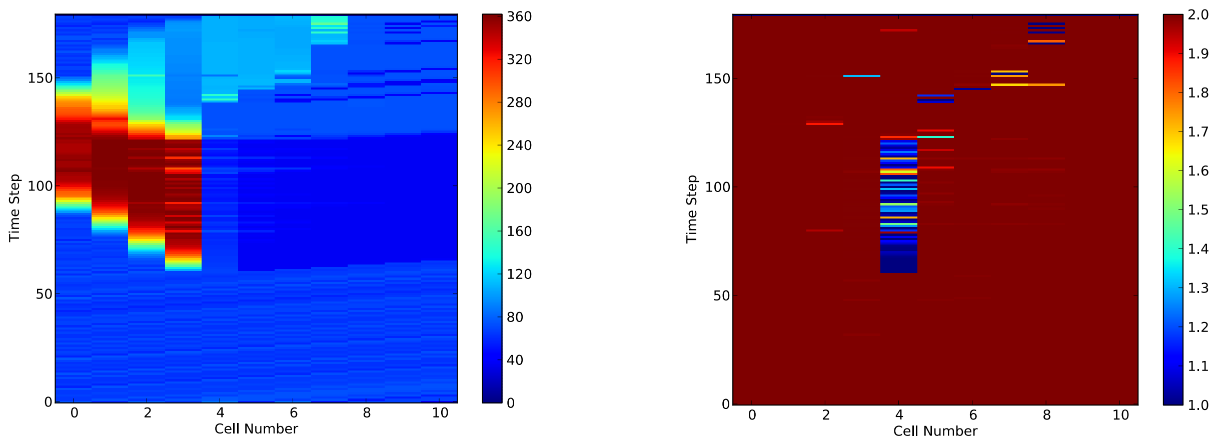
(b) Number of lanes open

Figure 6.3: Estimate of the multiple model particle filter with full penetration of GPS equipped vehicles

6.2 Estimation results

The algorithm is first tested by assuming a full penetration rate (i.e. every vehicle sends speed measurements), to get an upper bound on the performance of the estimator when the data is generated from a microscopic model. The simulation is conducted with multiple model particle filtering only, no smoothing is performed, and the results are shown in figure 6.3.

The results show that the density estimates are close to the true density values simulated by CORSIM. However, the estimate still has some error. This is expected because the estimator uses the LWR model as a prior, and weights the measurements with the prediction made by the macroscopic model. The results are necessarily worse than what would be expected if the true state and measurements are generated from the same macroscopic model. When the true traffic evolves according to the exact same model assumed in the estimator, which is never true in practice, the estimator can produce overly optimistic results. In contrast, this microscopic simulation test the algorithm by assuming the CORSIM traffic is the true state, and we calibrate our model to fit the traffic simulated CORSIM as best as possible. Consequently, the estimation is not as actuate as the macroscopic simulation, but it is a more reasonable approach to evaluate whether the proposed algorithm will work in practice.

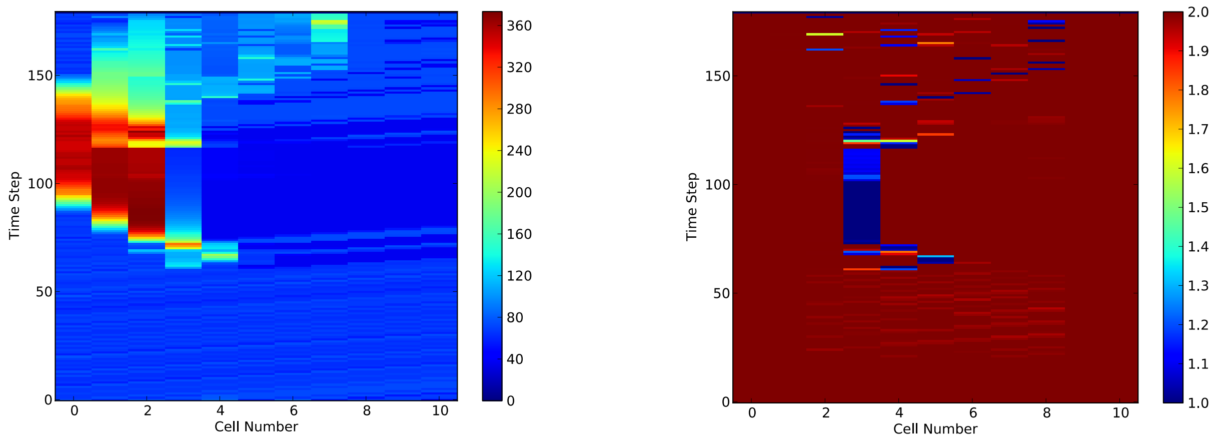


(a) Density

(b) Number of lanes open

Figure 6.4: Estimate of the multiple model particle filter, probe vehicle headway 20s

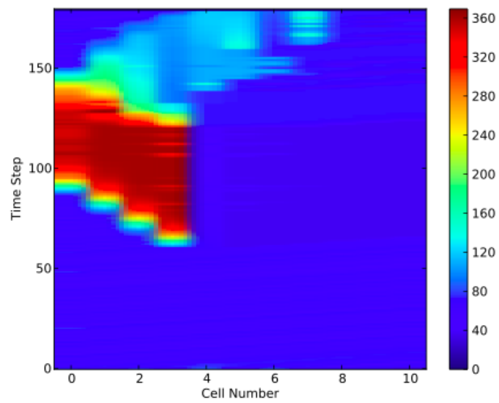
Next, a lower penetration rate of GPS vehicles is used to test the algorithm. When the headway between GPS vehicles is set as 20 and 60 seconds, the estimation results without smoothing are shown in figure 6.4 and 6.5 respectively. As the result shows, when the headway is 20s, the algorithm is able to correctly estimate the regime variable. When the headway is increased to 60s, the location the traffic incident is off by one cell. One reason for this is because when the headway is high, there are fewer sensors on the freeway, and there is not enough information for the algorithm to estimate the parameter with good accuracy. Another reason is due to the lane changing logic in CORSIM, which does not exist in the LWR model. In CORSIM, vehicles will switch lanes when they see a traffic incident ahead. Consequently, the density right before the incident in the blocked lane may be very low, due to vehicles waiting upstream to merge. In contrast, the LWR model is a single pipe model, where it is implicitly assumed the density is uniformly spread across the two lanes.



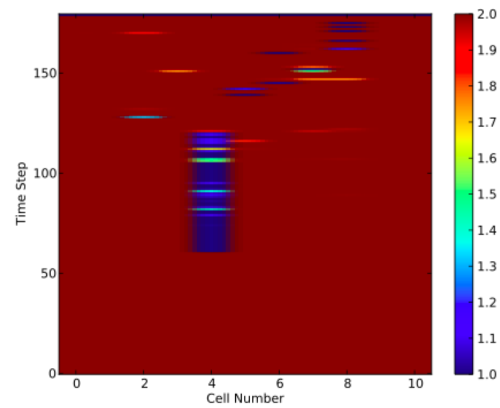
(a) Density (b) Number of lanes open

Figure 6.5: Estimate of the multiple model particle filter, probe vehicle headway 60s

The results for the multiple model particle smoothing algorithm are shown in figures 6.6 and 6.7. Figure 6.6 shows the simulation results when the headway between GPS vehicles is 20 seconds and figure 6.7 shows the results when the headway is 60 seconds. The ΔS for these two simulations are separately set as one and two. Compared to figures 6.4 and 6.5, the accuracy of both the state and parameter estimates improve with smoothing when the headways are large. Thus, when the penetration rates of probe vehicles are low, smoothing might be a meaningful way to improve estimates, without the need for additional probe data. The increased accuracy comes at the cost of a lag in the estimate, however.



(a) Density



(b) Number of lanes open

Figure 6.6: Estimate of the multiple model particle smoother, probe vehicle headway 20s, with smoothing, $\Delta S = 1$

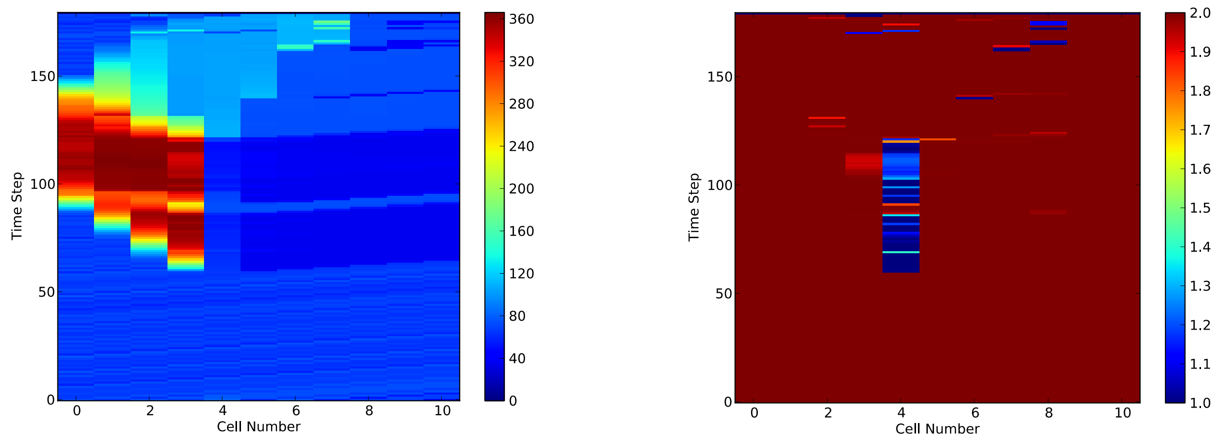
The estimation accuracy of the state vector x^n is quantitatively evaluated by computing the average error as follows:

$$error = \frac{1}{(i_{max}+1)(n_{max}+1)} \sum_{i=0}^{i_{max}} \sum_{n=0}^{n_{max}} \left\{ \frac{|\hat{\rho}_i^n - \rho_i^n|}{\rho_i^n} \right\}$$

where $\hat{\rho}_i^n$ is the estimated density and ρ_i^n is the true density at each time n and location i. The results are shown in table 6.5.

Error (density/mile/lane)	filter	smoother
20 sec headway	7.42	7.50
60 sec headway	11.71	8.26

Table 6.5: Error for density estimation



(a) Density (b) Number of lanes open

Figure 6.7: Estimate of the multiple model particle smoother, probe vehicle headway 60s, with smoothing, $\Delta S = 2$

We conclude that the proposed multiple model particle smoothing algorithm is able to detect the location and severity of traffic incident with synthetic traffic incident data generated by CORSIM even when the penetration rate decreases. At high penetration rates, smoothing and filtering produce similar estimates.

CHAPTER 7. CONCLUSIONS

This work developed two algorithms to jointly estimate the traffic state and the traffic parameters in an online setting. The multiple model particle filter and the multiple model particle smoother are proposed, and tested on the problem of detecting incidents as a benchmark problem.

This work also showed that the incident detection problem can be posed as a joint state and parameter estimation problem. A regime variable is used to model and location and severity of traffic incident, and it is embedded in the traffic flow model.

The simultaneous solving for the parameters and the traffic state has several advantages. First, knowledge of the incident can improve the quality of the traffic state estimate. Second, knowledge of the traffic state can improve the ability to detect incidents. Instead of separately solving for incidents and the traffic state, the two can be solved simultaneously using the algorithms developed in this work.

The algorithms developed in this work were tested in two numerical environments, including a microscopic and a macroscopic setting. The microscopic experiments were performed by simulating incidents in CORSIM, and highlight that the multiple model particle filter and smoother can achieve good performance, even using a simplified macroscopic model for the state evolution equations. It was also discovered that a smoothing algorithm with low probe data rates can perform similarly to a filtering algorithm with several times as much data. This is powerful because a small change in the algorithm design can effectively replace data, allowing the algorithms to be deployed even when probe data rates are low, or fixed sensors are sparse.

Several areas are open for future exploration. First, the output of the two algorithms is a posterior distribution on the regime variable, which indicates the number of open lanes. When the mass is not uniquely centered on one integer, several approaches can be used to transform the distribution into a best estimate of the number of open lanes. This investigation would be absolutely essential in order to get good practical performance in the field, without producing too many false positives or too many missed incidents.

Additionally, while this work laid the foundation for the design of the algorithms, the performance of the algorithms was only explored numerically. The synthetic example using data from a macroscopic model allows one to define an upper bound on the performance of the algorithm, and the CORSIM experiments demonstrate that the algorithm would have potential to work well in the field. Still, CORSIM is itself a simplification of true traffic, so additional testing with field data is needed.

Finally, the scalability of the algorithms should be investigated. Because the particle filter has significantly larger computational requirements compared to techniques like ensemble Kalman filtering, localized estimation may be necessary on large freeway networks.

References

[1] M. J. Lighthill and G. B. Whitham, “On kinematic waves II. A theory of traffic flow on long crowded roads,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[2] P. Richards, “Shock waves on the highway,” *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[3] C. F. Daganzo, “The cell transmission model, part II: network traffic,” *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.

[4] —, “The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory,” *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[5] Y. Wang and M. Papageorgiou, “Real-time freeway traffic state estimation based on extended Kalman filter: a general approach,” *Transportation Research Part B: Methodological*, vol. 39, no. 2, pp. 141–167, 2005.

[6] D. Work, O.-P. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, 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*, 2008, pp. 2141–2147.

[7] D. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. Bayen, “A traffic model for velocity data assimilation,” *Applied Mathematics Research Express*, vol. 2010, no. 1, pp. 1–35, 2010.

[8] L. Mihaylova, R. Boel, and A. Hegyi, “Freeway traffic estimation within recursive Bayesian framework,” *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.

[9] L. Mihaylova and R. Boel, “A particle filter for freeway traffic estimation,” in

Proceedings of the 43rd IEEE Conference on Decision and Control, 2004, pp. 2106–2111.

[10] "Mobile Millennium," http://traffic.berkeley.edu/.

[11] R. A. Reiss and W. M. Dunn, "Freeway incident management handbook," Federal Highway Administration, Tech. Rep. FHWA-SA-97-064, 1991.

[12] NHTSA, "National motor vehicle crash causation survey," U.S. Department of Transportation, Tech. Rep. DOT HS 811059, 2008.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[14] M. Szeto and D. Gazis, "Application of Kalman filtering to the surveillance and control of traffic systems." *Transportation Science,* vol. 6, no. 4, pp. 419–439, 1972.

[15] M. Cremer and M. Papageorgiou, "Parameter identification for a traffic flow model," *Automatica*, vol. 17, no. 6, pp. 837–843, 1981.

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

[17] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo, "An adaptive freeway traffic state estimator," *Automatica*, vol. 45, no. 1, pp. 10–24, 2009.

[18] X. Sun, L. Munoz, and R. Horowitz, "Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application," in *Proceedings of the American Control Conference*, vol. 3, 2004, pp. 2098 – 2103.

[19] J.-C. Herrera and A. Bayen, "Traffic flow reconstruction using mobile sensors and loop detector data," in 87th *Transportation Research Board Annual Meeting*, 2008.

[20] J.-C. Herrera and A. Bayen, "Incorporation of Lagrangian measurements in freeway traffic state estimation," *Transportation Research Part B: Methodological*, vol. 44, no. 4, pp. 460– 481, 2010.

[21] S. J. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[22] J. Sau, N. El Faouzi, A. Ben Assa, and O. De Mouzon, "Particle filter-based real-time esti- mation and prediction of traffic conditions," in *Proceedings of the 12th International Conference on Applied Stochastic Models and Data Analysis*, 2007.

[23] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, pp. 4629–4640, 2008.

[24] 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*, 2005, pp. 2164–2169.

[25] 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*, 2006, pp. 5213–5218.

[26] 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*, vol. 55, no. 5, pp. 1142–1157, 2010.

[27] —, "Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part II: Computational methods," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1158–1174, 2010.

[28] E. Parkany and C. Xie, "A complete review of incident detection algorithms & their deployment: what works and what doesn't," University of Massachusetts

Transportation Center, Tech. Rep. NETCR37, 2005.

[29] H. S. Mahmassani, C. Haas, S. Zhou, and J. Peterman, “Evaluation of incident detection methodologies,” Center for Transportation Research at The University of Texas at Austin, Tech. Rep. FHWA/TX–00/1795–1, 1998.

[30] K. N. Balke, “An evaluation of existing incident detection algorithms,” Texas A&M University, Texas Transportation Institute, Tech. Rep. FHWA/TX–93/1232–20, Nov 1993.

[31] S. C. Tignor and H. Payne, “Improved freeway incident detection algorithms,” *Public Roads,* vol. 41, no. 1, pp. 32–40, 1977.

[32] H. Payne and S. Tignor, “Freeway incident detection algorithms based on decision trees with states,” *Transportation Research Record*, no. 682, pp. 30–37, 1978.

[33] H. Payne, E. Helfenbein, and H. Knobel, “Development and testing of incident detection algorithms, volume 2: Research methodology and detailed results,” Technology Service Corporation, Tech. Rep. FHWA–RD–76–20, 1976.

[34] M. Levin, G. M. Krause, and J. Budrick, “Incident–detection algorithms. part 2. on–line evaluation,” *Transportation Research Record*, no. 722, pp. 58–64, 1979.

[35] A. R. Cook and D. E. Cleveland, “Detection of freeway capacity–reducing incidents by traffic–stream measurements,” *Transportation Research Record*, no. 495, pp. 1–11, 1974.

[36] Y. J. Stephanedes and A. P. Chassiakos,“Freeway incident detection through filtering,” *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 3, pp. 219–233, Sep. 1993.

[37] ——, “Application of filtering techniques for incident detection,” *Journal of Transportation Engineering*, vol. 119, no. 1, pp. 13–26, 1993.

[38] Y. J. Stephanedes, A. P. Chassiakos, and P. G. Michalopoulos, “Comparative

performance evaluation of incident detection algorithms," *Transportation Research Record*, no. 1360, pp. 50–57, 1992.

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

[40] J. Collins, C. Hopkins, and J. Martin, "Automatic incident detection–TRRL algorithm HIOCC and PATREG," Transport and Road Research Laboratory, Tech. Rep. 526, 1979.

[41] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box- Jenkins techniques," *Transportation Research Record*, no. 722, pp. 1–9, 1979.

[42] M. Levin and G. M. Krause, "Incident detection: a Bayesian approach," *Transportation Re- search Record*, no. 682, pp. 52–58, 1978.

[43] A. I. Gall and F. L. Hall, "Distinguishing between incident congestion and recurrent congestion: a proposed logic," *Transportation Research Record*, no. 1232, pp. 1–8, 1989.

[44] Y. J. Stephanedes and X. Liu, "Artificial neural networks for freeway incident detection," *Transportation Research Record*, no. 1494, pp. 91–97, 1995.

[45] P. G. Michalopoulos, R. D. Jacobson, C. A. Anderson, and T. B. DeBruycker, "Automatic incident detection through video image processing," *Traffic Engineering and Control*, vol. 34, no. 2, pp. 66–75, 1993.

[46] 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*, vol. 25, no. 3, pp. 347–360, 1980.

[47] E. Parkany and D. Bernstein, "Design of incident detection algorithms using vehicle–to– roadside communication sensors," *Transportation Research Record*, no. 1494, pp. 67–74, 1995.

[48] G.Newell,"Asimplifiedtheoryofkinematicwavesinhighwaytraffic,partI:Generaltheory," *Transportation Research Part B: Methodological*, vol. 27, no. 4, pp. 281–287, Aug. 1993.

[49] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Matematicheskii Sbornik*, vol. 89, no. 3, pp. 271–306, 1959.

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

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

[52] J. P. Kaipio and E. Somersalo, *Statistical and computational inverse problems*. Springer, 2005, vol. 160.

[53] H. Chen, H. A. Rakha, and S. Sadek, "Real–time freeway traffic state prediction: A particle filter approach," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 626–631.

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

[55] 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*, vol. 50, no. 2, pp. 174–188, 2002.