Multiple Model Particle Filter for Traffic Estimation and Incident Detection

Ren Wang, *Member, IEEE*, Daniel B. Work, *Member, IEEE*, and Richard Sowers

***Abstract*—This paper poses the joint traffic state estimation and incident detection problem as a hybrid state estimation problem, in which a continuous variable denotes the traffic state and a discrete model variable identifies the location and severity of an incident. A multiple model particle smoother is proposed to solve the hybrid estimation problem, in which the multiple model particle filter is used to accommodate the nonlinearity and switching dynamics of the traffic incident model, and the smoothing algorithm is applied to improve the accuracy of the estimate when data are limited. The proposed algorithms are evaluated through numerical experiments using CORSIM as the true model. The proposed algorithm is also compared with a standard macroscopic traffic estimator via particle filtering and the California incident detection algorithm. The results show that jointly estimating the state and incidents in one algorithm is better than two dedicated algorithms working independently.**

***Index Terms*—Particle filtering, traffic estimation, traffic incident detection.**

I. Introduction

A. A Hybrid State Estimation Problem

THE objective of traffic estimation is to monitor the traffic state. The traffic state (e.g., traffic density along the roadway) can be estimated with a traffic model and a nonlinear filter, where the traffic model is used to predict the traffic state given the initial and boundary conditions, and the nonlinear filter is used to improve the prediction by incorporating information from sensor measurements.

Most existing traffic estimation algorithms 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 model can perform well under normal traffic operating conditions, it will provide poor traffic state estimates when a traffic incident occurs.

This article is motivated by the fact that jointly estimating the traffic state and incidents can improve both traffic state estimates and incident detection capabilities. Clearly, knowledge of an incident can improve post-incident traffic state estimates. On the other hand, 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. In this article, the joint traffic state estimation and incident detection problem is posed as a hybrid state estimation problem. The system evolution and observation equations of the hybrid system are given by:

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

where the continuous variable x denotes the traffic state (e.g., a vector of densities along the roadway), and the discrete 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 traffic evolution equations are constructed from a scalar macroscopic traffic flow model denoted by f, which evolves the traffic state x^{n-1} at discrete time $n-1$ to time n. The term z^n is a vector of density and speed measurements, h^n is a nonlinear observation operator that relates the system state with the measurements, ω^n is the noise associated with the traffic model, and ν^n is the measurement noise. We use an additive noise model for both the evolution and observation equations.

Given the evolution observation system (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 $\{z^1, \ldots, z^n\}$. If γ^n can be uniquely determined, the location, severity, and duration of a traffic incident are known. Here, the severity of a traffic incident is specified by the number of blocked lanes.

One of the central difficulties for solving the joint traffic state estimation and incident detection problem is that γ^n may not have a unique solution. Thus, we formulate the problem in a Bayesian setting [1], where the solution of the problem is the posterior probability distribution of system state x^n and model variable γ^n. Because of the nonlinearity and switching dynamics of the traffic model, we propose a multiple model particle filter to solve the sequential estimation problem. A multiple model particle smoothing algorithm is proposed to improve the accuracy of the estimate when data is sparse.

Manuscript received December 13, 2014; revised December 22, 2015; accepted April 4, 2016. Date of publication June 6, 2016; date of current version November 23, 2016. This work was supported in part by the NEXTRANS University Transportation Center under Grant DTRT12-G-UTC05 and in part by the Initiative for Mathematical Sciences and Engineering, University of Illinois at Urbana-Champaign. The Associate Editor for this paper was Y. Wang.

R. Wang is with the Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA (e-mail: renwang2@illinois.edu).

D. B. Work is with the Department of Civil and Environmental Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA (e-mail: dbwork@illinois.edu).

R. Sowers is with the Department of Mathematics and the Department of Industrial and Enterprise Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA (e-mail: r-sowers@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2560769

B. Related Work

The main challenge of traffic state estimation is the integration of various types of sensor data such as flow, occupancy, or speed into a nonlinear traffic model. The existence of nonlinearities have led to the application of sophisticated nonlinear data fusion techniques such as the *ensemble Kalman filter* (EnKF) [2], [3], *unscented Kalman filter* [4], *mixture Kalman filter* [5], and *particle filter* (PF) [6]–[8] for traffic monitoring in the transportation literature. However, these algorithms do not account for changes in the dynamics on the highway caused by traffic incidents, which may result in poor traffic state estimates when an incident occurs.

A variety of approaches have been developed for traffic incident detection [9], such as the *California algorithms* (CA) [10], [11], probe-based methods [12], [13], the artificial intelligence based method [14], the neural network approach [15], and the wavelet-based method [16]. Among these approaches, the most well known ones are variants of the California algorithm [10], [11]. 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. Then, 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 occupancies. While the above incident detection algorithms have been shown to have good performance in terms of incident detection, a limitation is that they are not able to estimate the traffic state.

Macroscopic traffic flow model based incident detection methods have also been developed. The work [17] detects traffic incident by identifying when the measurements obtained from the field significantly deviates from the prediction by the traffic flow model. However, the incident does not change any properties on the macroscopic model, and the traffic estimates under an incident suffers as a result. The dynamic model approach [18] deploys a macroscopic traffic model and generates multiple models by instantiating a new equilibrium fundamental diagram for each incident severity. Then, a *multiple model* (MM) extended Kalman filter is used to select the most likely model (similarly incident severity) and to produce filtered traffic states. One drawback of the MM approach is that the model selection at the current time step is independent from the selected model at the previous time step. Given the fact that a traffic incident is a rare event and once it occurs, it will persist, the current selected system model should influence the transitional probabilities for the models at the next time step. Our preliminary work [19] has shown how an *interactive multiple model* (IMM) approach can help to improve the estimation accuracy by incorporating this feature in the evolution of the model variable.

Other works that are closely related to this article are the biparameter approach [20] and the *extended Kalman filter* (EKF) approach [21], [22], where the traffic state estimation and incident detection problem is posed as a joint state and parameter estimation problem. In [20], two continuous parameters are embedded in to a traffic flow model 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 [21], [22], 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 [22]. Different from [20]–[22], this article 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. Compared to the extended Kalman filter, no linearization is needed in this proposed estimation framework since the particle filter is capable of handling non-linear models.

C. Contributions and Outline

The contributions of this article are summarized as follows:

- A multiple model particle filter is proposed to handle the nonlinearity and switching dynamics of the traffic model with incidents, and to solve the joint traffic estimation and incident detection problem. The multiple model particle filter is further extended to a multiple model particle smoother to improve the estimation accuracy when sensor data is limited. The influence of an incident on the macroscopic traffic model is enhanced compared to our preliminary work [19] to include a speed reduction in addition to a capacity and jam density reduction.
- The proposed algorithms are evaluated through numerical experiments using a micro simulator to generate incidents and the true traffic state, as opposed to a perturbed macroscopic model as in [18] and [19]. The results show that the proposed algorithms are able to detect traffic incidents with good accuracy when the remaining lane(s) cannot accommodate all the traffic (e.g., when congestion is created).
- The proposed algorithms are compared with both a particle filter and the California algorithm. The results show that jointly estimating the traffic state and incidents performs better than performing each task independently.

The remainder of this article is organized as follows. In Section II, the traffic flow model with incidents is introduced. In Section III, the hybrid state estimation problem is presented and the multiple model particle filter and smoother are described. In Section IV, the proposed method is tested with synthetic traffic incident data simulated by CORSIM, and conclusions are presented in Section V.

II. Description of Traffic State and Incident Evolution Equations

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

A. Traffic Evolution Equation

The *Lighthill–Whitham–Richards Partial Differential Equation* (LWR PDE) [23], [24] is used to describe the evolution of the density $\rho(x,t) \in [0, \gamma(x,t)\rho_{\max}]$ at location x and at time t

on a roadway. The LWR PDE is chosen since it is a commonly used macroscopic traffic flow model in the transportation community. The model variable $\gamma(x,t)$ denotes the number of lanes open at location x and time t. This model expresses the conservation of vehicles on the roadway of length L, and is given by:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial \left(\rho(x,t) v\left(\rho(x,t), \gamma(x,t)\right)\right)}{\partial x} = 0, \quad (x,t) \in (0,L) \times (0,T) \tag{2}$$

with the following initial and boundary conditions:

$$\begin{aligned} \rho(x,0) &= \rho_0(x),\ \gamma(x,0) = \gamma_0(x) \\ \rho(0,t) &= \rho_l(t),\ \rho(L,t) = \rho_r(t) \\ \gamma(0,t) &= \gamma_l(t),\ \gamma(L,t) = \gamma_r(t) \end{aligned} \tag{3}$$

and where ρ_0, ρ_l, ρ_r, γ_0, γ_l, γ_r are the initial, left, and right boundary conditions for the traffic state and the model variable. When the incident is viewed as a local (e.g., the incident occurs at a point) unilateral constraint on the flux function $q(\rho,\gamma) = \rho \times v(\rho,\gamma)$ defined when all lanes are open, the well posedness of the model can be established [25]. To close the model, a constitutive relationship between density and velocity, denoted by v, must be specified. In this work, the following velocity function is used:

$$v(\rho,\gamma) = \begin{cases} v_{\max}(\gamma) & \text{if } \rho \le \rho_c(\gamma) \\ a(\gamma)\rho + b(\gamma) + c(\gamma)/\rho & \text{otherwise} \end{cases} \tag{4}$$

where $v_{\max}$ and ρ_c are the maximum speed and critical density of the road. The parameters a, b, and c determine the shape of the quadratic function and need to be calibrated from field data. These parameters are functions of γ because when a traffic incident occurs, the parameters at the incident location will change depending on how many lanes are blocked. Practically, it is hard to calibrate the offline parameters under incident scenarios for a specific road segment. In this work, we suggest using values from the literature to determine these parameters. Of course, if historical incident data is available for the test site, these parameters should be calibrated from historical incident data or estimated through online algorithms (e.g., the estimation method proposed by Wang and Papageorgiou [22]).

In this work, the maximum flow for the incident location is determined from the *Highway Capacity Manual* (HCM) [26], where the maximum flow of the remaining lane is given based on the number of lanes blocked by an incident and the total number of lanes available. For example, according to the HCM, the remaining capacities are separately 0.49 and 0.17 of the original capacity when one and two lanes are blocked on a three-lane road. Field data collected in [27] indicates the maximum speed under an incident is 18 mph. Fig. 1 illustrates the *fundamental diagram* (density-flow curve) under different traffic incidents for a three-lane freeway.

For numerical implementation, (2) is discretized using a Godunov scheme [28], yielding the *Cell Transmission Model* (CTM) [29], [30]. Specifically, the time and space domains are discretized by introducing a discrete time step 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:

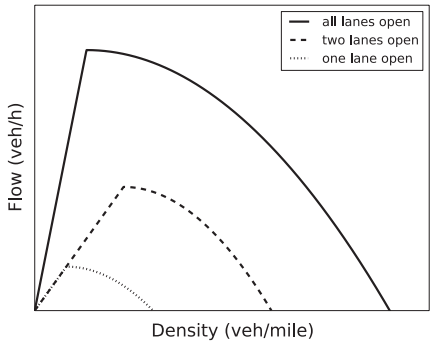


Fig. 1. Relationships between traffic density and flow under a traffic incident on a three-lane road. Note that the critical density for incident scenarios may increase or decrease (as reported in [21]) depending on the choice of the maximum speed and the maximum flow.

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta T}{\Delta x} G\left(\rho_{i-1}^n, \rho_i^n, \gamma_{i-1}^{n+1}, \gamma_i^{n+1}\right) - \frac{\Delta T}{\Delta x} G\left(\rho_i^n, \rho_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}\right). \tag{5}$$

In (5), ρ_i^n denotes the value of the traffic density at time step n and in cell i. The numerical flux $G(\rho_i^n, \rho_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}) = \min\{S(\rho_i^n, \gamma_i^{n+1}), R(\rho_{i+1}^n, \gamma_{i+1}^{n+1})\}$. The functions S and R are known as the sending and receiving functions, which are given by:

$$S(\rho,\gamma) = \begin{cases} q(\rho,\gamma) & \text{if } \rho < \rho_c(\gamma) \\ q\left(\rho_c(\gamma),\gamma\right) & \text{if } \rho \ge \rho_c(\gamma) \end{cases} \tag{6}$$

$$R(\rho,\gamma) = \begin{cases} q\left(\rho_c(\gamma),\gamma\right) & \text{if } \rho < \rho_c(\gamma) \\ q(\rho,\gamma) & \text{if } \rho \ge \rho_c(\gamma). \end{cases} \tag{7}$$

To ensure numerical stability, the time and space steps are coupled through the CFL condition [31]: $v_{\max}\frac{\Delta T}{\Delta x} \le 1$.

B. Incident Evolution Equations

In the problem described by (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, 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 [32] with transition probabilities defined by:

$$\pi_{kj} = p\{\gamma^n = j | \gamma^{n-1} = k\}, \quad k,j \in U, \tag{8}$$

where the set $U = \{1,2,\dots,u\}$ defines all possible incident conditions. The transition probability matrix is defined as $\bar{\Pi} = [\pi_{kj}]$, which is a $u \times u$ matrix satisfying

$$\pi_{kj} \ge 0 \quad \text{and} \quad \sum_{j=1}^{u} \pi_{kj} = 1. \tag{9}$$

Equation (8) indicates the probability of the transition from one model to another. We use $\Pi(\cdot)$ to denote the transition function of the model variable γ, which 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 specifies how many lanes will likely be open at each time step. The traffic model defined by (5) defines the evolution operator f in (1), while (8) defines the evolution of the model variable.

C. Observation Equation

In this work, traffic density measurements from inductive loops and speed measurements from GPS equipped probe vehicles are assumed to be available. The nonlinear operator h in (1) needs to be defined to link the system state to the measurements. The system state at time n is defined by the vector $x^n = [\rho_0^n, \ldots, \rho_{i_{\max}}^n]$. The observation operator h is given by:

$$h^n(x^n, \gamma^n) = H^n \begin{bmatrix} x^n \\ v(x^n, \gamma^n) \end{bmatrix}. \tag{10}$$

The matrix H^n is constructed based on the locations where the measurements are acquired. 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 equipped vehicles may change over time. The observation noise term in (1),

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

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

III. Hybrid-State Estimation

In this section, 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 smoother which can improve the accuracy of the estimate.

A. 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 vector $y^n = [x^n, \gamma^n]$. The estimation problem is formulated using the Bayesian approach [1]. This approach estimates the posterior probability density function $p(y^n|Z^n)$, where y^n is the augmented system state and Z^n are the measurements from time step one to time step n, which is defined as $Z^n = \{z^1, \ldots, z^n\}$. The system state y^n is recursively updated according to:

$$\begin{aligned} p(y^n|Z^{n-1}) &= \int p(y^n|y^{n-1}, \bar{\Pi}) p(y^{n-1}|Z^{n-1}) dy^{n-1} \\ p(y^n|Z^n) &= \frac{p(z^n|y^n) p(y^n|Z^{n-1})}{p(z^n|Z^{n-1})}. \end{aligned} \tag{11}$$

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 $p(y^{n-1}|Z^{n-1})$ is the posterior distribution at time $n-1$, and $p(y^n|y^{n-1}, \bar{\Pi})$ 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 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 measurements. The posterior distribution is proportional to

$$p(y^n|Z^n) \propto p(z^n|y^n) p(y^n|Z^{n-1}). \tag{12}$$

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. 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 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 time step and their previous weights. 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 input to the system evolution model for the next iteration.

B. 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 [32], 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 generated for likely system models. It 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 matches well with the measurements, then we believe the system is operating in model γ at time n.

The pseudocode of the multiple model particle filter is summarized in Algorithm 1, and consists of the following steps.

- *Initialization*: Generate M particles from the initial distribution of y^0 and assign each particle an equal weight. The notation l is used to index the particles. 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.
- *Model transition*: Calculate the model variable for all particles according to the transition matrix (8) and (9).
- *Prediction*: Calculate the prior distribution of system state x^n according to the traffic model f in (2) and (3).

- *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. Measurements from inductive loop detectors and GPS vehicles are assumed to be available at each time step n.
- *Resampling*: Resample particles based on their weights. Resampling is applied to avoid the degeneracy problem in the particle filter. The *systematic resampling algorithm* in [32] is used in this work.
- *Output*: The solution to this problem is a posterior distribution of the augmented system state y^n. If the distribution of 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.

Algorithm 1 Multiple model particle filter [32]

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**
  Model transition: $\gamma_l^n = \Pi(\gamma_l^{n-1})$ for all l
  Prediction: $x_l^n = f(x_l^{n-1}, \gamma_l^n) + \omega_l^{n-1}$ for all l
  Measurement processing:
  calculate the likelihood: $p(z^n|y_l^n)$ for all l
  update weights: $w_l^n = w_l^{n-1} p(z^n|y_l^n)$ 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

This multiple model particle filter method will work well when traffic sensors are dense, but the estimation accuracy may decrease if the number of sensors is limited.

On one hand, when a traffic incident occurs and if there are no sensors nearby, it will take time 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 correct 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 collapses.

On the other hand, when there are no incidents and the sensors are sparse, the particles generated by incident models may found match well with the measurements for the same reason discussed above. Consequently, particles generated by incident models will remain in the sample set, and the estimation accuracy of the algorithm will decrease. To address this problem, we apply the idea of fixed-lag smoothing and combine it with the multiple model particle filter.

C. 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 will help to provide a better estimation of the current system state. The fixed-lag approximation [33], [34] is described by:

$$p(Y^n|Z^T) \approx p\left(Y^n|Z^{\min\{n+\Delta S, T\}}\right) \tag{13}$$

where $Y^n = \{y^0, \ldots, y^n\}$, 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 Y^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 *multiple model particle smoother* (MMPS) is shown in Algorithm 2.

Algorithm 2 Fixed-lag multiple model particle smoother

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**
  Model transition: $\gamma_l^n = \Pi(\gamma_l^{n-1})$ for all l
  Prediction: $x_l^n = f(x_l^{n-1}, \gamma_l^n) + \omega_l^{n-1}$ for all l
  Measurement processing and smoothing:
  for $\tau = 1$ to $\Delta S + 1$ **do**
    calculate the likelihood:
    $p(z^{n+\tau-1}|x_l^n(\tau), \gamma_l^n(\tau))$, with $x_l^n(1) = x_l^n$, $\gamma_l^n(1) = \gamma_l^n$ for all l
    update weights:
    $w_l^n(\tau) = w_l^n(\tau - 1) p(z^{n+\tau-1}|x_l^n(\tau), \gamma_l^n(\tau))$ for all l, with $w_l^n(0) = 1/M$
    normalize weights:
    $\hat{w}_l^n(\tau) = w_l^n(\tau) / \sum_{l=1}^M w_l^n(\tau)$ for all l
    $w_l^n(\tau) = \hat{w}_l^n(\tau)$ for all l
    if $\tau \neq \Delta S + 1$: **then**
      $\gamma_l^n(\tau + 1) = \Pi(\gamma_l^n(\tau))$ for all l
      $x_l^n(\tau + 1) = f(x_l^n(\tau), \gamma_l^n(\tau + 1)) + \omega_l^{n+\tau-1}$
    end if
    $\tau = \tau + 1$
  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

The main difference between the multiple model particle smoother and the multiple model particle filter is the measurement processing stage. In the multiple model particle filter, 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 likelihoods calculated during the time period $\Delta S + 1$ [35]:

$$w^n \propto p(z^{n+\Delta S} | Z^{n+\Delta S-1}, \Gamma^{n-1}), \tag{14}$$

where $\Gamma^{n-1} = \{\gamma^0, \ldots, \gamma^{n-1}\}$. Accordingly, resampling is performed using the weight calculated from the measurements up to $Z^{n+\Delta S}$. During smoothing, each particle is evolved forward in time. Thus, the state x^n generated by a model variable γ^n is evaluated for 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 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 is known as the sample impoverishment problem [32].

IV. Microscopic Numerical Simulation

To test whether the proposed algorithms have potential to work in practice, the CORSIM microscopic simulation software is used to simulate a traffic incident on a three-lane freeway segment. 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 claim 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 model 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 discretized LWR model used in the estimator, which models only conservation of vehicles and a relationship between average speed and density.

The CORSIM simulation is 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). 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 50% is given for the remaining lanes at the incident location to model the phenomenon that drivers will slow down when they see an incident on the road. The value of the rubberneck factor is selected based on the HCM. When the rubberneck factor is set as 50%, the capacity of the remaining lanes is about 0.5 of the original capacity, which is close to the capacity drop (i.e. 0.49) suggested by the HCM. Although not presented here, our algorithms perform well under perturbations up to plus/minus 10% of the rubberneck factor.

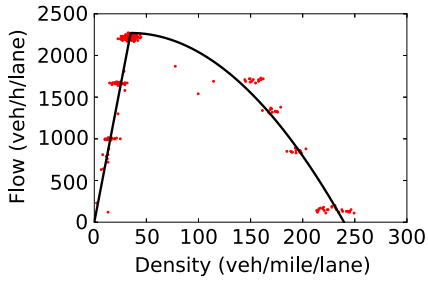


Fig. 2. Density–flow relationship. (Red dots) Measurements obtained from CORSIM. (Solid black line) Calibrated density–flow model.

A. Fundamental Diagram for CORSIM

To test the proposed algorithm in CORSIM, the fundamental diagram in the macroscopic LWR model 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 passing over a simulated inductive loop detector. Then, the traffic occupancy data is converted to traffic density data to construct the density-flow relationship. The red dots in Fig. 2 show the resulting density-flow relationship obtained from our CORSIM simulations. Note that on an experimental deployment, the density-flow data can be obtained by using historical data from the loop detectors.

The function $q(\rho, \gamma)$ is used to fit the data. The maximum flow is calibrated as 2210 veh/hour/lane according to the calibration procedure described in [36], which is the highest flow value observed from the data. The maximum speed $v_{\max}$ is set as 65 mph, which is the free flow speed specified in CORSIM. The jam density is calibrated as 239 veh/mile/lane by using a least squares fit.

These parameters are subject to change when an incident occurs. As described in Section II, the values from the HCM and [27] are used to determine the parameters used in this work, as summarized in Table I. With these parameters, the shape of the fundamental diagram can be determined. The parameters a,

TABLE I
TRAFFIC MODEL PARAMETERS

Parameters	three lanes	two lanes	one lane	unit
v_{max}	65	18	18	mile/hour
q_{max}	2210	1624	1127	veh/hour/lane
ρ_J	239	239	239	veh/mile/lane

TABLE II
SETUP FOR THE MACROSCOPIC MODEL AND NOISE MODEL

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

b, and c in (4) are calculated by solving the following system of equations:

$$\begin{cases} a(\rho_J)^2 + b(\rho_J) + c = 0 \\ -\frac{b}{2a} = \frac{q_{\max}}{v_{\max}} \\ \frac{4ac-b^2}{4a} = q_{\max}. \end{cases} \tag{15}$$

Other parameters used for the discretized LWR model and noise models within the estimation algorithms are summarized in Table II. 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.

B. Assumptions for Model Variable Evolution Equations

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 the 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 0.5% probability for the incident to be cleared, and 0.5% probability to have a second incident in the upstream of the existing incident. The second incident has an equal probability to occur at any upstream cell with any severity. Theoretically, we can allow a third or fourth incident to occur. However, since traffic incident is a rare event, in this work, we consider two incidents at maximum. Third, if there are two incidents at the current time step, there is a 99% probability for the incidents to remain in the next time step, and 1% probability for one of the two incidents 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 we assume a lower probability of an incident, a larger sample size may be needed.

C. Simulation and Error Metric Description

For the proposed multiple model particle filtering and smoothing algorithms, the sample size is set as $M = 2500$. Two inductive loop detectors are assumed 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.

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:

$$e_x = \frac{1}{(i_{\max}+1)(n_{\max}+1)} \sum_{i=0}^{i_{\max}} \sum_{n=0}^{n_{\max}} |\hat{\rho}_i^n - \bar{\rho}_i^n|$$

$$e_\gamma = \frac{1}{(i_{\max}+1)(n_{\max}+1)} \sum_{i=0}^{i_{\max}} \sum_{n=0}^{n_{\max}} |\hat{\gamma}_i^n - \bar{\gamma}_i^n| \tag{16}$$

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 (mean of the posterior distribution), and $\bar{\gamma}_i^n$ is the true model variable at each time n and location i.

D. Estimation Results With Different Penetration Rates

In the first set of simulations, the inflow in CORSIM is specified as 6000 veh/hour. The density and the model variable in the time and space domain for the true condition are shown in Fig. 3(a) and (b).

The left boundary condition assumed in the estimator is $5900 + \mathcal{N}(0, 150^2)$. The right boundary condition is in free flow. The algorithm is first tested by assuming penetration rates of 4% and 1%, and the estimation results without smoothing are shown in Fig. 3(c)–(f). As the result shows, when the penetration rate is 4%, the algorithm is able to correctly estimate the model variable. When the penetration rate is decreased to 1%, the estimation accuracy for both the traffic state and the model variable decreases, due to the reasons discussed in Section III-B.

Next, the multiple model particle smoother is tested when the penetration rate is 1% and the results are shown in Fig. 3(g) and (h). The ΔS for this simulation is set as three. Compared to Fig. 3(e) and (f), which have the same penetration rate (without smoothing), the accuracy of both the traffic state and model variable estimates improves with smoothing. Thus, when the penetration rates of probe vehicles are 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.

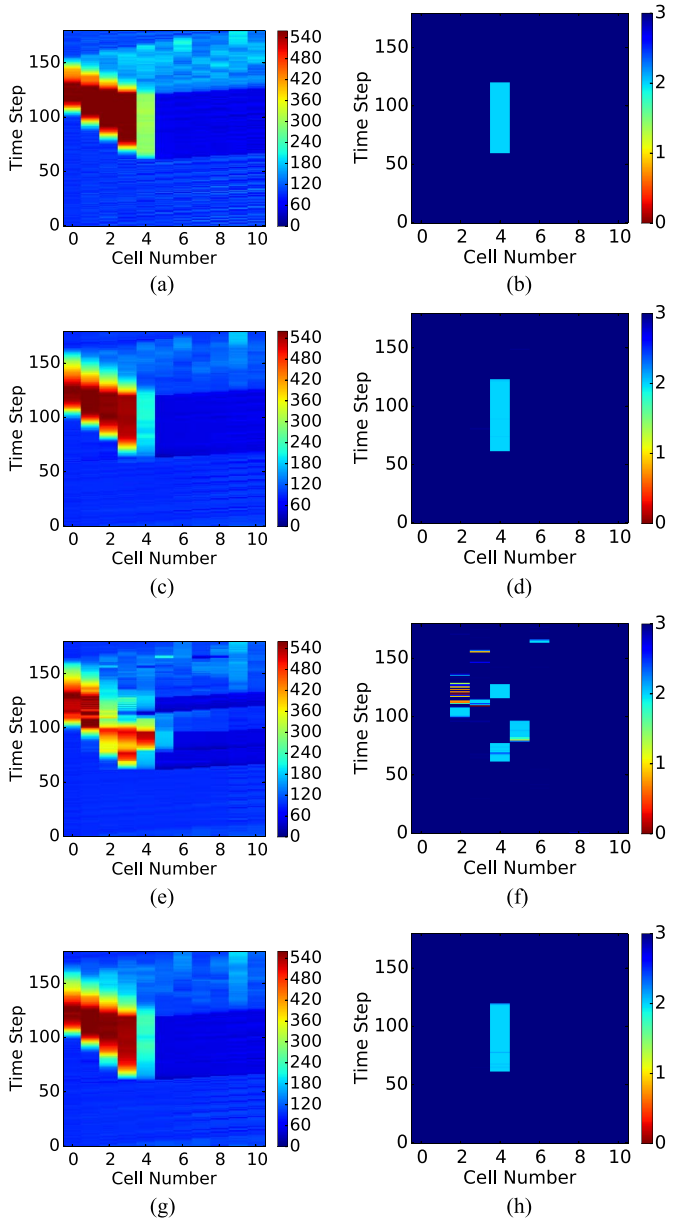


Fig. 3. True evolution of the traffic density and the model variable (first row). Estimate of the multiple model particle filter, penetration rate 4% (second row) and 1% (third row). Estimate of the multiple model particle smoother, penetration rate 1% and $\Delta S = 3$ (fourth row). The values of the (left) traffic state and (right) model variable estimate at each time and space domain are described by the color bar. The value shown is the mean of the posterior distribution. (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). (g) Density (veh/mile). (h) Model variable (lanes open).

To provide a more comprehensive analysis, the algorithms are tested by assuming four penetration rates. For each penetration rate, five tests are conducted with the multiple model particle filter and the multiple model particle smoother. The ΔS for these simulations is set as three. The results are summarized in Fig. 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.

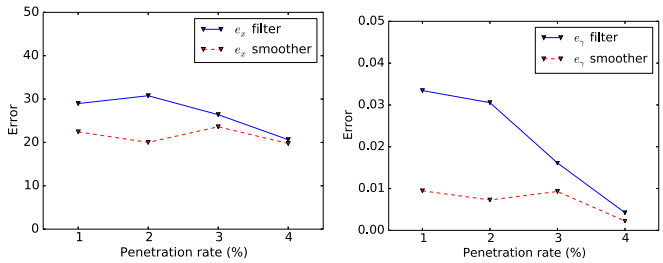


Fig. 4. Average error (five tests) for (left) density and (right) model variable estimates under different headways. The ΔS for these simulations is set as three.

The proposed algorithms are implemented in Python and run on a 2.7 GHz Intel Core i7 Macbook Pro. The source code is available for download [37]. Each one hour numerical experiment can be run in about 15 minutes when the MMPF is applied. When the multiple model particle smoothing algorithm is used with the $\Delta S = 3$, the experiment can be run in about 40 minutes. Thus, the proposed algorithms are suitable for real-time applications, although parallelization approaches [38] may be necessary for large networks.

E. Comparison With the Particle Filter, the California Algorithm, and the IMM EnKF for Different Inflows

The performance of the algorithms are tested with different boundary conditions and compared with a particle filter [6] and the California algorithm [10], which estimate the traffic state and incidents independently, and the IMM EnKF proposed in our preliminary work [19]. The penetration rate is 4% for all simulations. The particle filter in [6] is implemented with the traffic model described in Section II, but without any incident dynamics (i.e., all lanes are always assumed open). The California algorithm [10] is implemented and the occupancy measurements from the inductive loop detectors at cell one and cell nine are used. Table III shows the comparison between the algorithms for traffic estimation and incident detection when the inflow ranges from 1000 veh/hour to 6000 veh/hour.

When one lane is blocked, the remaining capacity for the three-lane road is about 3300 veh/hour. When the remaining lanes have enough capacity to accommodate all of the traffic (i.e., when inflow ranges from 1000 to 3000 veh/hour), the traffic incident does not generate significant congestion. As a result, the traffic state estimates of all techniques are accurate even though the algorithms assume all lanes are open. In this case, the California algorithm, the IMM EnKF, and the MMPF fail to detect the incident. 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. Similarly, for the IMM EnKF, an incident is reported if an incident model is selected for consecutive three time steps.

When the inflow exceeds the remaining capacity of the road, 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

TABLE III
TRAFFIC ESTIMATION AND INCIDENT DETECTION PERFORMANCE OF THE CA, PF, IMM ENKF, AND MMPF ALGORITHMS

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

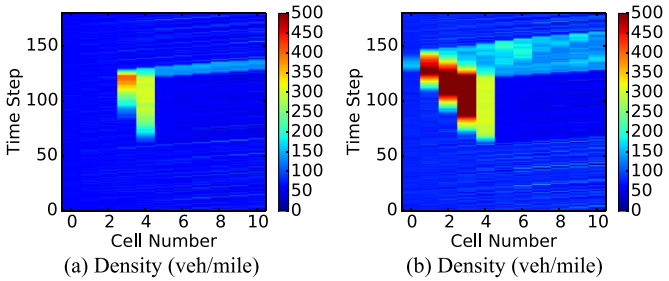


Fig. 5. True evolution of traffic density. (a) Inflow: 4000 veh/h. (b) Inflow: 5000 veh/h.

the PF continues to estimate the traffic assuming all lanes are open. Consequently, the PF collapses and provides bad state estimates. We conclude the MMPF and IMM EnKF perform better than a particle filter in terms of traffic estimation under incidents resulting in congestion.

Compared to the IMM EnKF, the state estimation accuracy of the MMPF is generally higher. This is because the solution to the MMPF is a posterior distribution that may contain state estimates generated from more than one model, while the IMM EnKF proposed in [19] is a model conditioned filter and the posterior distribution contains samples generated only by the most likely model. When the traffic model does not perfectly model 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 four minutes to estimate an hour of traffic, while the computation time for the MMPF is 15 minutes. Thus, the MMPF is more accurate for traffic state estimation, but the IMM EnKF may be suitable larger implementations.

The California algorithm is not able to detect the incident when the inflow is 4000 veh/hour or below. When the inflows are 5000 and 6000 veh/hour, the California algorithm detects the incidents 26 and 18 minutes late. Fig. 5 shows the true evolution of traffic density when the inflows are 4000 and 5000 veh/hour. As Fig. 5(a) shows, the California algorithm cannot detect the incident when the inflow is 4000 veh/hour since the incident information does not propagate to the upstream sensor at cell one. When the inflow is 5000 veh/hour, the California algorithm reports the incident after the congestion propagates to the sensor at cell one. In comparison, both the MMPF and the IMM EnKF are able to detect the traffic incident close to real-time for both cases by taking additional measurements from GPS vehicles.

V. CONCLUSION

This paper formulates the joint traffic estimation and incident detection problem as a hybrid state estimation problem. A multiple model particle filter and a multiple model particle smoother are proposed to detect the location and severity of traffic incidents. The algorithms are tested by synthetic traffic incident data generated by CORSIM and the results show the proposed methods are able to detect traffic incidents with good accuracy when the inflows are large.

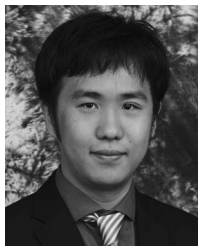
Several areas are open for future exploration. First, the output of the MMPF is a posterior distribution on the model variable, which indicates the number of open lanes. In this work, we use the model variable with the highest probability to infer the existence of an incident. When the mass is not uniquely centered on one integer, several other approaches can be used to transform the distribution into a best estimate of the number of open lanes. For example, a classifier could be trained to link the posterior distribution of x and γ to incidents.

Additionally, while this work laid the foundation for the design of the algorithms, the performance of the incident detection algorithms was only explored numerically. The CORSIM experiments demonstrate that the algorithm might 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. Moreover, because the particle filter has significantly larger computational requirements compared to nonlinear Kalman filtering approaches, more computationally efficient algorithms and parallelization approaches may be necessary on large freeway networks. In our future work, the scalability of the algorithms will be investigated and the estimation framework will be tested in field experiments.

REFERENCES

[1] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*. Berlin, Germany: Springer, 2005.
[2] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. M. Bayen, "A traffic model for velocity data assimilation," *Appl. Math. Res. Exp.*, vol. 2010, no. 1, pp. 1–35, 2010.
[3] S. Blandin, A. Couque, A. Bayen, and D. Work, "On sequential data assimilation for scalar macroscopic traffic flow models," *Physica D, Nonlinear Phenomena*, vol. 241, no. 17, pp. 1421–1440, 2012.
[4] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within particle filtering framework," *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.
[5] X. Sun, L. Muñoz, and R. Horowitz, "Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application," in *Proc. Amer. Control Conf.*, 2004, vol. 3, pp. 2098–2103.
[6] L. Mihaylova and R. Boel, "A particle filter for freeway traffic estimation," in *Proc. IEEE 43rd Conf. Decision Control*, 2004, vol. 2, pp. 2106–2111.

[7] J. Sau, N.-E. El Faouzi, A. B. Assa, and O. De Mouzon, "Particle filter-based real-time estimation and prediction of traffic conditions," *Appl. Stochastic Models Data Anal.*, vol. 12, pp. 406–413, 2007.
[8] N. Polson and V. Sokolov, "Bayesian analysis of traffic flow on interstate I-55: The LWR model," *Ann. Appl. Statist.*, vol. 9, no. 4, pp. 1864–1888, 2014.
[9] E. Parkany and C. Xie, "A complete review of incident detection algorithms & their deployment: What works and what doesn't," Univ. Massachusetts Transp. Center, Amherst, MA, USA, Tech. Rep. NETCR37, 2005.
[10] H. Payne, E. Helfenbein, and H. Knobel, "Development and testing of incident detection algorithms, volume 2: Research methodology and detailed results," Technol. Service Corp., Santa Monica, CA, USA, Tech. Rep. FHWA-RD-76-20, 1976.
[11] M. Levin, G. M. Krause, and J. Budrick, "Incident-detection algorithms. Part 2. On-line evaluation," *Transp. Res. Rec.*, vol. 722, pp. 58–64, 1979.
[12] E. Parkany and D. Bernstein, "Design of incident detection algorithms using vehicle-to-roadside communication sensors," *Transp. Res. Rec.*, vol. 1494, pp. 67–74, 1995.
[13] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: Automatic traffic accident detection and notification with smartphones," *Mobile Netw. Appl.*, vol. 16, no. 3, pp. 285–303, 2011.
[14] Y. J. Stephanedes and X. Liu, "Artificial neural networks for freeway incident detection," *Transp. Res. Rec.*, vol. 1494, pp. 91–97, 1995.
[15] D. Srinivasan, X. Jin, and R. L. Cheu, "Adaptive neural network models for automatic incident detection on freeways," *Neurocomputing*, vol. 64, pp. 473–496, Mar. 2005.
[16] Y. Jeong, M. Castro-Neto, M. K. Jeong, and L. D. Han, "A wavelet-based freeway incident detection algorithm with adapting threshold parameters," *Transp. Res. C, Emerging Technol.*, vol. 19, no. 1, pp. 1–19, 2011.
[17] A. Lemarchand, D. Koenig, and J. J. Martinez Molina, "Incident detection for an uncertain traffic model," *Fault Detection, Supervision Safety Tech. Process.*, vol. 8, no. 1, pp. 648–653, 2012.
[18] A. Willsky *et al.*, "Dynamic model-based techniques for the detection of incidents on freeways," *IEEE Trans. Autom. Control*, vol. 25, no. 3, pp. 347–360, Jun. 1980.
[19] R. Wang and D. Work, "An interactive multiple model ensemble Kalman filter for traffic estimation and incident detection," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2014, pp. 804–809.
[20] A. Dabiri and B. Kulcsár, "Freeway traffic incident reconstruction—A bi-parameter approach," *Transp. Res. C, Emerging Technol.*, vol. 58, pp. 585–597, Sep. 2015.
[21] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, 2005.
[22] Y. Wang *et al.*, "An adaptive freeway traffic state estimator," *Automatica*, vol. 45, no. 1, pp. 10–24, 2009.
[23] M. Lighthill and G. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London, A, Math. Phys. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.
[24] P. Richards, "Shock waves on the highway," *Oper. Res.*, vol. 4, no. 1, pp. 42–51, 1956.
[25] R. M. Colombo and P. Goatin, "A well posed conservation law with a variable unilateral constraint," *J. Different. Equations*, vol. 234, no. 2, pp. 654–675, 2007.
[26] W. Reilly, Highway Capacity Manual 2000, TR News, Washington, DC, USA, no. 193, 1997.
[27] B. Pan, U. Demiryurek, C. Shahabi, and C. Gupta, "Forecasting spatio-temporal impact of traffic incidents on road networks," in *Proc. IEEE Conf. Data Mining*, 2013, pp. 587–596.
[28] 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.
[29] C. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transp. Res. B, Methodol.*, vol. 28, no. 4, pp. 269–287, 1994.
[30] C. Daganzo, "The cell transmission model. Part II: Network traffic," *Transp. Res. B, Methodol.*, vol. 29, no. 2, pp. 79–93, 1995.
[31] R. LeVeque, *Numerical Methods for Conservation Laws*. Basel, Switzerland: Birkhäuser, 1992.
[32] B. Ristic, S. Arulampalm, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA, USA: Artech House, 2004.
[33] A. Doucet, N. De Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*. New York, NY, USA: Springer, 2001.
[34] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[35] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 613–624, Mar. 2001.
[36] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya, "Automatic calibration of the fundamental diagram and empirical observations on capacity," in *Proc. 88th Annu. Meet. Transp. Res. Board*, 2009, vol. 15, pp. 1–14.
[37] R. Wang, *Source Code for Multiple Model Particle Filter for Traffic Estimation and Incident Detection*. [Online]. Available: https://github.com/Lab-Work/MMPF_Traffic_Estimation_Incident_Detection
[38] L. Mihaylova, A. Hegyi, A. Gning, and R. K. Boel, "Parallelized particle and Gaussian sum particle filters for large-scale freeway traffic systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 36–48, Mar. 2012.



Ren Wang (M'15) received the B.S. degree in transportation engineering from Beijing University of Technology, Beijing, China, in 2010 and the M.S. degree in industrial engineering from University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2013. He is currently working toward the Ph.D. degree in the Department of Civil Engineering, University of Illinois at Urbana-Champaign.

His research interests include hybrid state estimation, traffic flow theory, data mining, and machine learning with application on transportation and infrastructure systems.

Daniel B. Work (M'14) received the B.S. degree in civil engineering from Ohio State University, Columbus, OH, USA, in 2006, and the M.S. and Ph.D. degrees in civil engineering from University of California, Berkeley, CA, USA, in 2007 and 2010, respectively.

He is currently an Assistant Professor with the Department of Civil and Environmental Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL, USA. His research interests include control, estimation, and optimization of cyber–physical systems, mobile sensing, and inverse modeling and data assimilation applied to problems in civil and environmental engineering.

Prof. Work was a recipient of a number of awards, including the CAREER Award from the National Science Foundation in 2014 and the IEEE ITSS Best Dissertation Award in 2011.

Richard Sowers received the B.S. degree in electrical engineering from Drexel University, Philadelphia, PA, USA, in 1986, and the M.S. degree in electrical engineering and the Ph.D. degree in applied mathematics from University of Maryland, College Park, MD, USA, in 1988 and 1991, respectively.

He is a Professor of industrial and enterprise systems engineering and mathematics with University of Illinois at Urbana-Champaign, Champaign, IL, USA.