Transportation Research Part C 71 (2016) 521–537

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Efficient multiple model particle filtering for joint traffic state estimation and incident detection

Ren Wang, Shimao Fan, Daniel B. Work *

Department of Civil and Environmental Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA

ARTICLE INFO

Article history:
Received 2 May 2016
Received in revised form 24 July 2016
Accepted 5 August 2016
Available online 10 September 2016

Keywords:
Traffic estimation
Traffic incident detection
Multiple model
Particle filter
Field implementation

ABSTRACT

This article proposes an *efficient multiple model particle filter* (EMMPF) to solve the problems of traffic state estimation and incident detection, which requires significantly less computation time compared to existing multiple model nonlinear filters. To incorporate the on ramps and off ramps on the highway, junction solvers for a traffic flow model with incident dynamics are developed. The effectiveness of the proposed EMMPF is assessed using a benchmark hybrid state estimation problem, and using synthetic traffic data generated by a micro-simulation software. Then, the traffic estimation framework is implemented using field data collected on Interstate 880 in California. The results show the EMMPF is capable of estimating the traffic state and detecting incidents and requires an order of magnitude less computation time compared to existing algorithms, especially when the hybrid system has a large number of rare models.

1. Introduction

The problems of traffic state estimation and incident detection are important challenges in the area of freeway traffic surveillance. Macroscopic model based traffic estimators are typically posed as a state estimation problem in which the traffic flow model serves as a state predictor that is corrected with real time measurements via a filter. Since the framework was first introduced by Gazis and Knapp (1971), a variety of improvements in the filtering algorithms have been developed (e.g., Wang and Papageorgiou, 2005; Work et al., 2010; Mihaylova and Boel, 2004; Van Hinsbergen et al., 2012; Mihaylova et al., 2012, see Blandin et al. (2012) for a recent review). On the other hand, incident detection algorithms are often developed using statistical techniques, ranging from the seminal California algorithm (Payne et al., 1976) to more recent machine learning approaches (see for example, Samant and Adeli, 2001; Yuan and Cheu, 2003; Gu et al., 2016; Wang et al., 2013; Srinivasan et al., 2005; Kinoshita et al., 2015, and the review Parkany and Xie, 2005).

Because the traffic evolution is influenced by the presence of incidents, and incident detection can be enhanced by knowledge of expected congestion, integrated approaches to jointly estimate the traffic state and the presence of incidents is desirable. Such connections were realized as early as 1973 (Nahi, 1973) as an open research direction. Later, the dynamic model approach (Willsky et al., 1980) was proposed to exploit the traffic dynamics to improve incident detection using a multiple model *extended Kalman filter* (EKF). More recently, the bi-parameter approach (Dabiri and Kulcsár, 2015), the EKF approach (Wang and Papageorgiou, 2005; Wang et al., 2009), and the development of a flow model to integrate trajectory data influenced by incidents (Colombo and Marcellini, 2016) have also been proposed. A review on related works on

* Corresponding author.
E-mail addresses: renwang2@illinois.edu (R. Wang), shimao@illinois.edu (S. Fan), dbwork@illinois.edu (D.B. Work).

http://dx.doi.org/10.1016/j.trc.2016.08.003
0968-090X/

traffic state estimation and traffic incident detection, and the advantages of solving the traffic state estimation and incident detection problems jointly can be found in Wang et al. (2016).

In Wang et al. (2016) and Wang and Work (2014), a general framework for jointly estimating the traffic state and location and severity of incidents was developed. The joint problem is posed as a hybrid state estimation problem, where the canonical evolution and observation equations are given as:

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

In the hybrid system (1), the variable γ^n is the model variable, which encodes the integer number of lanes open along the freeway at time n. The first equation describes the evolution of the model variable. The function $\Pi(\cdot)$ is a model transition function, which returns a new model variable given the current model variable according to the model transition probability (e.g., the probability of a traffic incident model to occur at time n given no incident at time $n-1$). The second equation describes the evolution of traffic, where f is a traffic flow model, and x^n denotes the traffic state (e.g., densities along the roadway) at time n. The model variable γ is an input to the flow model f because the evolution of traffic depends on whether there is an incident on the road. The function h^n is a nonlinear observation operator that relates the system state with the measurements z^n. The variable ω^n is a random variable representing the process noise associated with the traffic model, and v^n is a random variable that describes the measurement noise. An additive noise model is used for both the evolution and observation equations. The objective of the hybrid state estimation problem is to jointly estimate the traffic state x^n and the model variable γ^n given measurements z from time zero to time n sequentially in time.

The hybrid state estimation problem in the form of (1) has been studied by the estimation community for applications such as maneuver target tracking (Li and Jilkov, 2003, 2005; Ristic et al., 2004). When the system model f is nonlinear, the *multiple model particle filter* (MMPF) (Ristic et al., 2004), the *interactive multiple model* (IMM) *particle filter* (PF) (Tafazoli and Sun, 2006), and the IMM *ensemble Kalman filter* (EnKF) (Li and Jilkov, 2005) have been proposed to solve hybrid state estimation problems. The MMPF, IMMPF, and IMM EnKF are Monte Carlo based methods, which use sample distributions to approximately estimate the system states. The MMPF (Ristic et al., 2004) is computationally expensive when the hybrid system contains rare models (e.g., a traffic incident or a fault), because the algorithm assigns samples to each model proportional to the model transition probability. When the system contains rare models, the MMPF assigns an unnecessarily large number of samples to the likely models in order to generate a few samples for the rare models, which leads to an often prohibitive computational cost. The IMM based filter (Tafazoli and Sun, 2006) has been proposed to address this issue. The idea of an IMM method is to first compute the model probability for all models according to the model transition probability. Then, a filter is performed on each model with a fixed number of samples. Here, the model probability indicates how likely each model will occur, and the estimation result from the filter indicates how well the prediction of each model matches with the measurements from the field. These two pieces of information are combined together to infer the correct model and the state estimate. The IMM based filters always assign equal number of samples to all models. As a result, when a hybrid system contains rare models, it avoids the need for a large sample size in order to generate sufficient samples for all models. The works (Wang et al., 2016; Wang and Work, 2014) deployed the above ideas to solve the hybrid state estimation problem (1) for joint traffic state estimation and incident detection, where an IMM EnKF (Wang and Work, 2014) and a MMPF (Wang et al., 2016) were proposed.

One feature associated with the estimation problem (1) for tracking traffic and incidents is that the hybrid system contains a large number of rare models, because traffic incidents can occur at any location with different severities (e.g., one lane blocked, two lanes blocked). The number of models involved in the system is a function of the length and the number of lanes on the highway. For field implementation, the total number of models of the hybrid system can be large because the road network can contain many miles of highway and may include segments with many lanes. Moreover, when the traffic flow model is implemented on roads with entrance and exit ramps, the road often needs to be discretized into smaller cells when ramps are considered, so that the ramps are located at (or close to) the boundary between cells while maintaining a fixed spatial discretization throughout the network. As a result, even the IMM filters may not run in real time because running a filter for a large number of models requires a large computational cost. Later we show the MMPF and the IMM EnKF (Wang et al., 2016; Wang and Work, 2014) does not run in real-time when implemented with field data on a four to five lane stretch of freeway in California.

In this work, we propose an *efficient multiple model particle filter* to address this problem, which is significantly more computationally efficient compared to the earlier algorithms (Wang et al., 2016; Wang and Work, 2014). Different from the MMPF and the IMM filters where the model selection and the state estimation are coupled together, the EMMPF performs the two tasks separately. The proposed EMMPF has a model selection step and a model-conditioned filtering step. In the model selection step, the EMMPF deploys the idea of an IMM approach, where the model probability for each model is precomputed according to the model transition probability, and the model is selected by evaluating the pre-computed model probability and how well the estimated state by each model matches with the measurements. Unlike the IMM filters, where a filter is performed on each possible model to evaluate how well the state estimated by each model matches with the measurements, the EMMPF performs the evaluation by running a single sample prediction in each model. Then, after the model is selected, a filter (i.e., a particle filter) is performed only on the most likely model determined in the model selec-

tion step. In other words, the EMMPF only runs a single sample on each model to select the correct model, and performs a particle filter only on the selected model. As a result, significant computation can be saved for systems with many rare models since the filter is performed only on one model. In this work, the effectiveness of the proposed EMMPF is validated using both numerical simulations and in a field implementation.

The contributions of this article are summarized as follows. An EMMPF is proposed to solve the joint traffic state estimation and incident detection problem for road networks. The proposed EMMPF is tested on incident data collected on Interstate 880 in California, where traffic densities from inductive loops and speeds from GPS equipped vehicles are collected and used as input to the algorithm. The results show that the EMMPF runs in real time, and that it is able to jointly estimate the traffic state and detect incidents. The computational efficiency of the proposed algorithm is also compared with existing MMPF and IMM EnKF algorithms (Wang et al., 2016; Wang and Work, 2014). It is found that the EMMPF requires significantly less computation time, especially when the hybrid system contains a large number of models.

The reminder of the article is organized as follows. In Section 2, the single link macroscopic traffic flow model with incidents (Wang et al., 2016) is reviewed and extended to a network model by deriving junction models which accommodate incident dynamics. In Section 3, the efficient multiple model particle filter is proposed to solve the hybrid state estimation problem. In Section 4, a benchmark numerical example is used to test the effectiveness of the proposed EMMPF and to compare with the performance of the MMPF. The EMMPF is also tested for traffic estimation and incident detection using synthetic traffic incident data generated by CORSIM and compared with the IMM EnKF and the MMPF developed in Wang et al. (2016). In Section 5, the proposed EMMPF is implemented using field data collected on Interstate 880 in California, and future directions are summarized in Section 6.

2. Macroscopic traffic incident models

In this section, the junction solvers for the traffic incident model are developed to allow the traffic model to be deployed on road networks. In order to describe the junction problem, we first briefly review the traffic incident flow model (Wang et al., 2016), then present the new junction solvers for the traffic incident model.

2.1. Macroscopic traffic incident model

The evolution of traffic density $\rho(\chi,\tau)$ at location χ and at time τ is modeled by the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) (Lighthill and Whitham, 1955; Richards, 1956). A model variable γ is embedded into the traffic model to describe traffic evolution under incidents. The LWR PDE with incident dynamics on the roadway of length L during time T is given by:

$$\begin{aligned}&\frac{\partial\rho(\chi,\tau)}{\partial\tau}+\frac{\partial(\rho(\chi,\tau)v(\rho(\chi,\tau),\gamma(\chi,\tau)))}{\partial\chi}=0,\\&(\chi,\tau)\in(0,L)\times(0,T)\end{aligned}\tag{2}$$

with the following initial and weak boundary conditions:

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

where $\rho_0,\ \rho_l,\ \rho_r$, and $\gamma_0,\ \gamma_l,\ \gamma_r$ are the initial and boundary conditions for the traffic density and the model variable. The well posedness of (2) can be established (Colombo and Goatin, 2007) by recognizing the incident as a point constraint on the flux function $q(\rho,\gamma)=\rho\times v(\rho,\gamma)$ defined when all lanes are open.

The choice of the velocity function v depends on the assumptions on the relationship between density and velocity. A possible relationship is the quadratic–linear function proposed by Smulders (1990):

$$v(\rho,\gamma)=\begin{cases}v_{\max}(\gamma)\left(1-\dfrac{\rho}{\beta(\gamma)}\right) & \text{if } \rho\leqslant\rho_c(\gamma)\\ \dfrac{v_{\max}(\gamma)\rho_c(\gamma)(\rho_m(\gamma)-\rho)(\beta(\gamma)-\rho_c(\gamma))}{\rho\beta(\gamma)(\rho_m(\gamma)-\rho_c(\gamma))} & \text{otherwise.}\end{cases}\tag{4}$$

In (4), the variable $v_{\max}$ denotes the maximum speed. The parameter β determines the shape of the velocity function for the free flow regime. In particular, it controls how the average vehicle speed will change when the traffic density increases from zero to the critical density ρ_c. The variable ρ_m denotes the jam density. All parameters are a function of γ because when γ refers to an incident model, the number of open lanes will drop, and the parameters in the traffic model will change accordingly (see Wang et al. (2016) for details).

For numerical implementation, the time and space domain are discretized using a Godunov scheme (Godunov, 1959), which gives a cell transmission traffic incident model:

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

In (5), the variable ΔT, indexed by $n \in \{0, \ldots, n_{\max}\}$, denotes the discrete time step, and the variable Δx, indexed by $i \in \{0, \ldots, i_{\max}\}$, denotes the space step. The variable ρ_i^{n+1} denotes the traffic density at time step $n+1$ and in cell i. The flow (flux) that crosses the cell boundaries is determined by the numerical flux function G, given by:

$$G(\rho_i^n, \rho_{i+1}^n, \gamma_i^{n+1}, \gamma_{i+1}^{n+1}) = \min\{S(\rho_i^n, \gamma_i^{n+1}), R(\rho_{i+1}^n, \gamma_{i+1}^{n+1})\}. \tag{6}$$

The sending and receiving functions S and R are given by:

$$S(\rho, \gamma) = \begin{cases} q(\rho, \gamma) & \text{if } \rho < \rho_c(\gamma) \\ q(\rho_c(\gamma), \gamma) & \text{if } \rho \geqslant \rho_c(\gamma), \end{cases} \tag{7}$$

$$R(\rho, \gamma) = \begin{cases} q(\rho_c(\gamma), \gamma) & \text{if } \rho < \rho_c(\gamma) \\ q(\rho, \gamma) & \text{if } \rho \geqslant \rho_c(\gamma), \end{cases} \tag{8}$$

where the flow function is given as $q(\rho, \gamma) = \rho \times v(\rho, \gamma)$. Because the fundamental diagram changes in the presence of an incident, the sending and receiving functions will also change, and this consequently influences the number of vehicles that can move from one cell to the next. To ensure numerical stability, the CFL condition (LeVeque, 1992): $v_{\max} \frac{\Delta T}{\Delta x} \leqslant 1$ should be satisfied. Here $v_{\max}$ denotes the maximum speed when there is no incident.

2.2. Network problem for the traffic incident model

Because of the existence of on ramps and off ramps, the traffic model needs to be extended to networks. The road network is modeled as a graph consisting of links and junctions. On each link, the traffic evolves according to (2) when γ is known. At each junction, the flow allocation from incoming links to outgoing links needs to be defined. Existing allocation rules at the junction are extended to accommodate the presence of incidents at the junction. In this section, three types of junctions are considered: bottlenecks and lane additions, diverges, and merges. These are the most common freeway junctions and represent lane drops and lane additions, off-ramps, and on-ramps.

The junction solvers for traffic flow models have been widely studied (e.g., Daganzo, 1995; Garavello and Piccoli, 2006; Herty and Klar, 2003; Holden and Risebro, 1995; Jin, 2010; Lebacque, 2005). The precise form of the model is application specific, and depends on the ability of the modeller to obtain or the parameters in the model. From a mathematical standpoint, a main goal is to determine a unique solution such that the well posedness of the network model of PDEs can be analyzed. For simplicity of illustration, in this work the implemented models are based on maximizing the flow through the junction such that constraints on the merging (or diverging) flows are satisfied. The general approach to modify alternative junction solvers to include incident dynamics follows a similar procedure as the one presented next.

In this section, the model parameter γ is embedded into the three types of junction models, and the junction solvers for the traffic incident model are developed. We also show how the traffic model can be combined with the junction solvers to evolve the network traffic state in time.

2.2.1. Bottlenecks and lane additions: one incoming road and one outgoing road

The simplest junction contains two links labeled $m = 1$ (incoming) and $m = 2$ (outgoing) with a different number of lanes. This can model bottlenecks (shown in Fig. 1), or expansion zones where a lane is added to the roadway. We use $\rho_{i_{\max}}^{n,(1)}$, $\gamma_{i_{\max}}^{n+1,(1)}$ to denote the traffic state in the last cell of the incoming link, and $\rho_0^{n,(2)}$, $\gamma_0^{n+1,(2)}$ to denote the traffic state in the first cell of the outgoing link. The flow that can cross the junction over an interval ΔT is determined by solving the following optimization problem:

$$\begin{aligned} \underset{G^{(1)}, G^{(2)}}{\text{maximize}}: \quad & G^{(1)} \\ \text{subject to}: \quad & 0 \leqslant G^{(1)} \leqslant S^{(1)}\left(\rho_{i_{\max}}^{n,(1)}, \gamma_{i_{\max}}^{n+1,(1)}\right) \\ & 0 \leqslant G^{(2)} \leqslant R^{(2)}\left(\rho_0^{n,(2)}, \gamma_0^{n+1,(2)}\right) \\ & G^{(1)} = G^{(2)}, \end{aligned} \tag{9}$$

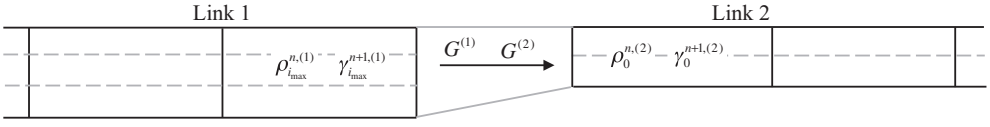

Fig. 1. Bottleneck junction with one incoming link and one outgoing link. The arrow indicates the flow from the upstream link to the downstream link. The term $G^{(m)}$ denotes the flow corresponding to link m.

where $G^{(1)}$ and $G^{(2)}$ separately represent the flow that leaves link 1 and the flow that enters link 2. The term $S^{(1)}$ denotes the sending function of last cell at link 1, and $R^{(2)}$ is the receiving function of the first cell at link 2. Here, the sending and receiving functions are the same as those defined in Eqs. (7) and (8), just with a link superscript to denote each link may have different fundamental diagram parameters. The optimal decision variables $G^{(1)*}$ and $G^{(2)*}$ to the optimization problem (9) are:

$$G^{(1)*} = G^{(2)*} = \min\left\{S^{(1)}, R^{(2)}\right\}. \tag{10}$$

2.2.2. Diverge: one incoming road and two outgoing roads

A diverge road junction contains three links $m = 1$ (incoming), $m = 2$ (outgoing) and $m = 3$ (outgoing), as shown in Fig. 2. We use $\rho_{i_{\max}}^{n,(1)}$, $\gamma_{i_{\max}}^{n+1,(1)}$ to denote the traffic state in the last cell of the incoming link, and $\rho_0^{n,(2)}$, $\gamma_0^{n+1,(2)}$ and $\rho_0^{n,(3)}$, $\gamma_0^{n+1,(3)}$ to denote the traffic state in the first cell of each outgoing link. The flow that crosses the junction, and the flow that enters each outgoing link can be determined by extending the model (Garavello and Piccoli, 2006) for incident dynamics. The resulting optimization problem is as follows:

$$\begin{aligned}
\underset{G^{(1)},G^{(2)},G^{(3)}}{\text{maximize}}: \quad & G^{(2)} + G^{(3)} \\
\text{subject to}: \quad & 0 \leqslant G^{(1)} \leqslant S^{(1)}\left(\rho_{i_{\max}}^{n,(1)}, \gamma_{i_{\max}}^{n+1,(1)}\right) \\
& 0 \leqslant G^{(2)} \leqslant R^{(2)}\left(\rho_0^{n,(2)}, \gamma_0^{n+1,(2)}\right) \\
& 0 \leqslant G^{(3)} \leqslant R^{(3)}\left(\rho_0^{n,(3)}, \gamma_0^{n+1,(3)}\right) \\
& G^{(1)} = G^{(2)} + G^{(3)} \\
& \alpha_d G^{(2)} = (1-\alpha_d) G^{(3)},
\end{aligned} \tag{11}$$

where $\alpha_d \in (0,1)$ is known as the split ratio, which determines the fraction of the flow out of link 1 that travels to link 3. In this work, we assume the flow to each outgoing link strictly obeys the split ratio specified by α_d, which is a common assumption of diverge models (Garavello and Piccoli, 2006). The optimal decision variables $G^{(1)*}$, $G^{(2)*}$, and $G^{(3)*}$ to the optimization problem (11) can be calculated as:

$$\begin{aligned}
G^{(2)*} &= \min\left\{R^{(2)}, (1-\alpha_d)/\alpha_d R^{(3)}, (1-\alpha_d) S^{(1)}\right\}, \\
G^{(3)*} &= \alpha_d/(1-\alpha_d) G^{(2)*}, \\
G^{(1)*} &= G^{(2)*} + G^{(3)*}.
\end{aligned} \tag{12}$$

2.2.3. Merge: two incoming roads and one outgoing road

A merge road junction contains three links $m = 1$ (incoming), $m = 2$ (incoming) and $m = 3$ (outgoing), as shown in Fig. 3. We use $\rho_{i_{\max}}^{n,(1)}$, $\gamma_{i_{\max}}^{n+1,(1)}$, and $\rho_{i_{\max}}^{n,(2)}$, $\gamma_{i_{\max}}^{n+1,(2)}$ to denote the traffic state in the last cells of the incoming links, and $\rho_0^{n,(3)}, \gamma_0^{n+1,(3)}$ to denote the traffic state in the first cell of the outgoing link. The flow that crosses the junction over the interval ΔT, and the flow that leaves each incoming link can be determined by solving the following optimization problem:

$$\begin{aligned}
\underset{G^{(1)},G^{(2)},G^{(3)}}{\text{maximize}} \quad & G^{(1)} + G^{(2)} \\
\text{subject to} \quad & 0 \leqslant G^{(1)} \leqslant S^{(1)}\left(\rho_{i_{\max}}^{n,(1)}, \gamma_{i_{\max}}^{n+1,(1)}\right) \\
& 0 \leqslant G^{(2)} \leqslant S^{(2)}\left(\rho_{i_{\max}}^{n,(2)}, \gamma_{i_{\max}}^{n+1,(2)}\right) \\
& 0 \leqslant G^{(3)} \leqslant R^{(3)}\left(\rho_0^{n,(3)}, \gamma_0^{n+1,(3)}\right) \\
& G^{(3)} = G^{(1)} + G^{(2)} \\
& \alpha_m G^{(1)} = (1-\alpha_m) G^{(2)},
\end{aligned} \tag{13}$$

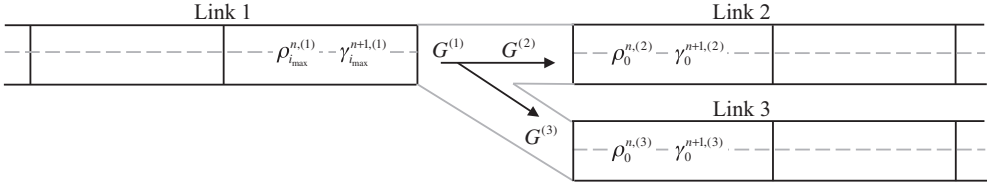

Fig. 2. Diverge junction with one incoming link and two outgoing links. The arrow indicates the flow from the upstream link to the downstream link. The term $G^{(m)}$ denotes the flow corresponding to link m.

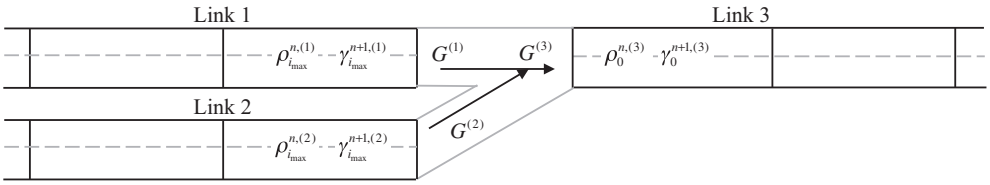

Fig. 3. Merge junction with two incoming links and one outgoing link. The arrow indicates the flow from the upstream link to the downstream link. The term $G^{(m)}$ denotes the flow corresponding to link m.

where $\alpha_m \in (0,1)$ denotes the merge ratio of traffic from the incoming links. The vehicles that enter the downstream link are composed of traffic from the two incoming links. The optimal decision variables $G^{(1)*}$, $G^{(2)*}$, and $G^{(3)*}$ to the optimization problem (13) can be computed as:

$$\begin{aligned} G^{(1)*} &= \min\left\{S^{(1)}, (1-\alpha_m)/\alpha_m S^{(2)}, (1-\alpha_m)R^{(3)}\right\}, \\ G^{(2)*} &= \alpha_m/(1-\alpha_m)G^{(1)*}, \\ G^{(3)*} &= G^{(1)*} + G^{(2)*}. \end{aligned} \tag{14}$$

2.2.4. Network traffic evolution

We briefly show how the traffic model can be combined with junction solvers to evolve the network traffic state. Consider a network with two links and a bottleneck junction for simplicity; a more general network with merges and diverges can be developed similarly. Given the traffic densities $\rho_i^{n,(m)}$, the model variables $\gamma_i^{n+1,(m)}$ for all cells i on each link m ($m = 1, 2$), and the link boundary conditions $\rho_0^{n,(1)}$, $\gamma_0^{n+1,(1)}$, $\rho_{i_{\max}}^{n,(2)}$, $\gamma_{i_{\max}}^{n+1,(2)}$, a single step network traffic model prediction can be performed as follows. First, the boundary conditions at the bottleneck junction are obtained using the densities and model parameters in last cell at link 1 and first cell of link 2: $\rho_{i_{\max}}^{n,(1)}$, $\gamma_{i_{\max}}^{n+1,(1)}$, $\rho_0^{n,(2)}$, and $\gamma_0^{n+1,(2)}$. Next, the junction problem (9) is solved according to (10) for $G_{(1)}^*$ and $G_{(2)}^*$. With the junction boundary follow determined, the last cell in link 1 is updated as follows:

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

and in the first cell of link 2 (outgoing link), the prediction follows:

$$\rho_0^{n+1,(2)} = \rho_0^{n,(2)} + \frac{\Delta T}{\Delta x}\left(G^{(2)*} - G(\rho_0^n, \rho_1^n, \gamma_0^{n+1}, \gamma_1^{n+1})\right).$$

For all other cells on link 1 and link 2, the cell transmission model (5) is applied.

3. Nonlinear filters for hybrid state estimation

In this section, we first describe the multiple model particle filter proposed in Wang et al. (2016) for joint traffic state estimation and incident detection to highlight the main computational bottleneck when many rare models are present. Next, the efficient multiple model particle filter proposed in this work is presented. The computational efficiency of the efficient multiple model particle over the multiple model particle filter is discussed.

3.1. Multiple model particle filter

Before presenting the multiple model filters, we first review the evolution equations of the model variable γ, and the nonlinear observation operator h defined in (1). The model variable γ is modeled as a u-state first-order Markov chain (Ristic et al., 2004) with transition probabilities defined by:

$$\pi_{(kj)} = p\{\gamma^n = j | \gamma^{n-1} = k\}, \quad k, j \in \Gamma, \tag{15}$$

where the set Γ defines all possible incident conditions ($\gamma \in \Gamma$). For example, consider a stretch of road with three lanes and four cells. Then $\gamma = [3,3,3,3]$ indicates there is no incident, $\gamma = [3,2,3,3]$ indicates there is an incident at cell one and the incident blocks one lane. The transition probability matrix is defined as $\bar{\Pi} = [\pi_{(kj)}]$, and the transition function is denoted as $\Pi(\cdot)$, which returns a new model variable given the current model variable according to the transition probability matrix $\bar{\Pi}$. More details on the transition function $\Pi(\cdot)$ can be found in Wang et al. (2016).

In this work, traffic density measurements from inductive loops and speed measurements from GPS equipped probe vehicles are used as measurements in the traffic estimation algorithms. The nonlinear operator h^n in (1) is given by:

$$h^n(x^n, \gamma^n) = H^n \begin{bmatrix} x^n \\ v(x^n, \gamma^n) \end{bmatrix}. \tag{16}$$

For networks, the variable x denotes the stacked link states (e.g., the density in each cell in the network), the variable γ denotes the number of lanes open in each cell, and v is the appropriate velocity function composed from the individual link velocity functions. The matrix H^n is constructed based on the locations of sensor measurements, and it is assumed the measurement vector z^n is arranged with the density measurements stacked on top of the speed measurements. The observation noise v^n in (1) is composed of v_{density} and v_{speed}, to denote the different error models for density and speed measurement. The measurement noise is assumed to follow a normal distribution $v^n \sim \mathcal{N}(0, V^n)$, where V^n is the measurement error covariance matrix.

The MMPF (Wang et al., 2016; Ristic et al., 2004) estimates an augmented system state $y^n = [x^n, \gamma^n]$ in a Bayesian setting (see Wang et al. (2016) for details on the Bayesian formulation). The algorithm is presented in Algorithm 1, and it works as follows. In the initial time step, the algorithm generates M particles of the state based on the initial knowledge of the system and assigns each particle equal weight. Here, the variable l denotes the index of particles, the variable $x^n_{(l)}$ denotes the traffic state at time step n of particle l. At each time step, the algorithm determines the model variable for all particles using the transition function $\Pi(\cdot)$. Then, each traffic state realization x^{n-1}_l is evolved forward in time using the network traffic flow model to compute the prior distribution at the next time step. After the measurements are received, the algorithm evaluates how well the predicted traffic state matches the measurements and updates the weight of each particle. The idea is that if the traffic state $x^n_{(l)}$ is generated by the correct model variable γ, it will match well with the measurements z^n. Finally, all the particles are resampled based on their weights. As a result, the particles generated by the correct model variable will be kept and duplicated, and the particles generated by the wrong model variables will be removed from the sample set.

Algorithm 1 Multiple model particle filter (Ristic et al., 2004)

Initialization ($n = 0$): generate M samples $y^0_{(l)}$ and assign equal weights $w^0_{(l)} = 1/M$, where $l = 1, \ldots, M$
for $n = 1$ to $n_{\max}$ **do**
  Model transition: $\gamma^n_{(l)} = \Pi\left(\gamma^{n-1}_{(l)}\right)$ for all l
  Prediction: $x^n_{(l)} = f\left(x^{n-1}_{(l)}, \gamma^n_{(l)}\right) + \omega^{n-1}_{(l)}$ for all l
  Measurement processing:
  calculate the likelihood: $p\left(z^n | y^n_{(l)}\right)$ for all l
  update weights: $w^n_{(l)} = w^{n-1}_{(l)} p\left(z^n | y^n_{(l)}\right)$ for all l
  normalize weights: $\hat{w}^n_{(l)} = w^n_{(l)} \Big/ \sum_{l=1}^{M} w^n_{(l)}$ for all l
  Resampling: multiply/suppress samples $y^n_{(l)}$ with high/low importance weights $\hat{w}^n_{(l)}$
  Output: posterior distribution of x^n and γ^n
  Reassign weights: $w^n_{(l)} = 1/M$ for all l
  $n = n + 1$
end for

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

3.2. Efficient multiple model particle filter

In this section, an EMMPF is proposed to approximately solve the hybrid state estimation problem (1), and is shown in Algorithm 2. Different from the MMPF shown in Algorithm 1, the proposed EMMPF is an IMM approach applied to the particle filter, where the IMM is used to estimate the correct model, and the PF is performed on the estimated model for traffic estimation. Unlike the IMM filters (Wang and Work, 2014; Li and Jilkov, 2005; Tafazoli and Sun, 2006), where a filter is performed on each model to evaluate how well the prediction matches with the measurements, the EMMPF only uses a single sample (e.g., the most likely sample or the mean sample) in each model to infer the correct model. Then, all particles are evolved forward in time using the most likely model determined in the model selection step. As a result, significant computation can be saved at the model selection step. However, the reduction on computation time is at a cost of using one sample for model selection, which may result in a higher false model selection rate. A brief summary and comparison of the MMPF, IMM PF, and the proposed EMMPF is provided in Table 1.

Algorithm 2 Efficient multiple model particle filter

Initialization: generate M samples $x_{(l)}^0$ and assign equal weights $w_{(l)} = 1/M$, where $l = 1, \ldots, M$. The single particle estimate (e.g., mean, most likely) from $x_{(l)}^0$ is denoted by $\tilde{x}^0$.

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

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

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

Predicted state: $x_\gamma^n = f(\tilde{x}^{n-1}, \gamma) + \mathrm{E}[\omega^{n-1}]$

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

Update the model probability: $\mu_\gamma^n = \mu_\gamma^n p_\gamma^n$

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

3. Model-conditioned particle filter:

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

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

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

Normalize weights: $\hat{w}_{(l)}^n = w_{(l)}^n \Big/ \sum_{l=1}^M w_{(l)}^n$ for all l

4. Resampling: multiply/suppress samples $x_{(l)}^n$ with high/low importance weights $\hat{w}_{(l)}^n$, then assign all particle with equal weights $w_{(l)}^n = 1/M$, where $l = 1, \ldots, M$

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

end for

In Algorithm 2, step one and step two describe how the model γ^n is selected. The term μ_γ^n is the probability associated with model γ at time step n. In step one, the model probability is reinitialized based on the model γ at the previous time step and the transition probability. Next, the algorithm predicts the traffic state by assuming γ is correct, and determines the likelihood of γ by evaluating how well the predicted system state x^n by model γ matches with the measurements z^n. Here, the system state $\tilde{x}^{n-1}$ is the single particle estimate from the posterior distribution $x_{(l)}^{n-1}$ from the last time step. In this work the mean of the posterior is used, which coincides with the most likely particle when the sample distribution tends to a Gaussian. The model probability μ_γ^n is updated by multiplying the initial model probability and the model likelihood. This procedure is repeated for all models defined in the system. In step two, the algorithm selects the model γ with the highest model probability as the correct model in the current time step. During model selection, the expected model noise $\mathrm{E}[\omega]$ is added during the model prediction step. This is because instead of using a filter (in which case, model noise should be incorporated), the EMMPF runs only one sample on each model to infer the correct model. As a result, we use the nominal predicted state from each model for model selection.

In step 3, a standard particle filter is performed on the selected model. Here, the prior distribution of $x_{(l)}^{n-1}$ is the posterior distribution of the system state from the previous time step. During the prediction step, we predict the future traffic state for all particles. Then, we compute the likelihood of each particle and update its weight. Next, the weights of the particles are normalized so that the total weight of all particles sum to one. Finally, the systematic resampling algorithm (Ristic et al., 2004) is performed to resample the distribution based on the weights of the particles. The selected model γ^n and the resulting posterior distribution of x^n are output and used as inputs to the algorithm for the next time step.

Table 1
A brief summary and comparison of the MMPF, IMM PF and EMMPF.

Filters	Number of samples assigned to each model	Model selection	State estimation
MMPF (Wang et al., 2016; Ristic et al., 2004)	Proportional to the model probability	Select the model from the posterior distribution of γ	Obtained from the posterior distribution of x
IMM PF (Tafazoli and Sun, 2006)	A fixed number (i.e., the number of samples required for running a filter)	Select the model using precomputed model probability, and model conditioned estimation results from the PF	Obtained from the model conditioned estimation results for the selected model
EMMPF	A single sample	Select the model using precomputed model probability, and the results from model-conditioned single sample prediction	Obtained by running a PF on the selected model

4. Hybrid systems and microsimulation benchmark problems

In this section, we first show the performance of the proposed EMMPF and compare it with the MMPF and the IMM EnKF (Wang and Work, 2014) using a benchmark problem in the multiple model filtering field. Next, the EMMPF is implemented for traffic estimation using synthetic traffic incident data generated by a micro-simulation software, and compared with the IMM EnKF and MMPF.

4.1. Testing of the EMMPF using a benchmark problem

We use the following nonlinear system to test the performance of the proposed EMMPF. The nonlinear system is a generalization of a problem used by a number of studies for testing the performance of filters (Tafazoli and Sun, 2006; Kadirkamanathan et al., 2002; Li and Kadirkamanathan, 2001; Arulampalam et al., 2002). The scalar system is given as follows:

$$\begin{aligned} x^n &= 0.5x^{n-1} + 25\frac{x^{n-1}}{1+(x^{n-1})^2} + 8\cos(1.2n) + \Upsilon^n + \omega^{n-1} \\ z^n &= \frac{(x^n)^2}{20} + v^n. \end{aligned} \tag{17}$$

Similar to the hybrid system (1), the system state is denoted by x^n, the measurement is denoted by z^n, and the model and measurement noises are denoted by ω and v. When $\Upsilon^n = 0$, the system (17) is considered in a normal operating mode (model). When Υ^n takes a value other than 0, the system is in a fault mode. In the following simulations, we assume the system operates in only two modes, one normal mode (where $\Upsilon^n = 0$), and one faulty mode (where Υ^n equals to a nonzero integer). Let p_f denote the transition probability from the normal model to the fault model. For simplicity, once the system is in a fault model, the transition probability from the fault model to the normal model is also assumed to be p_f. When the model transition probabilities are known, the transition probability matrix $\bar{\Pi}$ can be constructed.

The true system is in the normal mode for the first 30 time steps. Between time step 30 and 60, the system switches to the fault model, and switches back to the normal model after time step 60. The initial state $x^0 = 1$. The model noise and measurement nose are assumed to follow normal distributions, where $\omega^n \sim \mathcal{N}\left(0, 0.1^2\right)$ and $v^n \sim \mathcal{N}\left(0, 0.1^2\right)$. The model variable $\Upsilon \in \{0, 5\}$.

In the first set of experiments the number of samples M is set at 100 (equivalently ensembles for the IMM EnKF) and the value of p_f is selected from the set $\{0.5, 0.3, 0.1, 0.01, 0.001, 0.0001\}$ to understand how the performance of the algorithms compare when the fault mode becomes rare. The absolute mean state estimation error is computed as follows:

$$e_x = \frac{1}{n_{\max}}\sum_{0}^{n_{\max}}(|\hat{x}^n - \bar{x}^n|), \tag{18}$$

where $\hat{x}^n$ is the estimated state and $\bar{x}^n$ is the true state at each time step n.

The results for the EMMPF, MMPF, and IMM EnKF are summarized in Fig. 4. For the relatively large state transition values p_f, the MMPF and the EMMPF have similar performance. However, as p_f becomes smaller (i.e., the fault model becomes more rare), the MMPF performance degrades due to the insufficient sample size M to generate (any) particles in the fault model. When no fault model particles are generated, the MMPF is not able to detect the mode switch.

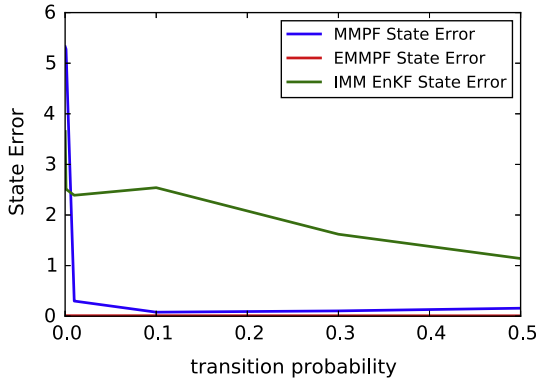

Fig. 4. State error as a function of the transition probability p_f for $M = 100$.

Compared to the EMMPF and the MMPF, the IMM EnKF has more (incorrect) fault model predictions. This occurs when the predicted state value of the normal model and the fault model are close, and after the measurement update step, it may be difficult to distinguish between the modes. Recall that in the EMMPF, the model selection step is based only on a model forward prediction, so the differences between the fault and non-fault modes are more pronounced. On the other hand, if the model noise and measurement noise are reduced (e.g., drawn from $\omega^n \sim \mathcal{N}\left(0, 0.001^2\right)$ and $v^n \sim \mathcal{N}\left(0, 0.001^2\right)$), the IMM EnKF performance is significantly improved and is similar to the performance of the EMMPF.

To further illustrate the performance challenges when the fault model becomes rare, consider when p_f is set as 0.0001 and the number of samples M is set as 100. As Fig. 5a and b shows, the EMMPF is able to detect the switch of the model and to correctly estimate the traffic state. In contrast, the standard MMPF (Fig. 5c and d) is not able to detect the model switch, because the fault model was not generated due to the low transition probability and the relatively small number of particles. The IMM EnKF also has degraded performance, but the cause is due to the additional transitions into the fault regime (Fig. 5e and f). Again, for lower model noises, the IMM EnKF achieves an accuracy comparable with the EMMPF.

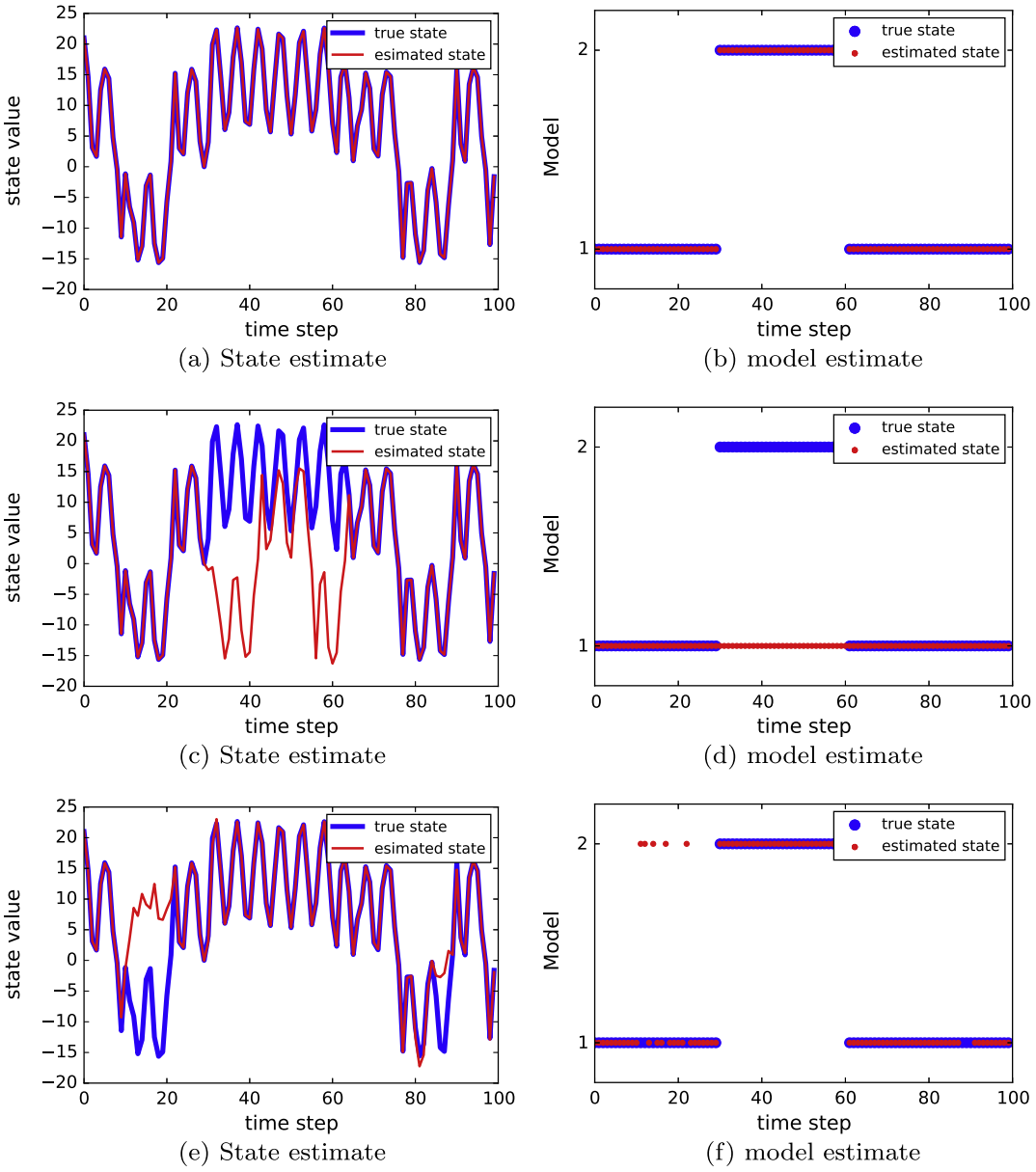

Fig. 5. State estimate and model estimate of the EMMPF (first row), the MMPF (second row), and the IMM EnKF (third row) for system (17). The sample size is set as 100.

In the final set of experiments, the performance and runtimes of the algorithms is compared as the number of particles increases (Fig. 6a and b). With the benchmark noise model, the EnKF is both the lowest quality estimator and also the slowest. The EMMPF offers improved estimation accuracy when the number of particles is small (similarly when the runtime of all algorithms is the smallest). Note the main focus of this numerical experiment is to emphasize that the model selection step of the EMMPF does not significantly degrade the accuracy of the estimate, and can actually enhance the accuracy when rare fault models are used and the number of particles is forced to remain small to meet realtime runtime constraints. When the number of (rare) fault models is increased, the number of particles required to achieve good fault detection grows by the number of modes and inversely proportional to the fault model probability.

It should also be mentioned that the performance of all algorithms depends on the value of Υ. When the fault Υ value is small (e.g., $\Upsilon = 1$), it means there is not a large difference between the normal model and the fault model. In this case, detection of the model switch is more challenging, because the difference between the predicted state under the fault model and the normal model is on the same order of the model and the measurement noise. When the fault value of Υ is large, the difference between the normal model and the fault model is transparent, and consequently the algorithms are more likely to identify the difference and detect the model switch.

4.2. CORSIM implementation

In this section, we test the EMMPF for joint traffic state estimation and incident detection using synthetic traffic data generated by the CORSIM traffic micro-simulation software. The algorithm is tested on a four mile long, three-lane freeway replicating the step up in Wang et al. (2016) so that the EMMPF can be compared to the algorithms proposed in our earlier work. During the one hour traffic simulation, one incident is created, and the speed measurements from GPS equipped vehicles are used as input to the estimation algorithm. The simulation setup, calibration of fundamental diagram parameters, and determination of the model and sensor noise models are the same as in Wang et al. (2016). The density and the model variable in the time and space domain for the true traffic conditions obtained from CORSIM are shown in Fig. 7a and b.

The EMMPF is implemented using the simulation set up in Wang et al. (2016) and compared with the estimation results by the IMM EnKF and the MMPF. In order to expect at least a particle in each of the incident modes (a minimum requirement to expect to detect an incident model when it occurs), the MMPF is run with $M = 2500$ samples. Since the EMMPF and IMM EnKF are model conditioned filters and consequently do not require large particle sizes in order to generate samples in all fault models, they are run with significantly fewer samples (i.e., $M = 100$). The algorithms are tested with the incident data from CORSIM for various inflows ranging from 1000 veh/h to 6000 veh/h with a fixed penetration rate of GPS equipped probe vehicles of 4%. For traffic state estimation on a single link, the absolute mean state estimation errors are computed as follows:

$$e_\rho = \frac{1}{(i_{\max}+1)(n_{\max}+1)} \sum_{i=0}^{i_{\max}} \sum_{n=0}^{n_{\max}} |\hat{\rho}_i^n - \bar{\rho}_i^n|, \tag{19}$$

where $\hat{\rho}_i^n$ is the estimated density (mean of the posterior distribution), and $\bar{\rho}_i^n$ is the true density. As shown in Table 2, compared to the MMPF and IMM EnKF, the EMMPF has a higher state estimation error, but overall, the EMMPF is able to detect the incident and track the resulting congestion, as shown in Fig. 8. Note that when the inflow is 6000 veh/h, the EMMPF has several false incident predictions after the incident is cleared. However, compared to the true state in Fig. 7, the performance of the EMMPF does not suffer too much. When the inflows are 4000 veh/h and 5000 veh/h, the EMMPF estimates the correct incident locations but an incorrect incident severity (e.g., the filter predict two lanes are blocked, and consequently overestimated the resulting congestion).

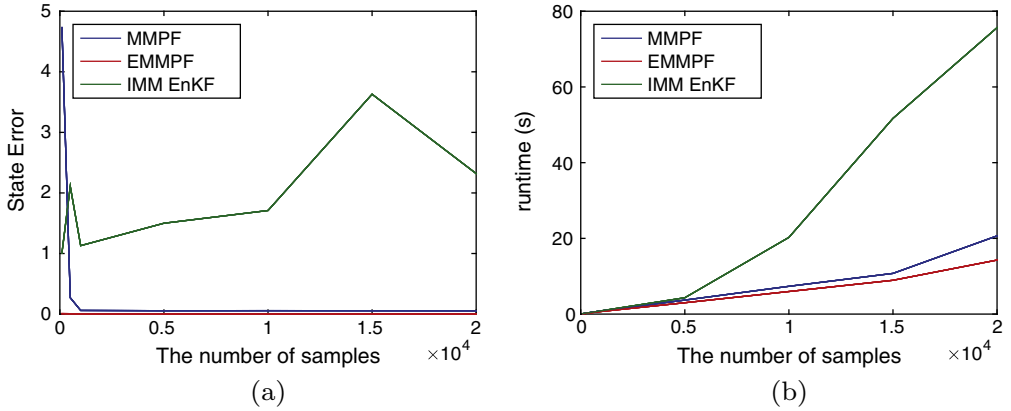

Fig. 6. (a) State error and (b) runtime as a function of M, for $p_f = 0.0001$.

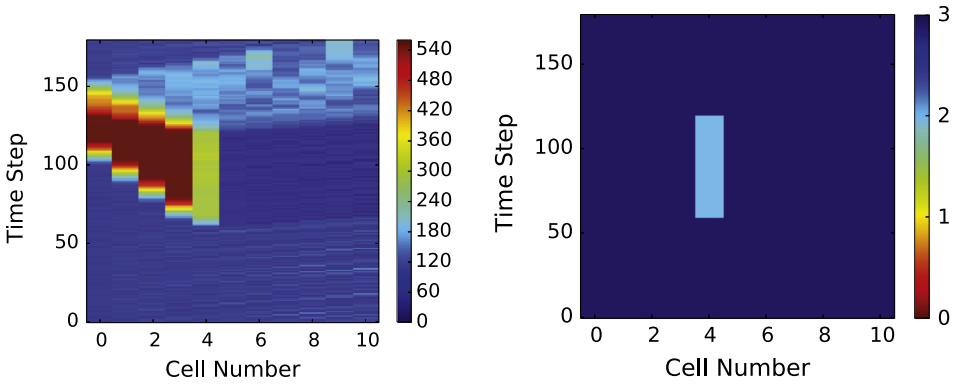

Fig. 7. True evolution of the traffic density and the model variable.

Table 2
Traffic estimation and incident detection performance of the EMMPF, IMM EnKF, and MMPF algorithms.

Inflow (veh/h)	State estimation error (veh/mile)			Incident detection		
	EMMPF	IMM EnKF	MMPF	EMMPF	IMM EnKF	MMPF
1000	3.7	4.8	3.3	Not detected	Not detected	Not detected
2000	5.1	6.0	4.2	Not detected	Not detected	Not detected
3000	7.9	9.4	8.3	Not detected	Not detected	Not detected
4000	51.8	14.6	11.3	Detected 3.0 min late	Detected 3.3 min late	Detected 1.6 min late
5000	57.6	24.4	17.9	Detected 3.0 min late	Detected 3.0 min late	Detected 3.0 min late
6000	33.5	19.5	19.7	Detected 1.6 min late	Detected 1.6 min late	Detected 1.6 min late

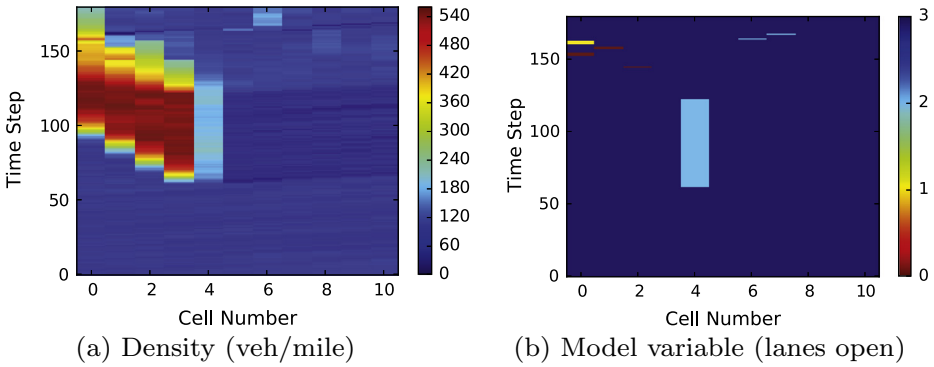

Fig. 8. Density and model variable estimates of the EMMPF, inflow = 6000 veh/hour, penetration rate of 4%.

It takes approximately 15 min for the MMPF and 4 min for the IMM EnKF to complete the one hour simulation. When the EMMPF is used, the computation time drops to approximately 25 s. In this simulation, there are 22 models (one normal model and 21 incident models) associated with the system. The results from this section shows that the EMMPF runs much faster than the IMM EnKF and the MMPF when the hybrid system has a number of rare models, at the expense of a modest reduction in accuracy on the traffic state estimation.

5. Field implementation

In this section, the proposed EMMPF is tested with field data collected on Interstate 880 in California. We first give a brief overview of the network setup and the experimental data used in the estimator. Then, the model calibration procedure and the experiment setup are described. Finally, the implementation results of the EMMPF are presented and discussed.

Note that the MMPF and IMM EnKF were also implemented with the field data to further compare the influence of the algorithm on the estimation results, however both algorithms are far too slow to complete even a single experiment for

analysis. The EMMPF runs in real time and it is able to jointly estimate the traffic state and detect an incident that occurred when the field data was collected.

5.1. Implementation overview

The proposed EMMPF algorithm is tested on a segment of Interstate 880 in California. Density measurements from inductive loops and speed measurements from GPS equipped vehicles are used as measurements in the traffic estimation algorithm, where density measurements are obtained from the *Caltrans Performance Measurement System* (PeMS) and the speed data is collected from the GPS devices deployed during the Mobile Century experiment (Herrera et al., 2010).

PeMS is a highway monitoring system and it collects and records 30 s loop detector data for all of California. Users can get access to the historical and real time loop detector data by visiting their website (http://pems.dot.ca.gov/). The speed measurements used in the field implementation are collected through the Mobile Century experiment (Herrera et al., 2010). The Mobile Century experiment collected speed data from approximately 77 GPS equipped vehicles which run continuously on the segment of highway for 6 h (10 am to 4 pm) on February 8, 2008. The penetration rate of GPS equipped vehicles is approximately 2% of the total traffic flow over the course of the data collection (Herrera et al., 2010).

The geometry of the highway segment used in this work is shown in Fig. 9. The segment of highway is six miles long, from postmile 21.3 to postmile 27.3. The algorithm is tested with six hours of data from 10 am to 4 pm. There are five on ramps and five off ramps on the road segment. In this work, the ramps are modeled as links with one cell. A unique feature of the dataset is that it contains an incident that occurred at postmile 26.64 from 10:27 am to 11:00 am, which serves as the benchmark incident in the experiment to be estimated by the algorithms. The incident was also recorded in the California Highway Patrol traffic incident feed which is also archived on PeMS and can be verified through their website (http://pems.dot.ca.gov/).

5.2. Model calibration

For the field implementation, there are five groups of parameters that need to be determined. The parameters include the discretization of the traffic model, the parameters (e.g., maximum flow, maximum speed, jam density) that determine the shape of the fundamental diagram, the transition probability between incident models and non-incident models, the model noise and measurement noise, and the boundary conditions for the entrance and exit ramps. The calibration is more challenging than the problem in Wang et al. (2016) since the underlying traffic is more complex and contains merging and diverging traffic, heavy congestion, and real driver behaviors. Moreover, determination of the ramp flows is significantly more challenging because there are no sensors at the ramps to create good historical estimates of the ramp flows. In this section, we describe how the model parameters used in the estimator are determined.

The discretization of the traffic flow models is summarized in Table 3. The highway is discretized into 60 cells, and each cell is 0.1 mile in length, so that the ramps are located close to the boundaries of cells. The total experiment simulation length is six hours. The time step ΔT is set as five seconds so that the CFL condition $v_{\max}\frac{\Delta T}{\Delta x} \leqslant 1$ is satisfied. As a result, there are total 60 cells and 4320 time steps in the discretization.

The fundamental diagram related parameters for the non-incident and incident scenarios are determined as follows. For the non-incident scenario, the density-flow data collected from inductive loops are used to calibrate the parameters. The maximum speed is calibrated using a least squares fit using the density-flow data collected from inductive loops under free flow conditions (Dervisoglu et al., 2009). The critical density ρ_c (veh/mile/lane), the shape parameter β (veh/mile/lane), and the jam density ρ_m (veh/mile/lane) as well as the fundamental diagram related parameters for ramps are manually calibrated since there is no field data available on the ramps. The resulting parameters are summarized in Table 4. Later, a sensitivity analysis is performed to show the performance of the EMMPF does not suffer within ±10% perturbations on the calibrated model parameters.

For the incident scenarios, the calibration is difficult since incidents are rare events and density-flow data for different incident severities is limited. In this work, the parameters for incident scenarios are determined using the HCM and a previous study (Pan et al., 2013), where the HCM values for the capacity loss are used to model the fundamental diagrams on the multiple lane highways when different numbers of lanes are blocked, and the work (Pan et al., 2013) is used to model the

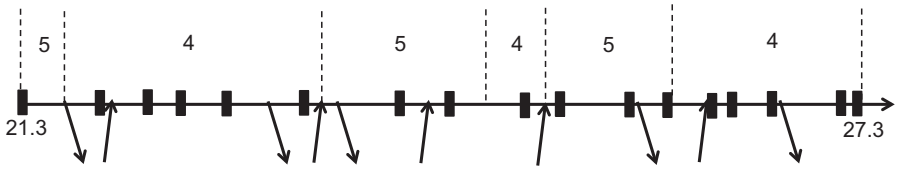

Fig. 9. Geometry of the segment of Interstate 880 in California used for field implementation. The black squares in the figure indicate loop detectors, and the arrows denote on-ramps and off-ramps. The numbers above each segment indicate the number of lanes of the highway. The two numbers at the left and right denote the starting and ending post mile of the highway segment. The traffic moves from the left to the right.

Table 3
Setup for model discretization.

Link length	6 miles
Simulation length	6 h
ΔT	5 s
Δx	0.1 miles
Number of cells	60
Number of time steps	4320

Table 4
Fundamental diagram related parameters.

Parameters	Highway	Ramp	Unit
$v_{\max}$	70	40	mile/h
ρ_c	24	40	veh/mile/lane
ρ_m	130	110	veh/mile/lane
β	10,000	10,000	veh/mile/lane

free flow speed under incidents. It is likely that the parameters from the references do not exactly correspond to the incident dynamics for this specific segment of highway, however, later we show our proposed EMMPF is able to estimate traffic state and detect incidents even when the incident related parameters are not optimally calibrated from the field.

The assumptions associated with the transition probability matrix are described as follows. We allow four incident severities: one lane blocked, two lanes blocked, three lanes blocked, and four lanes blocked. We assume there is at most one incident at a time, which results in a total 241 possible system models including the non-incident model. We also assume that if there is no incident at the current time step, there is 1.0×10^{-4} probability to have an incident at the next time step (i.e., approximately one to two incidents per day), and the probability of each possible incident model to occur is equal. If there is an incident at the current time step, then there is a 1.0×10^{-4} probability for the incident to be cleared for the next time step, otherwise, the incident will remain.

As was mentioned previously, there are two types of sensors are available to be used in the estimation algorithm, namely the inductive loops which collect traffic occupancy and traffic flow data, and the GPS equipped vehicles that provide vehicle speed measurements along their trajectories. Both the model noise and the measurement noises are assumed to be additive and follow a normal distribution. The noise models are summarized as follows: the model noise ω follows $\mathcal{N}(0, 20^2 \times I)$, the density measurement noise follows $\mathcal{N}(0, 40^2 \times I)$ and the speed measurement noise follows $\mathcal{N}(0, 20^2 \times I)$, where I is the appropriately sized identity matrix. Here, the unit of the density measurement noise is in vehicles per mile.

The boundary condition of the traffic model is the traffic density and the model variable (e.g., the number of lanes open at the boundary cell). In the implementation, the boundary conditions for the model variable is determined by the highway geometry and assumes all lanes are open. The density boundary conditions for the main freeway are estimated from historical data from the sensors located near both ends of the highway segment. Since no sensors are located on the ramps, the traffic density ramp boundary conditions are manually calibrated.

Ideally, the model parameters should be calibrated following a standard calibration procedure (e.g., Dervisoglu et al., 2009). However, we are not aware of an existing calibration procedure that has been developed for both normal fundamental diagrams and incident fundamental diagrams, specifically due to the difficulty to observe traffic at the location of the incident. As a result, we manually calibrate a subset of these parameters and perform a sensitivity analysis on the parameters as a proof of concept to show the proposed algorithms have the potential to work well with field data, and leave the development of automatic calibration procedures for future work.

5.3. Experiment description

We test the proposed EMMPF with the macroscopic traffic incident model for both traffic estimation and incident detection. At the initial time step, the prior distribution of the traffic density is assumed to follow a normal distribution, where the mean is set as 90 veh/mile and the standard deviation is 5% of the mean. Because we do not know the ground truth of the traffic evolution, the measurements from the inductive loops and GPS vehicle are used to provide a noisy and incomplete view of the traffic evolution, shown in Fig. 10a and b. The red area early in the day is the congestion caused by the incident, while the high density and slow speed in other regions correspond to non-incident related congestion.

To evaluate the performance of the proposed EMMPF algorithm, we select three loop detectors as a hold out test set, and the measurements are not used in the estimator. Later the estimated traffic state at these three locations are compared with the measurements from the sensors to evaluate the performance of the estimation algorithms. We choose one sensor in the upstream, one sensor in the downstream, and one sensor in the middle of the domain. In particular, the second sensor, the twelfth sensor, and the sixteenth sensor (from left to right in Fig. 9) are removed during the traffic estimation, and

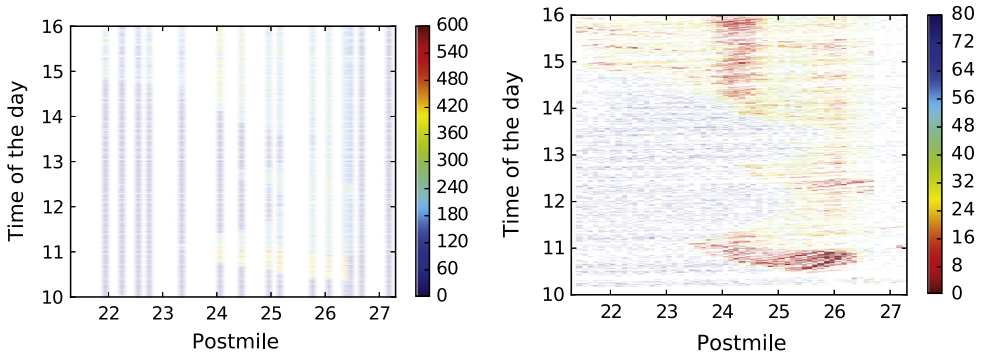

Fig. 10. Density measurements (a) and speed measurements (b). Missing values appear in white.

are subsequently used to evaluate the performance of the proposed algorithms. For the field implementation, the number of particles M is set as 100 particles.

5.4. EMMPF estimation results

The estimation results of the EMMPF are shown in Fig. 11. From the results we see that the proposed EMMPF is capable of detecting the incident and provides a good traffic state estimate. The absolute mean state errors e_ρ are computed at the locations of the three sensors and for the time steps for which measurements are available, and the true state is taken to be the value recorded by the sensor. The absolute mean density error is computed as 29.2 veh/mile.

The algorithms are implemented in Python and run on a 3.0 GHz Intel Core i7 Macbook Pro (Wang, 2016). The six hour experiment can be run in about four hours and 20 min. Thus, the proposed EMMPF is suitable for real time implementation. In comparison, the MMPF (Wang et al., 2016) requires at least 2,410,000 particles in order to expect at least one particle for all possible models. To update all 2,410,000 particles to the next time step alone takes approximately two hours to complete, resulting in a runtime of approximately one year to complete a single test. When the IMM EnKF (Wang and Work, 2014) is used, it requires 24,100 samples to run the EnKF with 100 samples in each model. It takes about 70 s for the IMM EnKF to complete one time step (5 s prediction). While this is a large improvement over the MMPF, the IMM EnKF is still an order of magnitude too slow to be implemented in real time. The computation time for the three algorithms are summarized in Table 5.

5.5. Sensitivity analysis on model parameters

A sensitivity analysis is performed to evaluate whether the estimation accuracy is sensitive to the calibrated model parameters. In this article, the critical density ρ_c and the jam density ρ_m are manually calibrated. Recall that the capacity drop specified in HCM is used to construct the fundamental diagrams for the incident scenarios, as a result, the critical density ρ_c also impacts the traffic operation for incident cases. In this section, we perturb the calibrated critical density and jam density by 10%, and evaluate how the estimation accuracy changes. The state estimation accuracy with the different combination of model parameters are summarized in Table 6.

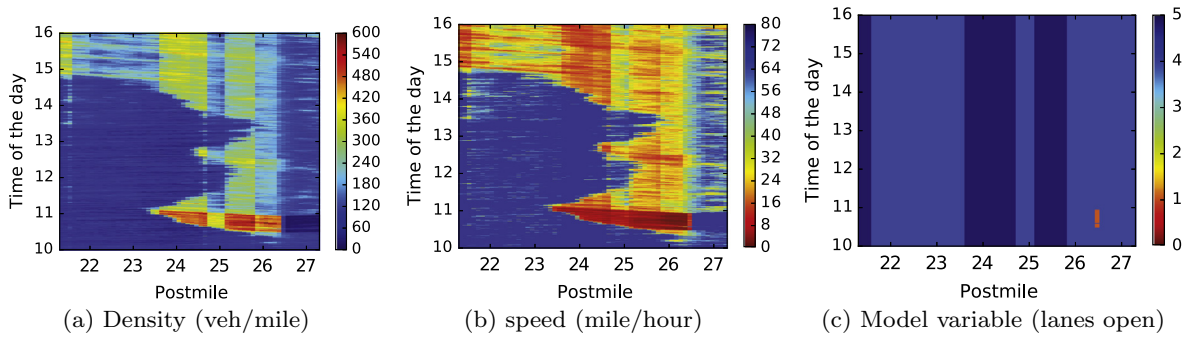

Fig. 11. Estimation results of the EMMPF for traffic density (a), traffic speed (b) and the model variable (c).

Table 5
Computation time of the MMPF, IMM EnKF and EMMPF.

Algorithm	Runtime compared to real time
MMPF	1460 times slower
IMM EnKF	14 times slower
EMMPF	0.7 times faster

Table 6
Estimation accuracy with different combinations of model parameters, which are perturbed by 10% of the nominal values in bold.

ρ_c (veh/mile/lane)	ρ_m (veh/mile/lane)	e_ρ (veh/mile)
24	**130**	29.2
24	143	29.7
24	117	28.4
22	130	27.7
26	130	30.9

The results of the sensitivity analysis show that our estimation algorithm is not very sensitive to the calibrated model parameters in terms of traffic estimation and incident detection. In fact, better estimation results can be achieved with some perturbations of these parameters. This makes sense because not all the traffic model parameters are calibrated from the field. It should also be noted that although the paper uses true positive and false positive to evaluate the performance of incident detection, the output of the algorithm indicates the existence of an incident for every time step (i.e., 5 s), which is significantly more precise compared to earlier incident detection work, which only indicate the occurrence of an incident and provide no information on how long the incident lasts.

Finally, we also provide a sensitivity analysis on the transition parameters. Specifically, the probability of remaining in a non-incident scenario is fixed at 0.9999 (similarly, the probability to remain in the same incident regime) in the experiments described above. To test the sensitivity of the same state transition probability, we also performed runs where the same state transition probability is $\{0.999, 0.9995, 0.9999, 0.99995, 0.99999\}$. The corresponding state estimation error is of similar magnitude i.e., $\{30.2, 31.3, 29.2, 29.0, 30.2\}$, and consequently a slightly improved choice of the transition probability would result in a marginal reduction of the estimation error. In all cases, an incident is detected in the correct incident location as reported from Caltrans and no false positives are reported. Although the location is consistent, the precise time and incident severity differ from one parameter choice to the next. In an extreme case, when the transition probability to stay in the same (non-incident state becomes large), the incident is detected but with an additional delay of 10 min. In summary, the state error and the incident location are not too sensitive to the parameters of the state transition matrix, although it does influence the estimated incident severity.

6. Future work

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

Another area for further development is on the estimation of multiple incidents, which is quite a bit more involved than the single incident scenario presented in this work due to the number of cases to consider. In many scenarios, a second incident may result in no change to the output of the system, rendering the second incident impossible to detect. For example, a small incident downstream of a major incident upstream will not be result in sufficient congestion to enable detection. A second incident in the congestion wave caused by a first incident is also difficult to detect if the resulting congestion from both scenarios is similar. Based on our preliminary experiments with multiple incidents, we believe that new ideas are required for major progress in the multi-incident detection.

Finally, it is possible to further enhance the accuracy of the incident detection capabilities by improving the determination of when an incident has occurred. For example, the estimation results from the EMMPF algorithm could be combined with social media to infer the existence of an incident. Such investigations would be essential in order to get good practical performance in the field, without producing too many false positives or too many missed incidents.

Acknowledgment

This work was supported by the NEXTRANS University Transportation Center under grant DTRT12-G-UTC05.

References

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. 50 (2), 174–188.
Blandin, S., Couque, A., Bayen, A., Work, D., 2012. On sequential data assimilation for scalar macroscopic traffic flow models. Physica D 241 (17), 1421–1440.
Colombo, R.M., Goatin, P., 2007. A well posed conservation law with a variable unilateral constraint. J. Diff. Equat. 234 (2), 654–675.
Colombo, R.M., Marcellini, F., 2016. A traffic model aware of real time data. Math. Models Methods Appl. Sci. 26 (03), 445–467.
Dabiri, A., Kulcsár, B., 2015. Freeway traffic incident reconstruction. A bi-parameter approach. Transp. Res. Part C: Emerg. Technol. 58, 585–597.
Daganzo, C.F., 1995. The cell transmission model, part II: network traffic. Transp. Res. Part B: Methodol. 29 (2), 79–93.
Dervisoglu, G., Gomes, G., Kwon, J., Horowitz, R., Varaiya, P., 2009. Automatic calibration of the fundamental diagram and empirical observations on capacity. In: Transportation Research Board 88th Annual Meeting.
Garavello, M., Piccoli, B., 2006. Traffic Flow on Networks. American Institute of Mathematical Sciences.
Gazis, D.C., Knapp, C.H., 1971. On-line estimation of traffic densities from time-series of flow and speed data. Transp. Sci. 5 (3), 283–301.
Godunov, S.K., 1959. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sb. 89 (3), 271–306.
Gu, Y., Qian, Z.S., Chen, F., 2016. From Twitter to detector: real-time traffic incident detection using social media data. Transp. Res. Part C: Emerg. Technol. 67, 321–342.
Herrera, J.C., Work, D., Herring, R., Ban, X., Jacobson, Q., Bayen, A., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: the mobile century field experiment. Transp. Res. Part C: Emerg. Technol. 18 (4), 568–583.
Herty, M., Klar, A., 2003. Modeling, simulation, and optimization of traffic flow networks. SIAM J. Sci. Comput. 25 (3), 1066–1087.
Holden, H., Risebro, N.H., 1995. A mathematical model of traffic flow on a network of unidirectional roads. SIAM J. Math. Anal. 26 (4), 999–1017.
Jin, W., 2010. Continuous kinematic wave models of merging traffic flow. Transp. Res. Part B: Methodol. 44 (8), 1084–1103.
Kadirkamanathan, V., Li, P., Jaward, M.H., Fabri, S.G., 2002. Particle filtering-based fault detection in non-linear stochastic systems. Int. J. Syst. Sci. 33 (4), 259–265.
Kinoshita, A., Takasu, A., Adachi, J., 2015. Real-time traffic incident detection using a probabilistic topic model. Inform. Syst. 54, 169–188.
Lebacque, J.P., 2005. Intersection modeling, application to macroscopic network traffic flow models and traffic management. In: Traffic and Granular Flow. Springer, pp. 261–278.
LeVeque, R.J., 1992. Numerical Methods for Conservation Laws. Birkhäuser.
Li, X.R., Jilkov, V.P., 2003. Survey of maneuvering target tracking. Part I. Dynamic models. IEEE Trans. Aerosp. Electron. Syst. 39 (4), 1333–1364.
Li, X.R., Jilkov, V.P., 2005. Survey of maneuvering target tracking. Part V. Multiple-model methods. IEEE Trans. Aerosp. Electron. Syst. 41 (4), 1255–1321.
Li, P., Kadirkamanathan, V., 2001. Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems. IEEE Trans. Syst., Man Cybernet. Part C, Appl. Rev. 31 (3), 337–343.
Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 229, pp. 317–345.
Mihaylova, L., Boel, R., 2004. A particle filter for freeway traffic estimation. In: Proceedings of the IEEE Conference on Decision and Control, pp. 2106–2111.
Mihaylova, L., Hegyi, A., Gning, A., Boel, R.K., 2012. Parallelized particle and Gaussian sum particle filters for large-scale freeway traffic systems. IEEE Trans. Intell. Transp. Syst.
Nahi, N.E., 1973. Freeway-traffic data processing. Proc. IEEE 61 (5), 537–541.
Pan, B., Demiryurek, U., Shahabi, C., Gupta, C., 2013. Forecasting spatiotemporal impact of traffic incidents on road networks. In: Proceeding of the IEEE Conference on Data Mining, pp. 587–596.
Parkany, E., Xie, C., 2005. A Complete Review of Incident Detection Algorithms & Their Deployment: What Works and What Doesn't Technical Report NETCR37. University of Massachusetts Transportation Center.
Payne, H.J., Helfenbein, E.D., Knobel, H.C., 1976. Development and Testing of Incident Detection Algorithms Technical Report FHWA-RD-76-20. Research Methodology and Detailed Results, vol. 2. Technology Service Corporation.
Richards, P.I., 1956. Shock waves on the highway. Oper. Res. 4 (1), 42–51.
Ristic, B., Arulampalm, S., Gordon, N., 2004. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House Publishers.
Samant, A., Adeli, H., 2001. Enhancing neural network traffic incident-detection algorithms using wavelets. Comput.-Aided Civ. Infrastruct. Eng. 16 (4), 239–245.
Smulders, S., 1990. Control of freeway traffic flow by variable speed signs. Transp. Res. Part B: Methodol. 24 (2), 111–132.
Srinivasan, D., Jin, X., Cheu, R.L., 2005. Adaptive neural network models for automatic incident detection on freeways. Neurocomputing 64, 473–496.
Tafazoli, S., Sun, X., 2006. Hybrid system state tracking and fault detection using particle filters. IEEE Trans. Control Syst. Technol. 14 (6), 1078–1087.
Van Hinsbergen, C.P.I.J., Schreiter, T., Zuurbier, F.S., C Van Lint, J.W., Van Zuylen, H.J., 2012. Localized extended Kalman filter for scalable real-time traffic state estimation. IEEE Trans. Intell. Transp. Syst. 13 (1), 385–394.
Wang, R., 2016. <https://github.com/renwang/emmpf.git>.
Wang, Y., Papageorgiou, M., 2005. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. Transp. Res. Part B: Methodol. 39 (2), 141–167.
Wang, R., Work, D.B., 2014. Interactive multiple model ensemble Kalman filter for traffic estimation and incident detection. In: Proceedings of the IEEE International Conference on Intelligent Transportation Systems, pp. 804–809.
Wang, Y., Papageorgiou, M., Messmer, A., Coppola, P., Tzimitsi, A., Nuzzolo, A., 2009. An adaptive freeway traffic state estimator. Automatica 45 (1), 10–24.
Wang, J., Li, X., Liao, S.S., Hua, Z., 2013. A hybrid approach for automatic incident detection. IEEE Trans. Intell. Transp. Syst. 14 (3), 1176–1185.
Wang, R., Work, D.B., Sowers, R., 2016. Multiple model particle filter for traffic estimation and incident detection. IEEE Trans. Intell. Transp. Syst., 1–10 http://dx.doi.org/10.1109/TITS.2016.2560769.
Willsky, A., Chow, E., Gershwin, S., Greene, C., Houpt, P., Kurkjian, A., 1980. Dynamic model-based techniques for the detection of incidents on freeways. IEEE Trans. Autom. Control 25 (3), 347–360.
Work, D., Blandin, S., Tossavainen, O.P., Piccoli, B., Bayen, A.M., 2010. A traffic model for velocity data assimilation. Appl. Math. Res. eXpress 2010 (1), 1–35.
Yuan, F., Cheu, R.L., 2003. Incident detection using support vector machines. Transp. Res. Part C: Emerg. Technol. 11 (3), 309–328.