---

**Proceedings Paper:**
Hawes, M., Amer, H.M. and Mihaylova, L. orcid.org/0000-0001-5856-2223 (2016) Traffic State Estimation via a Particle Filter with Compressive Sensing and Historical Traffic Data. In: Information Fusion (FUSION), 2016 19th International Conference on. 19th International Conference on Information Fusion, 05-08 Jul 2016, Heidelberg, Germany. IEEE . ISBN 978-0-9964-5274-8

---

# Traffic State Estimation via a Particle Filter with Compressive Sensing and Historical Traffic Data

Matthew Hawes, Hayder M Amer, Lyudmila Mihaylova

Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK

{m.hawes,hmamer1,l.s.mihaylova}@sheffield.ac.uk

***Abstract*—In this paper we look at the problem of estimating traffic states within segments of road using a particle filter and traffic measurements at the segment boundaries. When there are missing measurements the estimation accuracy can decrease. We propose two methods of solving this problem by estimating the missing measurements by assuming the current measurements will approach the mean of the historical measurements from a suitable time period. The proposed solutions come in the form of an $l_1$ norm minimisation and a relevance vector machine type optimisation. Test scenarios involving simulated and real data verify that an accurate estimate of the traffic measurements can be achieved. These estimated missing measurements can then be used to help to improve traffic state estimation accuracy of the particle filter without a significant increase in computation time. For the real data used this can be up to a 23.44$\%$ improvement in RMSE values.**

## I. Introduction

Due to the increasing number of vehicles on the roads traffic state estimation and prediction is an important challenge that has to be addressed. However, modelling the traffic along stretches of motorways/roads is a complex problem with many interacting components and random perturbations [1]–[3]. For example, consider drivers in a traffic jam. As drivers approaching an incident observe the congestion forming in front of them they begin to slow down. The drivers following them see this change in speed and react in turn, resulting in a reduction in speed moving further up the road.

Models of varying levels of detail can be used. Microscopic models, [4], deal with state of individual vehicles, whereas macroscopic models, [3], [5]–[10], consider mean velocities and densities aggregated over time. As a result, macroscopic models are often employed in real time applications [2].

One such macroscopic model for motorways/freeways is the cell transmission model (CTM) [11]. In the CTM a length of road is split into a sequence of links. Each link can then be further separated in segments of road known as cells. The interactions between neighboring cells is then modelled by sending and receiving functions, which along with a maximum number of vehicles allowed in each cell controls the movement of vehicles between cells.

In [1] a flexible stochastic compositional model (SCM) is presented for online modelling of traffic flows. This is an extension of CTM which uses a dynamic equation to describe how traffic speeds evolve in each of the cells. The SCM is flexible in terms of the time update step and cell sizes, which can vary with time if required as long as no single vehicle will miss the subsequent cell during a time step. In this model the random nature of traffic state evolution can also be explicitly accounted for via probability distributions that govern the sending and receiving functions as well as noise terms.

With such models it is possible to recursively estimate the traffic states using Kalman filters (KFs) [12]–[15]. Alternatively particle filters (PFs), [16], [17], have also been successfully applied to traffic estimation problems [2], [4] and shown to be powerful and scalable. In such work past observations and the system dynamics are used to obtain the conditional distribution of the traffic state. It has been shown that when we do not have measurements available at all of the road segment boundaries that the estimation accuracy can decrease at the boundaries without measurements [2]. This raises the question can we get an estimate of what these measurements would be in order to improve the overall estimation accuracy of the filter?

Compressive sensing (CS), [18], [19], and Bayesian compressive sensing (BCS), [20], are methods that can be applied to beat the Nyquist sampling rate. It has also been shown that CS based approaches can be used for matrix completion in order to fill in missing data entries [21]. This has been applied in context of the traffic estimation problem [22], [23]. In these works data from probe vehicles is used, i.e. taxis equipped with global positioning system (GPS) to give their locations and velocities. However, there is no way to control how many taxis are on the roads or which roads they are on. As a result the missing data problem for traffic state estimation arises.

In this work we make the assumption that the current traffic state will approach the mean of the historical traffic states from a suitable period of time (unlike for the previous work where the missing data in time and space is directly estimated). As a result the problem can be formulated as an $l_1$ norm minimisation of the difference between this historical mean and the current traffic state estimate. In order to ensure accurate estimated measurement are achieved a constraint is added to ensure that the estimated traffic state matches the traffic state measurements that are available at given cell boundaries. This

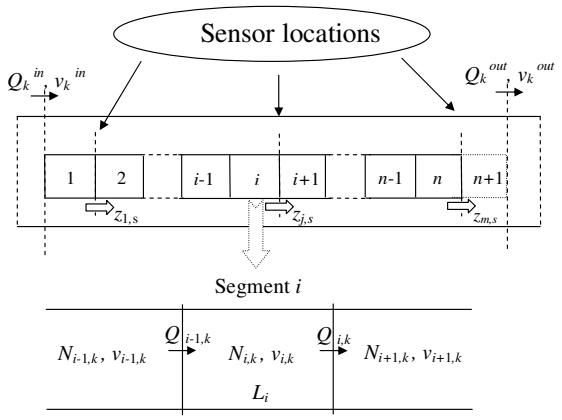


Fig. 1. Road segments and measurement points [2]. $Q_{i,k}$ is the number of vehicles crossing the boundary between segments $i$ and $i+1$ at time $k$, $N_{i,k}$ and $v_{i,k}$ the number of vehicles and average of the vehicles, respectively.

can then be further formulated in a relevance vector machine (RVM) type framework, [24], for improved efficiency. Note, the resulting algorithms require the historical measurements (or an estimate of them). As missing measurements are then estimated they can take the place of the missing measurements in historical data. The resulting CS and BCS based algorithms for estimating the missing measurements are tested with both simulated and real data and integrated with a PF for traffic state estimation.

The rest of this paper is structured in the following manner: Firstly, the traffic flow model is introduced in Section II (traffic model II-A and measurement model II-B). Then Section III introduces the methods of estimating the missing measurements. Two methods are considered, one based on CS (Section III-A) and the second on BCS (Section III-B). Section IV gives details of the PF used for traffic state estimation. Finally a performance evaluation is provided in Section V and conclusions are drawn in Section VI.

## II. Traffic Flow Model

### A. Traffic Model

In this work we make use of the SCM [1], where the road is split into segments as shown in Figure 1 and $L_i$ is the length of road segment $i$, where segment $i$ consists of $l_i$ lanes. We are interested in estimates of the traffic states at times $t_1, t_2, ..., t_k, ....$ The state vector is given by $\mathbf{x}_k = [\mathbf{x}_{1,k}^T, \mathbf{x}_{2,k}^T, ..., \mathbf{x}_{n,k}^T]^T$, $\mathbf{x}_{i,k} = [N_{i,k}, v_{i,k}]^T$, where $N_{i,k}$ and $v_{i,k}$ are the number of vehicles and their average speed, respectively and $n+1$ is the fictitious last road segment. Finally we assume vehicles have an average length of $A_l$.

We can describe the evolution of the traffic states using the following equations:

$$\mathbf{x}_{1,k+1} = f_1(Q_k^{in}, v_k^{in}, \mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \boldsymbol{\eta}_{1,k}), \quad (1)$$

$$\mathbf{x}_{i,k+1} = f_i(\mathbf{x}_{i-1,k}, \mathbf{x}_{i,k}, \mathbf{x}_{i+1,k}, \boldsymbol{\eta}_{i,k}), \quad (2)$$

$$\mathbf{x}_{n,k+1} = f_n(\mathbf{x}_{n-1,k}, \mathbf{x}_{n,k}, Q_k^{out}, v_k^{out}, \boldsymbol{\eta}_{n,k}), \quad (3)$$

where $f_i$ is specified by the traffic model and $\boldsymbol{\eta}_k$ allows for random fluctuations and modelling error. In equations (1)-(3) $Q_k^{in}$ and $Q_k^{out}$, are the vehicles entering the first segment and leaving the last segment within the time interval $\Delta t_k = t_{k+1} - t_k$ with average speeds $v_k^{in}$ and $v_k^{out}$, respectively. Note, these are the boundary conditions and not traffic states to be estimated. The traffic behaviour is modelled with forward and backward propagation of traffic perturbations. This model is summarised in Algorithm 1 and the interested reader can find further details in [1]. Note, in Algorithm 1 $S_{i,k}$ and $R_{i,k}$ are the sending and receiving functions, respectively. The sending functions determine the number of vehicles that can leave a road segment, while the receiving function determines the number that can enter. Finally, $\rho_{i,k+1}^{antic}$ is an anticipated traffic density as a result of mixing densities from two neighbouring cells, $\rho_{th}$ is a threshold value for the road traffic density and $v_{i,k+1}^{interm}$ is an intermediate traffic velocity (intermediate since it can be seen as kind of mixing velocities from neighboring cells).

### B. Measurement model

There are then sensors, e.g. magnetic loops, radar or video cameras, on the boundaries of various road segments. Measurements of the number of vehicles crossing the segment boundaries and their speeds are made at the discrete time points of interest, given by $t_s$. The result of this is the measurement vector given by $\mathbf{z}_s = [\mathbf{z}_{1,s}^T, \mathbf{z}_{2,s}^T, ..., \mathbf{z}_{m,s}^T]^T$ where there are measurements made at $m$ boundaries and $\mathbf{z}_{j,s} = [\bar{Q}_{j,s}, \bar{v}_{j,s}]^T$.

Given the measurement equation

$$\mathbf{z}_s = h(\mathbf{x}_s, \boldsymbol{\xi}_s), \quad (4)$$

where $h(.)$ is determined by the measurement model used. If we know the distribution of the initial state vector then the traffic state estimation problem becomes a recursive Bayesian estimation problem and can be solved with a PF (see Section IV). In this work we assume $\boldsymbol{\xi}_s = [\xi_{Q_{j,s}}, \xi_{v_{j,s}}]^T$ is a Gaussian measurement noise giving:

$$\mathbf{z}_{j,s} = \begin{pmatrix} \bar{Q}_{j,s} \\ \bar{v}_{j,s} \end{pmatrix} + \boldsymbol{\xi}_s. \quad (5)$$

## III. Missing Measurement Estimation

### A. Compressive Sensing

At a given time $t_s$, the actual measurements are given by $\mathbf{z}_s = [\mathbf{z}_{1,s}^T, \mathbf{z}_{2,s}^T, ..., \mathbf{z}_{n,s}^T]^T$, where $n$ is the number of road segment boundaries. However, not all of these measurements will be available at time $t_s$. Instead we can estimate these missing measurements. Firstly, consider a measurement matrix given by

$$\mathbf{m}_s = \mathbf{b}_s \circ \mathbf{z}_s, \quad (6)$$

where $\circ$ is the Hadamard product,

$$\mathbf{b}_s = [b_{s,1}, b_{s,2}, ..., b_{s,2n}]^T \quad (7)$$

**Algorithm 1** The Traffic Model [2]

1: *Forward wave*:
For $i = 1, 2, \ldots, n$

$$S_{i,k} = max\left(N_{i,k}\frac{v_{i,k}.\Delta t_k}{L_i} + \eta_{S_i,k},\ N_{i,k}\frac{v_{min}.\Delta t_k}{L_i}\right)$$

and set $Q_{i,k} = S_{i,k}$.
End For

2: *Backward wave*:
For $i = n, n-1, \ldots, 1$

$$R_{i,k} = N^{max}_{i+1,k} - N_{i+1,k} + Q_{i+1,k},$$

where

$$N^{max}_{i+1,k} = (L_{i+1}\ell_{i+1,k})/(A_\ell + v_{i+1,k}t_d),$$

if $S_{i,k} < R_{i,k}$, $Q_{i,k} = S_{i,k}$ else $Q_{i,k} = R_{i,k}$,
$v_{i,k} = Q_{i,k}L_i/(N_{i,k}\Delta t_k)$.
End For

3: Update the number of vehicles inside segments:
For $i = 1, 2, \ldots, n$

$$N_{i,k+1} = N_{i,k} + Q_{i-1,k} - Q_{i,k}.$$

End For

4: Update the density:
For $i = 1, 2, \ldots, n$

$$\begin{aligned}\rho_{i,k+1} &= N_{i,k+1}/(L_i\ell_{i,k+1}),\\ \rho^{antic}_{i,k+1} &= \alpha\rho_{i,k+1} + (1-\alpha)\rho_{i+1,k+1}.\end{aligned}$$

End For

5: Update of the speed:
For $i = 1, 2, \ldots, n$

$$v^{interm}_{i,k+1} = \begin{cases}\frac{v_{i-1,k}Q_{i-1,k} + v_{i,k}(N_{i,k}-Q_{i,k})}{N_{i,k+1}},\ N_{i,k+1} \neq 0,\\ v_f,\ \text{otherwise},\end{cases}$$

$$\begin{aligned}v^{interm}_{i,k+1} &= max(v^{interm}_{i,k+1}, v_{min})\\ v_{i,k+1} &= \beta_{k+1}v^{interm}_{i,k+1} + (1-\beta_{k+1})v^e(\rho^{antic}_{i,k+1})\\ &+ \eta_{v_i,k+1},\end{aligned}$$

$$\beta_{k+1} = \begin{cases}\beta^I, & |\rho^{antic}_{i+1,k+1} - \rho^{antic}_{i,k+1}| \geq \rho_{th},\\ \beta^{II} & \text{otherwise.}\end{cases}$$

End For

and

$$b_{s,i} = \begin{cases}1, & \text{measurements available,}\\ 0 & \text{measurements unavailable.}\end{cases} \quad (8)$$

This measurement matrix can now be used to gain an estimate of the current measurements $\hat{\mathbf{z}}_s$.

We assume that the current measurements at the segment boundaries will be close to the mean of the historical measurements (or corresponding estimates) over a suitable period of time, defined by the length $t_h$. This is given by

$$\tilde{\mathbf{z}}_s = \tilde{\boldsymbol{\phi}}_s\tilde{\boldsymbol{\theta}}_s, \quad (9)$$

where $\tilde{\boldsymbol{\theta}}_s = [1/t_h, 1/t_h, ..., 1/t_h]^T$ ($\tilde{\boldsymbol{\theta}}_s \in \mathbb{R}^{t_h \times 1}$) and $\tilde{\boldsymbol{\phi}}_s$ is the relevant historical measurements/estimates, given by

$$\tilde{\boldsymbol{\phi}}_s = [\boldsymbol{\phi}_{s-t_h}, \boldsymbol{\phi}_{s-t_h+1}, ..., \boldsymbol{\phi}_s] \quad (10)$$

and

$$\boldsymbol{\phi}_j = [\boldsymbol{\phi}^T_{1,j}, \boldsymbol{\phi}^T_{2,j}, ..., \boldsymbol{\phi}^T_{n,j}]^T, \quad (11)$$

$$\boldsymbol{\phi}_{i,j} = \begin{cases}\mathbf{z}_{i,j}, & \text{measurements available,}\\ \hat{\mathbf{z}}_{i,j} & \text{measurements unavailable.}\end{cases} \quad (12)$$

This gives us the following problem

$$\min ||\tilde{\mathbf{z}}_s - \hat{\mathbf{z}}_s||_1, \quad (13)$$

where $||.||_1$ is the $l_1$ norm. Minimising the $l_0$ norm would give the smallest amount of non-zero values for $\tilde{\mathbf{z}}_s - \hat{\mathbf{z}}_s$. However, this can not be achieved in practice and the $l_1$ norm is used as an approximation [18], [19]. Note, $||\tilde{\mathbf{z}}_s - \hat{\mathbf{z}}_s||_1$ can be written as follows

$$\begin{aligned}||\tilde{\mathbf{z}}_s - \hat{\mathbf{z}}_s||_1 &= ||\tilde{\boldsymbol{\phi}}_s\tilde{\boldsymbol{\theta}}_s - \tilde{\boldsymbol{\phi}}_s\hat{\boldsymbol{\theta}}_s||_1 \quad (14)\\ &= ||\tilde{\boldsymbol{\phi}}_s(\tilde{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_s)||_1 = |\tilde{\boldsymbol{\phi}}_s|||\tilde{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_s||_1.\end{aligned}$$

Therefore, as $|\tilde{\boldsymbol{\phi}}_s|$ is constant at a given time the minimisation in (13) can be achieved by $\min_{\hat{\boldsymbol{\theta}}_s} ||\tilde{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_s||_1$.

However, this will always aim to have $\tilde{\boldsymbol{\theta}}_s = \hat{\boldsymbol{\theta}}_s$. As a result a constraint has to be added to ensure that the estimated measurements do not disagree with the available measurements matrix. In other words we want to place a limit on $||\mathbf{m}_s - \mathbf{b}_s \circ \hat{\mathbf{z}}_s||_2 = ||\mathbf{m}_s - (\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)\hat{\boldsymbol{\theta}}_s||_2$, where $\mathbf{B} = [\mathbf{b}_s, \mathbf{b}_s, ..., \mathbf{b}_s]$ ($\mathbf{B} \in \mathbb{R}^{t_h \times 1}$). This results in:

$$\min_{\hat{\boldsymbol{\theta}}_s} ||\tilde{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_s||_1 \ \text{ subject to } \ ||\mathbf{m}_s - (\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)\hat{\boldsymbol{\theta}}_s||_2 \leq \varepsilon. \quad (15)$$

Here the constant $\varepsilon$ in the added constraint places a limit on the error between the available measurement vector and the corresponding estimated measurements. The final estimate is then given by

$$\hat{\mathbf{z}}_{s,CS} = \tilde{\boldsymbol{\phi}}_s\hat{\boldsymbol{\theta}}_{\boldsymbol{s}}. \quad (16)$$

### *B. Bayesian Compressive Sensing*

Alternatively, the problem can be formulated in a Bayesian framework. Firstly, we know

$$\mathbf{m}_s = (\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)\hat{\boldsymbol{\theta}}_s + \mathbf{e}_s, \quad (17)$$

where we assume $\mathbf{e}_s$ to be Gaussian noise with a variance $\sigma^2$. The solution is then found by evaluating

$$\hat{\boldsymbol{\theta}}_{s,BCS} = \max \mathcal{P}(\hat{\boldsymbol{\theta}}_s, \sigma^2, \mathbf{p}_s|\mathbf{m}_s, \tilde{\boldsymbol{\theta}}_s), \quad (18)$$

where $\mathbf{p}_s = [p_{s,1}, p_{s,2}, ..., p_{s,2n}]^T$are hyperparameters to be estimated.

As per (17) the likelihood is given by

$$\mathcal{P}(\mathbf{m}_s|\hat{\theta}_s,\sigma^2)=(2\pi\sigma^2)^{-n}\exp\Big\{-\frac{1}{2\sigma^2}||\mathbf{m}_s-(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)\hat{\theta}_s||_2^2\Big\}. \tag{19}$$

We further assume that the values of $\hat{\theta}_s$ will be likely to be close to those of $\tilde{\theta}_s$, which gives us the prior distribution

$$\begin{aligned}\mathcal{P}(\hat{\theta}_s|\mathbf{p}_s,\tilde{\theta}_s) &= (2\pi)^{-t_h/2}|\mathbf{P}_s|^{1/2} \\ &\times \exp\Big\{-\frac{1}{2}(\hat{\theta}_s-\tilde{\theta}_s)\mathbf{P}_s(\hat{\theta}_s-\tilde{\theta}_s)^T\Big\}.\end{aligned} \tag{20}$$

Here, $|\mathbf{P}_s|$ is the of determinant $\mathbf{P}_s=\text{diag}(\mathbf{p}_s)$.

Now place independent Gamma priors on the hyperparameters $p_{s,i}$ giving

$$\mathcal{P}(\mathbf{p}_s)=\prod_{i=1}^{2n}G(p_{s,i}|a,b). \tag{21}$$

A further Gamma prior can also be used for $\sigma^2$

$$\mathcal{P}(\sigma^2)=G(\sigma^{-2}|c,d), \tag{22}$$

where $a,b,c$ and $d$ are scale and shape priors.

With these definitions we can now find the solution to (18) by following a RVM type framework [24]. We know that

$$\mathcal{P}(\hat{\theta}_s,\sigma^2,\mathbf{p}_s|\mathbf{m}_s,\tilde{\theta}_s)=\mathcal{P}(\hat{\theta}_s|\mathbf{m}_s,\sigma^2,\mathbf{p}_s,\tilde{\theta}_s)\mathcal{P}(\mathbf{p}_s,\sigma^2|\mathbf{m}_s) \tag{23}$$

and

$$\begin{aligned}\mathcal{P}(\hat{\theta}_s|\mathbf{m}_s,\sigma^2,\mathbf{p}_s,\tilde{\theta}_s) &= \frac{\mathcal{P}_s(\mathbf{m}_s|\hat{\theta}_s,\sigma^2)\mathcal{P}(\hat{\theta}_s|\mathbf{p}_s,\hat{\theta}_s)}{\mathcal{P}(\mathbf{m}_s|\mathbf{p}_s,\sigma^2,\tilde{\theta}_s)} \\ &= (2\pi)^{-t_h/2}|\mathbf{\Sigma}_s|^{-1/2}\exp\Big\{-\frac{1}{2} \\ &\times (\hat{\theta}_s-\boldsymbol{\mu}_s)^T\mathbf{\Sigma}_s^{-1}(\hat{\theta}_s-\boldsymbol{\mu}_s)\Big\},\end{aligned} \tag{24}$$

where $\mathbf{\Sigma}_s$ and $\boldsymbol{\mu}_s$ are the covariance matrix and mean vector given by

$$\mathbf{\Sigma}_s=(\sigma^{-2}(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)^T(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)+\mathbf{P}_s)^{-1}, \tag{25}$$

and

$$\boldsymbol{\mu}_s=\mathbf{\Sigma}_s(\sigma^{-2}(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)^T\mathbf{m}_s+\mathbf{P}_s\tilde{\theta}_s), \tag{26}$$

respectively.

Following a similar method to [24] we have

$$\mathcal{P}(\sigma^2,\mathbf{p}_s|\mathbf{m}_s)\approx\mathcal{P}(\mathbf{m}_s|\mathbf{p}_s,\sigma^2,\tilde{\theta}_s)\mathcal{P}(\mathbf{p}_s)\mathcal{P}(\sigma^2), \tag{27}$$

where if we have $a=b=c=d=10^{-4}$ then $\mathcal{P}(\mathbf{p}_s)$ and $\mathcal{P}(\sigma^2)$ are non-informative [24]. As a result, maximising $\mathcal{P}(\sigma^2,\mathbf{p}_s|\mathbf{M}_s)$ is equivalent to maximising $\mathcal{P}(\mathbf{m}_s|\mathbf{p}_s,\sigma^2,\tilde{\theta}_s)$. This can be achieved by maximising

$$\begin{aligned}\mathcal{L}(\mathbf{p}_s,\sigma^2) &= \log\Big\{(2\pi\sigma^2)^{-t_h/2}|\mathbf{\Sigma}_s|^{\frac{1}{2}}|\mathbf{P}_s|^{\frac{1}{2}}\exp\Big(-\frac{1}{2} \\ &\times (\mathbf{m}_s^T\mathbf{C}\mathbf{m}_s+\tilde{\theta}_s^T\mathbf{D}_s\tilde{\theta}_s \\ &- 2\sigma^2\mathbf{m}_s^T(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)\mathbf{\Sigma}_s\mathbf{P}_s\tilde{\theta}_s)\Big)\Big\} \\ &= -\frac{1}{2}\Big(t_h\log(2\pi)+t_h\log\sigma^2-\log|\mathbf{\Sigma}_s|- \\ &\log|\mathbf{P}_s|+\sigma^{-2}||\mathbf{m}_s-(\mathbf{B}_s\circ\boldsymbol{\phi}_s)\boldsymbol{\mu}_s||_2^2 \\ &+ \boldsymbol{\mu}_s^T\mathbf{P}\boldsymbol{\mu}_s+\tilde{\theta}_s^T\mathbf{P}_s\tilde{\theta}_s-\tilde{\theta}_s^T\mathbf{P}_s\boldsymbol{\mu}_s\Big),\end{aligned} \tag{28}$$

where $\mathbf{C}_s=(\sigma^2\mathbf{I}+(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}})\mathbf{P}_s^{-1}(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)^T)^{-1}$ and $\mathbf{D}_s=\mathbf{P}_s-\mathbf{P}_s^T\mathbf{\Sigma}_s\mathbf{P}_s$.

By differentiating (28) with respect to $p_{s,i}$ and $\sigma^{-2}$ it possible to get the update equations for the precision hyperparameters and variance, respectively. This gives us

$$p_{s,i}^{new} = \frac{\gamma_{s,i}}{\mu_{s,i}^2+\tilde{\theta}_{s,i}-\tilde{\theta}_{s,i}\mu_{s,i}}, \tag{29}$$

$$\sigma_{new}^2 = \frac{||\mathbf{m}_s-(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)\boldsymbol{\mu}_s||_2^2}{t_h-\sum_i\gamma_{s,i}}, \tag{30}$$

where $\gamma_{s,i}=1-p_{s,i}\Sigma_{s,ii}$, $\Sigma_{s,ii}$ is the $i^{th}$ diagonal element of $\mathbf{\Sigma}_s$.

The optimisation is then achieved by iteratively finding $\mathbf{\Sigma}_s$ and $\boldsymbol{\mu}_s$, followed by $p_{s,i}^{new}$ and $\sigma_{new}^2$ until a convergence criterion is met. To obtain the final estimates of $\hat{\theta}_s$ the optimised values of $\mathbf{p}_s$ and $\sigma^2$ are put into (26) to give

$$\begin{aligned}\hat{\mathbf{z}}_{s,BCS} &= \Big(\frac{(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)^T(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}}_s)}{\sigma_{opt}^2}+\mathbf{P}_{s,opt}\Big)^{-1} \\ &\times \Big(\frac{(\mathbf{B}_s\circ\tilde{\boldsymbol{\phi}})^T\mathbf{m}_s}{\sigma_{opt}^2}+\mathbf{P}_{s,opt}\tilde{\theta}_s\Big).\end{aligned} \tag{31}$$

Either the estimates $\hat{\mathbf{z}}_{s,CS}$ or $\hat{\mathbf{z}}_{s,BCS}$ can then be used to replace the available measurements used with in a PF. Such a scheme is detailed in the next section.

## IV. Particle Filtering Framework for Traffic State Estimation

Here the aim is to find the posterior probability density function (PDF) of the state at time $t_k$ given a set of measurements up to the same point in time. In other words we want to evaluate $p(\mathbf{x}_k|\hat{\mathbf{Z}}^k)$, where $\hat{\mathbf{Z}}^k=[\hat{\mathbf{z}}_1,....,\hat{\mathbf{z}}_k]$ and $\hat{\mathbf{z}}_i$ for $i=1,...,k$ is estimated using (16) or (31). From Bayes rule

$$p(\mathbf{x}_k|\hat{\mathbf{Z}}^k)=\frac{p(\hat{\mathbf{z}}_k|\mathbf{x}_k)p(\mathbf{x}_k|\hat{\mathbf{Z}}^{k-1})}{p(\hat{\mathbf{z}}_k|\hat{\mathbf{Z}}^{k-1})}, \tag{32}$$

where

$$p(\mathbf{x}_k|\hat{\mathbf{Z}}^{k-1})=\int_{\mathbb{R}^{n_x}}p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\hat{\mathbf{Z}}^{k-1})d\mathbf{x}_{k-1} \tag{33}$$

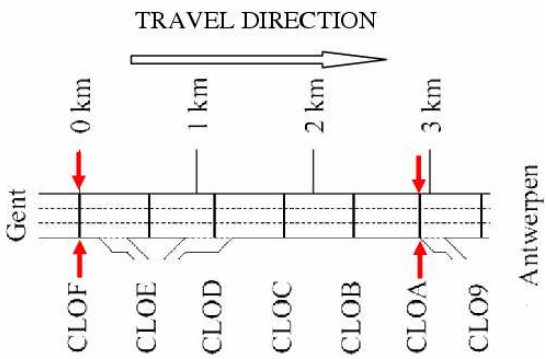


Fig. 2. Schematic of Belgium freeway considered [2]. CLOF-CLO9 show the locations of cameras used to make the traffic measurements

and $p(\hat{\mathbf{z}}_k|\hat{\mathbf{Z}}^{k-1})$ is a normalising constant. This means $p(\mathbf{x}_k|\hat{\mathbf{Z}}^{k})$ can be updated using the following proportionality relationship:

$$p(\mathbf{x}_k|\hat{\mathbf{Z}}^{k}) \propto p(\hat{\mathbf{z}}_k|\mathbf{x}_k)p(\mathbf{x}_k|\hat{\mathbf{Z}}^{k-1}). \tag{34}$$

This recursive estimation is computationally expensive which is why PFs are used to give an approximate solution [16], [17]. Algorithm 2 gives the PF (with $M_{pf}$) for traffic state estimation that is considered in this work. We refer the interested reader to [2] for further details. The difference between this work and the algorithm shown here is the inclusion of the measurement estimation step which has been detailed in the section above. Note the inclusion of $t_k \equiv t_s$ is to account for the fact that we do not necessarily have measurements available at every time step within the particle filter.

## V. Performance Evaluation

In this section we will provide a performance evaluation of the proposed algorithms. This will be formed of two parts: Firstly simulated and real data will be used to test how well (16) and (31) can fill in the missing measurements at road segment boundaries. Note, (16) is solved using cvx [25], [26]. Then a PF with and without estimated measurements will be used to estimate the traffic states for the real data. All comparisons are implemented in Matlab on a computer with an Intel Xeon CPU E3-1271 (3.60GHz) and 16GB of RAM.

The simulated data comes from the SUMO traffic simulator [27]. We simulated two 1km lanes of traffic travelling in one direction with a maximum speed of 25m/s. An induction loop was placed every 0.5km to take the segment boundary measurements every 30 seconds.

Figure 2 shows the section of freeway considered between Ghent and Antwerp in Belgium. The labels CLOF-CLO9 refer to the traffic cameras at road segment boundaries on the section of road being considered. Theses cameras record the number of vehicles passing the boundaries in 1 minute intervals and their average speeds.

**Algorithm 2** Particle Filter with CS/BCS Estimated Measurements for Traffic State Estimation [2]

1: Initialization: $k = 0$
For $l = 1, \ldots, M_{pf}$
Generate samples $\{\mathbf{x}_0^{(l)}\}$ from the initial distribution $p(\mathbf{x}_0)$ and initial weights $w_0^{(l)} = 1/M_{pf}$.
End For

2: Prediction step:
For $l = 1, \ldots, M_{pf}$,
sample $\mathbf{x}_k^{(l)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(l)})$ according to (4)-(11) for segments between two boundaries where measurements arrive
End For

3: Missing measurement estimation (only for $t_k \equiv t_s$): Obtain the estimated traffic measurements, using either

$$\min_{\hat{\boldsymbol{\theta}}_s} ||\tilde{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_s||_1 \text{ subject to } ||\mathbf{m}_s - (\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)\hat{\boldsymbol{\theta}}_s||_2 \leq \varepsilon,$$

$$\hat{\mathbf{z}}_s = \tilde{\boldsymbol{\phi}}_s \hat{\boldsymbol{\theta}}_{\boldsymbol{s}}.$$

or

$$\hat{\mathbf{z}}_s = \Big(\frac{(\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)^T(\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}}_s)}{\sigma_{opt}^2} + \mathbf{P}_{s,opt}\Big)^{-1} \times \Big(\frac{(\mathbf{B}_s \circ \tilde{\boldsymbol{\phi}})^T \mathbf{m}_s}{\sigma_{opt}^2} + \mathbf{P}_{s,opt}\tilde{\boldsymbol{\theta}}_s\Big).$$

4: Estimated measurement processing step (only for $t_k \equiv t_s$) compute the weights:
For $l = 1, \ldots, M_{pf}$
$w_s^{(l)} = w_{s-1}^{(l)} p(\hat{\mathbf{z}}_s|\mathbf{x}_s^{(l)})$,
End For
where the likelihood $p(\hat{\mathbf{z}}_s|\mathbf{x}_s^{(l)})$ is calculated by the model (5) from Section II-B.
For $l = 1, \ldots, M_{pf}$
Normalize the weights: $\hat{w}_s^{(l)} = w_s^{(l)} / \sum_{l=1}^{M_{pf}} w_s^{(l)}$.
End For

5: Output: $\hat{\mathbf{x}}_s = \sum_{l=1}^{M_{pf}} \hat{w}_s^{(l)} \mathbf{x}_s^{(l)}$,

6: Selection step (resampling) only for $t_k \equiv t_s$:
Multiply/ Suppress samples $\mathbf{x}_s^{(l)}$ with high/ low importance weights $\hat{w}_s^{(l)}$, in order to obtain $M$ random samples approximately distributed according to $p(\mathbf{x}_s^{(l)}|\hat{\mathbf{Z}}^{s})$, e.g. by residual resampling.
For $l = 1, \ldots, M_{pf}$,
$w_s^{(l)} = \hat{w}_s^{(l)} = 1/M_{pf}$,
End For

7: $k \leftarrow k + 1$ and return to step (1).

### A. Evaluation of CS and BCS based measurement estimation methods

For the SUMO simulator we assume that there are measurements available at all of the loop locations for $t_h = 25$ time instances. After this we then assume that there only measurements available at the first and last loop location.

TABLE I
Performance summary for the CS and BCS based measurement estimation methods with simulated data.

| Method | CS | BCS |
|---|---|---|
| Computation time (s) total (per snapshot) | 3.79 (0.39) | 0.81 (0.04) |
| $\max(\|\|\mathbf{z}_{1,s} - \hat{\mathbf{z}}_{1,F,s}\|\|_2)$ | 6.14 | 5.08 |
| $\overline{\|\|\mathbf{z}_{1,s} - \hat{\mathbf{z}}_{1,F,s}\|\|_2}$ | 1.89 | 2.21 |
| $\max(\|\|\mathbf{z}_{2,s} - \hat{\mathbf{z}}_{2,F,s}\|\|_2)$ | 5.16 | 5.15 |
| $\overline{\|\|\mathbf{z}_{2,s} - \hat{\mathbf{z}}_{2,F,s}\|\|_2}$ | 2.31 | 2.60 |

TABLE II
Performance summary for the CS and BCS based measurement estimation methods with real data.

| Method | CS | BCS |
|---|---|---|
| Computation time (s) total (per snapshot) | 6.44 (0.40) | 0.70 (0.03) |
| $\max(\|\|\mathbf{z}_{1,s} - \hat{\mathbf{z}}_{1,F,s}\|\|_2)$ | 9.74 | 8.04 |
| $\overline{\|\|\mathbf{z}_{1,s} - \hat{\mathbf{z}}_{1,F,s}\|\|_2}$ | 6.22 | 4.85 |
| $\max(\|\|\mathbf{z}_{2,s} - \hat{\mathbf{z}}_{2,F,s}\|\|_2)$ | 28.56 | 24.43 |
| $\overline{\|\|\mathbf{z}_{2,s} - \hat{\mathbf{z}}_{2,F,s}\|\|_2}$ | 15.56 | 9.62 |

For the real data we initially have measurements available at CLOE-CLOB and start at a time of 5pm. When the estimates have been found they can then replace the measurements at the current time instance and the process repeated for the desired length of time (20 time steps in total).

Firstly, 100 independent sets of estimates were found using (31) and a representative example selected. This was achieved using initial estimates of the hyperparameters as $p_{s,i} = (2n)^{-2}$, where $2n$ gives the number of measurements, and initial estimate of the variance for the Gaussian noise as $\sigma^2 = 0.1$. Then from the representative sample a value of $\varepsilon$ for use in (16) can be found to allow fair comparison.

Table I summarises the performance of the two methods for the simulated data, where $||\mathbf{z}_{1,s} - \hat{\mathbf{z}}_{1,F,s}||_2$ and $||\mathbf{z}_{2,s} - \hat{\mathbf{z}}_{2,F,s}||_2$, for $s = t_{init}, ..., t_{init+t_h}$ and $F = \{CS, BCS\}$, is used to indicate the estimation accuracy. Note, the subscript 1 refers to the measurements related to vehicle speed and the subscript 2 for the number of vehicles. From this we can see that both methods give a comparable performance in terms of estimation accuracy. For the CS based method this relates to the speed estimate always being within 3.66 m/s of the actual measurement and 3.54 of the actual vehicle count. Whereas, for the BCS based method the estimates are within 4.12m/s of the speed measurements and within 3.48 vehicles of the actual vehicle count. However, Table I shows that the BCS based method is computationally more efficient. This is also for the case with real data shown in Table II, where we can also see the BCS based method has also given an improved estimation accuracy compared to the CS based method. In this instance for the BCS based estimates are within 7.14km/h and 22.88 vehicles of the actual measurements. Whereas, for the CS based method the estimates are within 6.85km/h and 25 vehicles of the actual traffic measurements.

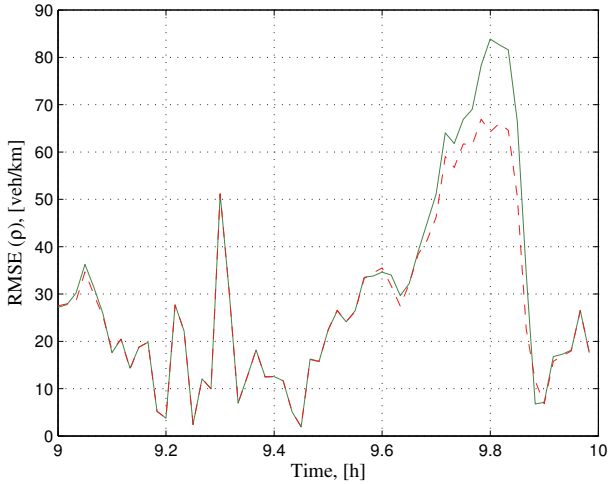


Fig. 3. Traffic density RMSE for CLOE, the solid line is for the PF using 2 measurements only and the dashed line with the BCS estimated measurements.

### *B. Evaluation of Traffic State Estimation performance*

Now we will consider how using estimated measurements effects the performance of estimating the traffic states within the road segments using a PF with $M_{pf} = 200$ particles. As we have previously shown that the BCS and CS based methods have a similar measurement estimation accuracy but that the BCS method is more efficient, we will only use the BCS based measurement estimation method in what follows.

The performance will be tested using the real data from Belgium over the period of an hour. We consider time steps of 10 seconds, where measurements are available every minute. The following parameters where used in the traffic model and PF: $v_{free} = 120km/h$, $v_{min} = 7.4km/h$, $\rho_{crit} = 20.89veh/km/lane$, $\rho_{jam} = 180veh/km$, $A_l = 0.01km$, $\sigma^2_{\xi_{Q_{j,s}}} = 1$ and $\sigma^2_{\xi_{Q_{j,s}}} = 3.24$.

Note, $r = 100$ independent Monte Carlo runs are completed. For a performance measure of the accuracy of the PF we consider the root mean square error (RMSE) as calculated in (35), where $j = 1$ for the speed related measurements and $j = 2$ for the number of vehicles/density related measurements. Here $\mathbf{z}_{i,k}$ is the actual measurements and $\hat{\mathbf{z}}_{i,k}$ the predicted measurements (found from traffic model and PF).

$$RMSE_{j,k} = \frac{1}{r}\sum_{i=1}^{r}(\mathbf{z}_{j,i,k} - \hat{\mathbf{z}}_{j,i,k})^T(\mathbf{z}_{j,i,k} - \hat{\mathbf{z}}_{j,i,k}). \quad (35)$$

We compare the performance for CLOE and CLOC to illustrate the effects on performance in segments were there was originally measurements available and unavailable respectively. Figures 3-6 show the changing RMSEs and Table III summarises the performances along with computation times. These show an improvement in estimation accuracy has been achieved by using the BCS measurement estimation method. This has been at the cost of a slight increase in computation time. However, the increase has not been significant enough to

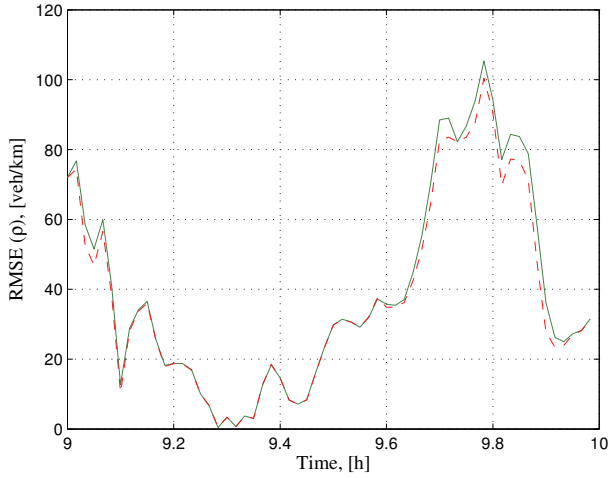


Fig. 4. Traffic density RMSE for CLOC, the solid line is for the PF using 2 measurements only and the dashed line with the BCS estimated measurements.

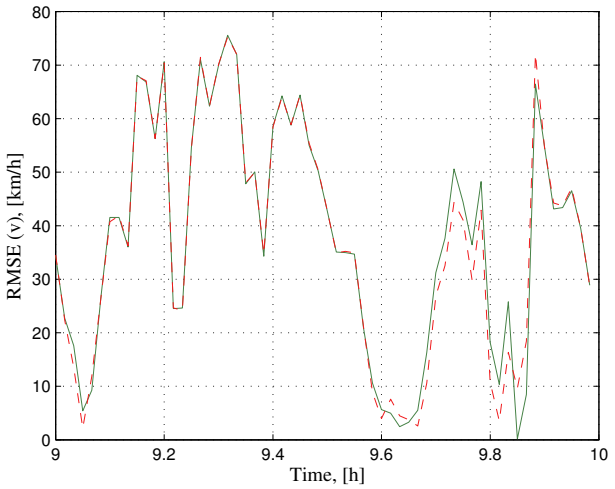


Fig. 5. Traffic velocity RMSE for CLOE, the solid line is for the PF using 2 measurements only and the dashed line with the BCS estimated measurements.

be a concern for real time implementation. The flow-density diagrams are plotted for the estimates from the PF with 2 measurements available and the BCS measurements available are shown in Figures 7-8. Both show the expected shape and are similar in appearance, further validating the effectiveness of the proposed method.

## VI. Conclusions

In this paper we have proposed two solutions to the problem of missing traffic measurements. We make the assumption that the current traffic measurements will be similar to the mean of the historical measurements from a suitable period of time. This can be assured by formulating the problem as an $l_1$ norm minimisation which is carried out subject to ensuring the estimates give an acceptable approximation of the available

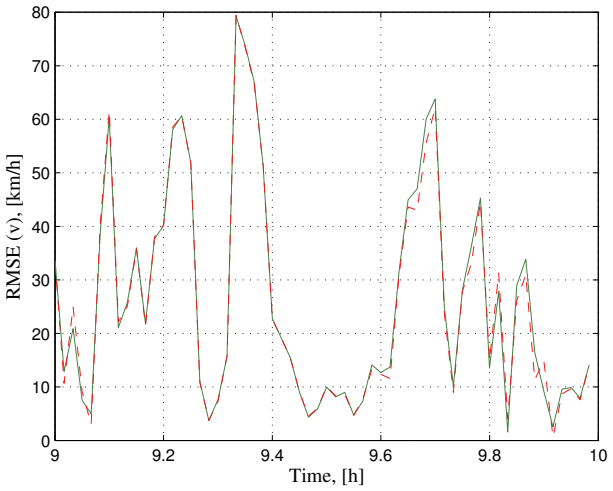


Fig. 6. Traffic velocity RMSE for CLOC, the solid line is for the PF using 2 measurements only and the dashed line with the BCS estimated measurements.

TABLE III
Performance summary for PF with 2 measurements available and the BCS estimated measurements.

| Example | 2 measurements | BCS estimated measurements |
|---|---|---|
| Computation time (s) total (per snapshot) | 6.66 (0.11) | 13.28 (0.47) |
| $\overline{RMSE_\rho}$ CLOE (CLOC) | 29.67 (39.53) | 27.51 (37.55) |
| $\overline{RMSE_v}$ CLOE (CLOC) | 38.11 (26.09) | 37.46 (25.78) |

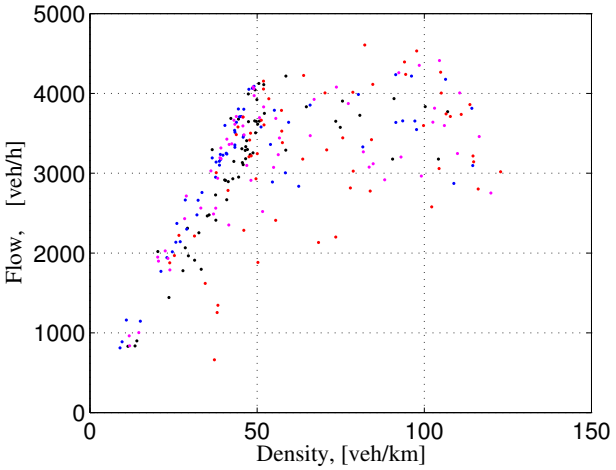


Fig. 7. Flow-density diagram for the PF with 2 measurements available.

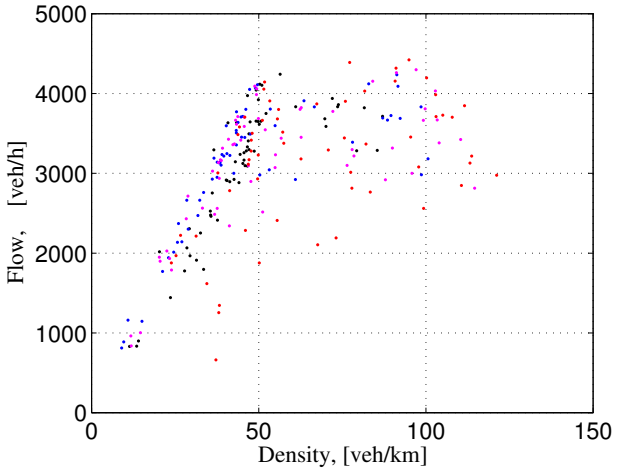


Fig. 8. Flow-density diagram for the PF with the BCS estimated measurements.

traffic measurements. Then we further formulate the problem in a Bayesian framework, deriving the a posterior distributions and marginal likelihood that are optimised using an RVM type framework. These methods can then be combined with a PF and SCM for traffic. The proposed methods are tested with simulated and real data to verify their effectiveness. We show that it is possible to get accurate estimates of the missing measurements which when used with the PF can give improved accuracy in terms of state estimation accuracy without a significant increase in computation time. For the real data considered in this paper up to a 23.44% improvement in RMSE values has been achieved.

## ACKNOWLEDGMENTS

We appreciate the support of the UK Engineering and Physical Sciences Research Council (EPSRC) via the project Bayesian Tracking and Reasoning over Time (BTaRoT) grant EP/K021516/1, EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems (TRAX) Grant agreement no.: 607400 and of the SETA project funded from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 688082. We also thank the Vlaams Verkeerscentrum Antwerpen, Antwerp, Belgium, for providing the data used in this study.

## REFERENCES

[1] R. Boel and L. Mihaylova, "A compositional stochastic model for real time freeway traffic simulation," *Transportation Research Part B: Methodological*, vol. 40, no. 4, pp. 319 – 334, 2006.

[2] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within particle filtering framework," *Automatica*, vol. 43, no. 2, pp. 290 – 300, 2007.

[3] M. Treiber and A. Kesling, *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin-Heidelberg: Springer, 2013.

[4] S. Hoogendoorn, S. Ossen, M. Schreuder, and B. Gorte, "Unscented particle filter for delayed car-following models estimation," in *procc. IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1598–1603.

[5] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, "Macroscopic modelling of traffic flow on the Boulevard Peripherique in Paris," *Transportation Research Part B: Methodological*, vol. 23, no. 1, pp. 29 – 47, 1989.

[6] S. Hoogendoorn and P. Bovy, "Exploiting structure in wavelet-based Bayesian compressive sensing," *Journal of Systems Cntrol Engineer-Proceedings of the Intnational Mechanical Engineers, Part I*, vol. 215(14), pp. 283–303, 2001.

[7] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham, "Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool metanet," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 282–292, 2002.

[8] M. van den Berg, A. Hegyi, B. D. Schutter, and J. Hellendoorn, "A macroscopic traffic flow model for integrated control of freeway and urban traffic networks," in *in procc. IEEE Conference on Decision and Control*, 2003, pp. 2774–2779.

[9] 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.

[10] S. Fan and D. B. Work, "A heterogeneous multiclass traffic flow model with creeping," *SIAM Journal on Applied Mathematics*, vol. 75, no. 2, pp. 813–835, 2015.

[11] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269 – 287, 1994.

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

[13] D. B. Work, O. P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble Kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *IEEE Conference on Decision and Control*, 2008, pp. 5062–5068.

[14] L. L. Ojeda, A. Y. Kibangou, and C. C. de Wit, "Adaptive Kalman filtering for multi-step ahead traffic flow prediction," in *American Control Conference*, 2013, pp. 4724–4729.

[15] P.-E. Mazar, O.-P. Tossavainen, and D. B. Work, "Computing travel times from filtered traffic states," *Discrete and Continuous Dynamical Systems - Series S*, vol. 7, no. 3, pp. 557–578, 2014.

[16] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[17] A. Doucet, N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, USA: Springer, 2001.

[18] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489 – 509, 2006.

[20] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[21] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[22] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, 2013.

[23] Y. Li, C. Tian, F. Zhang, and C. Xu, "Traffic condition matrix estimation via weighted spatio-temporal compressive sensing for unevenly-distributed and unreliable GPS data," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 1304–1311.

[24] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[25] C. Research, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, September 2012.

[26] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[27] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent developments and applications of SUMO - Simulation of Urban MObility," *International Journal on Advances in Systems abd Measurements*, vol. 5, pp. 128–138, 2012.