

Parallelized Particle and Gaussian Sum Particle Filters for Large-Scale Freeway Traffic Systems

Lyudmila Mihaylova, *Senior Member, IEEE*, Andreas Hegyi, Amadou Gning, and René K. Boel

Abstract—Large-scale traffic systems require techniques that are able to 1) deal with high amounts of data and heterogeneous data coming from different types of sensors, 2) provide robustness in the presence of sparse sensor data, 3) incorporate different models that can deal with various traffic regimes, and 4) cope with multimodal conditional probability density functions (pdfs) for the states. Often, centralized architectures face challenges due to high communication demands. This paper develops new estimation techniques that are able to cope with these problems of large traffic network systems. These are parallelized particle filters (PPFs) and a parallelized Gaussian sum particle filter (PGSPF) that are suitable for online traffic management. We show how complex pdfs of the high-dimensional traffic state can be decomposed into functions with simpler forms and how the whole estimation problem solved in an efficient way. The proposed approach is general, with limited interactions, which reduce the computational time and provide high estimation accuracy. The efficiency of the PPFs and PGSPFs is evaluated in terms of accuracy, complexity, and communication demands and compared with the case where all processing is centralized.

Index Terms—Freeway traffic state estimation, high-dimensional systems, multimodality, parallelized Gaussian sum particle filters (PGSPFs), parallelized particle filters (PPFs).

I. INTRODUCTION

MANAGEMENT of traffic systems and of large traffic networks has been subject to intensive studies, based on advanced models [19], [31], [33], sensor data fusion [9],

Manuscript received February 10, 2011; revised June 1, 2011; accepted August 27, 2011. Date of publication January 4, 2012; date of current version March 5, 2012. This work was supported by the European Community's Seventh Framework Program FP7/2007-2013 under Grant Agreement 238710 (Monte Carlo based Innovative Management and Processing for an Unrivalled Leap in Sensor Exploitation); by the U.K. Engineering and Physical Sciences Research Council (EPSRC) Project EP/E027253/1 and network activities from the EU COST action TU0702; by Project DWTC-CP/40 "Sustainability effects of traffic management" sponsored by the Belgian government; by the Belgian Program on Inter-University Poles of Attraction V/22 on "Dynamical Systems and Control: Computation, Identification and Modelling" initiated by the Belgian State; by the Prime Minister's Office for Science, Technology and Culture; and by the FP7 EC Project CON4COORD (INFSO-ICT-223844) and Project DISC (INFSO-ICT-224498). The work of R. Boel was supported by the U.K. Royal Academy of Engineering for the Distinguished Visiting Fellowship Award. The Associate Editor for this paper was H. J. W. C. Van Lint.

L. Mihaylova and A. Gning are with the School of Computing and Communication Systems, InfoLab21, Lancaster University, LA1 4WA Lancaster, U.K. (e-mail: mila.mihaylova@lancaster.ac.uk; e.gning@lancaster.ac.uk).

A. Hegyi is with the Faculty of Civil Engineering and Geosciences, Department of Transport and Planning, Delft University of Technology, 2600 Delft, The Netherlands (e-mail: a.hegyi@tudelft.nl).

R. K. Boel is with the Department of Electrical Energy, Systems and Automation, SYSTeMS, Ghent University, 9000 Ghent, Belgium (e-mail: rene.boel@ugent.be).

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

Digital Object Identifier 10.1109/TITS.2011.2178833

[12], and control [15], [35], [36]. The traffic systems require reliable online control [13], [25] to limit the effect of congestion to improve traffic safety and flow. One of the strategies is to integrate intelligence in the traffic systems [4] and advanced estimation methods that allow predicting future traffic behavior. The estimated traffic variables can then be used as if they were true traffic state variables in the feedback control loop.

Urban and freeway road traffic systems require processing of traffic data in a distributed way and transmission of the data to traffic control centers. These data are often used for traffic monitoring, control, and information dissemination. Traffic measurements from the sensors are corrupted by noise; some data may be missing; and often, the data are aggregated over a longer time period, or the detectors are located at large distances to each other. In this paper, we present parallelized particle filters (PPFs) and a parallelized Gaussian sum particle filter (PGSPF) that can cope with the aforementioned problems and are suitable for large networks due to their parallel implementation.

Various traffic state estimators have been investigated in the literature. In [39], an extended Kalman filter (EKF) is proposed for traffic vehicular flow estimation. This approach is evaluated for real traffic data in [37] and [38]. In [28], a particle filter (PF) is designed to estimate the traffic state (speed and density) based on sparse flow and speed measurements, and the PF is compared with an unscented Kalman filter (UKF). The EKF and UKF performance is compared for traffic state estimation in [17] with different filter settings and sensor configurations.

For general nonlinear systems with non-Gaussian noises, PPFs are among the most powerful approaches. The only potential disadvantage of particle filtering, compared with other methods, is the higher computational time. To reduce this computational time, different approaches for parallelized and distributed PPFs are proposed in the literature [8], [10], [18], [21], [27], [32]. They can be classified in two groups: i) algorithms transmitting particle values and their weights between the processing units (PUs) or ii) algorithms communicating a parametric approximation of the conditional density. Most of these implementations are for sensor-network-related problems and have the tendency to minimize communications.

In [32], two distributed PPFs are proposed with Gaussian mixture approximation of the posterior state density function. The parameters of the Gaussian mixture model are estimated using an expectation maximization algorithm, and then, the mixture parameters are exchanged, instead of particle weights. In other implementations, e.g., [5], the focus is on improved distributed resampling steps, and the emphasis is on increasing speed and reducing complexity.

However, particles have to be exchanged between the PUs, which can be particularly expensive in terms of communications when the number of particles is high.

Other related multiple particle filtering works are reported in [10], [14], and [20]. The method of [8] maintains a separate filter on each sensor node. Each PF has the same set of prior distributions. Under the assumption that the measurement errors are independent, the likelihood function can be represented as a product of the local likelihood functions. Each local likelihood function is updated at the sensor node based on local measurements.

In [8], methodologies for performing distributed particle filtering in a sensor network are presented based on likelihood factorization and on training of parametric models to approximate the likelihood factors. Distributed initialization of the state probability distribution of multiple targets is considered in [6].

In [10], a decomposition of the state space into separate subspaces is proposed, and different PFs are run in each of the subspaces. The PF performance strongly depends on the partitioning of the state space. In [20], two distributed PFs are proposed in cluster-based underwater sensor arrays.

In [34], a centralized mixture Kalman filter is employed to simultaneously detect the discrete traffic state (free-flow or congested) and track the traffic speed. The algorithm relies on a switching-mode model including several modes. Other works on Gaussian mixture model filtering have been proposed for positioning and navigation in [1] and [2] and for vehicular traffic in [29].

There are situations where the approach with Gaussian mixtures is preferred [2] for solving nonlinear estimation problems, and these are the cases when the following occur: 1) The system and/or measurement noises can be represented using Gaussian mixtures; 2) the measurement likelihood function is multimodal and can be approximated by a Gaussian mixture; 3) the state probability density function (pdf) of the traffic is multimodal, and then, the Gaussian mixture model filter and the Gaussian Sum PF [24] are the right choice.

Our observations of the likelihood of the traffic states in PFs [28] for vehicular traffic are that it exhibits multimodal behavior and this is another motivation for developing GSPFs to vehicular traffic problems.

The main contributions of this work can be summarized as follows: A solution to traffic estimation problems is proposed with parallelized and centralized particle and Gaussian sum PFs. The data are sparse: they are coming only from sensors at boundaries between traffic segments. Two parallelized PFs are derived with an importance sampling function affording parallelization-based on partitioning of the complex traffic network into simpler subnetworks. In the PFs, this is done by factorizing the complex state pdf into simpler functions. These state predictions are needed for online traffic management, and most of the required calculations can be locally performed, within the subnetworks, and measurement updates are performed only on some boundaries. This leads to efficient algorithms in terms of accuracy, reduced communications, and computational complexity per PU. We demonstrate the efficiency of the parallelized approaches over a testing example, in which the traffic network consists of a freeway link partitioned

into two parts. We compare the accuracy, the computational complexity, and the communication needs for the parallel filters.

The remaining part of this paper is organized as follows: Section II presents the theoretical background for the centralized Bayesian estimation. Section III gives the general formulation of the particle filtering approach. Section IV formulates the state space representation for the purposes of designing parallelized filters. Section V presents two parallelized PFs, whereas Sections VI and VII outline the centralized GSPF and the parallelized GSPF, respectively. The developed filters are tested over an example with traffic congestion, as described in Section VIII. Results are shown in Section IX, and conclusions are drawn in Section X.

II. BAYESIAN ESTIMATION

Consider the discrete-time nonlinear non-Gaussian system model

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{n}_k) \quad (2)$$

where $\mathbf{x}_k \in \mathcal{X}$ is the system state vector belonging to a high-dimensional state space \mathcal{X} and has to be estimated in time $k = 1, 2, \dots$; $\mathbf{z}_k \in \mathcal{Z}$ represents the measurement obtained at time k and \mathcal{Z} is the measurement space; and \mathbf{v}_k and \mathbf{n}_k are mutually independent noise vectors, i.e., the system noise and the measurement noise, respectively. The noises are also independent of the random initial state \mathbf{x}_0 .

Since the system and the measurements are stochastic, the exact state cannot be inferred from the measurements, and only the pdf $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ of the state \mathbf{x}_k can be determined, given all past and current measurements $\mathbf{z}_{1:k} \triangleq \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$. Hence, the goal of the state estimation problem is to determine the conditional (posterior) pdf $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ at each time step k .

The posterior pdf can be recursively determined according to the *prediction* and *update* steps.

1) *Prediction (prior)*:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (3)$$

2) *Update (posterior)*:

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_k) &= \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})} \\ &= \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k} \end{aligned} \quad (4)$$

where $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the state transition pdf, and $p(\mathbf{z}_k | \mathbf{x}_k)$ is the measurement likelihood function. Equation (4) represents the Bayes' rule.

III. GENERAL FORMULATION OF PARTICLE FILTERING

A state space trajectory $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\} \in \mathcal{X}$ is defined, which represents the evolution of traffic state $\mathbf{x} \in \mathcal{X}$

during the time interval from 0 to k . To circumvent the integration that is necessary for the evaluation of the denominator of the right-hand side of (4), in particle filtering, the pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is approximated by a *random measure* $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^N$, where $\{\mathbf{x}_{0:k}^i, i = 1, \dots, N\}$ is a set of support points with weights $\{w_k^i, i = 1, \dots, N\}$. Here, $\{\mathbf{x}_{0:k}^i\}_{i=1}^N$ denote the particles, and N is the number of the particles. The posterior state pdf at k is approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i). \quad (5)$$

Each particle with the respective weight $\{\mathbf{x}_{0:k}^i, w_k^i\}$ can be considered as a hypothesis that the real state trajectory is $\mathbf{x}_{0:k}^i$ with belief w_k^i . This hypothesis is updated in two steps.

- 1) *State update*: When k is increased by one, a new value is appended to $\mathbf{x}_{0:k}^i$ to form $\mathbf{x}_{0:k+1}^i$, according to state (1), using the assumption that the previous state was $\mathbf{x}_{0:k}^i$ and a given sample drawn from $\mathbf{v}_k^i \sim p(\mathbf{v}_k)$.
- 2) *Measurement update*: When a measurement arrives, its likelihood function is calculated and used for the update of weights $w_{k+1}^i, i = 1, \dots, N$.

In both steps, the principle of *importance sampling* plays a crucial role, which can be explained as follows (cf. [3]):

A. Importance Sampling

Suppose that we want to approximate the pdf $p(\mathbf{x})$, which is difficult to sample from, but for which a test pdf $\pi(\mathbf{x}) \propto p(\mathbf{x})$ exists that can be evaluated for a given \mathbf{x} . Let $\mathbf{x}^i \sim q(\mathbf{x}), i = 1, \dots, N$ be samples that are drawn (sampled) from another pdf $q(\mathbf{x})$, which is called *importance density* or *proposal distribution*. Then, an approximation of the pdf $p(\mathbf{x})$ is given by

$$p(\mathbf{x}) \approx \sum_{i=1}^N w^i \delta(\mathbf{x} - \mathbf{x}^i), \quad \text{with} \quad w^i \propto \frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)}, \quad \sum_i w^i = 1$$

where w^i is the normalized weight of the i th sample.

Given this principle, the weights in (5) are defined to be

$$w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}$$

where proposal distribution $q(\mathbf{x})$ can be arbitrarily chosen; nevertheless, the choice of $q(\mathbf{x})$ is a relevant step. It can be shown [3] that, if $q(\mathbf{x})$ is chosen to factorize as

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$$

then the weights are *recursively* determined by

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}. \quad (6)$$

This expression can be easily evaluated for a given triple of $\mathbf{x}_{k-1}^i, \mathbf{x}_k^i$, and \mathbf{z}_k since it contains the known measurement and the state model in the numerator and the user-defined proposal distribution q in the denominator.

TABLE I
SYSTEMATIC RESAMPLING

```

[{\mathbf{x}_k^{j*}, w_k^j, ij}_{i=1}^N] = RESAMPLE[{\mathbf{x}_k^i, w_k^i}_{i=1}^N]
- Initialize the cumulative density function:  $c_1 = w_k^1$ 
for  $i = 2 : N$  do
  - Construct the cumulative density function:
     $c_i = c_{i-1} + w_k^i$ 
end
- Let  $i = 1$ 
- Draw starting point:  $u_1 \sim \mathbb{U}[0, \frac{1}{N}]$ 
for  $j = 1 : N$  do
  - Let  $u_j = u_1 + (j-1)\frac{1}{N}$ 
  while  $u_j > c_i$  do
     $i = i + 1$ 
  end
  - Assign sample:  $\mathbf{x}_k^{j*} = \mathbf{x}_k^i$ 
  - Assign weight:  $w_k^j = N^{-1}$ 
  - Assign parent:  $i^j = i$ 
end

```

A frequently used proposal distribution is the transition prior:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i).$$

Using this in (6) results in a simple weight update rule, i.e.,

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i).$$

B. Degeneracy and Resampling

It has been proven that the variance of the weights can only increase over time. This means, in general, that, after a few iterations, all but one weight will be (close to) zero, which is called the degeneracy problem. Consequently, one particle will represent the entire pdf, which is, of course, undesirable. To prevent this, the particles are regularly resampled, i.e., particles with small weights are eliminated, and new particles are created at or around those with large weights (such that the approximation in (5) still holds). To decide when to resample, the effective number of particles $\widehat{N}^{\text{eff}} = 1 / \sum_{i=1}^N (w_k^i)^2$ is compared to some predefined threshold $N^{\text{threshold}}$.

There exist several efficient resampling algorithms of computational complexity $O(N)$ that typically map the newly created particles to existing ones with high weights, such as residual resampling [26] and systematic resampling [3].

In this paper, we will use systematic resampling as given by the algorithm ‘‘RESAMPLE’’ in Table I, and the generic (centralized) particle filtering algorithm is described by algorithm ‘‘PF’’ (see Table II).

IV. STATE SPACE PARTITIONING FOR PARALLELIZATION

When recursive Bayesian filtering methods are applied to large traffic networks, the computational complexity may become too high for running in real time on a single PU. One way to tackle this problem is the parallelization of the filters.

The basic idea for the parallelization is to utilize the possibility that a traffic network can be *simulated* in parallel. A natural way to parallelize the simulation of traffic is to divide the traffic network into several subnetworks (corresponding

TABLE II
GENERIC PARTICLE FILTER ALGORITHM

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N] = \text{PF}[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$ <p>for $i = 1 : N$ (for each particle) do</p> <ul style="list-style-type: none"> - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k^i \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ - Determine the weight update factors according to (6). - Normalize weights, $w_k^i = w_{k-1}^i / \sum_i w_{k-1}^i$ <p>if $\widehat{N}^{\text{eff}} < N^{\text{threshold}}$ then</p> <ul style="list-style-type: none"> - Resample particles. <p>end</p> <ul style="list-style-type: none"> - Calculate the expected state: $E[\mathbf{x}_k \mathbf{z}_{1:k}] = \sum_{i=1}^N w_k^i \mathbf{x}_k^i$. <p>end</p>

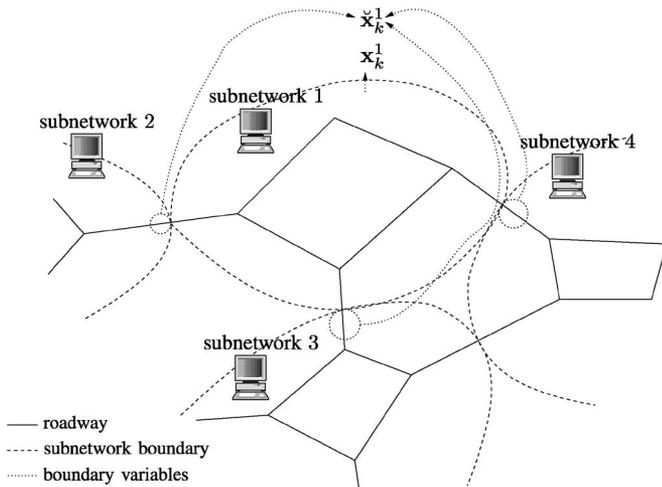


Fig. 1. Example of partitioning a traffic network into subnetworks for parallelized simulation/particle filtering.

to geographical regions), where each PU is responsible for one subnetwork and the *relevant* variables of the neighboring segments are communicated (as illustrated in Fig. 1). The state of the traffic network and the measurements can be correspondingly partitioned into S subvectors \mathbf{x}_k^s , $s = 1, \dots, S$, with $\mathbf{x}_k = [(\mathbf{x}_k^1)^T, (\mathbf{x}_k^2)^T, \dots, (\mathbf{x}_k^S)^T]^T$, and $\mathbf{z}_k = [(\mathbf{z}_k^1)^T, (\mathbf{z}_k^2)^T, \dots, (\mathbf{z}_k^S)^T]^T$. For each state subvector $\mathbf{x}_k^s \in \mathcal{X}^s$, $s = 1, \dots, S$, we have $\mathcal{X} = \{\mathcal{X}^1 \times \mathcal{X}^2 \times \dots \times \mathcal{X}^S\}$, and also for each measurement vector $\mathbf{z}_k^s \in \mathcal{Z}^s$, $s = 1, \dots, S$, where $\mathcal{Z} = \{\mathcal{Z}^1 \times \mathcal{Z}^2 \times \dots \times \mathcal{Z}^S\}$.

System (1) and (2) can now be described by

$$\mathbf{x}_k^s = f_k^s(\mathbf{x}_{k-1}^s, \check{\mathbf{x}}_{k-1}^s, \mathbf{v}_{k-1}^s) \quad (7)$$

$$\mathbf{z}_k^s = h_k^s(\mathbf{x}_k^s, \mathbf{n}_k^s) \quad (8)$$

where $s = 1, \dots, S$, and vector $\check{\mathbf{x}}_{k-1}^s$ collects all neighboring state variables that act as an input to subnetwork s . Note that not all states of the neighboring networks are communicated (only the states that serve as an input to subnetwork s).

Note also that, for (8), it is assumed that the measurements, taken in a subnetwork, depend only on the state in that subnetwork. This assumption holds for traffic systems since detectors typically measure traffic variables at one given location.

In addition, we assume independence of the state noises between subnetworks and independence of measurement noises between the subnetworks:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = \prod_{s=1}^S p(\mathbf{x}_k^s | \mathbf{x}_{k-1}^s, \check{\mathbf{x}}_{k-1}^s) \quad (9)$$

$$p(\mathbf{z}_k | \mathbf{x}_k) = \prod_{s=1}^S p(\mathbf{z}_k^s | \mathbf{x}_k^s). \quad (10)$$

V. APPROACHES OF PARALLELIZATION

1) *First Approach: A PF With Shared Particles:* In this approach, the PUs of different subnetworks share the same particles \mathbf{x}_k^i , and the particles are partitioned into subparticles $\mathbf{x}_k^{s,i}$ corresponding to subnetwork s . The PU corresponding to subnetwork s is responsible for the calculations for subparticles $\mathbf{x}_k^{s,i}$. This approach is functionally equivalent to the centralized approach, as previously presented, given that (9) and (10) hold.

In the state update step, subparticles $\mathbf{x}_k^{s,i}$ are now drawn from the distribution $q(\mathbf{x}_k^s | \mathbf{x}_{k-1}^s, \check{\mathbf{x}}_{k-1}^s, \mathbf{z}_k^s)$, which is based on local information only (including neighboring states). Now, choosing the proposal distribution $q(\mathbf{x}_k^s | \mathbf{x}_{k-1}^s, \check{\mathbf{x}}_{k-1}^s, \mathbf{z}_k^s)$ such that [cf. (9)]

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k) = \prod_{s=1}^S q(\mathbf{x}_k^s | \mathbf{x}_{k-1}^s, \check{\mathbf{x}}_{k-1}^s, \mathbf{z}_k^s) \quad (11)$$

and using (9) and (10) and that

$$p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}) = \sum_{j=1}^N p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j}) p(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i}) \quad (12)$$

$$p(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i}) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

the weight update (6) can be rewritten as

$$w_k^i \propto w_{k-1}^i \prod_{s=1}^S w_{k-1}^{s,i}, \quad \sum_i w_k^i = 1 \quad (14)$$

$$w_{k-1}^{s,i} = \frac{p(\mathbf{z}_k^s | \mathbf{x}_{k-1}^{s,i}) p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,i})}{q(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,i}, \mathbf{z}_k^s)}. \quad (15)$$

The consequence of (14) and (15) is that the state and measurement update can be locally performed (divided over S processors), except for the weight update.

For each time step k in the PF, the following communication has to take place.

- 1) The state variables $\check{\mathbf{x}}_{k-1}^{s,i}$ at the boundaries have to be sent to subnetwork s .
- 2) The weight update factors $w_{k-1}^{s,i}$ can be locally calculated, and only the results need to be communicated to a central PU to determine w_k^i .
- 3) The centrally calculated weights w_k^i are normalized, and sent back to the local PUs (after resampling, when necessary).
- 4) Resampling requires communication to a central PU, where all weights $w_k^{s,i}$ are collected and w_k^i are calculated

TABLE III
PARALLELIZED PF1 (SHARED PARTICLES)

```

[{\mathbf{x}_k^i, w_k^i}_{i=1}^N] = PPF1[{\mathbf{x}_{k-1}^i, w_{k-1}^i}_{i=1}^N, \mathbf{z}_k]
for s = 1 : S (for each subnetwork simultaneously) do
  for i = 1 : N (for each particle) do
    - Draw  $\mathbf{x}_k^{s,i} \sim q(\mathbf{x}_k^s | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,i}, \mathbf{z}_k^s)$ 
    - Determine the weight update factors according
      to (15) and send them to the central PU.
    At the central PU:
    - Determine  $w_k^i$  according to (14).
    - Normalize weights,  $w_k^i = w_k^i / \sum_i w_k^i$ 
    if  $\widehat{N}^{eff} < N^{threshold}$  then
      - Resample particles (determine the mapping
         $i^j$ )
    end
    - Send  $i^j$  and  $w_k^i$  to each local PU.
  end
end
for s = 1 : S (for each subnetwork) do
  - Update the particles according to
     $\mathbf{x}_k^{s,j} \leftarrow \mathbf{x}_k^{s,i^j}, j = 1, \dots, N$ .
  - Calculate the estimate of the state of
    subnetwork s :  $E[\mathbf{x}_k^s | \mathbf{z}_{1:k}^s] = \sum_{i=1}^N w_k^i \mathbf{x}_k^{s,i}$ .
end

```

according to (14). For the residual resampling [26] and the systematic resampling [3] methods, it is not necessary to communicate the particles themselves since these methods use only weights as the input and produce as a result new particles as a selection from the existing old particles (with some particles selected several times and others not at all). Therefore, after resampling only, the selected indices are communicated back to the PUs. For the resampling algorithms that create particles at new locations in the state space, such as regularization [11] and the MCMC scheme [7], the particles themselves also have to be communicated, and consequently, more communication is necessary.

The pseudocode for this PPF1 is given in Table III.

2) *Second Approach: A PF With Separate Particles:* In this approach, the particles of the different subnetworks are not shared, and only the neighboring traffic states are communicated over the boundaries to each subnetwork s . We can represent the predictive state pdf in the form

$$p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}) = \int_{\check{\mathbf{x}}_{k-1}^{s,j}} \left\{ p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j}) p(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i}) \right\} d\check{\mathbf{x}}_{k-1}^{s,j}. \quad (16)$$

Applying Monte Carlo sampling to the product $p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,i}) p(\check{\mathbf{x}}_{k-1}^{s,i} | \mathbf{x}_{k-1}^{s,i})$ with a proposal distribution $q(\check{\mathbf{x}}_{k-1}^{s,i} | \mathbf{x}_{k-1}^{s,i})$ results in the approximation

$$p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}) \approx \sum_j \frac{p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j}) p(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i})}{q(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i})}. \quad (17)$$

Note that, since the pdf of the communicated state variable is independent of $\mathbf{x}_{k-1}^{s,i}$ (by assumption)

$$p(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_{k-1}^{s,i}) = p(\check{\mathbf{x}}_{k-1}^{s,j}). \quad (18)$$

TABLE IV
PARALLELIZED PF2 (SEPARATE PARTICLES)

```

[{\mathbf{x}_k^i, w_k^i}_{i=1}^N] = PPF2[{\mathbf{x}_{k-1}^i, w_{k-1}^i}_{i=1}^N, \mathbf{z}_k]
for s = 1 : S (for each subnetwork) do
  for i = 1 : N (for each particle) do
    - Draw  $\check{\mathbf{x}}_{k-1}^{s,j_i} \sim p(\check{\mathbf{x}}_{k-1}^s)$ 
    - Draw  $\mathbf{x}_k^{s,i} \sim q(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j_i}, \mathbf{z}_k^s)$ 
    - Determine the weight update factors according
      to (20).
    - Normalize weights,  $w_k^{s,i} = w_k^{s,i} / \sum_i w_k^{s,i}$ 
    if  $\widehat{N}^{eff,s} < N^{threshold,s}$  then
      - Resample particles.
    end
    - Calculate the estimate of the state of
      subnetwork s :  $E[\mathbf{x}_k^s | \mathbf{z}_{1:k}^s] = \sum_{i=1}^N w_k^{s,i} \mathbf{x}_k^{s,i}$ .
  end
end

```

Using this relation and taking only one sample from $\check{\mathbf{x}}_{k-1}^{s,j_i} \sim p(\check{\mathbf{x}}_{k-1}^s)$ for each i , and choosing $q(\check{\mathbf{x}}_{k-1}^{s,j} | \mathbf{x}_k^{s,i}) = p(\check{\mathbf{x}}_{k-1}^{s,j})$, (17) simplifies to

$$p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}) \approx p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j_i}). \quad (19)$$

In this approach, the weights are locally updated since there are no centralized particles [cf. (14)], i.e.,

$$w_k^{s,i} = w_{k-1}^{s,i} \frac{p(\mathbf{z}_k^{s,i} | \mathbf{x}_k^{s,i}) p(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j_i})}{q(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j_i}, \mathbf{z}_k^s)}. \quad (20)$$

This form means that, for the PF, the first $\check{\mathbf{x}}_{k-1}^{s,j_i}$ needs to be sampled from $p(\check{\mathbf{x}}_{k-1}^s)$, and then, $\mathbf{x}_k^{s,i}$ according to the importance density $q(\mathbf{x}_k^{s,i} | \mathbf{x}_{k-1}^{s,i}, \check{\mathbf{x}}_{k-1}^{s,j_i}, \mathbf{z}_k^s)$. The parallelized PF algorithm with separate particles (called PPF2) is given in Table IV.

In this approach, there is no central PU, and there is only communication between the neighboring PUs. The communication takes place in each time step when the neighboring state variables $\check{\mathbf{x}}_{k-1}^{s,j_i}$ are sent to subnetwork s , after these quantities are drawn from $\check{\mathbf{x}}_{k-1}^{s,j_i} \sim p(\check{\mathbf{x}}_{k-1}^s)$. In this approach, resampling does not require communication since it can be locally performed at each PU.

The advantages of this approach over the PF approach with shared particles are given here.

- 1) It requires fewer particles since the dimension of the state space for each particle is reduced by factor S (assuming that all subnetworks have the same number of states).
- 2) For each subnetwork, a different number of particles can be used. This can be an advantage when a different accuracy is required for the different subnetworks.

A disadvantage of this approach is that an approximation is introduced in the interaction (joint pdf) of the local states with the states in neighboring subnetworks [as given by (18)].

VI. GAUSSIAN SUM PARTICLE FILTER FOR TRAFFIC FLOW ESTIMATION

The GSPF [22], [24] is a recently developed filtering technique in which the state filtering and prediction probability

density functions are approximated by finite mixtures of Gaussian components. When the object model and observation model noises are non-Gaussian, they can also be represented as Gaussian mixtures. The GSPF is essentially a bank of Gaussian PFs [23]. The GSPF reduces the estimation problem solution to updating a Gaussian mixture, where the mean, covariance, and weights are updated upon the receipt of each new observation.

The filtering state pdf can be approximated as follows:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \sum_{g=1}^G w_{k,g} \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{\mathbf{x}_k,g}, \boldsymbol{\Sigma}_{\mathbf{x}_k,g}) \quad (21)$$

by G Gaussian mixture components, having a vector mean and a covariance matrix $\boldsymbol{\mu}_{\mathbf{x}_k,g}$ and $\boldsymbol{\Sigma}_{\mathbf{x}_k,g}$, respectively. In (21), $w_{k,g}$ are the normalized weights, i.e., $\sum_{g=1}^G w_{k,g} = 1$.

The predictive state pdf can be given by (from [24, eq. (7)])

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) &= \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &\approx \int \sum_{g=1}^G w_{k-1,g} \mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{k-1,g}, \boldsymbol{\Sigma}_{\mathbf{x}_{k-1},g}) \\ &\quad \times p(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1} \\ &= \sum_{g=1}^G w_{k-1,g} \int \mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{k-1,g}, \boldsymbol{\Sigma}_{\mathbf{x}_{k-1},g}) \\ &\quad \times p(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \end{aligned} \quad (22)$$

According to [24, Th. 2, p. 2603], for an additive noise in the system model, the predictive state pdf $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ approaches the sum

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) \approx \sum_{g=1}^G \bar{w}_{k,g} \mathcal{N}(\mathbf{x}_{k-1}; \bar{\boldsymbol{\mu}}_{k,g}, \bar{\boldsymbol{\Sigma}}_{\mathbf{x}_{k-1},g}) \quad (23)$$

where $\bar{w}_{k,g}$ are the normalized weighting coefficients, and $\bar{\boldsymbol{\mu}}_{k,g}$ and $\bar{\boldsymbol{\Sigma}}_{\mathbf{x}_{k-1},g}$ are the mean and covariance matrix for the predictive state pdf, respectively, to distinguish them from those of the filtering pdf (21). The integral on the right-hand side of (22) is approximated by a Gaussian sum.

The whole centralized GSPF algorithm with the prediction and measurement update steps for time $k = 1, \dots, K$ (here K indicates the time steps) is given in Table V.

The output state estimate $\hat{\mathbf{x}}_k = E[\mathbf{x}_k | \mathbf{z}_{1:k}]$ and its covariance $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}_k} = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$ are calculated from the following Gaussian mixtures:

$$\hat{\mathbf{x}}_k = \sum_{g=1}^G w_{k,g} \boldsymbol{\mu}_{k,g} \quad (24)$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{x}_k} = \sum_{g=1}^G w_{k,g} [\boldsymbol{\Sigma}_{\mathbf{x}_k,g} + (\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{k,g})(\hat{\mathbf{x}}_k - \boldsymbol{\mu}_{k,g})^T] \quad (25)$$

respectively.

It should be noted that, whereas the resampling step in the PFs is performed with respect to particles, the resampling step in the GSPF is for the Gaussian mixing components. During the resampling step of the GSPF, mixands with insignificant weights are discarded, e.g., with residual resampling, whereas

TABLE V
CENTRALIZED GAUSSIAN SUM PARTICLE FILTER

Initialize: $\boldsymbol{\mu}_{\mathbf{x}_0,g}, \boldsymbol{\Sigma}_{\mathbf{x}_0,g}, w_{0,g} = \frac{1}{G}, g = 1, \dots, G$, at $k = 0$.

for $k = 1, 2, \dots, K$,

Prediction

for $g = 1, \dots, G$,

draw $\{\mathbf{x}_{k-1,g}^j\}_{j=1}^N \sim \mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{\mathbf{x}_{k-1},g}, \boldsymbol{\Sigma}_{\mathbf{x}_{k-1},g})$, **end**

for $g = 1, \dots, G$,

sample $\{\mathbf{x}_{k,g}^j\}_{j=1}^N \sim p(\mathbf{x}_k | \mathbf{x}_{k-1,g}^j)$, from (1).

end

for $g = 1, \dots, G$, update weights $\bar{w}_{k,g} = w_{k-1,g}$, **end**

for $g = 1, \dots, G$, from $\{\mathbf{x}_{k,g}^j\}_{j=1}^N$ obtain sample mean $\bar{\boldsymbol{\mu}}_{\mathbf{x}_k,g}$ and sample covariance $\bar{\boldsymbol{\Sigma}}_{\mathbf{x}_k,g}$,

end

The state predictive density is approximated as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \sum_{g=1}^G \bar{w}_{k,g} \mathcal{N}(\mathbf{x}_k; \bar{\boldsymbol{\mu}}_{\mathbf{x}_k,g}, \bar{\boldsymbol{\Sigma}}_{\mathbf{x}_k,g}). \quad (26)$$

Measurement Update

for $g = 1, \dots, G$, draw samples

$$\{\mathbf{x}_{k,g}^j\}_{j=1}^N \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_{\mathbf{x}_k,g}, \boldsymbol{\Sigma}_k; \bar{\boldsymbol{\mu}}_{\mathbf{x}_k,g}, \bar{\boldsymbol{\Sigma}}_{\mathbf{x}_k,g}).$$

end

for $g = 1, \dots, G, j = 1, \dots, N$ compute the weights $W_{k,g}^j \propto p(\mathbf{z}_k | \mathbf{x}_{k,g}^j)$ where the likelihood $p(\mathbf{z}_k | \mathbf{x}_{k,g}^j)$ is calculated from (2).

end

for $g = 1, \dots, G, j = 1, \dots, N$ estimate the mean and covariance by

$$\boldsymbol{\mu}_{\mathbf{x}_k,g} = \frac{\sum_{j=1}^N W_{k,g}^j \mathbf{x}_{k,g}^{(j)}}{\sum_{j=1}^N W_{k,g}^{(j)}}, \quad (27)$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_k,g} = \frac{\sum_{j=1}^N W_{k,g}^j (\mathbf{x}_{k,g}^j - \boldsymbol{\mu}_{\mathbf{x}_k,g})(\mathbf{x}_{k,g}^j - \boldsymbol{\mu}_{\mathbf{x}_k,g})^T}{\sum_{j=1}^N W_{k,g}^j} \quad (28)$$

end

for $g = 1, \dots, G$, update the weights

$$\bar{w}_{k,g} = \bar{w}_{k-1,g} \frac{\sum_{j=1}^N W_{k,g}^j}{\sum_{g=1}^G \sum_{j=1}^N W_{k,g}^j}, \quad (29)$$

end

- Normalize the weights

$$w_{k,g} = \bar{w}_{k,g} / \sum_{g=1}^G \bar{w}_{k,g}. \quad (30)$$

Output: the state estimate and its covariance calculated respectively from (24) and (25).

Resampling: resample the weights $w_{k,g}$ as in [24].

for $g = 1, \dots, G$ draw a number $\ell \in \{1, \dots, G\}$ with probabilities proportional to $w_{k,1}, \dots, w_{k,G}$ and let

$$\{\boldsymbol{\mu}_{k,g}, \boldsymbol{\Sigma}_{\mathbf{x}_k,g}\} = \{\boldsymbol{\mu}_{k+1,\ell}, \boldsymbol{\Sigma}_{\mathbf{x}_k,\ell}\},$$

- Set $w_{k,g} = \frac{1}{G}$.

end

- Set $k - 1 \rightarrow k$

mixands with significant weights are *duplicated*. Notice that the resampling procedure applied to the GSPF slightly differs from the described in [24] because of the fact that we are not representing the system noise as a Gaussian mixture.

VII. PARALLELIZED GAUSSIAN SUM PARTICLE FILTER FOR TRAFFIC FLOW ESTIMATION

The parallelized GSPF approach communicates only the statistics (mean and covariance) of the states on the boundaries.

TABLE VI
PARALLELIZED GAUSSIAN SUM PARTICLE FILTER

<p>Initialize: $\mu_{\mathbf{x}_{0,g}}, \Sigma_{\mathbf{x}_{0,g}}, w_{0,g} = \frac{1}{G}, g = 1, \dots, G$, at $k = 0$.</p> <p>for $k = 1, 2, \dots, K$,</p> <p>for $s = 1 : S$ (for each subnetwork)</p> <p>Prediction</p> <p>for $g = 1, \dots, G$, draw</p> $\{\tilde{\mathbf{x}}_{k-1}^{s,j}\}_{j=1}^N \sim p(\tilde{\mathbf{x}}_{k-1}^s) \quad (31)$ $\{\mathbf{x}_{k-1}^j\}_{j=1}^N \sim \mathcal{N}(\mathbf{x}_{k-1}; \mu_{\mathbf{x}_{k-1,g}}, \Sigma_{\mathbf{x}_{k-1,g}}) \quad (32)$ <p>end</p> <p>for $g = 1, \dots, G$, sample</p> $\{\mathbf{x}_{k,g}^j\}_{j=1}^N \sim p(\mathbf{x}_{k,g} \mathbf{x}_{k-1,g}^j, \tilde{\mathbf{x}}_{k-1}^{s,j}), \text{ from (1).}$ <p>end</p> <p>for $g = 1, \dots, G$, update weights $\tilde{w}_{k,g} = w_{k-1,g}$. end</p> <p>for $g = 1, \dots, G$, from $\{\mathbf{x}_{k,g}^j\}_{j=1}^N$ obtain sample mean $\bar{\mu}_{\mathbf{x}_{k,g}}$ and sample covariance $\bar{\Sigma}_{\mathbf{x}_{k,g}}$.</p> <p>end</p> <p>The state predictive density is approximated as</p> $p(\mathbf{x}_k \mathbf{z}_{1:k-1}) = \sum_{g=1}^G \tilde{w}_{k,g} \mathcal{N}(\mathbf{x}_k; \bar{\mu}_{\mathbf{x}_{k,g}}, \bar{\Sigma}_{\mathbf{x}_{k,g}}).$ <p>Measurement Update</p> <p>for $g = 1, \dots, G$, draw samples</p> $\{\mathbf{x}_{k,g}^{(j)}\}_{j=1}^N \sim \mathcal{N}(\mathbf{x}_k; \bar{\mu}_{\mathbf{x}_{k,g}}, \bar{\Sigma}_{\mathbf{x}_{k,g}}).$ <p>end</p> <p>for $g = 1, \dots, G$, $j = 1, \dots, N$ compute the weights $W_{k,g}^j \propto p(\mathbf{z}_k \mathbf{x}_{k,g}^j)$ where the likelihood $p(\mathbf{z}_k \mathbf{x}_{k,g}^j)$ is calculated from (2).</p> <p>end</p> <p>for $g = 1, \dots, G$, $j = 1, \dots, N$ estimate the mean and covariance from (27) and (28)</p> <p>end</p> <p>for $g = 1, \dots, G$, update the weights with (29) end</p> <p>- Normalize the weights with (30).</p> <p>Output: the state estimate and its covariance are calculated respectively from (24) and (25).</p> <p>Resampling</p> <p>- Resample the weights $w_{k,g}$ as in [24].</p> <p>for $g = 1, \dots, G$ draw a number $\ell \in \{1, \dots, G\}$ with probabilities proportional to $w_{k,1}, \dots, w_{k,G}$ and let $\{\mu_{k,g}, \Sigma_{\mathbf{x}_{k,g}}\} = \{\mu_{k,\ell}, \Sigma_{\mathbf{x}_{k,\ell}}\}$,</p> <p>- Set $w_{k,g} = \frac{1}{G}$. end</p> <p>end // of S loop (for each subnetwork)</p> <p>- Set $k - 1 \rightarrow k$</p>

The PGSPF is presented in Table VI. The implemented PGSPF is similar to PPF2, with separate particles. Only estimates of the boundary conditions are transmitted between the boundaries of each subnetwork. To sample from (31), a Gaussian mixture is used, similarly to [30].

In contrast with PPFs, where particles or statistics are transmitted between central processing unit (CPU), the parallelized GSPF requires transmission of the parameters of the GSPF (the mean, variance, and weights of each Gaussian component).

VIII. TESTING SCENARIO

In this section, we apply the developed parallelized filters to a traffic problem consisting of state estimation for a traffic freeway link composed of two sublinks. The purpose of this example is to compare the estimation accuracy, computational load, and communication requirements for the centralized and parallelization approaches.

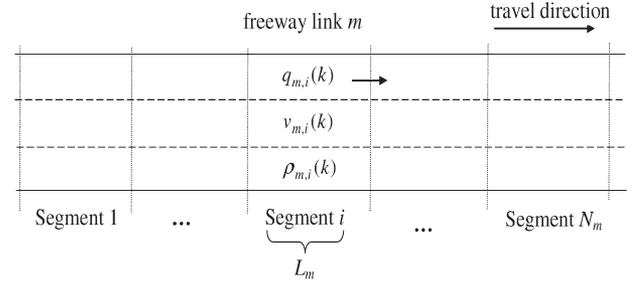


Fig. 2. In the METANET model, a freeway link is divided into segments. The main variables of the model are the following: The average outflow of a segment $q_{m,i}(k)$, average speed $v_{m,i}(k)$, average density $\rho_{m,i}(k)$, and segment length L_m .

The next section describes the freeway model of the traffic states.

A. Freeway Model: The Metanet Model

Consider a freeway link m that is subdivided into N_m segments, each with length L_m and λ_m lanes, and a discrete time step with length T (in hours). Traffic dynamics is described in terms of the aggregated variable *speed* $v_{m,i}(k)$ (in kilometers per hour), *flow* $q_{m,i}(k)$ (in vehicles per hour), and *density* $\rho_{m,i}(k)$ (in vehicles per kilometer per lane), where i is the segment index. The relevant variables are shown in Fig. 2.

The METANET model equations are given by the fundamental relationship between speed, density, and flow

$$q_{m,i}(k) = \rho_{m,i}(k) v_{m,i}(k) \lambda_m \quad (33)$$

the law of conservation of vehicles

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m} (q_{m,i-1}(k) - q_{m,i}(k)) + \xi_{m,i}^\rho(k) \quad (34)$$

and a heuristic relationship of the speed dynamics

$$v_{m,i}(k+1) = v_{m,i}(k) + \frac{T}{\tau} (V(\rho_{m,i}(k)) - v_{m,i}(k)) + \frac{T}{L_m} v_{m,i}(k) (v_{m,i-1}(k) - v_{m,i}(k)) - \frac{\eta T}{\tau L_m} \frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} + \xi_{m,i}^v(k) \quad (35)$$

$$V(\rho_{m,i}(k)) = v_{\text{free},m} \exp \left[-\frac{1}{a_m} \left(\frac{\rho_{m,i}(k)}{\rho_{\text{crit},m}} \right)^{a_m} \right] \quad (36)$$

where $\xi_{m,i}^\rho(k)$ and $\xi_{m,i}^v(k)$ are random variables representing the random (unmodeled) dynamics in the speed and density evolution. Although (34) is an exact relationship and, therefore, a modeling error is not present, we include the random variable $\xi_{m,i}^\rho(k)$ to allow a state filter to correct the number of vehicles in the network. This noise model formulation is the same as in [17] and [39]. Furthermore, $v_{\text{free},m}$ is the free-flow speed in segment m , $\rho_{\text{crit},m}$ is the critical density (the density at or above which traffic becomes unstable), and τ , η , a_m , and κ are the model fitting parameters without direct physical meaning.

An extension was introduced to be able to express the different anticipation behavior of the drivers at the head and the tail of a traffic jam (i.e., a shock wave) [16]. The parameter η in (35) is replaced by the density dependent $\eta_{m,i}(k)$ according to

$$\eta_{m,i}(k) = \begin{cases} \eta_{\text{high}}, & \text{if } \rho_{m,i+1}(k) \geq \rho_{m,i}(k) \\ \eta_{\text{low}}, & \text{if } \rho_{m,i+1}(k) < \rho_{m,i}(k). \end{cases}$$

B. Boundary Conditions

Variables $q_{m,0}$, $v_{m,0}$, and ρ_{m,N_m+1} are *boundary variables*, which incorporate the influence of upstream and downstream segments from the considered link. Usually, $q_{m,0}$ and $v_{m,0}$ can be directly measured; in practice, density ρ_{m,N_m+1} is not directly measured and must be estimated. Even though $q_{m,0}$ and $v_{m,0}$ can be directly measured, the measurements will be corrupted by errors. Therefore, we will consider all boundary variables as extra states of the system, and we will estimate them from the measurement data, similarly to the other state variables. This approach is also recommended in [39]. The dynamic evolution of the boundary variables is described by a random walk

$$\begin{bmatrix} q_{m,0}(k+1) \\ v_{m,0}(k+1) \\ \rho_{m,N_m+1}(k+1) \end{bmatrix} = \begin{bmatrix} q_{m,0}(k) \\ v_{m,0}(k) \\ \rho_{m,N_m+1}(k) \end{bmatrix} + \begin{bmatrix} \xi_{m,0}^q(k) \\ \xi_{m,0}^v(k) \\ \xi_{m,N_m+1}^\rho(k) \end{bmatrix} \quad (37)$$

where $\xi_{m,0}^q(k)$, $\xi_{m,0}^v(k)$, $\xi_{m,N_m+1}^\rho(k)$ are stochastic variables.

C. Measurements

The most frequently used traffic measurement devices typically measure speed and flow. For the segments that are equipped with sensors, the measurement equations are given as follows:

$$y_{m,i}^q(k) = q_{m,i}(k) + n_{m,i}^q(k) \quad (38)$$

$$y_{m,i}^v(k) = v_{m,i}(k) + n_{m,i}^v(k) \quad (39)$$

where $n_{m,i}^q(k)$, and $n_{m,i}^v(k)$ are the measurement noises for the flow and the speed, respectively. Note that, in practice, traffic systems provide measurements with a larger sampling time than the model time step. Typically, the measurement sampling time step is 1 or 5 min, whereas the model time step is 10 s. Including this fact in the development of a PF is straightforward, but for the sake of simplicity, we assume that, for each model time step, a measurement is available.

D. State-Space Representation

To bring (33)–(37) into the state-space representation required by the various filters, state \mathbf{x}_k is defined as¹ $\mathbf{x}_k = [\rho_1(k), \dots, \rho_N(k), v_1(k), \dots, v_N(k), v_0(k), q_0(k), \rho_{N+1}]^T$, and the measurement vector $\mathbf{z}_k = [y_{m,i}^q(k)^T, y_{m,i}^v(k)^T]^T$ collects the

¹Link index m is omitted in the rest of this section, assuming that all the variables introduced hereafter refer to the same link.

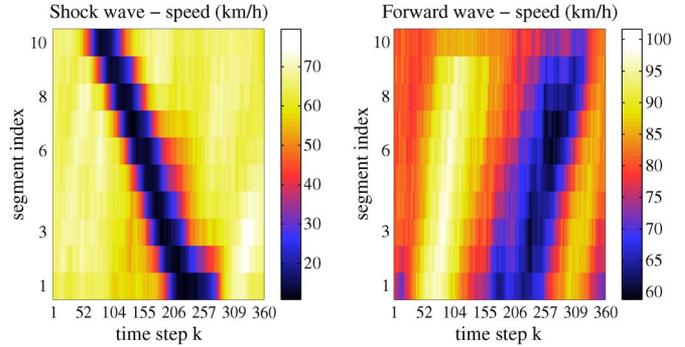


Fig. 3. (Left) shock wave and (right) forward wave scenario used for the evaluation of the filters. The travel direction is from segment 1 to 10. The colors indicate the speed. Note the difference in color bar scales: The shock wave scenario includes a wider range of speed since it also contains congested traffic.

flow and speed measurements from (38) and (39) for the segments equipped with sensors.

E. Experiment Design

1) *Layout*: The network for the testing example consists of a two-lane freeway link of ten segments of 1 km each. For the two parallel approaches, this link is divided into two sublinks (“subnetworks”) consisting of the first and last five segments, respectively.

2) *Scenario*: Two different scenarios are used to evaluate the filters: one with downstream propagating waves (in free-flow) and one with an upstream propagating shock wave, as shown in Fig. 3. These scenarios are defined by selecting the upstream and downstream boundary conditions. The motivation to select these two scenarios is to have both conditions where information propagates forward and where information propagates backward over the sublink boundaries. The state and measurement noises are taken to be Gaussian (although any other distribution could be taken) with state noise variances, $\text{var}(\xi_{m,i}^v(k)) = 0.5 \text{ (km/h)}^2$, $\text{var}(\xi_{m,i}^\rho(k)) = 0.5 \text{ (veh/km/lane)}^2$, measurement noise variances $\text{var}(n_{m,i}^v(k)) = (2 \text{ km/h})^2$, and $\text{var}(n_{m,i}^q(k)) = (150 \text{ veh/h})^2$.

3) *Parameters*: The following model parameters are used: $T = 10 \text{ s}$, $\tau = 18 \text{ s}$, $a = 1.867$, $\eta_{\text{high}} = 65 \text{ km}^2/\text{h}$, $\eta_{\text{low}} = 30 \text{ km}^2/\text{h}$, $\kappa = 40 \text{ veh/km/lane}$, $\rho_{\text{crit},m} = 33.5 \text{ veh/km/lane}$, $v_{\text{min}} = 7 \text{ km/h}$, and $v_{\text{free},m} = 102 \text{ km/h}$.

For the experiment investigating the information exchange over the subnetwork boundaries, it was assumed that segments 1 and 10 are measured (the two ends of the complete link), and only the segments of the downstream sublink (sublink 2) are measured for the shock wave scenario. In this way, the upstream sublink gets information about the incoming backward propagating shock wave only from the downstream sublink and not from the measurements, and the performance of the upstream link will depend on the information from the downstream neighbor link.

Similarly, for the forward wave scenario, only the upstream sublink is measured (and segment 10) to investigate the communication over the sublink boundary in case of a forward

propagating wave (corresponding to downstream propagating information).

4) *Filter Setup*: The PFs are set up according to the algorithms in Sections III and V. For the proposal distribution, the transition prior is used, and the PF, PPF1, and PPF2 are investigated for different numbers of particles in the range $N \in \{20, 50, 100, 200, 500, 1000\}$. The performance of the GSPF and PGSPF presented in Sections VI and VII has also been evaluated with a different number of particles and Gaussian mixture components $G = 1, 3, 8, 16, 24, 32$, but due to space limitations, we present results only with $N = 100$. The resampling threshold is chosen to be $N^{\text{threshold}} = 0.3$. The state noise v_k^i , which is sampled during the operation of the filters, is taken to have the same realization for the three different filters for better comparability.

5) *Performance Measures*: The performance of the filters is evaluated by the following three performance measures:

- 1) *Estimation accuracy*. For each filter, the root mean square error (RMSE) is determined of the expected value of the particles $\tilde{x}_k = E(x_k^i)$ relative to the state vector \hat{x}_k in the reference scenarios. The RMSE is determined for the speed and the density separately, according to

$$J_{\text{RMSE},\rho} = \sum_i \sum_k \sqrt{\frac{(\hat{\rho}_{i,k} - \tilde{\rho}_{i,k})^2}{KN_m}}$$

where $\hat{\rho}_{i,k}$ and $\tilde{\rho}_{i,k}$ are the density components of the real state and the expected state, respectively, of the particles for segment i at time k ; K is the number of simulation steps; and N_m is the number of segments within the traffic link. $J_{\text{RMSE},v}$ is similarly calculated.

- 2) *Communication*. The communication needs of the filters are evaluated on the basis of the number of communicated real numbers (doubles) for a complete run of the simulation. Depending on the filter, this communication may include the communication of the measurements to the PU, the communication of boundary states and weights between the PUs, and the communication of the weights to and from the central PU.
- 3) *CPU time*. As a measure for the computational demand, the time that each filter needs for a complete run is determined.

IX. RESULTS AND DISCUSSION

A. Performance Evaluation of the Centralized PF, Parallelized PF1, and Parallelized PF2

Fig. 4 shows results for the estimation accuracy as a function of the number of particles for the centralized PF (top), PPF1 from Table III (middle), and PPF2 from Table IV (bottom). In these experiments, the shock wave scenario was used. The experiments were repeated ten times, and the figure presents the averages and standard deviations of these experiments. For all filters, the performance gets better (lower error) when the number of particles increases. The performance of the centralized filter and the performance of approach 1 are similar

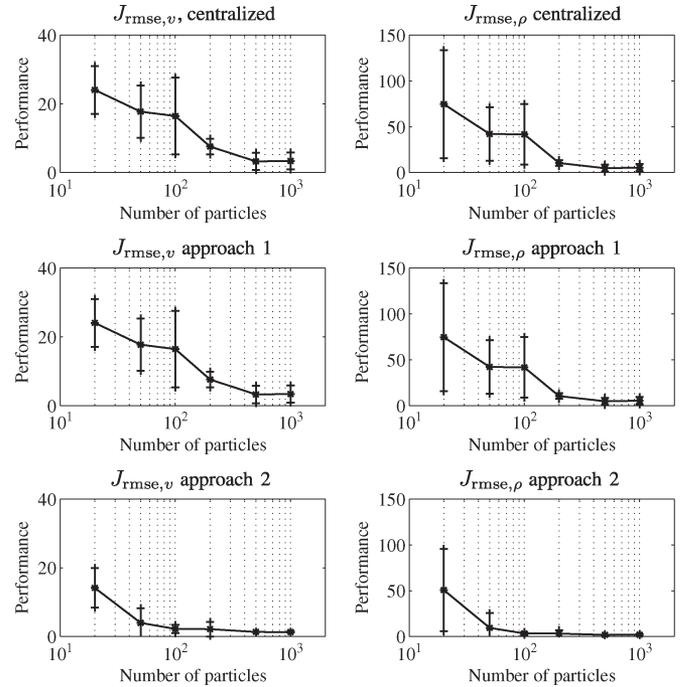


Fig. 4. RMSE as a function of the number of particles for the (top) centralized filter, (middle) approach 1, and (bottom) approach 2 (for segment 1 of the considered traffic link). The dots connected by the solid line indicate the mean, and the vertical lines with the “+” signs the standard deviation over ten Monte Carlo experiments. Note the logarithmic scale of the horizontal axis.

TABLE VII
NUMBER OF COMMUNICATED DOUBLES FOR ONE TIME CYCLE (FROM k TO $k + 1$) FOR EACH APPROACH AS A FUNCTION OF THE NUMBER OF PARTICLES N

N	Centralized PF	PPF1	PPF2
20	4	124	44
50	4	304	104
100	4	604	204
200	4	1204	404
500	4	3004	1004
1000	4	6004	2004

TABLE VIII
CPU TIME (IN SECONDS) FOR THE CENTRALIZED PF, PPF1, AND PPF2, FOR ONE TIME CYCLE (FROM k TO $k + 1$) FOR ONE PARTICLE

Centralized PF	PPF1	PPF2
0.0023	0.0020	0.00185

because the two filters are functionally equivalent, and the same noise realizations are used.

Interestingly, the average performance of approach 2 is significantly better for all numbers of particles. From this, it can be concluded that the improvement following from the fact that the same number of particles covers a smaller state space (i.e., a state space with lower dimensions) is more important than the deterioration following from the approximation of the pdfs made at the boundaries of the sublinks.

The communicated doubles for the three types of PFs are given in Table VII and the CPU time in Table VIII. The experiments have been performed on a personal computer with Intel Pentium 4 CPU, 3.40 GHz, 2 GB of random-access memory, and with MATLAB environment.

TABLE IX
CPU TIME (IN SECONDS) FOR EACH APPROACH
FOR ONE TIME CYCLE (FROM k TO $k + 1$)

Centralized PF $N = 1000$	PPF1 $N = 1000$	PPF2 $N = 1000$	GSPF $GN = 960$	PGSPF $GN = 960$
2.3	2	1.85	1.0156	0.9570

The number of communicated doubles can be analytically calculated based on the relationships given here. For PPF1, the corresponding analytical expression is $n_z + 6N$, where n_z is the number of measurements and “6” comes from $2 + 4$ (where 2 is (speed and flow) and 4 is the 2×2 weights communicated to the central PU and back to the local PUs). For PPF2, the analytical expression is $n_z + 2N$, with “2” indicating the speed and flow linked with N particles.

For both parallelization approaches, the CPU time required by one of the PUs corresponding to one sublink is clearly less than the CPU of the centralized filter. However, based on the number of floating-point operations, it would be expected that the parallelized filters have a computational demand around 50% of the centralized filter since the same operations are executed by two PUs, instead of one. The difference between the expectation and the simulation results can be explained by the overhead CPU time that is needed by all filters during code execution, such as the time needed to call the state transition functions, which is called the same number of times for all PUs. For larger problems, it can be expected that the CPU time will not be dominated by this common overhead, and the efficiency improvement will be higher (see Table IX).

B. Performance Evaluation of the Centralized and Parallelized GSPFs

The performance of the developed PGSPF and centralized GSPF is studied with a different number of Gaussian mixture components and a different number of particles, compared with the baseline centralized PF and with PPF1 and PPF2 over the same scenarios as in Section IX-A. To have a thorough comparison of the filters, we choose GN , which is the product of Gaussian mixture components with the number of particles in the GSPF and PGSPF to be different: less, equal, and more than the number of particles in the PF. In Figs. 5–8, we are showing RMSE results from the shock wave scenario. The results presented are with $N = 1000$ for the centralized PF and $N = 50$ for the centralized GSPF and PGSPF. The speed and density RMSEs (for segment 1) are shown in Figs. 5 and 6, respectively, for the centralized GSPF, compared with the centralized PF, and the respective RMSEs for the PGSPF are given in Figs. 7 and 8. In terms of accuracy, both the centralized GSPF and PGSPF, with $G > 3$ Gaussian mixture components, outperform the centralized PF.

The communicated doubles for each approach as a function of the Gaussian mixture components and for $N = 100$ particles are shown in Table X, and they are less than the communicated doubles of PPF1 and PPF2 reported in Table VII. As evident from Tables VII and X, the GSPF approaches outperform the centralized PF and the parallelized PFs. In sublink 1, PGSPF is sending $n_z + (2 + 4)G$, where we have 2 for the boundary conditions (mean values, for the speed and flow for G mixture

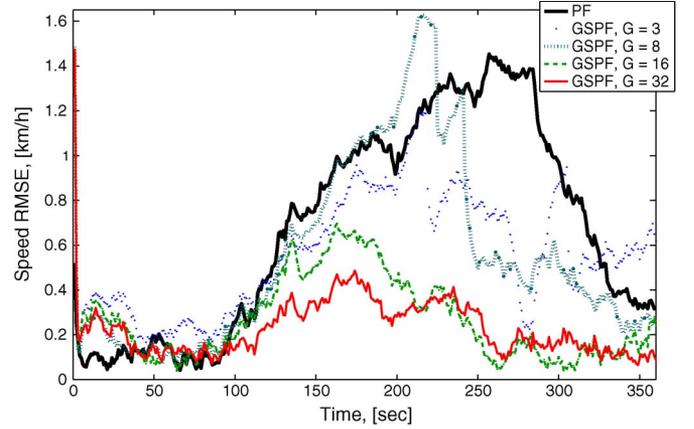


Fig. 5. Speed RMSEs obtained with the centralized PF ($N = 1000$ particles, centralized GSPF with $N = 50$ particles, and a different number of Gaussian mixture components. The results are with the shock wave traffic scenario shown in Fig. 3 (left) for segment 1 and ten Monte Carlo runs.

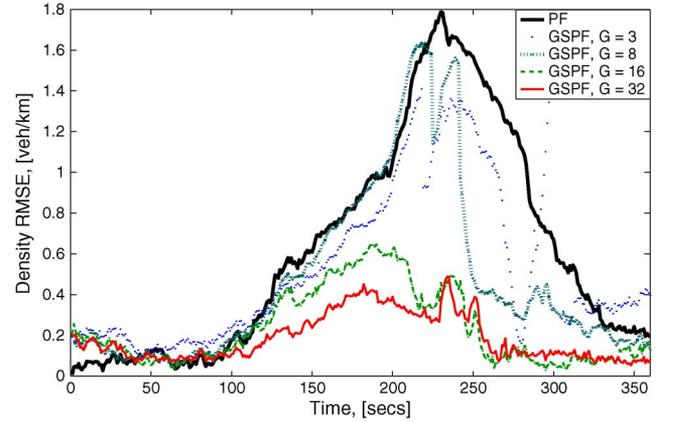


Fig. 6. Density RMSEs obtained with the centralized PF ($N = 1000$ particles, GSPF with $N = 50$ particles, and a different number of Gaussian mixture components. The results are with the shock wave traffic scenario shown in Fig. 3 (left) for segment 1 and ten Monte Carlo runs.

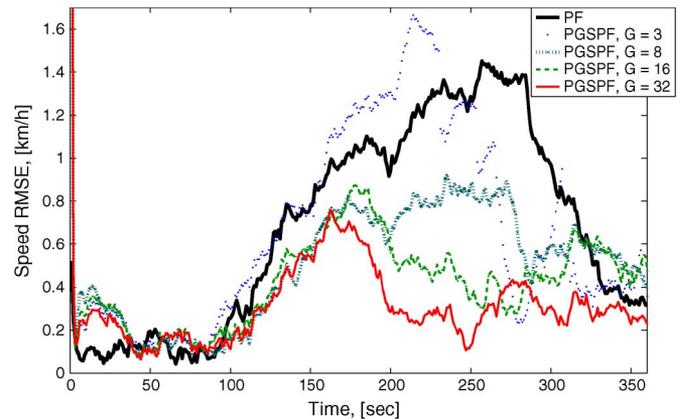


Fig. 7. Speed RMSEs obtained with the centralized PF ($N = 1000$ particles, PGSPF with $N = 50$ particles, and a different number of Gaussian mixture components. The results are with the shock wave traffic scenario shown in Fig. 3 (left) for segment 1 and ten Monte Carlo runs.

components) and 2×2 are for the covariance matrices of the speed and flow for G components. Sublink 2 sends $n_z + (1 + 1)G$, which corresponds to the boundary density and covariance of this density.

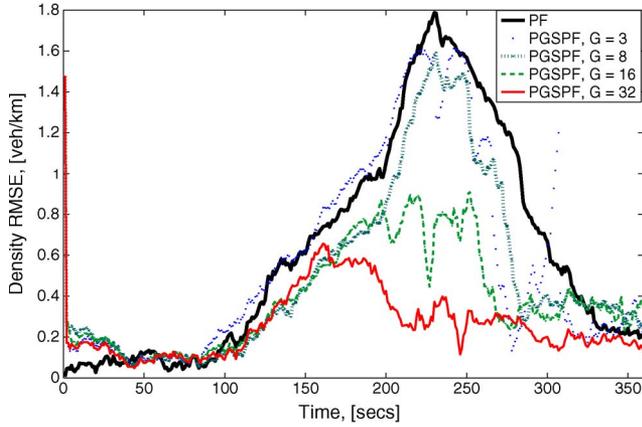


Fig. 8. Density RMSEs obtained with the centralized PF ($N = 1000$ particles, PGSPF with $N = 50$ particles, and a different number of Gaussian mixture components. The results are with the shock wave traffic scenario shown in Fig. 3 (left) for segment 1 and ten Monte Carlo runs.

TABLE X

NUMBER OF COMMUNICATED DOUBLES FOR EACH APPROACH AS A FUNCTION OF NUMBER G OF GAUSSIAN MIXTURE COMPONENTS, WITH $N = 100$ PARTICLES, FOR ONE TIME CYCLE (FROM k TO $k + 1$)

G	Centralized GSPF: as the Centralized PF	PGSPF
1	4	12
3	4	28
8	4	68
16	4	132
24	4	196
32	4	260

TABLE XI

CPU TIME FOR EACH APPROACH IN SECONDS (FOR FIVE SEGMENTS) AS A FUNCTION OF NUMBER G OF GAUSSIAN MIXTURE COMPONENTS FOR ONE TIME CYCLE (FROM k TO $k + 1$) FOR ONE PARTICLE

G	Centralized GSPF	PGSPF
1	0.0012	0.0011
3	0.0035	0.0032
8	0.0080	0.0079
16	0.0153	0.0149
24	0.0258	0.0252
32	0.0313	0.0308

The CPU time results are presented in Table XI per particle, with a different number G of Gaussian mixture components. To have a better understanding of the CPU time of the developed filters, we also show the results in Table IX, with $N = 1000$ particles for the centralized PF, PPF1, and PPF2, and with approximately the same overall number of particles $GN = 16 * 60 = 960$. Comparing the results from Table XI with the results from Tables VIII and IX, one can see that the CPU time of the GSPF and PGSPF is smaller than the CPU time of the centralized and parallelized PFs.

The computational complexity of the centralized PF is proportional to the product of the number of particles N and the number of segments N_m , i.e., NN_m , whereas the computational complexity of the centralized GSPF is proportional to NN_mG . The experiments show that the GSPF and PGSPF provide high estimation accuracy with a smaller number of particles, compared with centralized PF.

With the GSPF, having $N = 100$ particles and $G = 8$ Gaussian mixture components, we can achieve comparable accuracy to a PF working with $N = 1000$ particles.

In large-scale road networks, other unknown variables need to be estimated, such as turn fractions and inflows. It is straightforward to include these additional variables in the estimation problem, as done in [39]. According to the analytical expressions for the computational complexity and communication needs (previously presented), the developed approaches are scalable in the sense that the computational demand and the communication demand remains constant per PU as long as the subnetwork size remains constant. An exception is the communication of PPF1 to the central PU, which linearly grows with the number of subnetworks. However, the main performance gain remains since it is due to the parallelization and to the efficient representation of the state pdf of the proposed filters.

X. CONCLUSION

This work has proposed a new approach to deal with high-dimensional nonlinear estimation problems. The developed techniques have provided state estimates that are necessary for online traffic control. The complex probability density function of the huge traffic network has been represented by local probability density functions in a simpler form. A contribution of this work has also been in the way the proposal distribution was defined, which affords parallelization of this high-dimensional estimation problem. PPFs and PGSPFs have been developed. The state prediction is locally performed within traffic segments, and only estimates for the boundary traffic states are transmitted on certain locations (where the sensors are positioned). The approaches are evaluated with a freeway state estimation problem for two scenarios (with and without congestions) and compared to a centralized PF and a centralized GSPF. The two proposed PPFs are with shared and separate particles, respectively.

The estimation accuracy and computational load per PU of PF approach 1 (with shared particles) for parallelization are equal to those of centralized PF with the same number of particles, whereas approach 2 for PF parallelization (separate particles) significantly outperforms centralized PF with the same number of particles. Thus, the main conclusion is that, despite the approximation used in approach 2, the performance of the PF transmitting estimates of the boundary states is superior to parallelized PF1 with shared particles. Naturally, the communication needs of the parallelized approaches are higher than those of the centralized PF, but the communication demand should not be a problem for the current data networks.

The accuracy of both centralized and parallelized GSPFs is much higher than the accuracy of the PFs for a comparable number of components. Another advantage of the parallelized GSPF is that they require transmission of the estimated boundary states and their covariances via the boundaries, which provides decomposition by region and by node and substantially decreases the computational load and communication demands. The proposed parallelized particle filtering approaches have potential and can be extended to solve other high-dimensional problems, e.g., in video processing.

REFERENCES

- [1] S. Ali-Löyty, "Efficient Gaussian mixture filter for hybrid positioning," in *Proc. PLANS IEEE/ION*, 2008, pp. 60–66.
- [2] S. Ali-Löyty and N. Sirola, "Gaussian mixture filter and hybrid positioning," in *Proc. ION GNSS*, Fort Worth, TX, Sep. 2007, pp. 562–570.
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [4] L. D. Baskar, B. De Schutter, and J. Hellendoorn, "Hierarchical model-based predictive control for intelligent vehicle highway systems: Regional controllers," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Madeira Island, Portugal, 2010, pp. 249–254.
- [5] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms and architectures for distributed particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2442–2450, Jul. 2005.
- [6] M. Borkar and J. H. McClellan, "Decentralized state initialization with delay compensation for multimodal sensor networks," *J. VLSI Signal Process.*, vol. 48, no. 1/2, pp. 109–125, Aug. 2007.
- [7] B. P. Carlin, N. G. Polson, and D. S. Stoffer, "A Monte Carlo approach to nonnormal and non-linear state-space modelling," *J. Amer. Stat. Assoc.*, vol. 87, no. 418, pp. 493–500, Jun. 1992.
- [8] C. Coates, "Distributed particle filtering for sensor networks," in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, Apr. 2004, pp. 99–107.
- [9] O. de Mouzon and N.-E. El Faouzi, "Real-time data fusion of road traffic and ETC data for road network monitoring," in *Proc. SPIE—Multisens., Multisource Inf. Fusion: Arch., Algorithms, Appl.*, B. V. Dasarthy, Ed., 2007, vol. 6571, p. 65 710J.
- [10] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *Proc. IEEE 32nd ICASSP*, 2007, pp. III-1181–III-1184.
- [11] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001, ch. Improving Regularised Particle Filters.
- [12] N.-E. El Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges—A survey," *Inform. Fusion*, vol. 12, no. 1, pp. 4–10, Jan. 2011.
- [13] A. H. Ghods, L. Fu, and A. Rahimi-Kian, "An efficient optimization approach to real-time coordinated and integrated freeway traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 873–884, Dec. 2010.
- [14] D. Gu, "Distributed particle filter for target tracking," in *Proc. Int. Conf. Robot. Autom.*, 2007, pp. 3856–3861.
- [15] A. Hegyi, T. Bellemans, and B. De Schutter, "Freeway traffic management and control," in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. New York: Springer-Verlag, 2009, pp. 3943–3964.
- [16] A. Hegyi, B. De Schutter, and J. Hellendoorn, "Optimal coordination of variable speed limits to suppress shock waves," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 102–112, Mar. 2005.
- [17] A. Hegyi, D. Girimonte, R. Babuška, and B. De Schutter, "A comparison of filter configurations for freeway traffic state estimation," in *Proc. 9th IEEE Intell. Transp. Syst. Conf.*, Toronto, ON, Canada, Sep. 2006, pp. 1029–1034.
- [18] A. Hegyi, L. Mihaylova, R. Boel, and Zs. Lendek, "Parallelized particle filtering for freeway traffic state tracking," in *Proc. Eur. Control Conf.*, Kos, Greece, Jul. 2–5, 2007, pp. 2442–2449.
- [19] S. P. Hoogendoorn, V. L. Knoop, and H. J. van Zuylen, "Robust control of traffic networks under uncertain conditions," *J. Adv. Transp.*, vol. 42, no. 3, pp. 357–377, Jul. 2008.
- [20] Y. Huang, W. Liang, H.-B. Yu, and Y. Xiao, "Target tracking based on a distributed particle filter in underwater sensor networks," *Wireless Commun. Mobile Comput.*, vol. 8, no. 8, pp. 1023–1033, Oct. 2008.
- [21] G. Ing, "Distributed particle filters for object tracking in sensor networks," M.S. thesis, Dept. Elect. Comput. Eng., McGill Univ., Montreal, QC, Canada, Dec., 2005.
- [22] J. H. Kotecha and P. M. Djuri, "Gaussian sum particle filtering for dynamic state space models," in *Proc. ICASSP*, Salt Lake City, UT, 2001, pp. 3465–3468.
- [23] J. H. Kotecha and P. M. Djuri, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- [24] J. H. Kotecha and P. M. Djuri, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.
- [25] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Study on fast model predictive controllers for large urban traffic networks," in *Proc. 12th Int. IEEE ITSC*, St. Louis, MO, Oct. 2009, pp. 691–696.
- [26] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical system," *J. Amer. Stat. Assoc.*, vol. 93, no. 443, pp. 1032–1044, Sep. 1998.
- [27] S. Maskell, K. Weekes, and M. Briers, "Distributed tracking of stealthy targets using particle filters," in *Proc. IEE Semin. Target Tracking: Algorithms Appl.*, Birmingham, U.K., Mar. 2006, pp. 13–20.
- [28] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within recursive Bayesian framework," *Automatica*, vol. 43, no. 2, pp. 290–300, Feb. 2007.
- [29] L. Mihaylova, A. Gning, V. Doychinov, and R. Boel, "Parallelised Gaussian mixture filtering for vehicular traffic flow estimation," in *Lecture Notes in Informatics—Proc. Ser. Gesellschaft für Informatik (GI)*, S. Fischer, E. Maehle, and R. Reischuk, Eds., Luebeck, Germany, 2009, vol. P-154, pp. 2321–2333.
- [30] R. Mittelman and E. L. Miller, "Nonlinear filtering using a new proposal distribution and the improved fast gauss transform with tighter performance bounds," *IEEE Trans. Signal Process.*, vol. 56, no. 12, pp. 5746–5757, Dec. 2008.
- [31] D. Ngoduy, S. P. Hoogendoorn, and R. Liu, "Continuum traffic flow modelling of cooperative traffic systems," *J. Phys. A, Stat. Mech. Appl.*, vol. 388, no. 13, pp. 2705–2716, Jul. 2009.
- [32] X. Sheng, Y. H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Proc. 4th Int. Conf. IPSN*, 2005, pp. 181–188.
- [33] Q. Song and X. Wang, "Efficient routing on large road networks using hierarchical communities," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 132–140, Mar. 2011.
- [34] X. Sun, L. Munoz, and R. Horowitz, "Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application," in *Proc. Amer. Control Conf.*, Boston, MA, 2004, pp. 2098–2103.
- [35] J. W. C. van Lint, "Online learning solutions for freeway travel time prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 38–47, Mar. 2008.
- [36] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
- [37] Y. Wang and M. Papageorgiou, "An adaptive freeway traffic state estimator and its real-data testing—Part I: Basic properties," in *Proc. 8th Int. Conf. Intell. Transp. Syst.*, Vienna, Austria, Sep. 13–16, 2005, pp. 531–536.
- [38] Y. Wang and M. Papageorgiou, "An adaptive freeway traffic state estimator and its real-data testing—Part II: Adaptive capabilities," in *Proc. 8th Int. IEEE Conf. Intell. Transp. Syst.*, Vienna, Austria, Sep. 13–16, 2005, pp. 537–542.
- [39] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B*, vol. 39, no. 2, pp. 141–167, Feb. 2005.



Lyudmila Mihaylova (SM'08) received the M.Sc. degree in systems and control engineering, the M.Sc. degree in applied mathematics in informatics, and the Ph.D. degree in systems and control engineering from the Technical University of Sofia, Sofia, Bulgaria, in 1989, 1991, and 1996, respectively.

She is currently a Reader in advanced signal processing with the School of Computing and Communication Systems, InfoLab21, Lancaster University, Lancaster, U.K. She is the Editor-in-Chief of the *Open Transportation Journal* and an Associate

Editor for the *Elsevier Signal Processing Journal*. Her work involves the development of novel Bayesian techniques, e.g., for high-dimensional problems (including for vehicular traffic flow estimation and for image processing), localization, and positioning in sensor networks. She has published book chapters and numerous journal and conference proceeding papers. Her research is funded by grants from the Engineering and Physical Sciences Research Council, the European Union (EU), the Ministry of Defense, and industry. Her research interests are nonlinear filtering, sequential Monte Carlo methods, statistical signal processing, and sensor data fusion.

Dr. Mihaylova is a member of the International Society of Information Fusion. She has given a number of invited tutorials, including for the COST-NEARCTIS workshop, and is involved in the organization of international conferences/workshops.



Andreas Hegyi received the M.Sc. degree in electrical engineering and the Ph.D. degree from Delft University of Technology (TU Delft), Delft, The Netherlands, 1998 and 2004, respectively.

In 2005, he visited Ghent University, Ghent, Belgium, as a Postdoctoral Researcher, where he worked on traffic state estimation with particle filters. He is currently an Assistant Professor with the Faculty of Civil Engineering and Geosciences, Department of Transport and Planning, TU Delft. His current research interests include intelligent trans-

portation systems, integrated/coordinated control of freeway and urban traffic networks, model predictive control, and traffic state estimation with data fusion and filtering techniques.



Amadou Gning received the Ph.D. degree in technologies of information and systems from the Université de Technologie de Compiègne, Compiègne, France, in 2006.

Since 2007, he has been with the School of Computing and Communication Systems, InfoLab21, Lancaster University, Lancaster, U.K. His research interests are sensor data fusion, which cover theoretical tools such as nonlinear filtering, sequential Monte Carlo methods, statistical signal processing, distributed sensor networks, and interval analysis.



René K. Boel received the B.S. degrees in electro-mechanical engineering and nuclear engineering from Ghent University, Ghent, Belgium, in 1969 and 1970, respectively, and the M.Sc. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1972 and 1974, respectively.

He is currently a Professor with the Department of Electrical Energy, Systems and Automation, SYSTeMS, Ghent University. He was a Research Fellow of the F.W.O.-Vlaanderen Flemish Founda-

tion for Scientific research (F.W.O.-Vlaanderen) until September 2000. He has held temporary appointments with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley; the Department of Technical Cybernetics NTH, Trondheim Norway; the Mathematics Department, Katholieke Universiteit Leuven, Leuven, Belgium; the Mathematics and Statistics Department, Bell Laboratories, Murray Hill, NJ; the Department of Computing and Control, Imperial College London, London, U.K.; the Department of Systems Engineering, Australian National University, Canberra, Australia; the Department of Electrical and Computer Engineering, University of Newcastle, Newcastle, Australia; and the Electrical and Electronic Engineering Department, Melbourne University, Melbourne, Australia. His current research interests are distributed control and estimation for stochastic, discrete event, and hybrid systems, with applications to traffic and power systems.