Proceedings of the European Control Conference 2007
Kos, Greece, July 2-5, 2007



# Parallelized particle filtering for freeway traffic state tracking

A. Hegyi, L. Mihaylova, R. Boel, Zs. Lendek

*Abstract*— **We consider parallelized particle filters for state tracking (estimation) of freeway traffic networks. Particle filters can accurately solve the state estimation problem for general nonlinear systems with non-Gaussian noises. However, this high accuracy may come at the cost of high computational demand.**

**We present two parallelized particle filtering algorithms where the calculations are divided over several processing units (PUs) which reduces the computational demand per processing unit. Existing parallelization approaches typically assign sets of particles to PUs such that each full particle resides at one PU. In contrast, we partition *each particle* according to a partitioning of the network into subnetworks based on the topology of the network.**

**The centralized case and the two proposed approaches are evaluated with a benchmark problem by comparing the estimation accuracy, computational complexity and communication needs.**

**This approach is in general applicable to systems where it is possible to partition the overall state into subsets of states, such that most of the interaction takes place within the subsets.**

**Keywords: Parallel particle filters, freeway traffic state tracking.**

## I. Introduction

To manage urban and freeway road traffic, traffic data is collected in traffic control centers in many countries. This data is often used for traffic monitoring, control, and information dissemination. The quality of the data is for several reasons often not as good as one would wish. Direct traffic measurements from the sensors are corrupted by noise, or some data may be missing, and in some countries the data is aggregated over a longer time period, or the detectors are located at large distances to each other[1].

In this paper we present a particle filtering (PF) method that can cope with the above mentioned problems and is suitable to large networks by the possibility of parallel implementation. State estimation filters provide an estimation of the traffic state combining the knowledge of the system behavior with the available (sometimes sparse) measurements to estimate the state.

Several traffic state tracking filters have been investigated in the literature. In [17] an extended study is presented of estimation schemes with the extended Kalman filter (EKF). This approach is evaluated for real traffic data in [15], [16]. In [10] a PF is applied to estimate the traffic state (speed and density) based on flow and speed measurements at the boundaries of the stretch. In a similar study the PF is compared with an unscented Kalman (UKF) filter in [11]. The performance of the EKF and the UKF is compared for traffic state estimation in [7] for several filter settings. In [14] a mixture Kalman filter is employed to simultaneously detect the discrete traffic state (free-flow or congested) and track the traffic speed.

For general nonlinear systems with non-Gaussian noises particle filters are more suitable than most of the existing approaches. The only potential disadvantage of particle filtering compared with the other methods is the higher computational complexity. To reduce complexity, different approaches for parallelized and distributed particle filters are proposed in the literature [4], [9], [13]. They can be classified in two groups: i) algorithms transmitting particle values and their weights between the processing units (PUs) or ii) communicating a parametric approximation. Most of these implementations are for sensor network related problems and have the tendency to minimize communications. In [13] two distributed particle filters are proposed with Gaussian mixture approximation of the belief 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. Other implementations, e.g., in [2] 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 communication) if the number of particles is high.

In this paper we develop a particle filtering approach for traffic networks, where the computational demand per PU is reduced through parallelization. We present two approaches, both using a parallelization scheme which is based on the topological partitioning of a traffic network into subnetworks.

We demonstrate the parallelized approach with a benchmark problem, in which the traffic network consists of a freeway stretch partitioned into two substretches. We compare the accuracy, the computational complexity, and the communication needs for the three filters.

## II. State tracking by particle filtering

In state estimation problems, the state-space representation of the dynamical system is used. This describes the evolution of the system state $x_k$ over time, and the measurements $z_k$ as

A. Hegyi and Zs. Lendek are with TU Delft, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands. a.hegyi@tudelft.nl, zs.lendek@tudelft.nl

R. Boel is with the University of Ghent, Department of Electrical Energy, Systems and Automation, SYSTeMS. rene.boel@ugent.be

L. Mihaylova is with the Lancaster University, Department of Communication Systems, InfoLab21. mila.mihaylova@lancaster.ac.uk

[1]For example, in Belgium inductive loops are typically placed only near on- and off-ramps on the freeways, which may be several km's apart.

ISBN: 978-960-89028-5-5

a function of the state:

$$x_k = f(x_{k-1}, v_{k-1}), \quad (1)$$
$$z_k = h(x_k, n_k), \quad (2)$$

where $v_k$ is the state noise, $n_k$ the measurement noise, and $k$ the sample step counter. These equations define a probability density function (pdf) for the state transition $p(x_k|x_{k-1})$ and for the measurement $p(z_k|x_k)$.

Since the system and the measurements are stochastic, the exact state cannot be inferred from the measurements, only the pdf of the state $p(x_k|z_{1:k})$ can be determined given all measurements $z_{1:k}$ from sample step 1 to $k$. So, the goal of the state estimation problem is to determine $p(x_k|z_{1:k})$. Although it is possible to use Bayes' rule to express this conditional density in terms of the state transition pdf $p(x_k|x_{k-1})$, and the measurement pdf $p(z_k|x_k)$, it requires the evaluation of several integrals, which is not possible (analytically) in general, nor is it efficient to evaluate them numerically [12]. For these reasons we will focus on particle filtering, which provides a trade-off between accuracy and efficiency for the state estimation problem.

In the next section we introduce the general formulation of particle filtering and present two approaches for parallelization. Although the parallelization is explained for traffic networks, the same approach can be followed for other processes where the overall state can be partitioned into subsets of states where the interaction between the states takes mainly place *within* one subset.

### A. General formulation particle filtering

The goal of the state estimation is to determine the pdf $p(x_k|z_{1:k})$ at each time step $k$. According to Bayes' rule this may be written as

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \quad (3)$$

To circumvent the integration that is necessary for the evaluation of the right hand side of (3), in particle filters the pdf $p(x_k|z_{1:k})$ is approximated by a *random measure* $\{x^i_{0:k}, w^i_k\}^I_{i=1}$, where $\{x^i_{0:k}, i = 0,\ldots,I\}$ is a set of support points with weights $\{w^i_k, i = 0,\ldots,I\}$, where each $\{x^i_{0:k}, w^i_k\}^I_{i=1}$ is called a particle, and $I$ is the number of particles. The posterior density at $k$ is approximated as

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{I} w^i_k \delta(x_{0:k} - x^i_{0:k}). \quad (4)$$

Each particle $\{x^i_{0:k}, w^i_k\}$ can be considered as a hypothesis that the real state trajectory is $x^i_{0:k}$ with belief $w^i_k$. This hypothesis is updated in two steps:

1) **state update.** When $k$ is increased by one, a new value is appended to $x^i_{0:k}$ to form $x^i_{0:k+1}$, according to the state equation (1), using the assumption that the previous state was $x^i_k$ and a given sample drawn from $v^i_k \sim p(v_k)$.
2) **measurement update.** When a measurement arrives, the belief in the particles will in general change, which is reflected by the updates weights $w^i_{k+1}$. The belief in the particle changes according to how well it explains the measurement.

In both steps the principle of *importance sampling* plays a crucial role, which can be explained as follows (cf. [1]).

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

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

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

Given this principle, the weights in (4) are defined to be

$$w^i_k \propto \frac{p(x^i_{0:k}|z_{1:k})}{q(x^i_{0:k}|z_{1:k})},$$

where the proposal distribution $q(x)$ can be chosen arbitrarily, nevertheless the choice of $q(x)$ is a relevant step. It can be shown [1] that if $q(x)$ is chosen to factorize as

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k}) q(x_{0:k-1}|z_{1:k-1})$$

then the weights are determined *recursively* by

$$w^i_k \propto w^i_{k-1} \frac{p(z_k|x^i_k) p(x^i_k|x^i_{k-1})}{q(x^i_k|x^i_{k-1}, z_k)}. \quad (5)$$

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

A frequently used proposal distribution is the prior:

$$q(x_k|x^i_{k-1}, z_k) = p(x_k|x^i_{k-1}).$$

Using this in (5) results in a simple weight update rule:

$$w^i_k \propto w^i_{k-1} p(z_k|x^i_k).$$

*2) 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 undesired. To prevent this, the particles are regularly resampled, i.e., particles with small weights are eliminated, and new particles are created at or around the ones with large weights (such that the approximation in (4) still holds). To decide when to resample, the effective number of particles $\widehat{I^{\text{eff}}} = 1/\sum_{i=1}^{I} (w^i_k)^2$ is compared with some predefined threshold $I^{\text{threshold}}$.

There exist several efficient resampling algorithms of $O(I)$ that typically map the newly created particles to existing ones with high weights, such as the residual resampling [8] and the systematic resampling [1]. In this paper we will use systematic resampling as given by the algorithm 'RESAMPLE', and the basic (centralized) particle filtering algorithm is described by the algorithm 'PF' as shown in the frame below.

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

$[\{x_k^i, w_k^i\}_{i=1}^I] =$ PF$[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^I, z_k]$
**for** $i = 1 : I$ *(for each particle)* **do**
  - Draw $x_k^i \sim q(x_k^i | x_{k-1}^i, z_k)$
  - Determine the weight update factors according to (5).
  - Normalize weights, $w_k^i = w_k^i / \sum_i w_k^i$
  **if** $\widehat{I^{eff}} < I^{threshold}$ **then**
    - Resample particles.
  **end**
  - Calculate the expected state: $E[x_k] = \sum_{i=1}^I w_k^i x_k^i$.
**end**

### B. Parallelization

When particle filtering is applied for the state tracking of 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 particle filtering. In this section we present two approaches for parallelizing particle filtering.

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 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 $x_k^s, s = 1, ..., S$ with $x_k = [(x_k^1)^{\mathrm{T}}, (x_k^2)^{\mathrm{T}}, \ldots, (x_k^S)^{\mathrm{T}}]^{\mathrm{T}}$, and $z_k = [(z_k^1)^{\mathrm{T}}, (z_k^2)^{\mathrm{T}}, \ldots, (z_k^S)^{\mathrm{T}}]^{\mathrm{T}}$. The system (1)-(2) can now be described by

$$x_k^s = f_k^s(x_{k-1}^s, \hat{x}_{k-1}^s, v_{k-1}^s), \tag{6}$$
$$z_k^s = h_k^s(x_k^s, n_k^s), \tag{7}$$

where $s = 1, ..., S$, and the vector $\hat{x}_{k-1}^s$ collects all neighboring

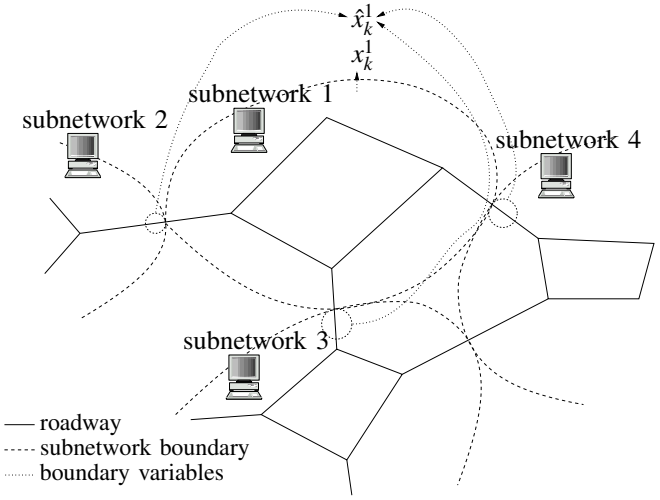


Fig. 1. An example of partitioning a traffic network into subnetworks for parallelized simulation/particle filtering.

state variables that act as an input to subnetwork $s$. Note that not all states of the neighboring networks are communicated, only the variables that serve as a input to subnetwork $s$.

Note also that for (7) it is assumed that the measurements taken in a subnetwork only depend on the state in that subnetwork. This assumption holds for traffic since detectors typically measure the traffic variables at one given location.

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

$$p(x_k | x_{k-1}) = \prod_{s=1}^{S} p(x_k^s | x_{k-1}^s, \hat{x}_{k-1}^s), \tag{8}$$
$$p(z_k | x_k) = \prod_{s=1}^{S} p(z_k^s | x_k^s). \tag{9}$$

*1) First approach: shared particles:* In this approach the PUs of different subnetworks share the same particles $x_k^i$, and the particles are partitioned into subparticles $x_k^{s,i}$ corresponding to subnetwork $s$. The PU corresponding to subnetwork $s$ is responsible for the calculations for subparticles $x_k^{s,i}$. This approach is functionally equivalent to the centralized approach, as presented above, given that (8) and (9) hold.

In the state update step, the subparticles $x_k^{s,i}$ are now drawn from the distribution $q(x_k^s | x_{k-1}^{s,i}, \hat{x}_{k-1}^{s,i}, z_k^s)$ which is based on local information only (including neighboring states). Now, choosing the proposal distribution $q(x_k^s | x_{k-1}^s, \hat{x}_{k-1}^s, z_k^s)$ such that (c.f. (8))

$$q(x_k | x_{k-1}^i, z_k) = \prod_{s=1}^{S} q(x_k^s | x_{k-1}^{s,i}, \hat{x}_{k-1}^{s,i}, z_k^s), \tag{10}$$

and using (8)-(9) and the facts that

$$p(x_k^{s,i} | x_{k-1}^{s,i}) = \sum_{j=1}^{I} p(x_k^{s,i} | x_{k-1}^{s,i}, \hat{x}_{k-1}^{s,j}) p(\hat{x}_{k-1}^{s,j} | x_{k-1}^{s,i}),$$
$$p(\hat{x}_{k-1}^{s,j} | x_{k-1}^{s,i}) = \begin{cases} 1 \text{ if } i = j, \\ 0 \text{ otherwise.} \end{cases} \tag{11}$$

the weight update equation (5) can be rewritten as

$$w_k^i \propto w_{k-1}^i \prod_{s=1}^{S} w_{k-1}^{s,i} \quad , \quad \sum_i w_k^i = 1, \tag{12}$$

$$w_{k-1}^{s,i} = \frac{p(z_k^s|x_k^{s,i})p(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,i})}{q(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,i},z_k^s)}. \tag{13}$$

The consequence of (12)–(13) is that the state and measurement update can be performed locally (divided over $S$ processors) except for the weight update.

For each time step $k$ in the PF the following communication has to take place:

- The state variables $\hat{x}_{k-1}^{s,i}$ at the boundaries have to be sent to subnetwork $s$.
- The weight update factors $w_{k-1}^{s,i}$ can be calculated locally, and only the results need to be communicated to a central PU to determine $w_k^i$.
- The centrally calculated weights $w_k^i$ are normalized, and sent back to the local PUs (after resampling, when necessary).
- Resampling requires communication to a central PU where all weights $w_k^{p,i}$ are collected and the $w_k^i$ are calculated according to (12). For the residual resampling [8] and the systematic resampling [1] methods it is not necessary to communicate the particles themselves, since these methods use only the weights as the input and produce as a result the new particles as a selection from the existing old ones (with some selected multiple times, others not at all). Therefore, after resampling only the indices describing the selection are communicated back to the PUs. For the resampling algorithms that create particles at new locations in the state space, such as regularization [5] and the MCMC scheme [3] the particles themselves also have to be communicated, and consequently more communication is necessary.

The pseudo code for the algorithm is given by 'PF1' in the frame below.

*2) Second approach: separate particles:* In this approach the particles of the different subnetworks are not shared, only the statistics of the neighboring traffic state is communicated over the boundaries to each subnetwork $s$. Consequently (11) does not hold anymore, but instead

$$p(x_k^{s,i}|x_{k-1}^{s,i}) = \int_{\hat{x}_{k-1}^{s,j}} \{p(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j})p(\hat{x}_{k-1}^{s,j}|x_{k-1}^{s,i})\}\, d\hat{x}_{k-1}^{s,j}. \tag{14}$$

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

$$p(x_k^{s,i}|x_{k-1}^{s,i}) \approx \sum_j \frac{p(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j})p(\hat{x}_{k-1}^{s,j}|x_{k-1}^{s,i})}{q(\hat{x}_{k-1}^{s,j}|x_{k-1}^{s,i})}. \tag{15}$$

Note that since the pdf of the communicated state variables is independent of $x_{k-1}^{s,i}$ (by assumption),

$$p(\hat{x}_{k-1}^{s,j}|x_{k-1}^{s,i}) = p(\hat{x}_{k-1}^{s,j}). \tag{16}$$

$[\{x_k^i, w_k^i\}_{i=1}^I] =$ PF1$[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^I, z_k]$
**for** $s = 1:S$ *(for each subnetwork simultaneously)* **do**
  **for** $i = 1:I$ *(for each particle)* **do**
    - Draw $x_k^{s,i} \sim q(x_k^s|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,i},z_k^s)$
    - Determine the weight update factors according to (13) and send them to the central PU.
    At the central PU:
    - Determine $w_k^i$ according to (12).
    - Normalize weights, $w_k^i = w_k^i / \sum_i w_k^i$
    **if** $\widehat{I^{eff}} < I^{threshold}$ **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 $x_k^{s,j} \leftarrow x_k^{s,i^j},\ j = 1,\ldots,I.$
  - Calculate the estimate of the state of subnetwork $s: E[x_k^s] = \sum_{i=1}^I w_k^i x_k^{s,i}$.
**end**

$[\{x_k^i, w_k^i\}_{i=1}^I] =$ PF2$[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^I, z_k]$
**for** $s = 1:S$ *(for each subnetwork)* **do**
  **for** $i = 1:I$ *(for each particle)* **do**
    - Draw $\hat{x}_{k-1}^{s,j_i} \sim p(\hat{x}_{k-1}^s)$
    - Draw $x_k^{s,i} \sim q(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j_i},z_k^s)$
    - Determine the weight update factors according to (18).
    - Normalize weights, $w_k^{s,i} = w_k^{s,i} / \sum_i w_k^{s,i}$
    **if** $\widehat{I^{eff,s}} < I^{threshold,s}$ **then**
      - Resample particles.
    **end**
    - Calculate the estimate of the state of subnetwork $s: E[x_k^s] = \sum_{i=1}^I w_k^{s,i} x_k^{s,i}$.
  **end**
**end**

Using this relation and taking only one sample from $\hat{x}_{k-1}^{s,j_i} \sim p(\hat{x}_{k-1}^s)$ for each $i$, and choosing $q(\hat{x}_{k-1}^{s,j}|x_k^{s,i}) = p(\hat{x}_{k-1}^{s,j})$, (15) simplifies to

$$p(x_k^{s,i}|x_{k-1}^{s,i}) \approx p(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j_i}). \tag{17}$$

In this approach the weights are updated locally since there are no centralized particles (c.f. (12)):

$$w_k^{s,i} = w_{k-1}^{s,i} \frac{p(z_k^{s,i}|x_k^{s,i})p(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j_i})}{q(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j_i},z_k^s)}. \tag{18}$$

This form means for the particle filter that first $\hat{x}_{k-1}^{s,j_i}$ needs to be sampled from $p(\hat{x}_{k-1}^s)$ and then $x_k^{s,i}$ according to the the importance density $q(x_k^{s,i}|x_{k-1}^{s,i},\hat{x}_{k-1}^{s,j_i},z_k^s)$.The algorithm is given by 'PF2' in the frame below.

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 $\hat{x}_{k-1}^{s,j_i}$ are sent to subnetwork $s$, after these quantities are drawn from $\hat{x}_{k-1}^{s,j_i} \sim p(\hat{x}_{k-1}^{s})$. In this approach resampling does not require communication since it can be performed locally at each PU.

The advantages of this approach over the approach where the particles are shared are the following:

- it can be expected that this approach requires less particles since the dimension of the state space is reduced by a factor $S$ (assuming that all subnetworks have the same number of states).
- 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 (16)).

## III. Benchmark

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

The benchmark network is small, but the same approach can be applied to networks of any size. Before the presentation of the benchmark problem, the freeway model is described here.

*3) Freeway model – the METANET model:* Consider a freeway link $m$ that is subdivided into $N_m$ segments, each with a length $L_m$ and $\lambda_m$ lanes, and a discrete time step with length $T$ (h). Traffic dynamics is described in terms of the aggregated variables *speed* $v_{m,i}(k)$ (km/h), *flow* $q_{m,i}(k)$ (veh/h), and *density* $\rho_{m,i}(k)$ (veh/km/lane), where $i$ is the segment index. In Fig. 2 the relevant variables are shown.

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

$$q_{m,i}(k) = \rho_{m,i}(k) v_{m,i}(k) \lambda_m , \tag{19}$$

the law of conservation of vehicles

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}(q_{m,i-1}(k) - q_{m,i}(k)) + \xi_{m,i}^{\rho}(k) \tag{20}$$

and a heuristic relationship of the speed dynamics

$$v_{m,i}(k+1) = v_{m,i}(k) + \frac{T}{\tau}\left(V(\rho_{m,i}(k)) - v_{m,i}(k)\right) + \frac{T}{L_m} v_{m,i}(k)\left(v_{m,i-1}(k) - v_{m,i}(k)\right) - \frac{\eta T}{\tau L_m}\frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} + \xi_{m,i}^{v}(k), \tag{21}$$

$$V(\rho_{m,i}(k)) = v_{\text{free},m} \exp\left[-\frac{1}{a_m}\left(\frac{\rho_{m,i}(k)}{\rho_{\text{crit},m}}\right)^{a_m}\right], \tag{22}$$

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 (20) is an exact relationship and therefore 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 [7]. 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 $\tau$, $\eta$, $a_m$, $\kappa$, are 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) [6]. The parameter $\eta$ in (21) is replaced by the density dependent $\eta_{m,i}(k)$ according to:

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

### A. Boundary conditions

The variables $q_{m,0}$, $v_{m,0}$, $\rho_{m,N_m+1}$ 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 measured directly, whereas in practice the density $\rho_{m,N+1}$ is not measured directly and must be estimated. Even though $q_{m,0}$ and $v_{m,0}$ can be measured directly, 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 [17]. The dynamic evolution of the boundary variables is described by a random walk:

$$\begin{bmatrix} q_{m,0}(k+1) \\ v_{m,0}(k+1) \\ \rho_{m,N_m+1}(k+1) \end{bmatrix} = \begin{bmatrix} q_{m,0}(k) \\ v_{m,0}(k) \\ \rho_{m,N_M+1}(k) \end{bmatrix} + \begin{bmatrix} \xi_{m,0}^{q}(k) \\ \xi_{m,0}^{v}(k) \\ \xi_{m,N_M+1}^{\rho}(k) \end{bmatrix}, \tag{23}$$

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

### B. Measurements

The most frequently used traffic measurement devices typically measure speed and flow. For the segments that are

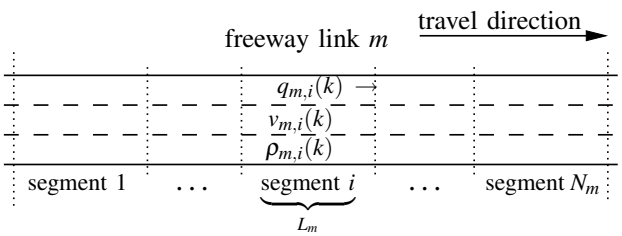


Fig. 2. In the METANET model, a freeway link is divided into segments. The main variables in the model are the average outflow of a segment $q_{m,i}(k)$, the average speed $v_{m,i}(k)$, the average density $\rho_{m,i}(k)$, and the segment length $L_m$.

equipped with sensors the measurement equations are:

$$y^q_{m,i}(k) = q_{m,i}(k) + n^q_{m,i}(k), \tag{24}$$

$$y^v_{m,i}(k) = v_{m,i}(k) + n^v_{m,i}(k), \tag{25}$$

where $n^q_{m,i}(k)$, and $n^v_{m,i}(k)$ 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 minutes, while the model time step is 10s. Including this fact in the development of a PF is straightforward, but for the sake of simplicity we assume that each model time step a measurement is available.

### C. State space representation

To bring equations (19)–(23) into the state-space representation required by the various filters, the state $x_k$ is defined as[2] $x_k = [\rho_1(k),\ldots,\rho_N(k),v_1(k),\ldots,\ v_N(k),v_0(k),q_0(k),\rho_{N+1}]^{\mathrm{T}}$, and the measurement vector $z_k = [y^q_{m,i}(k)^T, y^v_{m,i}(k)^T]^T$ collects the flow and speed measurements from (24) and (25) for the segments equipped with sensors.

### D. Experiment design

*1) Lay-out:* The benchmark network consists of a 2-lane freeway link of 10 segments of 1 km each. For the two parallel approaches this link is divided into two sublinks ("subnetworks") consisting of respectively the first and the last five segments.

*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, $\mathrm{var}(\xi^v_{m,i}(k)) = 0.5\,(\mathrm{km/h})^2, \mathrm{var}(\xi^\rho_{m,i}(k)) = 0.5\,(\mathrm{veh/km/lane})^2$, and measurement noise variances $\mathrm{var}(n^v_{m,i}(k)) = 4\,(\mathrm{km/h})^2$, and $\mathrm{var}(n^q_{m,i}(k) = 22500\,(\mathrm{veh/h})^2$.

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

*4) Detector configurations:* Several detector configurations are investigated. Segments that have detectors provide speed and flow measurements. For most experiments it is assumed that all segments are equipped with detectors.

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.

[2]The link index $m$ is omitted in the rest of this section assuming that all the variables introduced hereafter refer to the same link.

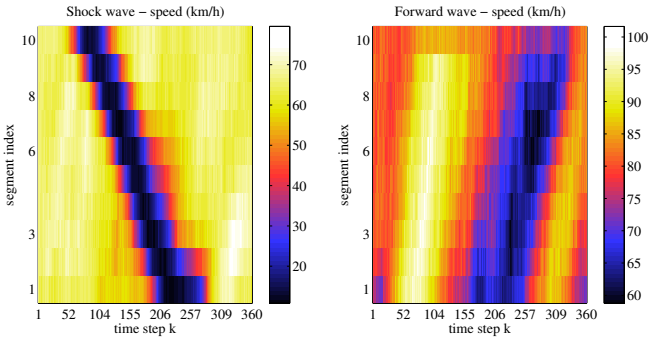


Fig. 3. The shock wave (left) and the forward wave (right) scenario, used for the evaluation of the filters. The travel direction is from segment 1 to 10. The colors indicate the speed. Please note the difference in color bar scales: the shock wave scenario includes a wider range of speed since it also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

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).

*5) Filter setup:* The particle filters are set up according to the algorithms in Section II-A and II-B. For the proposal distribution the prior is used, and the filters are investigated for several numbers of particles in the range $I \in \{20, 50, 100, 200, 500, 1000\}$. The resampling threshold is chosen to be $I^{\mathrm{threshold}} = 0.3$. The state noise $v^i_k$, which is sampled during the operation of the filters, is taken to have the same realization for the three different filters, for better comparability.

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

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

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

  where $\hat{\rho}_{i,k}$ and $\tilde{\rho}_{i,k}$ are the density components of respectively the real state and the expected state of the particles for segment $i$ at time $k$, and $K$ is the number of simulation steps. $J_{\mathrm{RMSE},v}$ is calculated similarly.
- **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.

- **CPU time.** As a measure for the computational demand the time that each filter needs for a complete run is determined.

## IV. Results & Discussion

Fig. 4 shows the results for the tracking accuracy as a function of the number of particles, for the centralized filter (top), approach 1 (middle), and approach 2 (bottom). In these experiments all segments were measured and the shock wave scenario was used. The experiments were repeated 10 times and the figure shows the averages and the standard deviations of these experiments. For all filters the performance get better (lower error) when the number of particles increases. The performance of the centralized filter and the performance of approach 1 is equal because the two filters are functionally equivalent and the same noise realization is 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 pdf's made at the boundaries of the sublinks.

Since there is no visible improvement between $I = 500$ and $I = 1000$ we select $I = 500$ for the other experiments.

Fig. 5 shows the CPU time needed by the different filters to complete a full scenario, as a function of the number of particles, again averaged over 10 runs. The averaging is necessary here as the measured CPU time may vary with some internal operations of the PC, such as memory swapping. The simulations were executed on a 2800 MHz Intel Pentium IV PC.

The shown values are normalized by the number of particles, and the curves are nearly flat which indicates that the required CPU depends linearly on the number of particles (as expected). 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 2 PUs instead of 1. 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.

The effect of the approximation introduced in approach 2 is investigated by the experiments with the shock wave and the forward wave scenarios. The detector locations are selected such that there are no detectors near the boundary in sublink 1 for the shock wave scenario, and no detectors near the boundary in sublink 2 for the forward wave scenario. In this way the information about the waves is communicated through the boundary of the two PUs and not via the measurements.

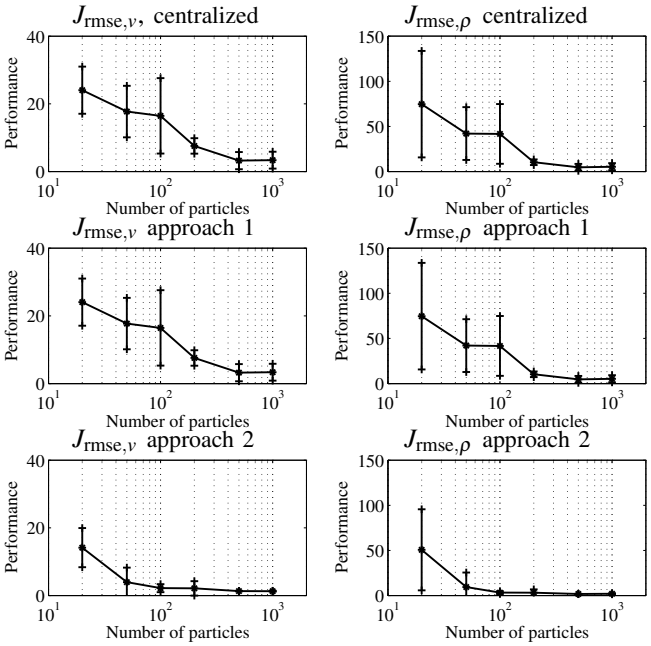


Fig. 4. The filter root mean square error as a function of the number of particles for the centralized filter (top), approach 1 (middle), and approach 2 (bottom). The dots connected by the solid line indicate the mean, and the vertical lines with the '+'-signs the standard deviation over the 10 experiments. All segments are measured. Note the logarithmic scale of the horizontal axis.

The performance of approach 2 is compared with the centralized filter (or the functionally equivalent approach 1) in Table I. Also in this case, the performance is averaged over 10 experiments and the standard deviation is shown in parentheses. The performance of approach 2 is significantly better, so we can conclude that the communication between the sublinks is sufficient to track forward and backward propagating waves.

Finally, the communication needs are shown in Table II. For the centralized filter only the measurements are communicated, which are independent of the number of particles. For the parallelized approaches, communication also takes place between the boundary of the sublinks, and to the central PU in case of approach 1, which increases the communication needs significantly. Nevertheless, these amounts of communication should not impose a problem for running these filters in real-time.

## V. Conclusions

We proposed two approaches for the parallelization of the particle filters. The approaches are evaluated with a freeway state tracking problem for two scenarios. The performance in terms of tracking accuracy, computational load per PU are equal (approach 1) or significantly better (approach 2) than for the centralized filter with the same number of particles.

| detector location (segment index) | scenario | centralized | | approach 1 | | approach 2 | |
|---|---|---|---|---|---|---|---|
| | | $J_{\text{rmse},v}$ | $J_{\text{rmse},\rho}$ | $J_{\text{rmse},v}$ | $J_{\text{rmse},\rho}$ | $J_{\text{rmse},v}$ | $J_{\text{rmse},\rho}$ |
| 1,x,x,x,x,6,7,8,9,10 | shock wave | 11.4 (2.5) | 14.6 (4.6) | 11.4 (2.5) | 14.6 (4.6) | 4.6 (2.3) | 4.7 (2.9) |
| 1,2,3,4,5,x,x,x,x,10 | forward wave | 3.1 (0.26) | 1.8 (0.16) | 3.1 (0.26) | 1.8 (0.16) | 2.8 (0.18) | 1.7 (0.11) |

TABLE I

THE PERFORMANCE AND STANDARD DEVIATION (IN PARENTHESES) FOR DIFFERENT DETECTOR LOCATIONS AND SCENARIOS, WITH $I = 500$ AND ALL EXPERIMENTS ARE REPEATED 10 TIMES. THE 'X' INDICATES AN UNMEASURED SEGMENT.

| $I$ | centralized | approach 1 | approach 2 |
|---|---|---|---|
| 20 | 7180 | 57440 | 28720 |
| 50 | 7180 | 132830 | 61030 |
| 100 | 7180 | 258480 | 114880 |
| 200 | 7180 | 509780 | 222580 |
| 500 | 7180 | 1263680 | 545680 |
| 1000 | 7180 | 2520180 | 1084180 |

TABLE II

THE NUMBER OF COMMUNICATED DOUBLES FOR EACH APPROACH AS A FUNCTION OF THE NUMBER OF PARTICLES $I$.

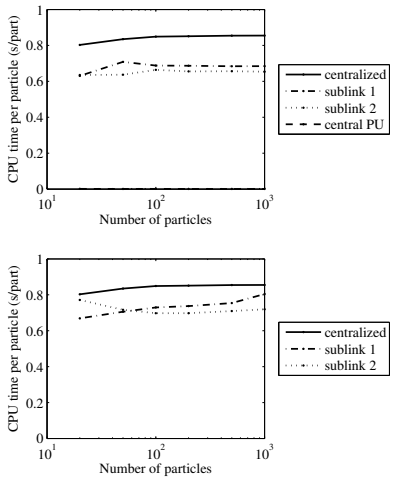


Fig. 5. The CPU time per particle for approach 1 (top) and approach 2 (bottom) as a function of the number of particles. The solid line is the CPU time for the centralized filter. All segments are measured. The simulations were executed on a 2800 MHz Intel Pentium IV PC. Note the logarithmic scale of the horizontal axis.

So the main conclusion is that despite the approximation used in approach 2, the performance of the filter was superior to the others for the experiments we carried out. Naturally, the communication needs of the parallelized approaches are higher than for the centralized filter, but the communication demand should not be a problem for the current data networks.

## VI. ACKNOWLEDGMENTS

Financial support by the project DWTC-CP/40 "Sustainability effects of traffic management", sponsored by the Belgian government is gratefully acknowledged, as well as by the Belgian Programme on Inter-University Poles of Attraction V/22 on "Dynamical Systems and Control: Computation, Identification and Modelling" initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture.

## REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[2] M. Bolic, P.M. Djuric, and S. Hong. Resampling algorithms and architectures for distributed particle filter. *IEEE Trans. Signal Processing*, 53:2442–2450, 2005.

[3] B. P. Carlin, N. G. Polson, and D. S. Stoffer. A Monte Carlo approach to nonnormal and non-linear state-space modelling. *Journal of the American Statistical Association*, 87(418):493 – 500, 1992.

[4] C. Coates. Distributed particle filtering for sensor networks. In *Proc. of the Int. Symp. Information Processing in Sensor Networks*, Berkeley, California, April 2004.

[5] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*, chapter Improving Regularised Particle Filters. Springer-Verlag, New York, 2001.

[6] A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):102–112, March 2005.

[7] A. Hegyi, D. Girimonte, R. Babuška, and B. De Schutter. A comparison of filter configurations for freeway traffic state estimation. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems 2006*, pages 1029–1034, Toronto, Canada, September 17–20 2006.

[8] J. S. Liu and R. Chen. Sequential Monte Carlo methods fory dynamical system. *Journal of the American Statistical Association*, 93:1032 – 1044, 1998.

[9] S. Maskell, K. Weekes, and M. Briers. Distributed tracking of stealthy targets using particle filters. In *Proc. of IEE seminar on target tracking: algorithms and applications*, pages 13–20. IEE, Birmingham, UK, March 2006.

[10] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 2106–2111, Atlantis, Paradise Island, Bahamas, 2004.

[11] L. Mihaylova, R. Boel, and A. Hegyi. Freeway traffc estimation within particle filtering framework. *Automatica*, 43(2):290–300, 2007.

[12] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter*. Artech House, 2004. ISBN: 1-58053-631.

[13] 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 *4th Intl. Conf. on Information Processing in Sensor Networks (IPSN)*, pages 181–188, 2005.

[14] X. Sun, L. Mu noz, and R. Horowitz. Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application. In *Proceedings of the American Control Conference*, pages 2098–2103, Boston, USA, 2004.

[15] Y. Wang and M. Papageorgiou. An adaptive freeway traffic state estimator and its real-data testing–Part I: Basic properties. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 531–536, Vienna, Austria, September 13–16, 2005.

[16] Y. Wang and M. Papageorgiou. An adaptive freeway traffic state estimator and its real-data testing–Part II: Adaptive capabilities. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 537–548, Vienna, Austria, September13–16 2005.

[17] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2):141–167, 2005.