

RESEARCH

Open Access

# Localization of acoustic sources using a decentralized particle filter

Florian Xaver<sup>1\*</sup>, Gerald Matz<sup>1</sup>, Peter Gerstoft<sup>2</sup> and Christoph Mecklenbräuer<sup>1</sup>

## Abstract

This paper addresses the decentralized localization of an acoustic source in a (wireless) sensor network based on the underlying partial differential equation (PDE). The PDE is transformed into a distributed state-space model and augmented by a source model. Inferring the source state amounts to a non-linear non-Gaussian Bayesian estimation problem for whose solution we implement a decentralized particle filter (PF) operating within and across clusters of sensor nodes. The aggregation of the local posterior distributions from all clusters is achieved via an enhanced version of the maximum consensus algorithm. Numerical simulations illustrate the performance of our scheme.

**Keywords:** source localization, acoustic wave equation, distributed state-space model, sequential Bayesian estimation, decentralized particle filter, argumentum-maximi consensus algorithm

## 1 Introduction

### Background and state of the art

In this paper, we use a physics-based model and a Bayesian approach to develop a decentralized particle filter (PF) for acoustic source localization in a sensor network (SN). In a decentralized PF, the processing is done locally at the sensors without using a fusion center. Thereby, the estimated position is known at every sensor in consequence of this decentralized process.

The problem formulation in this paper is motivated by indoor localization of an acoustic source. A hallway is modeled including basic boundary conditions for windows (membranes) and walls.

The source localization problem has been studied, e.g., in [1-3], [[4], p. 4089 ff], [[5], p. 746 ff] and [6], all of which use a sequential Bayesian estimator [7] to infer the source position states from observations using multiple sensors. These papers build on a state-space transition equation describing the global source state trajectory over time and the measurement equation between these states and the measurements. The underlying model of the physical process is modeled in the measurement equation. A decentralized approach aims

at identifying global source states that are common to all decentralized units. Each decentralized unit typically consists of a sensor and a Bayesian estimator associated with the sensor's neighborhood.

A different approach consists of incorporating the partial differential equation (PDE) describing the dynamics of the physical process. In source tracking applications, this implies that the field itself becomes part of the state, which thus is distributed over all space. For instance, the acoustic wave field is described by a hyperbolic PDE for pressure and hence the state vector comprises the spatio-temporal pressure field. This approach is used in (ocean) acoustic models [8,9] and geophysical models [[4], p. 4089 ff], [10-13]. For localization, the model is augmented with a source model providing a relation between global source states, e.g., position, and distributed field states, i.e., pressure.

Our approach belongs to the realm of the second approach. The novel aspects include the formulation of a source model suitable for distributed processing, the design of a distributed particle for the estimation of the posterior distribution of field and source states, and the development of a modified version of the maximum consensus (MC) algorithm [14] for the maximum a-posteriori (MAP) estimation of the source location. For several loosely connected agents, a consensus algorithm

\* Correspondence: florian.xaver@nt.tuwien.ac.at

<sup>1</sup>Institute of Telecommunications (ITC), Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 1040 Vienna, Austria

Full list of author information is available at the end of the article

makes the agents to converge to a group decision based on local information.

### Contributions and outline

We consider inference problems in space  $\mathbf{r}$  and time  $t$  which are modeled via partial differential equations (PDE) of the form [15,16]

$$\mathcal{L}\{p(\mathbf{r}, t)\} = s(\mathbf{r}, t), \quad (1a)$$

$$\mathcal{B}\{p(\mathbf{r}, t)\} = 0, \quad (1b)$$

where  $\mathcal{L}$  denotes the PDE operator;  $\mathcal{B}$  the boundary/initial conditions;  $p(\mathbf{r}, t)$  the quantity of interest, and  $s(\mathbf{r}, t)$  the source term. If the PDE parameters, the source term, and the boundary/initial conditions are known, determining  $p(\mathbf{r}, t)$  is the forward problem. In contrast, inverse problems amount to estimating PDE parameters or states like source locations from measurements of  $p(\mathbf{r}, t)$ .

Discretization of (1) and extending to a stochastic process leads to the state transition equation

$$\mathbf{x}_{k+1} = g_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \quad k \in \mathbb{Z}, \quad (2)$$

where  $k$  denotes discrete time,  $\mathbf{x}_k$  is the state vector (incorporating samples of  $p(\mathbf{r}, t)$ ),  $\mathbf{u}_k$  is the input vector (here corresponding to the sources),  $\mathbf{w}_k$  is a random noise vector (process noise), and  $g_k$  is the state transition mapping. The state transition equation is complemented by the measurement equation

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k). \quad (3)$$

Here,  $\mathbf{y}_k$  is the observation,  $\mathbf{v}_k$  denotes the measurement noise, and the mapping  $h_k$  characterizes the measurement. Taken together, (2) and (3) constitute the state-space model, see Section II and [17].

In the Gaussian case, Bayesian estimation based on the state-space model (2), (3) leads to various kinds of Kalman filters [1,7,18]. Here, the Bayesian estimator builds on the particle filter (PF) framework due to (i) various possible geometries and (ii) the non-linearity of the state-space model. After discussing a centralized PF for source localization and tracking in Section III, we develop a decentralized implementation of the PF by splitting the nodes of the sensor networks (SN) into clusters. The clustered SN architecture entails a corresponding decomposition of the state-space model, and the decentralized PF performs intra-cluster computation and inter-cluster communication on the decomposed state-space model (see Section IV).

The decentralized PF yields local posterior distributions within each cluster. Localization of the acoustic sources amounts to finding the maxima of the global posterior distribution. To this end, we propose a

modified maximum consensus algorithm in Section V. After a summary in Section VI, in Section VII, we describe extensive numerical simulations that illustrate the properties and performance of our source localization method.

## II System model

In this section, we develop a state-space model from the PDE of the spatio-temporal acoustic field using the finite difference method (FDM) [15,19] to obtain a discretization in space and time.

### A Forward model–spatio-temporal field

In the following, we consider an acoustic problem characterized by the hyperbolic PDE (scalar wave equation) [16,19,20]:

$$\frac{1}{c^2} \partial_t^2 p(\mathbf{r}, t) - \nabla^2 p(\mathbf{r}, t) = s(\mathbf{r}, t), \quad \mathbf{r} \in \Omega, \quad (4a)$$

Here,  $p(\mathbf{r}, t)$  denotes pressure,  $\partial_t$  is the partial derivative with respect to time,  $\nabla^2$  the Laplace operator,  $c$  the sound speed,  $s(\mathbf{r}, t)$  is the source, and  $\Omega \subset \mathbb{R}^2$  is the 2-D region of interest. Hereafter, let the initial conditions be

$$p(\mathbf{r}, t) = 0, \quad \mathbf{r} \in \Omega, \quad t = 0, \quad (4b)$$

$$\partial_t p(\mathbf{r}, t) = 0, \quad \mathbf{r} \in \Omega, \quad t = 0. \quad (4c)$$

From three basic boundary conditions

$$\frac{1}{c} \partial_t p(\mathbf{r}, t) - \nabla p(\mathbf{r}, t) \cdot \mathbf{n} = 0, \quad \mathbf{r} \in \partial\Omega_1, \quad (4d)$$

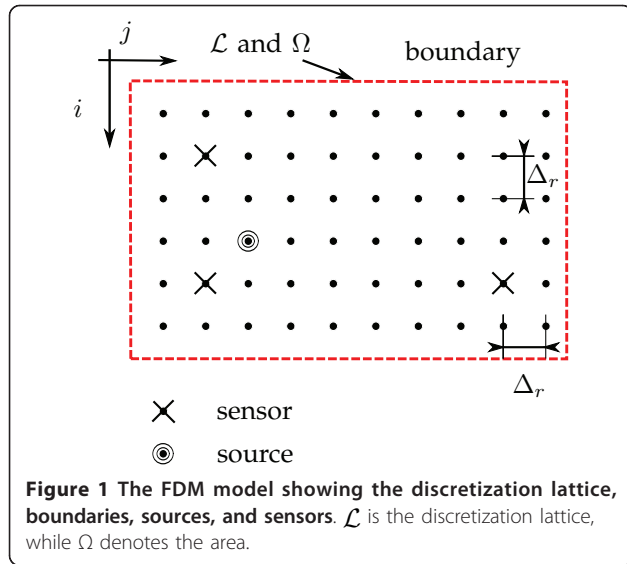
$$p(\mathbf{r}, t) = 0, \quad \mathbf{r} \in \partial\Omega_2, \quad (4e)$$

$$\partial_t p(\mathbf{r}, t) = 0, \quad \mathbf{r} \in \partial\Omega_3, \quad (4f)$$

we use (4d) and (4f) for modeling a hallway.  $\partial\Omega_1$  is the transparent part of the boundary of  $\Omega$  (with normal vector  $\mathbf{n}$ ) modeling an infinite domain for the behind uncovered area. The boundary  $\partial\Omega_2$  (disjoint from  $\partial\Omega_1$ ) models windows, whereas  $\partial\Omega_3$  (disjoint from  $\partial\Omega_1$  and  $\partial\Omega_2$ ) models walls. The choice of these boundary conditions indeed affects the resulting state-space model but does not change the general formulation of the decentralized approach.

### B Finite difference method

To obtain a space-time-discrete model, the differential operators are approximated by finite differences, see Figure 1. We assume a rectangular region in two dimensions (i.e.,  $\mathbf{r} = (x, y)$ ) and use a spatial sampling set given by the finite square lattice  $\mathcal{L} = \{(i\Delta_r, j\Delta_r) : i = 1, \dots, I, j = 1, \dots, J\}$ , where  $\Delta_r$  is the



spatial sampling interval. For simplicity, we assume identical sampling intervals in both coordinates, but using different sampling intervals for each coordinate is straightforward (Different sampling intervals influence the accuracy of the field approximation only but not the principal features of the decentralized estimator). For simplicity, we assume that there are  $R$  sensors whose locations form a subset  $\mathcal{R}$  of the lattice  $\mathcal{L}$ .

For the Laplace operator, we then obtain the discrete approximation

$$\begin{aligned} \nabla^2 p(i\Delta_r, j\Delta_r, t) \approx & \frac{1}{\Delta_r^2} [p((i-1)\Delta_r, j\Delta_r, t) \\ & + p((i+1)\Delta_r, j\Delta_r, t) + p(i\Delta_r, (j-1)\Delta_r, t) \\ & + p(i\Delta_r, (j+1)\Delta_r, t) - 4p(i\Delta_r, j\Delta_r, t)]. \end{aligned}$$

Similarly, for the second-order temporal derivative, we have

$$\begin{aligned} \partial_t^2 p(i\Delta_r, j\Delta_r, k\Delta_t) \approx & \frac{1}{\Delta_t^2} [p(i\Delta_r, j\Delta_r, (k-1)\Delta_t) \\ & - 2p(i\Delta_r, j\Delta_r, k\Delta_t) + p(i\Delta_r, j\Delta_r, (k+1)\Delta_t)]. \end{aligned}$$

Here,  $k$  is the discrete time index, and  $\Delta_t$  is the temporal sampling period. It is upper bounded by  $\Delta_r/c$  to ensure numerical stability. The right choice of  $\Delta_t$  is beyond the scope of our paper, so that we refer our reader to [16].

### C Forward model

We introduce the auxiliary function  $q(x, y, t) = \partial_t p(x, y, t)$  and define the pressure vector  $\mathbf{p}_k = \text{vec}\{\mathbf{P}_k\}$  with  $[\mathbf{P}_k]_{ij} = p(i\Delta_r, j\Delta_r, k\Delta_t)$ . The source vector  $\mathbf{s}_k$  and the pressure derivative vector  $\mathbf{q}_k$  are defined similarly. Applying the FDM to (4) then leads to the following linear system of equations:

$$\begin{bmatrix} \mathbf{q}_{k+1} \\ \mathbf{p}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \mathbf{I} \end{bmatrix}}_{\Phi_{\text{FDM}}} \begin{bmatrix} \mathbf{q}_k \\ \mathbf{p}_k \end{bmatrix} + \Delta_t c^2 \begin{bmatrix} \mathbf{s}_k \\ \mathbf{0} \end{bmatrix}. \quad (5)$$

The diagonal matrix  $\Phi_{11}$  results from the boundary condition (4d). Its diagonal elements are

$$[\Phi_{11}]_{ii} = \begin{cases} 1 - 2\kappa & \text{for nodes on the boundary } \partial\Omega_1, \\ 1, & \text{else} \end{cases}$$

where  $\kappa = c/\Delta_r$ . Also the diagonal matrix

$$[\Phi_{21}]_{ii} = \begin{cases} 1 & \text{for inner nodes and} \\ & \text{nodes on the boundary } \partial\Omega_1, \\ 0 & \text{for nodes on the boundary } \partial\Omega_3 \end{cases}$$

depends on the boundary condition (4f). Similarly, the sparse matrix  $\Phi_{12}$  stems from (4a) and is given by

$$[\Phi_{12}]_{ij} = \begin{cases} -4\kappa^2, & i = j, \\ 2\kappa^2, & |i - j| = 1 \text{ for nodes on } \partial\Omega_1, \\ \kappa^2, & |i - j| = 1 \vee |i - j| = I \text{ for inner nodes,} \\ 0 & \text{else.} \end{cases}$$

### D Source model

We assume that there are  $S$  sources whose positions form a subset  $\mathcal{S}$  of the discretization lattice  $\mathcal{L}$ , i.e.,  $s[i, j, k] = \sum_{l=1}^S s_0[k - k_l] \delta(i - i_l, j - j_l)$ , where  $s_0[k]$  is a known waveform, but the positions  $(i_l, j_l)$  and activation times  $k_l$  are unknown. These unknowns are captured via the integer variables  $n[i, j, k]$  that describe, for a lattice point  $(i, j)$ , the time between the source occurrence and the current time instant  $k$ , i.e., for the  $l$ th source there is  $n[i, j, k] = \max\{k - k_l, 0\}$ . If there is no source at position  $(i, j)$ , then  $n[i, j, k] = 0$ .

Clearly, the source life span satisfies the state transition equation

$$n[i, j, k + 1] = \begin{cases} n[i, j, k] + 1, & (i, j) \in \mathcal{S}_k \\ 0, & \text{else,} \end{cases}$$

where  $\mathcal{S}_k = \{(i, j) | k \geq k_l\}$  is the set of sources active at time  $k$ . Arranging the variables  $n[i, j, k]$  into a vector  $\mathbf{n}_k$  similarly to  $\mathbf{p}_k$ ,  $\mathbf{q}_k$ , and  $\mathbf{s}_k$ , we obtain

$$\mathbf{n}_{k+1} = \mathbf{n}_k + \delta_{\mathcal{S}_k}, \quad (6)$$

where the elements of  $\delta_{\mathcal{S}_k}$  are zero or one depending on whether a source is active at the corresponding position and at time instant  $k$ , i.e.,

$$[\delta_{\mathcal{S}_k}]_{i+(j-1)I} = \begin{cases} 1, & (i, j) \in \mathcal{S}_k, \\ 0, & \text{else.} \end{cases} \quad (7)$$

Note that the state vector  $\mathbf{n}_k$  has at most  $S$  non-zero elements. Using the convention  $s_0[0] = 0$ , the source vector  $\mathbf{s}_k$  in (5) is rewritten as

$$\mathbf{s}_k = s_0 [\mathbf{n}_k], \quad (8)$$

thereby linking the state Equation 6 and the forward model (5).

### E Noise model

So far, no process noise has been considered and specified. Since the source function depends on time and space, these are the only quantities that suffer from noise and are modeled in the following: The temporal noise models the perturbation of a source's life span by an additional term in (6), while this is not possible for the spatial perturbation. This is due to the fact that the position of sources is coded into the sub-vector  $\mathbf{n}_k$  by placing its elements. From a practical perspective, this is done by a time-dependent matrix  $\mathbf{D}_k$  which displaces the elements of a vector to other positions (jitter) according to the mapping between grid and sub-vector  $\mathbf{n}_k$ .

Equation 6 becomes

$$\mathbf{n}_{k+1} = \mathbf{D}_k(\mathbf{n}_k + \delta_{S_k} + \delta_{S_k} \odot \mathbf{n}'). \quad (9)$$

Here,  $\mathbf{n}'$  is a random integer perturbation,  $\odot$  is the Hadamard (element-wise) product, and the  $l$ th column of the displacement matrix  $\mathbf{D}_k$  is given by  $\mathbf{e}_{l+d(l)}$ , with the canonical column unit vector

$$[\mathbf{e}_l]_n = \begin{cases} 1, & l = n, \\ 0, & \text{else,} \end{cases}$$

and a random integer jitter  $d(l)$  whose probability mass is concentrated about zero.

Because of linearity, (9) is rewritten as

$$\mathbf{n}_{k+1} = \mathbf{D}_k \mathbf{n}_k + \mathbf{D}_k \delta_{S_k} + \mathbf{D}_k \text{diag}\{\delta_{S_k}\} \mathbf{n}'. \quad (10)$$

### F Augmented state-space model

We next combine the state-space model (5) with (8) and (10) to obtain an augmented state-space model for the extended state vector

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{q}_k \\ \mathbf{p}_k \\ \mathbf{n}_k \end{bmatrix}.$$

This gives the state transition equation

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \mathbf{G}_k \mathbf{n}'_k \quad (11)$$

with

$$\Phi_k = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0} \\ \Delta_t \mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_k \end{bmatrix}, \quad \Gamma_k = \begin{bmatrix} \Delta_t c^2 \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_k \end{bmatrix}, \quad (12)$$

and

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{D}_k \text{diag}\{\delta_{S_k}\} \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} s_0 [\mathbf{n}_k] \\ \mathbf{0} \\ \delta_{S_k} \end{bmatrix}. \quad (13)$$

Note that non-linearity is inherent in (11).

To complete the state-space model, the measurement equation is introduced. Since the actual observations are given by noisy samples of the pressure field at the sensor positions  $(i'_l, j'_l) \in \mathcal{R}$ , the measurement equation is

$$\mathbf{y}_k = \tilde{\mathbf{C}} \mathbf{x}_k + \mathbf{v}_k = \mathbf{C} \mathbf{p}_k + \mathbf{v}_k, \quad (14)$$

where  $\mathbf{v}_k$  denotes measurement noise and

$$\tilde{\mathbf{C}} = [\mathbf{0} \ \mathbf{C} \ \mathbf{0}], \quad \mathbf{C} = \begin{bmatrix} \mathbf{e}_{i'_1+(j'_1-1)l}^T \\ \vdots \\ \mathbf{e}_{i'_r+(j'_r-1)l}^T \end{bmatrix},$$

with  $\mathbf{e}_l$  denoting the  $l$ th unit vector.

### III Bayesian estimation

Our aim is to perform sequential Bayesian estimation of the state vector  $\mathbf{n}_k$  that characterizes the source positions and activation times.  $\mathbf{n}_k$  is one of the state vectors  $\mathbf{x}_k$  in (11). The data  $\mathbf{y}_k$  is specified in (14). A PF approach [7], i.e., a Monte Carlo approach based on importance sampling, is pursued. This approach exploits that our state-space model (11)-(14) is a hidden Markov model (cf. Figure 2), where (11) implies a state transition distribution  $f(\mathbf{x}_k | \mathbf{x}_{k-1})$  and (14) leads to a measurement distribution (likelihood function)  $f(\mathbf{y}_k | \mathbf{x}_k)$ , which both are assumed known in the following.

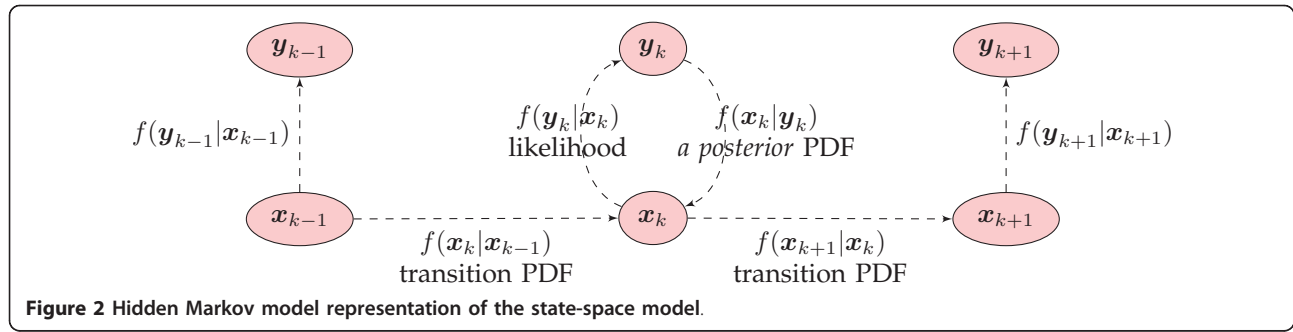
#### A Particle filter

To perform Bayesian estimation (e.g., MAP or MMSE) of (part of) the state vector  $\mathbf{x}_k$  given the past observations  $\mathbf{y}_{1:k} = [\mathbf{y}_1^T \dots \mathbf{y}_k^T]^T$ , the posterior distribution  $f(\mathbf{x}_k | \mathbf{y}_{1:k})$  is computed sequentially.

Using the Bayesian theorem and the fact that  $\mathbf{y}_{k+1}$  and  $\mathbf{y}_{1:k}$  are statistically independent (due to the Markov chain assumption) given  $\mathbf{x}_{k+1}$ , we have

$$\begin{aligned} f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k+1}) &= f(\mathbf{x}_{k+1} | \mathbf{y}_{k+1}, \mathbf{y}_{1:k}) \\ &= \frac{f(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}, \mathbf{y}_{1:k}) f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k})}{f(\mathbf{y}_{k+1} | \mathbf{y}_{1:k})} \\ &= \frac{f(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}) f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k})}{\int f(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}) f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k}) d\mathbf{x}_{k+1}}, \end{aligned} \quad (15)$$

which is known as the update step. While the measurement PDF  $f(\mathbf{y}_{k+1} | \mathbf{x}_{k+1})$  in (15) is known,  $f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k})$  needs to be computed via the so-called prediction step,



$$f(\mathbf{x}_{k+1} | \mathbf{y}_{1:k}) = \int f(\mathbf{x}_{k+1} | \mathbf{x}_k) f(\mathbf{x}_k | \mathbf{y}_{1:k}) d\mathbf{x}_k. \quad (16)$$

Here, the transition PDF  $f(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is known and  $f(\mathbf{x}_k | \mathbf{y}_{1:k})$  has been computed in the previous time step.

Since the integral in (16) typically is infeasible, it is usually approximated using a Monte Carlo technique known as importance sampling. The approximate sequential computation of the posterior distribution  $f(\mathbf{x}_k | \mathbf{y}_{1:k})$  based on importance sampling using the transition PDF  $f(\mathbf{x}_k | \mathbf{x}_{k-1})$  as importance (or, proposal) distribution  $q(\mathbf{x}_k)$  leads to the particle filter. Here, the desired PDFs are approximated in terms of particles, i.e., samples  $\mathbf{x}_k^{[l]}$  and associated weights  $\omega_k^{[l]}$ , hence

$$f(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{l=1}^L \omega_k^{[l]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[l]}), \quad (17)$$

where  $L$  is the number of particles. The new samples for the subsequent time instant are generated using the proposal distribution

$$q(\mathbf{x}_{k+1}) = f(\mathbf{x}_{k+1} | \mathbf{x}_k = \mathbf{x}_k^{[l]}),$$

where for the generation of each new particle  $\mathbf{x}_{k+1}^{[l]}$  the previous particle  $\mathbf{x}_k^{[l]}$  is chosen randomly with probability  $\omega_k^{[l]}$ . Sampling from  $q(\mathbf{x}_{k+1})$  can be achieved by generating a noise realization  $\mathbf{w}_k^{[l]}$  and invoking the state transition Equation 11, i.e.,

$$\mathbf{x}_{k+1}^{[l]} = \Phi_k^{[l]} \mathbf{x}_k^{[l]} + \Gamma_k^{[l]} \mathbf{u}_k^{[l]} + \mathbf{G}_k^{[l]} \mathbf{n}_k^{[l]}. \quad (18)$$

$\mathbf{u}_k^{[l]}$  can be computed from the particle  $\mathbf{x}_k^{[l]}$  according to (13). The dependency of the matrices on  $k$  issues from spatial noise.

The unnormalized weight for each new particle is

$$\tilde{\omega}_{k+1}^{[l]} = \omega_k^{[l]} f(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}^{[l]}) = \omega_k^{[l]} f_v(\mathbf{y}_{k+1} - \tilde{\mathbf{C}} \mathbf{x}_{k+1}^{[l]}), \quad (19)$$

where  $f_v(\mathbf{v}_k)$  is the distribution of the measurement noise and we used the measurement Equation 14. For i.

i.d. Gaussian measurement noise with variance  $\sigma_v^2$

$$\tilde{\omega}_{k+1}^{[l]} = \omega_k^{[l]} \exp\left(-\frac{1}{2\sigma_v^2} \|\mathbf{y}_{k+1} - \tilde{\mathbf{C}} \mathbf{x}_{k+1}^{[l]}\|^2\right).$$

Once all unnormalized weights have been obtained, the actual weights are computed via the normalization  $\omega_{k+1}^{[l]} = \tilde{\omega}_{k+1}^{[l]} / \sum_{l=1}^M \tilde{\omega}_{k+1}^{[l]}$ . Particle filters suffer from a general problem termed sample degeneracy, i.e., after some time only few particles have non-negligible weights. This problem is circumvented using resampling [21]. With sampling importance resampling (SIR), new samples are drawn from the distribution  $\sum_{l=1}^L \omega_k^{[l]} \delta(\mathbf{x}_k - \mathbf{x}_k^{[l]})$  and all weights are identical, i.e.,  $\omega_k^{[l]} = 1/L$ .

To obtain initial particles  $\mathbf{x}_0^{[l]}$ , samples of the state vector are needed.  $S$  random realizations of source positions and activation times are generated according to the prior distributions. Then, we apply the noise-free version of the state-space model (11)  $k_{\text{start}}$  times, i.e.,

$$\mathbf{x}_0^{[l]} = \Phi^{k_{\text{start}}} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{n}_0^{[l]} \end{bmatrix} + \sum_{\ell=0}^{k_{\text{start}}-1} \Phi^{k_{\text{start}}-1-\ell} \Gamma \mathbf{u}_\ell^{[l]}, \quad (20)$$

where  $\mathbf{n}_0^{[l]}$  and  $\mathbf{u}_\ell^{[l]}$  are determined by the realizations of the source parameters (cf. (13) and Section II-D). The random variable  $k_{\text{start}}$  denotes the time duration between source occurrence and activation of the estimator.

## B Source localization

Using (17), the posterior PDF of  $\mathbf{n}_k$  (i.e., the last  $IJ$  elements of  $\mathbf{x}_k$ ) is approximated as

$$f(\mathbf{n}_k | \mathbf{y}_{1:k}) \approx \sum_{l=1}^L \omega_k^{[l]} \delta(\mathbf{n}_k - \mathbf{n}_k^{[l]}). \quad (21)$$

(Note that  $\mathbf{n}_k$  contains all information about position and activation time of the sources.)



The probability  $P\{\mathcal{S}_k|\gamma_{1:k}\}$  for sources to be active at the coordinate set  $\mathcal{S}_k$  at time  $k$  is obtained via marginalization:

$$P\{\mathcal{S}_k|\gamma_{1:k}\} = \sum_{l \in \Lambda_k} \omega_k^{[l]}, \quad \Lambda_k = \left\{ l : Q(\mathbf{n}_k^{[l]}) = \delta_{\mathcal{S}_k} \right\}. \quad (22)$$

Here, the function  $Q : \mathbb{R}^J \rightarrow \{0, 1\}$  sets all entries of  $\mathbf{n}_k^{[l]}$  to 1 which are unequal to 0. In the case of one source and a SIR PF with  $w_k^{[l]} = 1/L$ , the probability for a source at position  $(i, j)$  at time  $k$  is approximately obtained as

$$P_s(i, j, k) = P\{\text{source at } (i, j, k) | \gamma_{1:k}\} = \frac{L_{i,j,k}}{L}, \quad (23)$$

where  $L_{i,j,k}$  is the number of particles for which  $[\mathbf{n}_k^{[l]}]_{i+(j-1)L} > 0$ .

#### IV Decentralized scheme

The particle filter developed in the previous section is centralized in nature since it requires all pressure measurements and the observation modalities described by the globally assembled likelihood function and operates on the full state vector  $\mathbf{x}_k$  in a fusion center. Additionally, the computed estimates are inherently unknown on the individual sensor nodes. In a SN context, such constraints are undesirable since they imply a large communication overhead to collect the measured data, a high computational effort due to the high-dimensional state vector, a feedback to the sensor nodes to spread the estimates, and a central knowledge of measurement noise. Therefore, a decentralized scheme that distributes the data collection and computational costs among several clusters of sensor nodes is developed. This is achieved by splitting the state-space model (11), (14) into lower-dimensional sub-models (each corresponding to a cluster), cf. with [22,23]. Due to the sparsity of the state-space matrices  $\Phi$  and  $\Gamma$ , these sub-models are only loosely coupled, thus a decentralized PF that requires

little communication between the clusters can be developed.

#### A SN clusters and partitioned state-space model

We start with partitioning the region of interest  $\Omega$  into  $M$  disjoint subregions  $\Omega^{(m)}$ . The sampling lattice corresponding to each subregion is given by  $\mathcal{L}^{(m)} = \mathcal{L} \cap \Omega^{(m)}$  with its boundary nodes  $\partial\mathcal{L}^{(m)}$ , see Figure 3. The sensors within each subregion form clusters, denoted by  $\mathcal{R}^{(m)} = \mathcal{R} \cap \Omega^{(m)} \subset \mathcal{L}^{(m)}$ . To each subregion, we associate a subset of elements of the state vector  $\mathbf{x}_k$  given by

$$\mathbf{x}_k^{(m)} = \begin{bmatrix} \mathbf{q}_k^{(m)} \\ \mathbf{p}_k^{(m)} \\ \mathbf{n}_k^{(m)} \end{bmatrix} \quad (24)$$

where

$$\mathbf{p}_k^{(m)} = [p(i\Delta_r, j\Delta_r, k\Delta_t)]_{(i,j) \in \mathcal{L}^{(m)}}$$

and the superscript  $^{(m)}$  refers to region  $m$ .

Except for  $\Phi_{12}$ , all of the blocks in the state-space matrices  $\Phi_k$  and  $\Gamma_k$  are diagonal or zero (cf. (12)). Thus, there is no coupling between the sub-vectors  $\mathbf{p}_k^{(m)}$  from different subregions and similarly for the sub-vector  $\mathbf{q}_k^{(m)}$ . Coupling between state vectors from different regions, induced by the non-diagonal structure of  $\Phi_{12}$ , is between the sub-vectors  $\mathbf{q}_k^{(m)}$  in one subregion and the sub-vectors  $\mathbf{p}_k^{(m)}$  in the adjacent subregions (in fact, this coupling is limited to samples at the boundaries of the subregions). The same applies for the sub-vectors  $\mathbf{n}_k^{(m)}$  due to the spatial noise. This gives

$$\begin{aligned} \mathbf{x}_{k+1}^{(m)} &= \Phi_k^{(m)} \mathbf{x}_k^{(m)} + \xi_k^{(m)} \\ &\quad + \Gamma_k^{(m)} \mathbf{u}_k^{(m)} + \gamma_k^{(m)} \\ &\quad + \mathbf{G}_k^{(m)} \mathbf{n}_k^{(m)}, \\ \gamma_k^{(m)} &= \mathbf{C}^{(m)} \mathbf{p}_k^{(m)} + \mathbf{v}_k^{(m)}. \end{aligned} \quad (25)$$

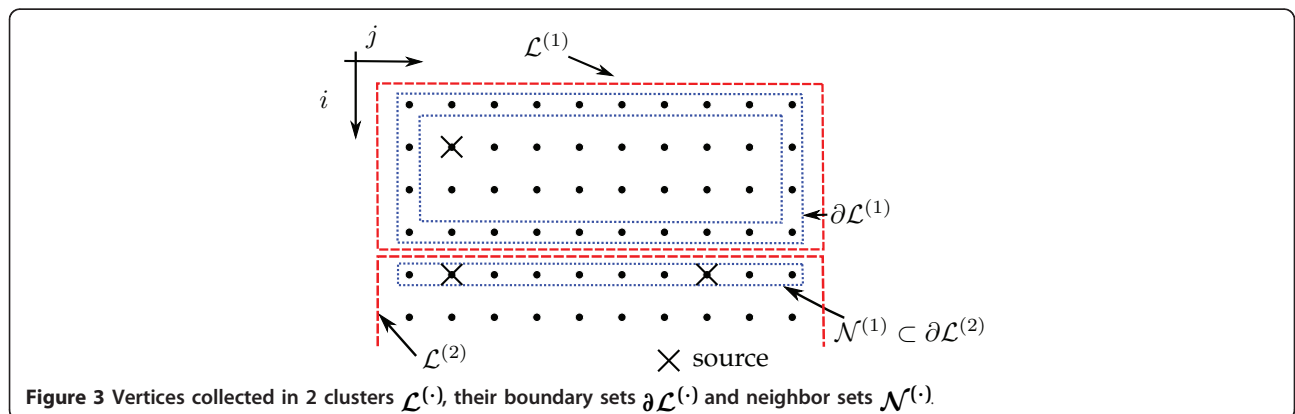


Figure 3 Vertices collected in 2 clusters  $\mathcal{L}^{(\cdot)}$ , their boundary sets  $\partial\mathcal{L}^{(\cdot)}$  and neighbor sets  $\mathcal{N}^{(\cdot)}$ .

This coupling Equation 25 is only possible for the time-independent part of these matrices. However, for uncorrelated noise between clusters, the time-dependent part, i.e.,  $D_k$ , is calculated separately according to Section II-E on every cluster at each time step, see below.

The coupling terms between neighboring subregions are given by

$$\xi_k^{(m)} = \sum_{m' \in \mathcal{N}^{(m)}} T_k^{(m,m')} x_k^{(m')}, \quad (26)$$

with

$$T_k^{(m,m')} = \begin{bmatrix} \mathbf{0} & \Phi_{12}^{(m,m')} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_k^{(m,m')} \end{bmatrix}, \quad (27)$$

and, analogously,

$$\gamma_k^{(m)} = \sum_{m' \in \mathcal{N}^{(m)}} R_k^{(m,m')} u_k^{(m')}, \quad (28)$$

with

$$R_k^{(m,m')} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_k^{(m,m')} \end{bmatrix}. \quad (29)$$

Here,  $\mathcal{N}^{(m)}$  is the set of subregions adjacent to  $\Omega^{(m)}$ , and  $\Phi_{12}^{(m,m')}$  is obtained from  $\Phi_{12}$  by extracting the rows and columns corresponding to  $\mathcal{L}^{(m)}$  and  $\mathcal{L}^{(m')}$ . The off-diagonals of  $\Phi_{12}$  are extremely sparsely populated; in fact, (26) contains only few non-zero terms corresponding to adjacent pressure samples and the change of sources from one to another cluster.  $D_k^{(m,m')}$  is generated from every cluster  $m'$  such that the composition of all submatrices  $D_k^{(m)}$  and  $D_k^{(m,m')}$  equals  $D_k$ . From a practical perspective, elements of  $D_k^{(m)}$  are calculated separately on every cluster by means of spatial noise with additional triggering of a message to neighbor clusters whenever a source hop (migration) from one cluster to another is detected (this takes over the purpose of  $D_k^{(m,m')}$  and supersedes (28)). Furthermore, the coupling term  $\xi_k^{(m)}$  means that pressure samples at subregion boundaries are exchanged between neighboring clusters in order to compute the finite differences.

Boundary conditions do not play a role in the decomposition step as long as (i) they do not depend on adjacent neighbors and (ii) their numerical solution fits into (5). In the first situation, an additional term  $\Phi_{11}^{(m,m')}$  or  $\Phi_{21}^{(m,m')}$  arises in matrix  $T_k^{(m,m')}$ .

## B Decentralized particle filter

For the decentralized PF, we need to distribute the sampling (particle generation) step and the weight computation step. Based on the local particles and weights, each cluster can then compute posterior source probabilities in a similar manner as in Section III-B.

1) *Particle Generation*: Sub-particles  $x_k^{[l,m]}$  within cluster  $\mathcal{R}^{(m)}$  are generated according to (25), cf. also (18),

$$\begin{aligned} x_{k+1}^{[l,m]} &= \Phi_k^{(m)} x_k^{[l,m]} + \xi_k^{[l,m]} \\ &\quad + \Gamma_k^{(m)} u_k^{[l,m]} + \gamma_k^{[l,m]} \\ &\quad + G_k^{[l,m]} n_k^{[l,m]}. \end{aligned} \quad (30)$$

Here,  $x_k^{[l,m]}$  is a randomly chosen previous particle and  $n_k^{[l,m]}$  is a (local) noise vector realization. Furthermore,  $\xi_k^{[l,m]} = \sum_{m' \in \mathcal{N}^{(m)}} T_k^{(m,m')} x_k^{[l,m']}$  and  $\gamma_k^{[l,m]} = \sum_{m' \in \mathcal{N}^{(m)}} R_k^{(m,m')} u_k^{[l,m]}$ , respectively. In order to compute the latter, only elements of  $x_k^{[l,m]}$  that correspond to pressure samples from the boundaries of adjacent subregions are exchanged, and in the event of source hopping from one to another cluster, a message is sent.

2) *Weights*: Assuming independent measurement noise in the individual subregions, i.e.,  $f_v(v_k) = \prod_{m=1}^M f_{v^{(m)}}(v_k^{(m)})$ , the weight update (19) is computed in each cluster as

$$\tilde{\omega}_{k+1}^{[l]} = \omega_k^{[l]} \prod_{m=1}^M \tilde{\omega}_k^{[l,m]}, \quad (31)$$

where the partial weights

$$\tilde{\omega}_k^{[l,m]} = f_{v^{(m)}}(y_{k+1}^{(m)} - \tilde{C}^{(m)} x_{k+1}^{[l,m]})$$

are computed within each cluster and then are shared among all clusters to obtain the final unnormalized weight [24] and [25] are treating the issue of computation of the global factorizable likelihood by means of distributed protocols. If these take longer than the time span between two estimator iterations, the particle filter converts to a particle predictor.

3) *(Re)sampling*: A remaining problem with the decentralized PF is that the sampling (particle generation) step (30) requires that the clusters pick local particles  $x_k^{[l,m]}$   $m = 1, \dots, M$ , that correspond to the same global particle  $x_k^{[l]}$ . This choice is made at random according to the weights  $\omega_k^{[l]}$ . The same problem occurs for the resampling procedure. Since a central random number generator whose output is distributed to each cluster incurs a large communication overhead, we propose to use identical pseudo-number generators in all clusters

and initialize those with the same seed, thereby ensuring that all clusters perform the same (re)sampling (cf. with [24] and [26]).

### V Decentralized source localization

The PF yields the posterior PDF of the sources' position and life span. To obtain the current MAP position estimates

$$(\hat{i}_k, \hat{j}_k) = \arg \max_{(i,j) \in \mathcal{L}} P_s(i, j, k), \quad (32)$$

the maximum and the maximizing state of the posterior PDF  $P_s(i, j, k)$  in (23) must be found. In the decentralized scheme, each cluster disposes only of the local posterior PDF for the state sub-vector  $\mathbf{x}_k^{(m)}$ . To find the global maximizing state, each cluster determines the local maximizing state and afterward the clusters use a distributed consensus protocol to determine the global maximum. For simplicity, this procedure is here developed for one source.

For the centralized PF, the posterior probability for a source to be active at time  $k$  at position  $(i, j)$  is given by (23). In the decentralized case, each cluster determines a similar probability according to

$$P_s^{(m)}(i, j, k) = \begin{cases} \frac{L_{i,j,k}^{(m)}}{L}, & (i, j) \in \mathcal{L}^{(m)}, \\ 0, & \text{else,} \end{cases}$$

where  $L_{i,j,k}^{(m)}$  denotes the number of particles  $\mathbf{x}_k^{[l,m]}$  for which  $[\mathbf{n}_k^{[l,m]}]_{i+(j-1)I} > 0$ . Since the probabilities  $P_s^{(m)}(i, j, k)$  have disjoint support, the maximization underlying the MAP estimates (32) is

$$P_{k,\max} = \max_{(i,j) \in \mathcal{L}} P_s(i, j, k) = \max_m P_{k,\max}^{(m)}$$

with

$$P_{k,\max}^{(m)} = \max_{(i,j) \in \mathcal{L}^{(m)}} P_s^{(m)}(i, j, k). \quad (33)$$

While the local maxima with regard to  $\mathcal{L}^{(m)}$  can be determined within each cluster, the global maximization with regard to  $m$  requires communication between the clusters. Since sharing the local maxima among all clusters via broadcast transmissions requires a large coordinated transmission, we compute the global maximum via the maximum consensus (MC) algorithm [14]. For the MC algorithm, we assume that only neighboring clusters communicate with each other. Thus, each cluster sends to the adjacent clusters a message which contains the local maximum and the position for which the local maximum is achieved. In the subsequent steps, each cluster compares the incoming "maximum"

messages with their current estimate of the global position and retain the most likely and its associated position. In the next iteration, this message will be sent to the neighboring clusters.

Denote the current estimate of the maximum  $P_{k,\max}$  for cluster  $m$  by  $\hat{P}_{k,\max}^{(m)}$  and let  $(\hat{i}_k^{(m)}, \hat{j}_k^{(m)})$  be the associated position estimate (initially,  $\hat{P}_{k,\max}^{(m)} = P_{k,\max}^{(m)}$ ). In our MC algorithm, termed *argumentum-maximi* consensus (AMC), at time instant  $k$ , each cluster performs the following steps:

1) Send a message containing the estimates  $\hat{P}_{k,\max}^{(m)}$  and  $(\hat{i}_k^{(m)}, \hat{j}_k^{(m)})$  to the neighbor clusters  $\mathcal{N}^{(m)}$ .

2) Receive corresponding messages from the neighbor cluster, if a neighbor  $m' \in \mathcal{N}^{(m)}$  remains silent, then  $\hat{P}_{k,\max}^{(m')} = \hat{P}_{k-1,\max}^{(m')}$ .

3) Update the maximum probability and position as

$$\hat{P}_{k+1,\max}^{(m)} = \hat{P}_{k,\max}^{(m_0)} \quad (\hat{i}_{k+1}^{(m)}, \hat{j}_{k+1}^{(m)}) = (\hat{i}_k^{(m_0)}, \hat{j}_k^{(m_0)}),$$

with  $m_0 = \arg \max_{m' \in \{m\} \cup \mathcal{N}^{(m)}} P_{k,\max}^{(m')}$ .

4) If  $\hat{P}_{k+1,\max}^{(m)} \neq \hat{P}_{k,\max}^{(m)}$  to go 1), otherwise go to 2).

When the maximum is fixed, all clusters converge to the true maximum after some iterations (depending on the diameter of the cluster communication graph). Here, the position of the maximum moves as the distributed PF evolves and the AMC will then allow the clusters to jointly track the maximum.

### VI Algorithm summary

#### A Dimensions and trade-offs

Since we are estimating the 2-D position and activation time for each of the  $S$  sources, the number of unknowns equals  $3S$ . This is relevant for the choice of the number of particles, cf. [4]. For the calculation of the forward model (state transition), however, the dimension of the state vector  $\mathbf{x}_k$  is relevant which equals  $3IJ$ . In the decentralized case, the computational complexity of the forward model is distributed across all clusters.

We now face the behavior of a high number of clusters. Generally, the volume of a polytope (cluster)  $\mathcal{L}^{(m)}$  with edge lengths  $e_i(m)$  in a  $d$ -dimensional lattice  $\mathcal{L} \subset \mathbb{Z}^d$  is given by  $|\mathcal{L}^{(m)}| = \prod_{i=1}^d e_i^{(m)}$  while its  $(d-1)$ -dimensional surface equals  $|\partial \mathcal{L}^{(m)}| = 2 \sum_{j=1}^d \partial_j \prod_{i=1}^d e_i^{(m)}$ .

Generally, the dimension per cluster of the equation system to be calculated is  $3|\mathcal{L}^{(m)}|$  which, in comparison, equals in the centralized case  $3|\mathcal{L}|$ .

In our 2-D problem, let the lattice  $\mathcal{L}$  be partitioned into  $M = M_i M_j$  clusters of same size,  $M_i$  clusters in  $i$ -direction and  $M_j$  clusters in  $j$ -direction. Then,  $e_1 = I/M_i$  and  $e_2 = J/M_j$ . Furthermore, the volume



$|\mathcal{L}^{(m)}| = IJ/M_i M_j$ . When  $M \rightarrow \infty$ , then the dimension of the equation system, which specifies the amount of computation, becomes in  $\mathcal{O}(1/M)$ [27]. Thus the computational effort per cluster decreases when the number of clusters increases. On the other hand, an increasing number of clusters leads to a larger number of boundaries and hence to a larger communication overhead (i. e., message exchange between adjacent clusters).

**Algorithm 1:** Global initialization

```

generate priors  $\mathcal{X}_0$ ; // Equation (20)
decompose  $\mathcal{X}_0$  to  $\{\mathcal{X}_0^{(m)}\}$ ; // Equation (24)
choose seed  $s_0$  (Section IV-B3);
for  $m = 1$  to  $M$  parallel do

```

DD-SIR-PF( $\mathcal{X}_0^{(m)}, s_0$ ) of cluster  $m$ ;

**Algorithm 2:** DD-SIR-PF(): Decentralized distributed SIR particle filter of cluster  $m$

**input :**  $\mathcal{X}_0^{(m)}, s_0$

$k \leftarrow 1$ ;

wait while no signal sensed and no wake-up call;

send wake-up call to other clusters;

**while** *estimating* **do**

**observe:**  $\mathcal{Y}_k^{(m)}$

$\{\bar{\mathcal{W}}_k^{(m)}, \mathcal{X}_k^{(m)}\} \leftarrow \text{SI}(\mathcal{X}_{k-1}^{(m)}, \mathcal{Y}_k^{(m)})$ ;

transmit  $\{\bar{\mathcal{W}}_k^{(m)}, \mathcal{P}_k^{(m)}, \hat{p}_{k-1, \max}^{(m)}, \hat{\mathcal{S}}_{k-1}^{(m)}\}$ ;

wait until reception from other clusters;

$\{\mathcal{W}_k, \mathcal{X}_k^{(m)}\} \leftarrow$

modify  $(\bar{\mathcal{W}}_k^1, \dots, \bar{\mathcal{W}}_k^M, \mathcal{X}_k^{(m)}, \mathcal{P}_k^{(N^{(m)})})$

calculate  $\{\hat{p}_{k, \max}^{(m)}, \hat{\mathcal{S}}_k^{(m)}\}$ ; //

Equation (33)

$\mathcal{X}_k^{(m)} \leftarrow \text{resampling}(\mathcal{W}_k, \mathcal{X}_k^{(m)}, s_0)$ ;

$\mathcal{W}_k^{(m)} \leftarrow \{1/L\}_{\ell=1}^L$ ;

$k \leftarrow k + 1$ ;

## B Communication between clusters

The variables that are broadcast by cluster  $m$  are summarized by the set

$$\left\{ \bar{\mathcal{W}}_k^{(m)}, \mathcal{P}_k^{(m)}, \mu_k^{(i,m)}, \hat{p}_{k, \max}^{(m)}, \hat{\mathcal{S}}_k^{(m)} \right\}. \quad (34)$$

The first subset  $\bar{\mathcal{W}}_k^{(m)} = \{\bar{w}_k^{[1,m]}, \dots, \bar{w}_k^{[L,m]}\}$  collects the local PF weights, while  $\mu_k^{(i,m)}$  collects all pressure sub-state particles on the

boundary. The third,  $\mu_k^{(i,m)}$ , signifies a message about sources which migrate across boundaries from one cluster to another. Every message includes the new location and the current time duration since the occurrence of the sources. The last two terms stem from the AMC algorithm where  $\hat{\mathcal{S}}_k^{(m)} = (\hat{i}_k^{(m)}, \hat{j}_k^{(m)})$ .

Note that the cardinality of (34) which is a measure of the amount of transmission per cluster is given by the sum

$$L \quad (\bar{\mathcal{W}}_k^{(m)} \text{ to all clusters})$$

$$+ |\partial \mathcal{L}^{(m)}| L \quad (\mathcal{P}_k^{(m)} \text{ to adjacent clusters})$$

$$+ 2M \quad (\hat{p}_{k, \max}^{(m)} \text{ and } \hat{\mathcal{S}}_k^{(m)} \text{ to adjacent clusters})$$

Here, the  $\mu_k^{(i,m)}$  messages are disregarded. The amount of transmission in the decentralized case to adjacent neighbors for  $M_i \rightarrow \infty$  and  $M_j \rightarrow \infty$  is in  $\mathcal{O}(1/M_i)$  and  $\mathcal{O}(1/M_j)$ , respectively. The transmission of weights is in  $\mathcal{O}(M)$  for  $M \rightarrow \infty$ , while the overall communication load is in  $\mathcal{O}(M^2)$ .

Note that there is no approximation compared to the centralized method and thus neither source coding nor approximations reducing the weight communication have been considered. For the communication of the weights, either the graph needs to be fully connected or the clusters need to act as relay. A summary is drawn in Table 1.

## C Algorithm

The algorithm of the decentralized and distributed SIR PF together with the AMC is drawn in Algorithms 1-4. Compare it with that one in [28] and note that the for-loop can be parallelized.

The joint setup of the computational nodes is shown in Algorithm 1 which consists of the calculation of the priors and the synchronization of the pseudo-random generator. Subsequently, each individual PF is launched (Algorithm 2). Two important sub-routines are plotted in their own tableaux:

- Algorithm 3 calculates particles and sends messages when a source jumps over to another cluster.

**Table 1 Necessary message exchange**

	Neighbor	Not neighbor
$p_k$	Boundary elements	
$n_k$	Source migration*	
$w_k^{[L,m]}$	All	(All if not relaying/forwarding)
$\hat{\mathcal{S}}_k^{(m)}$	All	
$p_{k, \max}^{(m)}$	All	

\*Source migration denotes the information that a source changes from one cluster to another.

- Algorithm 4 adds states from the neighbor clusters according to (25) and calculates the overall weight (31).

**Algorithm 3:**  $SI(\cdot)$ : sample importance part

**Input:**  $\mathcal{X}_{k-1}^{(m)}, \gamma_k^{(m)}$

**output:**  $\{\bar{\mathcal{W}}_k^{(m)}, \mathcal{X}_k^{(m)}\}$

**for**  $i = 1$  **to**  $L$  **do**

Draw  $\mathbf{x}_k^{[l,m]} \sim f(\mathbf{x}_k^{(m)} | \mathbf{x}_{k-1}^{(m)})$ ;

**if** source(s) cross(es) boundary **then**

send message to adjacent cluster

$\bar{\omega}_k^{[l,m]} \leftarrow f(\gamma_k^{(m)} | \mathbf{x}_k^{[l,m]})$ ;

**Algorithm 4:**  $modify(\cdot)$ : contribution of the neighbors.  $T^{(m)}$  is a mapping from neighbors' pressure sub-states to the own sub-states with  $T^{(m)}\mathcal{P}_k^{(\mathcal{N}^{(m)})}$  assembles to  $\{\xi_k^{[l,m]}\}_{l=1}^L$ .

**input:**  $\{\bar{\mathcal{W}}_k^1, \dots, \bar{\mathcal{W}}_k^M, \mathcal{X}_k^{(m)}, \mathcal{P}_k^{(\mathcal{N}^{(m)})}\}$

**output:**  $\{\mathcal{W}_k, \mathcal{X}_k^{(m)}\}$

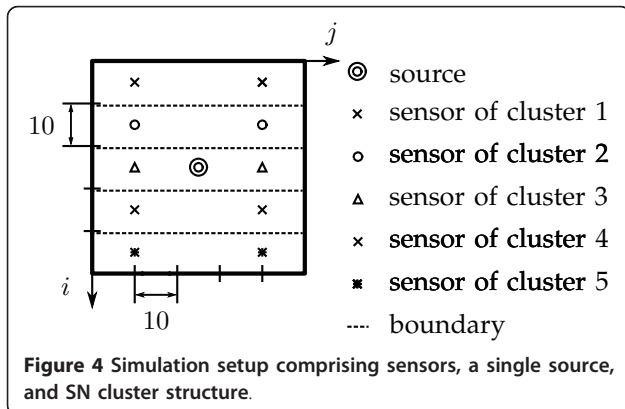
$\mathcal{X}_k^{(m)} \leftarrow \mathcal{X}_k^{(m)} + T^{(m)}\mathcal{P}_k^{(\mathcal{N}^{(m)})}$ ; // Equation (27)

$\hat{\mathcal{W}}_k \leftarrow \bar{\mathcal{W}}_k^1 \dots \bar{\mathcal{W}}_k^M$ ; // Equation (31)

normalize  $\hat{\mathcal{W}}_k$ ;

## VII Simulations

In this section, we present simulations illustrating the performance of the proposed Algorithms 1-4. The configuration used in the simulations is shown in Figure 4 with parameters in Table 2 ( $\mathcal{N}\{\mu, \sigma^2\}$  denotes the Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ ). In particular, we used  $M = 5$  subregions  $\Omega^{(m)}$  corresponding to 5 clusters each with 2 sensors. We considered a single source located in  $\Omega^{(3)}$  at the lattice point  $(i_0, j_0) = (25, 25)$ ,



**Table 2 Parameters for simulated hallway**

FDM	$\Delta_r$	371 ns
	$\Delta_r$	12.24 cm
	$l \times J$	$50 \times 50$
Speed	$c$	340 m/s
Noise	$\mathbf{w}$	i.i.d. $\mathcal{N}\{0, 100pPa/s^2\}$
	$\mathbf{v}$	i.i.d. $\mathcal{N}\{0, 100pPa\}$
Source	$s_0(t)$	ricker( $t - 16.7$ ms)
	$(i_0, j_0)$	(25, 25)
Sensors	Setup	Figure 4

25); it is modeled by choosing the source function as  $s_0[n] = s_0(n\Delta_t)$  where  $s_0(t)$  is a time-shifted Ricker wavelet. A Ricker wavelet [29] is defined by the negative second derivative of a Gaussian function such that

$$\text{ricker}(t) = (1 - 2\pi^2 v^2 t^2) \exp(-\pi^2 v^2 t^2). \quad (35)$$

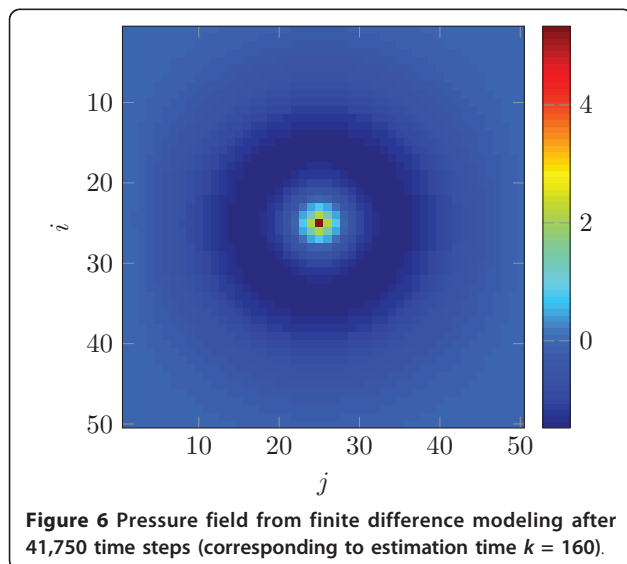
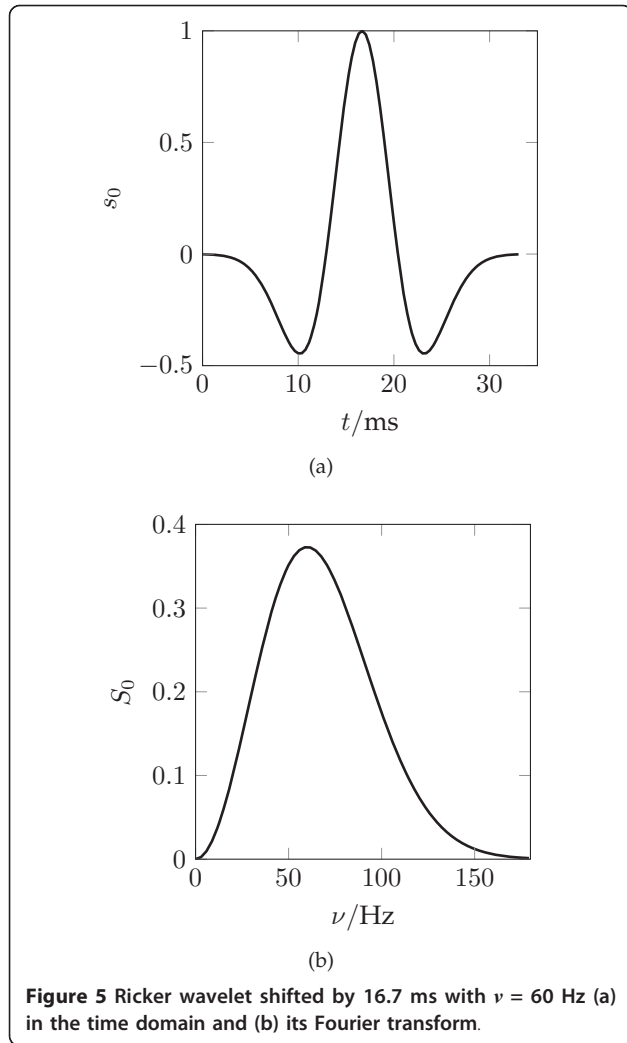
Here,  $v$  is approximately the peak frequency. A Ricker wavelet shifted by 16.7 ms with  $v = 60$  Hz is used, i.e.  $s_0(t) = \text{ricker}(t - 16.7 \text{ ms})$ , see Figure 5. The acoustic pressure field is simulated using the FDM introduced in Section II. A snapshot of the field at time  $k = 160$  is shown in Figure 6.

The parameters used in the decentralized PF are summarized in Table 3 ( $\mathcal{U}\{a, b\}$  represents a discrete uniform PDF with support  $[a, b]$ ). For the fixed source position, we used a discrete uniform distribution on the  $50 \times 50$  lattice. The spatio-temporal noise and the observation noise are drawn from a Gaussian distribution. The PF is initialized at time  $k = 0$ , and the source is assumed to become active at time instant  $k < 0$ . The maximum value of the random variable  $k_{\text{start}}$  is a prior and is proportional to the maximal possible time duration between source arise and first detection (cf. (20)). Larger values of  $k_{\text{start}}$  necessitate a larger number of particles to cover the time interval  $[-k_{\text{start}}, 0]$  and thus to achieve the same approximation accuracy.

### A Estimation of posterior PDF

For the centralized PF, Figure 7a shows an example of the posterior PDF  $P_s(i, j, k)$  for the source position obtained with the centralized particle filter at time instant  $k = 160$  (cf. (23)). For comparison, Figure 7b shows the result obtained with the decentralized PF, i.e., the composition  $\sum_{m=1}^5 P_s^{(m)}(i, j, k)$  of the local posterior PDF obtained by each cluster. It is seen that the centralized and the decentralized PF obtain similar results, and both yield a posterior PDF which is well concentrated about the true position  $(i_0, j_0) = (25, 25)$  of the source.

Figure 8a, b shows the MAP and MMSE of the source's  $i$  coordinate and  $j$  coordinate, respectively. The



**Table 3** PF parameters

Particles	$L$	20,000
Space/time jitter	$x, y$	$\mathcal{N}\{0, \Delta_r^2/8^2\}$
	$t$	$\mathcal{N}\{0, \Delta_t^2/8^2\}$
	$\mathbf{v}$	i.i.d. $\mathcal{N}\{0, 5 \text{ mPa}\}$
Priors	$k_{\text{start}}$	$\mathcal{U}\{0, 41345\}$
	$i, j$	$\mathcal{U}\{0, 50\}$

MAP estimates  $(\hat{i}_k, \hat{j}_k)$  are given by (32); the MMSE estimates  $(\hat{i}_k^{\text{MMSE}}, \hat{j}_k^{\text{MMSE}})$  are obtained as conditional means of the source coordinates obtained with the conditional posterior PMF  $P_s(i, j, k)$  for given  $k$ . Since the prior of the source location is a discrete uniform distribution, MMSE estimates at  $k = 0$  equal  $(\hat{i}_k^{\text{MMSE}}, \hat{j}_k^{\text{MMSE}}) = (I/2, J/2) = (25, 25)$ . Hence, in this specific case, the MMSE estimates outperform the MAP estimates for small  $k$ . After a certain number of PF iterations (around  $k > 6$ ), however, the MAP estimates match the true source position better than the MMSE estimates. The variance of  $P_s(i, j, k)$  for any given  $k$  (which can be interpreted as MMSE) is shown in Figure 9 and corroborates that for small-to-medium  $k$ , the  $i$  coordinate estimate is more reliable; this can be attributed to the specific sensor arrangements which favors better  $i$ -resolution (cf. Figure 4).

### B Decentralized MAP source localization

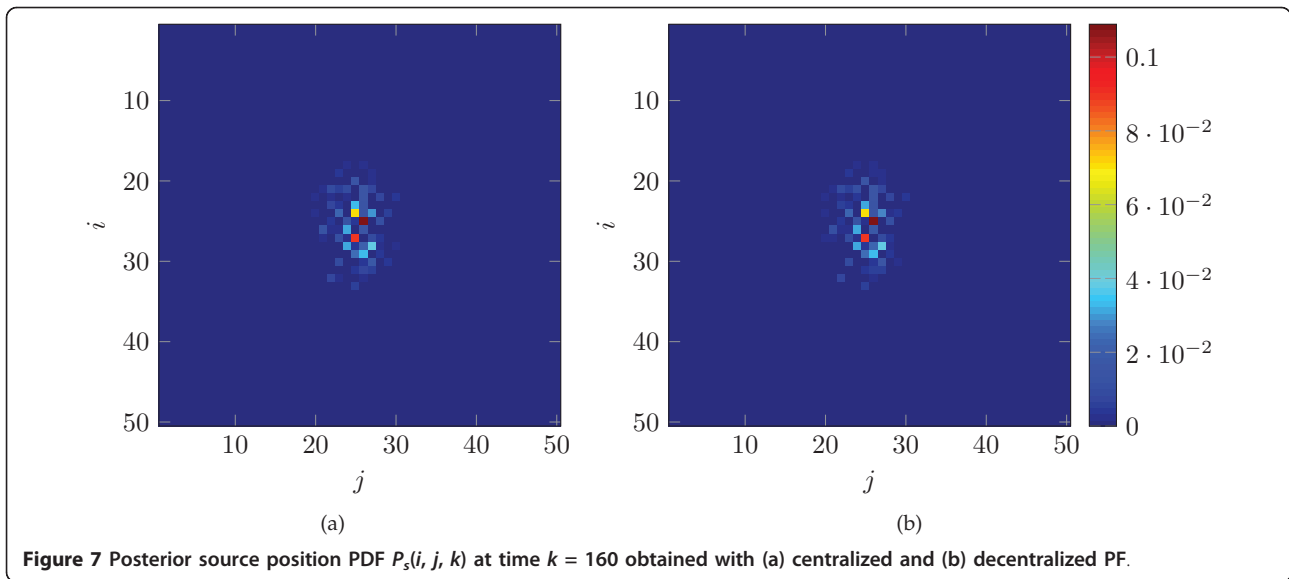
This subsection illustrates the decentralized source localization using the AMC algorithm proposed in Section V (simulation setup unchanged). Recall that with AMC, each cluster has estimates  $\hat{P}_{k,\text{max}}^{(m)}$  of the MAP probability and  $(\hat{i}_k^{(m)}, \hat{j}_k^{(m)})$  of the associated position. Figure 10a shows the local MAP probabilities  $P_{k,\text{max}}^{(m)}$  (cf. (33)) for all five clusters; clearly, only the third cluster builds up a distinguished maximum over time, which indicates that the source is located within  $\Omega^{(3)}$ .

All clusters track the global MAP probability, Figures 10b and 11, and eventually agree on the source position provided by cluster 3 whose behavior over time resembles the global estimates using the centralized PF (cf. Figure 8a, b).

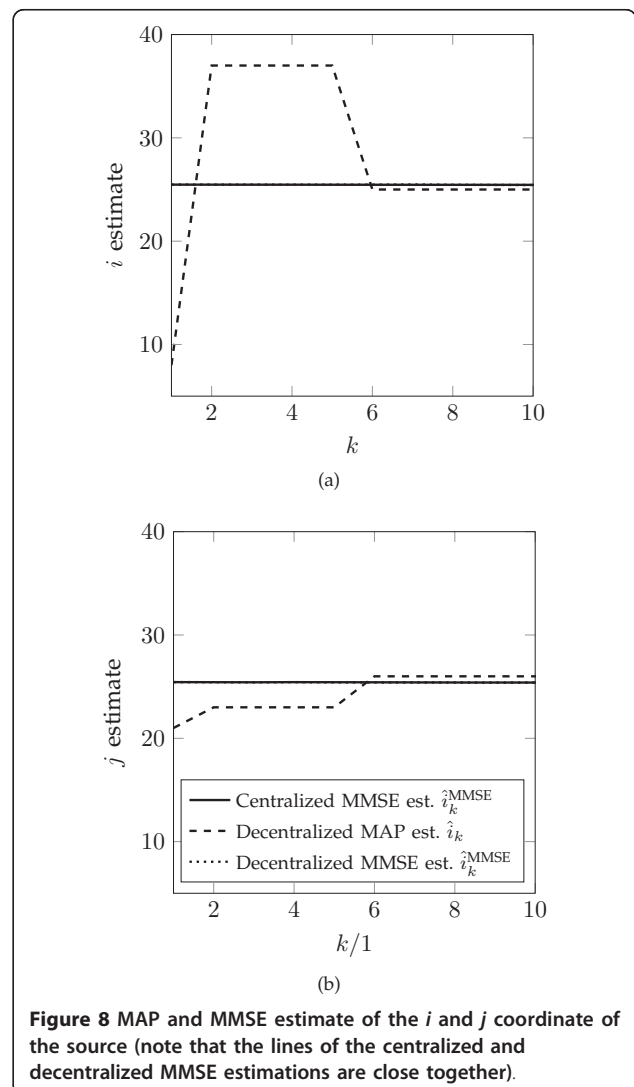
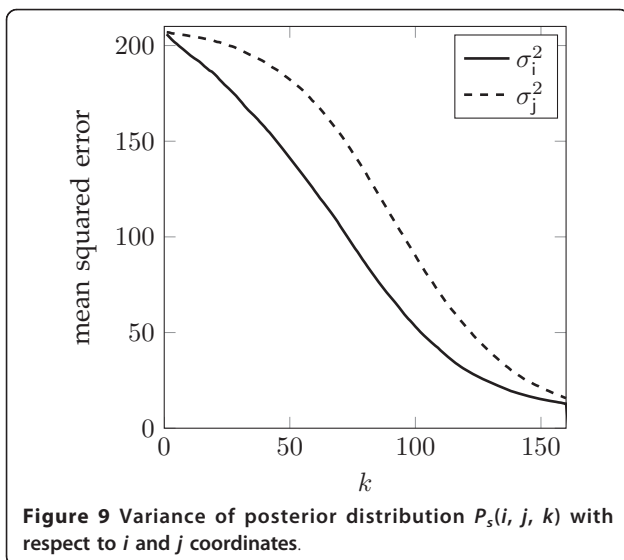
After about 6 iterations, the PF achieves a localization accuracy on the order of the lattice spacing  $\Delta_r$ . These estimates could be further improved (with higher computational complexity) by refining the discretization lattice and increasing the number of particles.

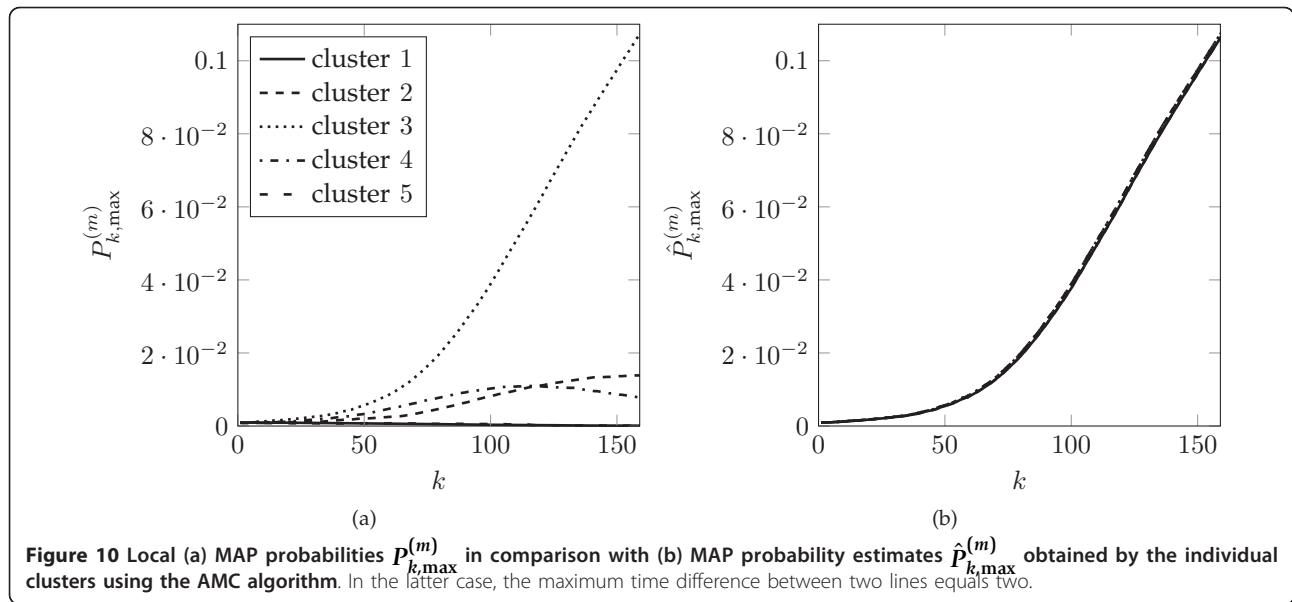
### VIII Conclusions

We proposed a scheme for the localization of multiple acoustic sources in a sensor network (SN). The method uses an augmented non-linear non-Gaussian state-space

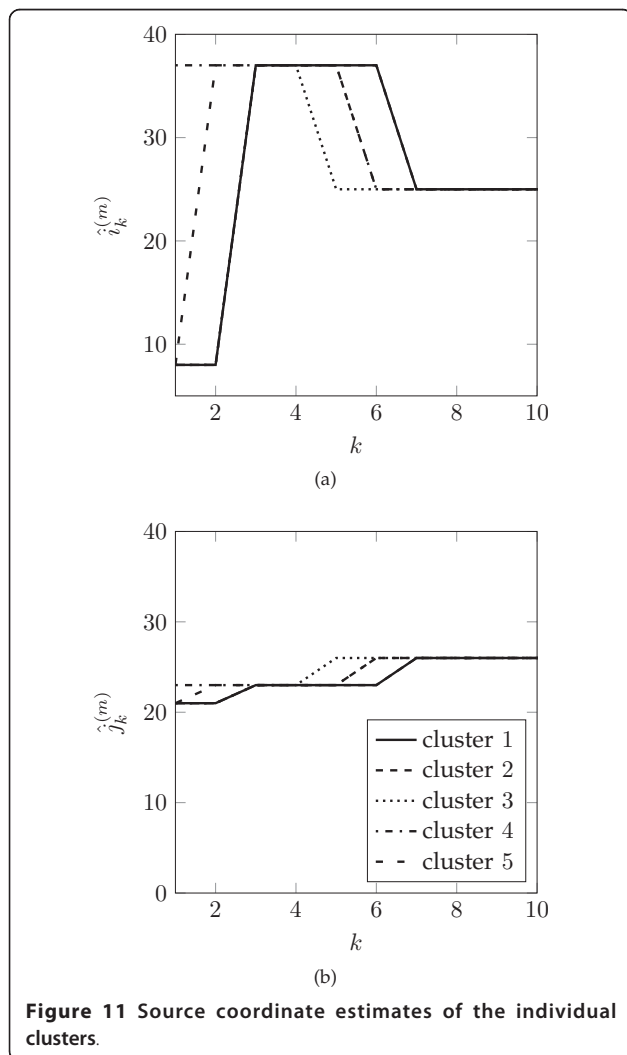


model for the acoustic field and on a particle filter (PF) for sequential Bayesian estimation of source positions. This state-space representation for the wave equation gives additional prior physical knowledge and incorporates perturbations and distortion like echoes, thereby resulting in improved estimation accuracy. In addition to the source positions, our PF implicitly provides an estimate of the acoustic field itself. We further developed a decentralized PF in which the computational complexity is distributed over several clusters of the SN. The decentralized PF exploits the sparsity of the matrices involved in the state-space model. In fact, the loose coupling between the components of the state vector allows separate and parallel computation of equation sub-systems of much smaller dimension in each cluster





**Figure 10** Local (a) MAP probabilities  $P_{k,\max}^{(m)}$  in comparison with (b) MAP probability estimates  $\hat{P}_{k,\max}^{(m)}$  obtained by the individual clusters using the AMC algorithm. In the latter case, the maximum time difference between two lines equals two.



**Figure 11** Source coordinate estimates of the individual clusters.

heads. To determine the global MAP estimate of the position of a source, we proposed an argumentum-maximi-consensus algorithm in which the clusters exchange their best MAP probability and source position.

#### Acknowledgements

This work is funded by Grant ICT08-44 of "Wiener Wissenschafts-, Forschungs- und Technologiefonds" (WWTF). This work in part was previously presented at the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Nov. 2010.

#### Author details

<sup>1</sup>Institute of Telecommunications (ITC), Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 1040 Vienna, Austria <sup>2</sup>Marine Physical Laboratory, Scripps Institution of Oceanography, University of California, San Diego, CA, USA

#### Competing interests

The authors declare that they have no competing interests.

Received: 19 January 2011 Accepted: 12 September 2011

Published: 12 September 2011

#### References

1. B Ristic, S Arulampalam, N Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, (Artech House, Boston) (2004)
2. O Hlinka, P Djuric, F Hlawatsch, Time-space-sequential distributed particle filtering with low-rate communications, in *Proceedings 43rd Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA) (2009)
3. O Hlinka, O Slučiak, F Hlawatsch, PM Djurić, M Rupp, Likelihood consensus: Principles and application to distributed particle filtering, in *Proceeding 44th Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, CA) (Nov. 2010)
4. P van Leeuwen, Particle filtering in geophysical systems. *Mon Weather Rev.* **137**, 4089 (2009). doi:10.1175/2009MWR2835.1
5. C Yardim, P Gerstoft, WS Hodgkiss, Tracking of geoacoustic parameters using kalman and particle filters. *J Acoust Soc Am.* **125**, 746 (2009). doi:10.1121/1.3050280
6. A Ihler, J Fisher III, R Moses, A Willsky, Nonparametric belief propagation for self-localization of sensor networks. *Sel Areas Commun, IEEE J.* **23**(4), 809–819 (2005)
7. S Kay, in *Fundamentals of Statistical Signal Processing, Estimation Theory*, vol. 1. (Pearson Education, New Jersey, 1993)



8. F Xaver, CF Mecklenbräuer, P Gerstoft, G Matz, Distributed state and field estimation using a particle filter, in *Proceeding 44th Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, CA, Nov. 2010)
9. J Candy, E Sullivan, Model-based identification: An adaptive approach to ocean-acoustic processing. *Oceanic Eng IEEE J.* **21**(3), 273–289 (2002)
10. L Rossi, B Krishnamachari, C Kuo, Distributed parameter estimation for monitoring diffusion phenomena using physical models, in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)* (2004)
11. T Zhao, A Nehorai, Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks. *IEEE Trans Signal Process.* **55**(4), 1511 (2007)
12. F Sawo, Nonlinear state and parameter estimation of spatially distributed systems, Ph.D. dissertation, Universität Karlsruhe (2009)
13. F Sawo, M Huber, U Hanebeck, Parameter identification and reconstruction for distributed phenomena based on hybrid density filter, in *Information Fusion, 2007 10th International Conference on*, pp. 1–8 (2007)
14. D Bauso, L Giarré, R Pesenti, Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Sys Control Lett.* **55**(11), 918–928 (2006). doi:10.1016/j.sysconle.2006.06.005
15. A Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, (Society for Industrial and Applied mathematics, Philadelphia, 2005)
16. F Jensen, W Kuperman, M Porter, H Schmidt, *Computational ocean acoustics*, American Institute of Physics Press, New York, 1994)
17. T Kailath, *Linear Systems* (Prentice-Hall, New Jersey, 1980)
18. A Doucet, S Godsill, C Andrieu, On sequential monte carlo sampling methods for bayesian filtering. *Stat Comput.* **10**(3), 197–208 (2000). doi:10.1023/A:1008935410038
19. R Mattheij, S Rienstra, J ten Thije Boonkkamp, *Partial Differential Equations: Modeling, Analysis, Computation* (Society for Industrial and Applied mathematics, Philadelphia, 2005)
20. M Zhdanov, *Geophysical Inverse Theory and Regularization Problems* (Elsevier Science Ltd, Amsterdam, 2002)
21. JD Hol, TB Schön, F Gustafsson, On resampling algorithms for particle filters, in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE, Sept., Ed. IEEE*, pp. 79–82 (2006)
22. F Sawo, K Roberts, U Hanebeck, Bayesian estimation of distributed phenomena using discretized representations of partial differential equations. in *3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 16–23 (2006)
23. F Sawo, V Klumpp, U Hanebeck, Simultaneous state and parameter estimation of distributed-parameter physical systems based on sliced gaussian mixture filter, in *Information Fusion, 2008 11th International Conference on*, 1–8 (2008)
24. S Farahmand, S Roumeliotis, G Giannakis, Particle filter adaptation for distributed sensors via set membership, in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE*, 3374–3377 (2010)
25. B Oreshkin, M Coates, Asynchronous distributed particle filter via decentralized evaluation of gaussian products, in *Proceedings of the Sixth International Conference of Information Fusion (IEEE, Edinburgh, Scotland, July 2010)*
26. M Coates, Distributed particle filters for sensor networks, in *Proceedings of the 3rd international symposium on Information Processing in Sensor Networks (ACM, 2004)*, pp. 99–107
27. D Knuth, Big omicron and big omega and big theta. *ACM Sigact News.* **8**(2), 18–24 (1976). doi:10.1145/1008328.1008329
28. M Arulampalam, S Maskell, N Gordon, T Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002). doi:10.1109/78.978374
29. H Ryan, A choice of wavelets. *CSEG Recorder* (09 Sept. 1994)

doi:10.1186/1687-1499-2011-94

**Cite this article as:** Xaver et al.: Localization of acoustic sources using a decentralized particle filter. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:94.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)