Decentralized Variational Filtering for Target Tracking in Binary Sensor Networks

Jing Teng, Hichem Snoussi, *Member*, *IEEE*, and Cédric Richard, *Senior Member*, *IEEE*

Abstract—The prime motivation of our work is to balance the inherent trade-off between the resource consumption and the accuracy of the target tracking in wireless sensor networks. Toward this objective, the study goes through three phases. First, a cluster-based scheme is exploited. At every sampling instant, only one cluster of sensors that located in the proximity of the target is activated, whereas the other sensors are inactive. To activate the most appropriate cluster, we propose a nonmyopic rule, which is based on not only the target state prediction but also its future tendency. Second, the variational filtering algorithm is capable of precise tracking even in the highly nonlinear case. Furthermore, since the measurement incorporation and the approximation of the filtering distribution are jointly performed by variational calculus, an effective and lossless compression is achieved. The intercluster information exchange is thus reduced to one single Gaussian statistic, dramatically cutting down the resource consumption. Third, a binary proximity observation model is employed by the activated slave sensors to reduce the energy consumption and to minimize the intracluster communication. Finally, the effectiveness of the proposed approach is evaluated and compared with the state-of-the-art algorithms in terms of tracking accuracy, internode communication, and computation complexity.

Index Terms—Variational methods, Bayesian inference, sensor networks, Monte Carlo methods.

1 Introduction

Wireless sensor networks (WSNs) represent an entirely new way of looking at signal collecting and processing. Hundreds or thousands of sensor nodes are deployed in a large geographical area to form a dense wireless ad hoc network, which provides access to information anytime, anywhere by performing high-level distributed sensing, processing, and disseminating data. Because of the reliability, flexibility, cost-effectiveness, and ease of deployment, WSNs promise to revolutionize our life in a wide range of application domains, including military, civil, and ecological areas, etc., [1], [2]. In spite of the diverse applications, WSNs face a number of unique technical challenges due to their inherent energy and bandwidth limitations, ad hoc deployment, and unattended operation, etc., [3], [4]. Unfortunately, very little previous works on distributed systems can be applied to WSNs, since the underlying assumptions have changed dramatically. Therefore, innovative energy-aware, scalable, and robust algorithms for distributed signal processing in WSNs are highly required. In this paper, we examine the design of an energy-aware decentralized Bayesian inference method, which is applied to the target tracking, one of the most basic issues of ad hoc WSNs.

Considering the temporal states of a target as the complex dynamics of a random variable, we can thus treat the target tracking as an optimal filtering problem [5], which consists of recursive updating the posterior distribution of an unobserved target state $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ of dimension n_x, given the sequence of observed data $z_{1:t}$. The dynamical system is characterized by a Markov state evolution model $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ and an observation model $p(z_t \mid \boldsymbol{x}_t)$. In the Bayesian context, the task of target tracking can be formulated as recursive calculating the predictive distribution $p(\boldsymbol{x}_t|z_{1:t-1})$ and the posterior distribution $p(\boldsymbol{x}_t|z_{1:t})$ as follows:

$$\begin{aligned} &\textbf{Prediction :}\\ &p(\boldsymbol{x}_t|z_{1:t-1}) = \int_{\mathbb{R}^{n_x}} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|z_{1:t-1})d\boldsymbol{x}_{t-1},\\ &\textbf{Update :}\\ &p(\boldsymbol{x}_t|z_{1:t}) = \frac{p(z_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|z_{1:t-1})}{p(z_t|z_{1:t-1})}. \end{aligned} \tag{1}$$

One can note from the filtering scheme, shown in Fig. 1, that based on the state evolution model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the estimation of the target state is updated by incorporating the observation model $p(z_t|\boldsymbol{x}_t)$. Therefore, the modeling of the state evolution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and the observation $p(z_t|\boldsymbol{x}_t)$ has great impact on the accuracy and the energy efficiency of target tracking solution.

If the transition dynamics and the observation model are both Gaussian, the classic Kalman Filter (KF) provides an optimal Bayesian solution [6]. Hasu and Koivo [7] proved the feasibility of applying the decentralized KF in WSNs. KF is computationally efficient, but its performance is limited by the nonuniversal Gaussian modeling assumptions. The Extended Kalman Filtering (EKF) and the Unscented Kalman Filtering (UKF) [8] have been proposed to improve the KF flexibility. However, deviation of the transition dynamics and the linear/Gaussian modeling assumptions cause severe degradation in its tracking performances.

- *J. Teng is with the School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China. E-mail: jing0404@gmail.com.*
- *H. Snoussi is with Charles Delaunay Institute, LM2S, University of Technology of Troyes, 12 rue Marie Curie, BP 2060, 10000 Troyes Cedex, France. E-mail: hichem.snoussi@utt.fr.*
- *C. Richard is with the Laboratory Fizeau, UMR CNRS 6525, Observatory of Côte d'Azur, University of Nice Sophia-Antipolis, Parc Valrose, 06108 Nice Cedex 02, France. E-mail: cedric.richard@unice.fr.*

Manuscript received 31 Oct. 2008; revised 26 Jan. 2010; accepted 25 Apr. 2010; published online 25 June 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-10-0437.
Digital Object Identifier no. 10.1109/TMC.2010.117.

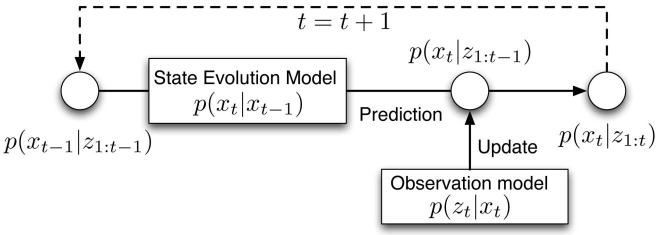

Fig. 1. Bayesian filtering scheme.

As for most nonlinear even non-Gaussian models, the closed-form analytic expression of the posterior distribution $p(\boldsymbol{x}_t|z_{1:t})$ is untractable [9]. To overcome this obstacle, a nonparametric Monte Carlo sampling-based method, called Particle Filtering (PF) [10], has gained popularity recently. The PF approximates the probability distribution by a set of weighted samples. It has the flexibility to accommodate nonlinear dynamics and multimodal observation models, but at the cost of more computation and storage requirements. In addition, the huge amount of the involved particles impeded its implementation in WSNs. Different methods to approximate the particles distribution were thus proposed in the literature [9], [11], [12], [13], in order to make the application of PF in WSNs possible. In [13], a message approximation scheme based on the greedy KD-tree was proposed. In [12], a full collaborative strategy for multiple targets tracking in wireless sensor networks, based on a Gaussian Mixture Model (GMM) message approximation scheme, was proposed. Kotecha and Djuric [9], [11] described the Gaussian Particle Filtering (GPF) algorithm [9] and the Gaussian Sum Particle Filtering (GSPF) algorithm [11], which approximate the posterior distribution by a single Gaussian distribution and a weighted sum of Gaussian distributions, respectively. The main drawback of these methods is the error propagation through the sensor network, when approximating the particle representation by a few number of Gaussian statistics. Recently, a variational approach that respects the communication constraints of sensor networks has been proposed in [14], which reduces the temporal dependence to a single Gaussian distribution. Contrary to the approximation methods mentioned above, the efficiency of the variational approximation relies on the fact that the online update of the filtering distribution and its compression are simultaneously performed. In addition, the variational approach has the nice property to be model-free, ensuring the robustness of data processing.

As the key impediments to successful WSN applications are the stringent energy constraints of sensor nodes, we extend the application of the variational approach to the context of Binary Sensor Networks (BSNs). A coarse but energy-efficient binary proximity observation model is employed by the sensors to detect the target. Besides, a cluster-based scheme is proposed, where sensors are statically divided into clusters, and each cluster consists of a single Cluster Head (CH) and a bunch of slave sensors. At every sampling instant, only one cluster of sensors is triggered to track the target. Resource consumption of the network is thus restricted to the activated cluster, where intracluster communication is dramatically reduced due to the binary proximity observation model. Concerning the intercluster information exchange, the belief estimate must be communicated between successive clusters, namely the handoff operation, when the activated cluster changes. By adopting the variational filtering (VF) algorithm, the intercluster communication is minimized, significantly reducing the required communication bandwidth and the energy consumption. Furthermore, based on the target tendency predicted by the VF, a nonmyopic cluster activation rule minimizes the occurrence of handoff operations. Therefore, the cluster-based VF algorithm for target tracking in BSNs ensures tracking accuracy with a minimum resource allocation.

In the following section, we will first formulate the target tracking problem in Section 2, where an analysis of the General State Evolution Model and the Binary Proximity Observation Model is given. Section 3 is about a detailed description of the variational filtering algorithm. The new target tracking algorithm balances the trade-off between the tracking accuracy and the resource consumption in BSNs. In Section 4, performances of the proposed algorithm are studied by computer simulations and are compared with the state-of-the-art algorithms [9], [15], [16]. Finally, in Section 5, we conclude and give some perspectives of this work.

2 Problem Formulation

As mentioned above, target tracking necessitates the modeling of the state evolution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and the observation $p(z_t|\boldsymbol{x}_t)$. In order to implement the VF algorithm in BSNs, we define the two specific models to ensure the estimation precision and the energy efficiency.

2.1 General State Evolution Model

Instead of the kinematic parametric model [17], [18], [2], which is usually used in tracking problems, we employ a General State Evolution Model (GSEM) [19], [14], [20]. This model is more appropriate to practical nonlinear and non-Gaussian situations, where no a priori information on the target velocity or its acceleration is available. The target temporal position $\boldsymbol{x}_t$ is assumed to follow a Gaussian model, where the expectation $\boldsymbol{\mu}_t$ and the precision matrix $\boldsymbol{\lambda}_t$ are both random. The randomness of the expectation and the precision is used here to further capture the uncertainty of the target state distribution. A practical choice of these distributions is a Gaussian distribution for the expectation $\boldsymbol{\mu}_t$ and a Wishart distribution for the precision matrix $\boldsymbol{\lambda}_t$. In other words, the hidden state $\boldsymbol{x}_t$ is extended to an augmented state $\boldsymbol{\alpha}_t = (\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)$, yielding a hierarchical model as follows:

$$\begin{cases} \boldsymbol{x}_t & \sim & \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t), \\ \boldsymbol{\mu}_t & \sim & \mathcal{N}(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}}), \\ \boldsymbol{\lambda}_t & \sim & \mathcal{W}_{n_x}(\boldsymbol{\lambda}_t|\bar{\boldsymbol{V}}, \bar{n}), \end{cases} \tag{2}$$

where the fixed hyperparameters $\bar{\boldsymbol{\lambda}}$, $\bar{n}$, and $\bar{\boldsymbol{V}}$ are the random walk precision matrix, the degrees of freedom, and the precision of the Wishart distribution, respectively. The dimension of the Wishart distribution equals that of the target state n_x. The GSEM is depicted in Fig. 2.

In fact, the marginal state distribution is obtained by integrating over the mean and precision matrix as follows:

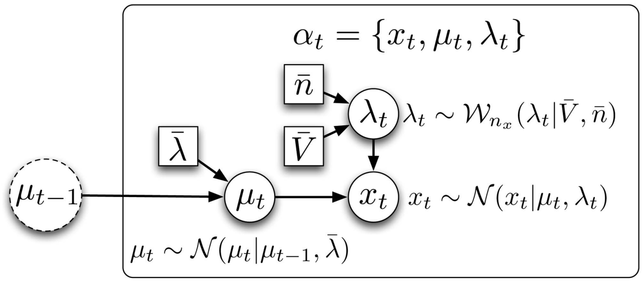
$\alpha_t = \{x_t, \mu_t, \lambda_t\}$
$\bar{n}$
λ_t
$\lambda_t \sim \mathcal{W}_{n_x}(\lambda_t|\bar{V}, \bar{n})$
$\bar{V}$
$\bar{\lambda}$
μ_{t-1}
μ_t
x_t
$x_t \sim \mathcal{N}(x_t|\mu_t, \lambda_t)$
$\mu_t \sim \mathcal{N}(\mu_t|\mu_{t-1}, \bar{\lambda})$

Fig. 2. Graphical model of the general state evolution model. Nodes denoted by circles correspond to hidden random variables, while nodes denoted by squares correspond to parameters of the model.

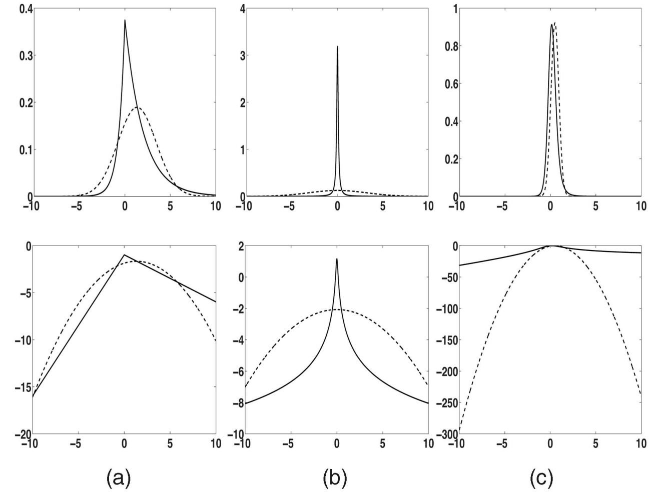

Fig. 3. Examples of the generalized hyperbolic distributions: (a) hyperbolic case, (b) cauchy case, and (c) student case. Pdfs appear on the top row and log densities on the bottom row. The dashed line corresponds to the Gaussian distribution with same mean and variance.

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \int\int \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)p(\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t|\boldsymbol{x}_{t-1})d\boldsymbol{\mu}_t d\boldsymbol{\lambda}_t, \qquad (3)$$

where the integration with respect to the precision matrix leads to the known class of scale mixture distributions introduced by Barndorff-Nielsen [21] for the scalar case. Low values of the degrees of freedom $\bar{n}$ reflect the heavy tails of the marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. In fact, varying the hyperparameters of model (2) yields a wide range of tail behaviors, from Gaussian tails to the heavy tails of the Student t-distributions. In order to illustrate the properties of scale mixture distributions, also referred to as Generalized Hyperbolic distributions, we take some examples of one-dimensional distributions with different hyperparameters. Fig. 3 represents the logarithm of Generalized Hyperbolic distributions for different hyperparameters values. One can note the model flexibility to cover a wide range of tail behaviors, which allows discrete jumps in the target trajectory.

2.2 Binary Proximity Observation Model

Considering the extremely constrained resource in the WSN, the Received Signal Strength Indicator (RSSI) technology [22], which has been proposed for hardware constrained systems, is adopted in the sensors to detect the target. It determines the distance between a receiver and a transmitter, based on the knowledge of a path-loss model. However, multipath reflections, nonline-of-sight conditions, and other shadowing effects lead to erroneous distance estimates. Therefore, a white Gaussian error ϵ_y^i is introduced to model the sensed observation y_t^i at sensor i of instant t:

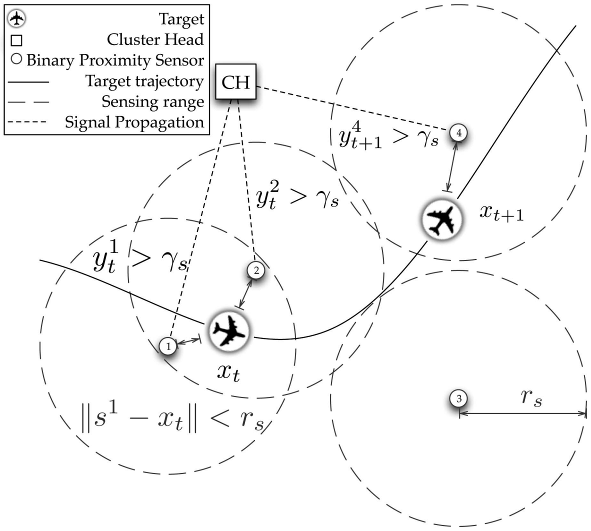

Fig. 4. The binary proximity observation model is described by a simple example. With respect to the first sensor, the target is within its sensing range at instant t. Observation "1" is thus transmitted to the CH. The same principle holds true for the second sensor. Concerning the third and fourth sensors, they keep silence at instant t. The CH then assigns "0" to the observation of them after waiting a given time slot. The situation at instant $t+1$ can be similarly deduced.

$$y_t^i \sim \mathcal{N}\big(y_t^i|\nu^i(\boldsymbol{x}_t), \sigma_y^{-2}\big),$$
$$\text{where} \quad \nu^i(\boldsymbol{x}_t) = \Psi_0 - 10\zeta\log\frac{\|\boldsymbol{s}^i - \boldsymbol{x}_t\|}{d_0}, \qquad (4)$$

where σ_y^2 is the variance of ϵ_y^i. The signal power $\nu^i(\boldsymbol{x}_t)$ is a one-to-one mapping to the distance $\|\boldsymbol{s}^i - \boldsymbol{x}_t\|$ traveled by the signal, where $\boldsymbol{s}^i$ is the location of the sensor i. The other denotations are, respectively, d_0 the reference distance, Ψ_0 the known received signal power in dBm at d_0, and ζ is the known path-loss distance exponent, which takes value in the range $[2,\ 4]$ ($\zeta = 2$ for propagation in free space, $\zeta = 4$ for relatively lossy environments and for the case of full specular reflection from the earth surface [23]). To further reduce the communication burden, a minimalist approach, the binary proximity observation model (BPOM) [24], [25], [26], [27], is employed. It makes a binary decision according to the strength of the perceived signal y_t^i, and only 1 bit is transmitted for further processing. The observation model is presented in Fig. 4, where the signal detection threshold γ_s is defined according to (4), namely $\gamma_s = \Psi_0 - 10\zeta\log(r_s/d_0)$, where r_s is the sensor detection range. If the observed signal y_t^i at sensor i is above the predefined threshold γ_s, it transmits "1" back to the CH that is responsible for signal processing; otherwise, no information about the target is transmitted. Due to the noisy wireless links, the signal received at the CH is assumed to be corrupted by normally distributed error ϵ_z^i. The realistic range measurement is thus formulated as follows:

$$z_t^i = \begin{cases} \beta^i + \epsilon_z^i, & \text{if } y_t^i \geq \gamma_s, \\ \epsilon_z^i, & \text{otherwise}, \end{cases} \qquad (5)$$

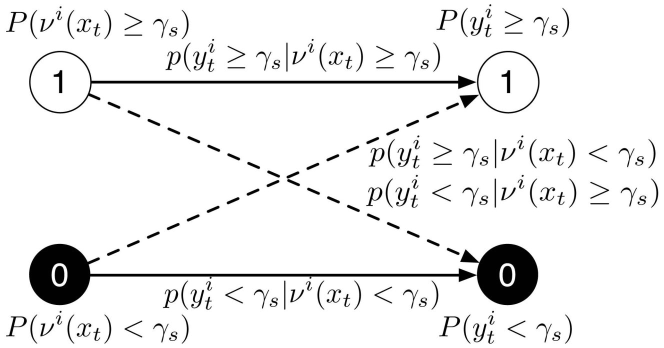
$P(\nu^i(x_t) \geq \gamma_s)$
$P(y_t^i \geq \gamma_s)$
$p(y_t^i \geq \gamma_s|\nu^i(x_t) \geq \gamma_s)$
1
1
$p(y_t^i \geq \gamma_s|\nu^i(x_t) < \gamma_s)$
$p(y_t^i < \gamma_s|\nu^i(x_t) \geq \gamma_s)$
0
0
$p(y_t^i < \gamma_s|\nu^i(x_t) < \gamma_s)$
$P(\nu^i(x_t) < \gamma_s)$
$P(y_t^i < \gamma_s)$

Fig. 5. Probability model of the sensed observation y_t^i with false alarms, where the dashed lines with arrows denote the false alarms.

$$\text{where} \quad \epsilon_z^i \sim \mathcal{N}\big(\epsilon_z^i|0,\sigma_z^{-2}\big),$$

and β^i is the attenuation coefficient associated with the sensor i. The Bayesian filtering framework for target tracking requires the construction of the distribution $p(z_t|\boldsymbol{x}_t)$. To track the target $\boldsymbol{x}$, the available temporal observation at the activated CH is denoted by $z_t = \{z_t^i\}_{i=1}^{m_t}$, where m_t is the number of slave sensors in the activated cluster, which is not constant during the tracking process. Assuming that the noise samples ϵ_z^i and the observed measurements y_t^i are independently distributed, we have

$$\begin{aligned} p(z_t|\boldsymbol{x}_t) = \prod_{i=1}^{m_t} \big[&\mathcal{N}\big(z_t^i|\beta^i,\sigma_z^{-2}\big)P\big(y_t^i \geq \gamma_s\big) \\ &+ \mathcal{N}\big(z_t^i|0,\sigma_z^{-2}\big)P\big(y_t^i < \gamma_s\big)\big], \\ \text{where} \quad P\big(y_t^i \geq \gamma_s\big) &= \int_{\gamma_s}^{\infty} \mathcal{N}\big(y_t^i|\nu^i(\boldsymbol{x}_t),\sigma_y^{-2}\big)dy_t^i, \\ \text{and} \quad P\big(y_t^i < \gamma_s\big) &= 1 - P\big(y_t^i \geq \gamma_s\big). \end{aligned} \tag{6}$$

An important problem introduced by the definition of the observation model $p(z_t|\boldsymbol{x}_t)$ is the false alarm. One can note from the formulation (4) that the mapping between $\boldsymbol{x}_t$ and y_t^i is not deterministic, due to the shadowing effect of ϵ_y^i. That is, if $\nu^i(\boldsymbol{x}_t)$, the one-to-one mapping to the true distance $\parallel \boldsymbol{s}^i - \boldsymbol{x}_t \parallel$, is greater than the threshold γ_s, the observed measurement y_t^i is not necessarily greater than γ_s. In fact, $P(y_t^i \geq \gamma_s)$ can also be formulated by

$$\begin{aligned} P\big(y_t^i \geq \gamma_s\big) = \big[&p\big(y_t^i \geq \gamma_s|\nu^i(\boldsymbol{x}_t) \geq \gamma_s\big)P\big(\nu^i(\boldsymbol{x}_t) \geq \gamma_s\big) \\ &+ p\big(y_t^i \geq \gamma_s|\nu^i(\boldsymbol{x}_t) < \gamma_s\big)P\big(\nu^i(\boldsymbol{x}_t) < \gamma_s\big)\big], \end{aligned}$$

as shown in Fig. 5. According to (6), the probability of false alarm $p(y_t^i \geq \gamma_s|\nu^i(\boldsymbol{x}_t) < \gamma_s)$ has already been naturally incorporated during the integral. Similarly, the symmetric probability of false alarm $p(y_t^i < \gamma_s|\nu^i(\boldsymbol{x}_t) \geq \gamma_s)$ is incorporated in the calculation of $P(y_t^i < \gamma_s)$.

Despite of the energy efficiency, tracking in BSNs suffers from poor estimation performances [24], [28], [29] because of the coarse information exploited. In recent years, substantial efforts have been dedicated to this problem. Djurić et al. [30], [16] apply the particle filtering method in the context of BSN, using a centralized manner, which yields acceptable tracking performance. Ribeiro and Giannakis [31], [32] condensed the observation down to a single bit. Instead of simply sensing the presence of the target, sensors still need to estimate their distances to the target, and then compare the timely observation with the estimation of the previous instant. A binary decision is thus made to denote the target moving manner, and transmitted to the central fusion unit for further processing. An implementation of the Sign of Innovation extended Kalman Filtering (SOI-KF) algorithm in WSN was proposed in [15]. The two algorithms using the binary observation model will be compared with our proposed algorithm in Section 4.

2.3 Problem Statement

To sum up, the hidden state $\boldsymbol{x}_t$ of the target is extended by the GSEM (2) to an augmented state $\boldsymbol{\alpha}_t = \{\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t\}$. Based on the BPOM (6) introduced above, our purpose consists of recursive computing the predictive distribution $p(\boldsymbol{\alpha}_t|z_{1:t-1})$ and updating the marginal posterior distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$ according to (1). Then, any quantity of interest, denoted by $f(\boldsymbol{\alpha}_{0:t})$, could be estimated by its a posteriori expectation:

$$\begin{aligned} I(f) &= \int f(\boldsymbol{\alpha}_{0:t})p(\boldsymbol{\alpha}_{0:t}|z_{1:t})d\boldsymbol{\alpha}_{0:t} \\ &= \boldsymbol{E}_{p(\boldsymbol{\alpha}_{0:t}|z_{1:t})}[f(\boldsymbol{\alpha}_{0:t})], \end{aligned}$$

where $z_{1:t} \equiv \{z_1, z_2, \ldots, z_t\}$ denotes the collection of observations gathered until time t.

3 Tracking in Binary Sensor Networks

3.1 Variational Approach

The nonlinear and the non-Gaussian aspect of the GSEM (2) and the BPOM (6) lead to intractable integrals when evaluating the marginals $p(\boldsymbol{\alpha}_t|z_{1:t})$ and $p(\boldsymbol{\alpha}_t|z_{1:t-1})$. An interesting solution is to resort to the Monte Carlo method, using a point-mass distribution of a set of weighted samples (called particles) to approximate the posterior distribution. Whereas the key contribution of this paper is the use of the variational filtering instead of the classical PF. The advantage of the variational approach will be naturally revealed after the following technical description. The variational approach consists of approximating the marginal posterior distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$ by a separable distribution $q(\boldsymbol{\alpha}_t) = \prod_i q(\boldsymbol{\alpha}_t^i)$ that minimizes the Kullback-Leibler (KL) divergence error:

$$\begin{aligned} D_{\mathrm{KL}}(q\|p) &= \int q(\boldsymbol{\alpha}_t)\log\left[\frac{q(\boldsymbol{\alpha}_t)}{p(\boldsymbol{\alpha}_t|z_{1:t})}\right]d\boldsymbol{\alpha}_t, \\ \text{where} \quad q(\boldsymbol{\alpha}_t) &= \prod_i q\big(\boldsymbol{\alpha}_t^i\big) = q(\boldsymbol{x}_t)q(\boldsymbol{\mu}_t)q(\boldsymbol{\lambda}_t). \end{aligned}$$

To minimize the KL divergence subject to the constraint $\int q(\boldsymbol{\alpha}_t)d\boldsymbol{\alpha}_t = \prod_i \int q(\boldsymbol{\alpha}_t^i)d\boldsymbol{\alpha}_t^i = 1$, the Lagrange multiplier method is used:

$$D_{\mathrm{KL}}(q\|p) = \int \prod_i q\big(\boldsymbol{\alpha}_t^i\big)\left[\sum_i \log q\big(\boldsymbol{\alpha}_t^i\big) - \log p(\boldsymbol{\alpha}_t|z_{1:t})\right]d\boldsymbol{\alpha}_t,$$

differentiating it with respect to $q(\boldsymbol{\alpha}_t^i)$,

$$\frac{\partial D_{\mathrm{KL}}(q\|p)}{\partial q(\boldsymbol{\alpha}_t^i)} = \log q\big(\boldsymbol{\alpha}_t^i\big) - \langle \log p(z_{1:t}, \boldsymbol{\alpha}_t)\rangle_{\prod_{j\neq i}} + 1 + \lambda_i,$$

where λ_i is a Lagrange multiplier introduced to ensure that $q(\boldsymbol{\alpha}_t^i)$ is normalized. The approximate distribution is thus yielded [33] as follows:

$$q(\boldsymbol{\alpha}_t^i) \propto \exp\langle \log\, p(z_{1:t}, \boldsymbol{\alpha}_t)\rangle_{\prod_{j\neq i} q(\boldsymbol{\alpha}_t^j),} \tag{7}$$

where $\langle\cdot\rangle_{q(\boldsymbol{\alpha}_t^j)}$ denotes the expectation operator relative to the distribution $q(\boldsymbol{\alpha}_t^j)$. Therefore, these dependent parameters can be jointly and iteratively updated.

3.1.1 Update

Taking into account the separable approximate distribution $q(\boldsymbol{\alpha}_{t-1})$ at time $t-1$, the filtering distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$ is sequentially approximated according to the following scheme: (see Appendix A for details):

$$\begin{aligned} &\hat{p}(\boldsymbol{\alpha}_t|z_{1:t}) \propto p(z_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t, \boldsymbol{\lambda}_t|\boldsymbol{\mu}_t)q_p(\boldsymbol{\mu}_t),\\ &\text{where}\quad q_p(\boldsymbol{\mu}_t) = \int p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1})q(\boldsymbol{\mu}_{t-1})d\boldsymbol{\mu}_{t-1}. \end{aligned} \tag{8}$$

Therefore, through a simple integral with respect to $\boldsymbol{\mu}_{t-1}$, the filtering distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$ can be sequentially updated. Considering the GSEM proposed in (2), the evolution of $\boldsymbol{\mu}_{t-1}$ is Gaussian, namely $p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}})$. Defining $q(\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}^*, \boldsymbol{\lambda}_{t-1}^*)$, $q_p(\boldsymbol{\mu}_t)$ is also Gaussian (see Appendix B.1 for details), with the following parameters:

$$\begin{aligned} &q_p(\boldsymbol{\mu}_t) \sim \mathcal{N}(\boldsymbol{\mu}_t^p, \boldsymbol{\lambda}_t^p),\\ &\text{where}\quad \boldsymbol{\mu}_t^p = \boldsymbol{\mu}_{t-1}^* \quad\text{and}\quad \boldsymbol{\lambda}_t^p = \left({\boldsymbol{\lambda}_{t-1}^*}^{-1} + \bar{\boldsymbol{\lambda}}^{-1}\right)^{-1}. \end{aligned} \tag{9}$$

The temporal dependence is, hence, reduced to the incorporation of only one Gaussian component approximation $q(\boldsymbol{\mu}_{t-1})$. The update and the approximation of the filtering distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$ are jointly performed, yielding a natural and adaptive compression [14], [20]. According to (7), variational calculus leads to closed-form expressions of $q(\boldsymbol{\mu}_t)$ and $q(\boldsymbol{\lambda}_t)$ (see Appendices B.1 and B.2), by substituting the deduction (9) into (8):

$$q(\boldsymbol{\mu}_t) \sim \mathcal{N}(\boldsymbol{\mu}_t^*, \boldsymbol{\lambda}_t^*), \quad q(\boldsymbol{\lambda}_t) \sim \mathcal{W}_{n_x}(\boldsymbol{V}^*, n^*),$$

where the parameters are iteratively updated until convergence, according to the following scheme:

$$\begin{aligned} \boldsymbol{\mu}_t^* &= \boldsymbol{\lambda}_t^{*-1}\left(\langle\boldsymbol{\lambda}_t\rangle\langle\boldsymbol{x}_t\rangle + \boldsymbol{\lambda}_t^p\boldsymbol{\mu}_t^p\right),\\ \boldsymbol{\lambda}_t^* &= \langle\boldsymbol{\lambda}_t\rangle + \boldsymbol{\lambda}_t^p,\\ n^* &= \bar{n} + 1,\\ \boldsymbol{V}^* &= \left(\langle\boldsymbol{x}_t\boldsymbol{x}_t^T\rangle - \langle\boldsymbol{x}_t\rangle\langle\boldsymbol{\mu}_t\rangle^T - \langle\boldsymbol{\mu}_t\rangle\langle\boldsymbol{x}_t\rangle^T\right.\\ &\qquad \left.+ \langle\boldsymbol{\mu}_t\boldsymbol{\mu}_t^T\rangle + \bar{\boldsymbol{V}}^{-1}\right)^{-1}. \end{aligned} \tag{10}$$

Note that $\langle\cdot\rangle$ denotes the expectation relative to the distribution $q(\cdot)$. The expectations and the precision matrices of $q(\boldsymbol{\mu}_t)$ and $q(\boldsymbol{\lambda}_t)$ have closed forms, which are easily derived as follows:

$$\langle\boldsymbol{\mu}_t\rangle = \boldsymbol{\mu}_t^*,\ \langle\boldsymbol{\lambda}_t\rangle = n^*\boldsymbol{V}^*,\ \langle\boldsymbol{\mu}_t\boldsymbol{\mu}_t^T\rangle = \boldsymbol{\lambda}_t^{*-1} + \boldsymbol{\mu}_t^*\boldsymbol{\mu}_t^{*T}. \tag{11}$$

However, the state $\boldsymbol{x}_t$ does not have a tractable form approximate distribution. By combining (7) and (8) (see Appendix B.3), we have the following form:

$$q(\boldsymbol{x}_t) \propto p(z_t|\boldsymbol{x}_t)\mathcal{N}(\langle\boldsymbol{\mu}_t\rangle, \langle\boldsymbol{\lambda}_t\rangle). \tag{12}$$

Thus, the GSEM (2) and the BPOM (6) are naturally incorporated to update $q(\boldsymbol{x}_t)$. This form immediately suggests an importance sampling (IS) procedure, where samples are drawn from the Gaussian distribution $\mathcal{N}(\langle\boldsymbol{\mu}_t\rangle, \langle\boldsymbol{\lambda}_t\rangle)$, and are weighted according to their likelihoods (taking into account the binary quantization in the BPOM):

$$\boldsymbol{x}_t^{(i)} \sim \mathcal{N}(\langle\boldsymbol{\mu}_t\rangle, \langle\boldsymbol{\lambda}_t\rangle),\ w_t^{(i)} \propto \prod_{j=1}^{m} p\left(z_t^j|\boldsymbol{x}_t^{(i)}\right). \tag{13}$$

Then, the expectations relative to $q(\boldsymbol{x}_t)$ can be approximated by the Monte Carlo method:

$$\langle\boldsymbol{x}_t\rangle = \sum_{i=1}^{N} w_t^{(i)}\boldsymbol{x}_t^{(i)},\ \langle\boldsymbol{x}_t\boldsymbol{x}_t^T\rangle = \sum_{i=1}^{N} w_t^{(i)}\boldsymbol{x}_t^{(i)}\boldsymbol{x}_t^{(i)T}. \tag{14}$$

3.1.2 Prediction

According to the Bayesian inference framework shown in Fig. 1, besides updating the filtering distribution $p(\boldsymbol{x}_t|z_{1:t})$, the predictive distribution $p(\boldsymbol{x}_t|z_{1:t-1})$ also needs to be calculated. In addition, the variational filtering algorithm is executed on a distributed cluster base, where only one cluster of sensors in the network is activated at each instant to detect and process the target data. If the sensors chosen to be activated are the closer to the target, the more information could be obtained, leading to the more precise estimation in return. Therefore, the predictive target distribution $p(\boldsymbol{x}_t|z_{1:t-1})$ performs a crucial role in activating sensors for the next instant. In fact, $p(\boldsymbol{x}_t|z_{1:t-1})$ can be efficiently updated by variational inference. Taking into account the separable approximate distribution $q(\boldsymbol{\alpha}_{t-1}) \propto p(\boldsymbol{\alpha}_{t-1}|z_{1:t-1})$, the predictive distribution is written as

$$p(\boldsymbol{\alpha}_t|z_{1:t-1}) \propto p(\boldsymbol{x}_t, \boldsymbol{\lambda}_t|\boldsymbol{\mu}_t)q_p(\boldsymbol{\mu}_t). \tag{15}$$

The exponential form solution, which minimizes the KL divergence between the predictive distribution $p(\boldsymbol{\alpha}_t \mid z_{1:t-1})$ and the separable approximate distribution $q_{t|t-1}(\boldsymbol{\alpha}_t)$, yields Gaussian distributions for the target state and its mean, while a Wishart distribution for the precision matrix:

$$\begin{aligned} &q_{t|t-1}(\boldsymbol{x}_t) \propto \mathcal{N}\left(\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}, \langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}}\right),\\ &q_{t|t-1}(\boldsymbol{\mu}_t) \propto \mathcal{N}\left(\boldsymbol{\mu}_{t|t-1}^*, \boldsymbol{\lambda}_{t|t-1}^*\right),\\ &q_{t|t-1}(\boldsymbol{\lambda}_t) \propto \mathcal{W}_{n_x}\left(\boldsymbol{V}_{t|t-1}^*, n_{t|t-1}^*\right), \end{aligned}$$

where the parameters are updated according to the same iterative scheme (10), except that the expectations of the predictive target state are now evaluated by the following expressions:

$$\begin{aligned} \langle\boldsymbol{x}_t\rangle_{q_{t|t-1}} &= \langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}},\\ \langle\boldsymbol{x}_t\boldsymbol{x}_t^T\rangle_{q_{t|t-1}} &= \langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}}^{-1} + \langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}^T. \end{aligned} \tag{16}$$

Compared to the classical PF, the computational cost and the memory requirements are dramatically reduced by the variational approximation in the prediction phase. In fact, the expectations involved in the computation of the predictive distribution have closed forms, avoiding the use of the Monte Carlo integration.

3.2 Cluster-Based VF Algorithm

The main advantage of the variational approach is the compression of the statistics required to update the filtering distribution between two successive instants. This implicit compression makes the VF algorithm much more adapted to distributed implementation in WSNs. In fact, the VF

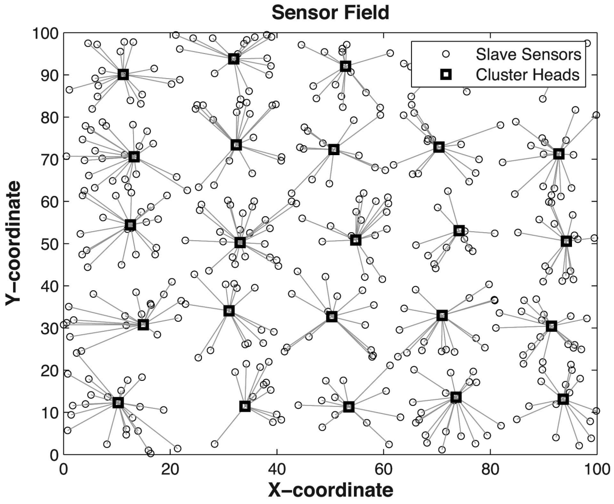

Fig. 6. Initialization scenario of a sensor field, where the lines indicate the intracluster communication.

algorithm is executed on a cluster base in our work, where clusters are statically formed in advance. Instead of an identical hardware configuration for all sensors, a hierarchical BSN is formed by the following:

- Slave sensors of identical sensing range r_s are randomly and densely deployed through the span of the network with a density ρ_s $(\mathrm{sensors/m^2})$. Each slave sensor belongs to only one cluster and is able to directly communicate with its cluster head via a single hop. By employing the BPOM (6), they identify themselves and report their one-bit observations to the corresponding cluster heads.
- CHs are sparsely placed with a density ρ_{CH} $(\mathrm{CHs/m^2})$ $(\rho_{CH} \ll \rho_s)$. The configuration and the deployment of the CHs guarantee the communication among neighboring CHs in one hop. Furthermore, each CH is capable of routing and forwarding the messages. Therefore, all the CHs can communicate with each other in multihop. CHs also have sufficient memory and computational resources to update the target belief by the VF algorithm using the observations collected by their slaves.

Fig. 6 describes such an initialization scenario of a sensor field. Initially, there is no target in the monitored area. To minimize energy consumption, all slave sensors are set to "Sleeping" mode. In contrast to the passive state of slave sensors, CHs are set to "Sensing" mode to detect the appearance of unidentified objects. Once an unidentified object is detected, the corresponding CH activates its slave sensors immediately to track the target, while the remaining sensors are kept in the "Sleeping" state. In particular, other noninvoked CHs are kept in the "Sensing" mode to cope with exceptions or any possible intrusions.

An overview of the VF process during consecutive sampling instants is presented in Fig. 7. The efficiency of the decentralized variational tracking algorithm depends on the relevance of the selected clusters in charge of processing the data and updating the filtering distribution. In the following, a nonmyopic cluster activation protocol based on the predictive distribution is described.

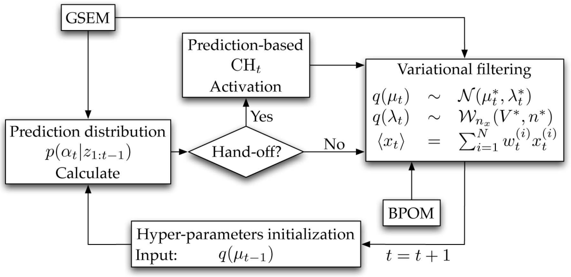
GSEM
Prediction-based
CH_t
Activation
Variational filtering
$q(\mu_t) \sim \mathcal{N}(\mu_t^*, \lambda_t^*)$
$q(\lambda_t) \sim \mathcal{W}_{n_x}(V^*, n^*)$
$\langle x_t \rangle = \sum_{i=1}^N w_t^{(i)} x_t^{(i)}$
Yes
Prediction distribution
$p(\alpha_t|z_{1:t-1})$
Calculate
Hand-off?
No
BPOM
Hyper-parameters initialization
Input: $q(\mu_{t-1})$
$t = t+1$

Fig. 7. VF process during consecutive sampling instants.

3.2.1 Nonmyopic Cluster Activation

The resource saving of cluster-based scheme lies in the fact that only one cluster is activated at each sampling instant to update the filtering distribution. Therefore, the cluster activation phase has a great importance not only in minimizing resource consumption but also in tracking accuracy. Various sensor activation strategies have been proposed in [34], [35]:

1. Naive activation, where all the sensors are active.
2. Randomized activation, where a random subset of the sensors is active.
3. Selective activation, where a subset of the sensors is activated according to some performance metric.
4. Duty-cycled activation, where sensors are active for some duty cycle.

By carefully choosing the activation parameters, selective activation strategy can provide great improvements in energy usage with quasi-optimal tracking quality [34]. Also, a commonly used solution is to activate the CH that is nearest to the target [36]. This strategy would incur unnecessary energy expenditure. First, all the CHs need to measure the distances between the target and themselves at every sampling instant; then, a comparison among them is required to choose the nearest one. The possibility of distributed signal processing is thus prevented and excessive communication is required. Second, when the target passes among the overlap of several clusters, frequent changes of the activated cluster occur, resulting in excessive communication of the temporal dependence information and leading to unacceptable additional energy and bandwidth consumption. To reduce the unnecessary energy expenditure, we propose to activate cluster based on the prediction of the target $\boldsymbol{x}_{t|t-1} = \langle \boldsymbol{x}_t \rangle_{q_{t|t-1}}$. If the predicted target position $\boldsymbol{x}_{t|t-1}$ remains in the vicinity of CH_{t-1}, then $\mathrm{CH}_t = \mathrm{CH}_{t-1}$. Otherwise, if $\boldsymbol{x}_{t|t-1}$ is going beyond the range of the current cluster, a new CH_t is activated based on the target position prediction $\boldsymbol{x}_{t|t-1}$ and its future tendency:

$$\begin{aligned} \mathrm{CH}_t &= \arg\max_{j=1,\dots,k} \left(\frac{\cos\ \theta_t^j}{d_t^j} \right), \\ \text{where} \quad d_t^j &= \|\boldsymbol{x}_{t|t-1} - \boldsymbol{p}_{\mathrm{CH}}^j\|, \\ \text{and} \quad \theta_t^j &\equiv \mathrm{angle}\big(\overrightarrow{\langle \boldsymbol{x}_{t-1}\rangle \boldsymbol{x}_{t|t-1}}, \overrightarrow{\langle \boldsymbol{x}_{t-1}\rangle \boldsymbol{p}_{\mathrm{CH}}^j}\big), \end{aligned} \tag{17}$$

where k is the number of CHs in the neighborhood of CH_{t-1} and $\boldsymbol{p}_{\mathrm{CH}}^j$ is the position of the jth neighboring CH. It is shown in Fig. 8 that when the angle θ_t^j between the

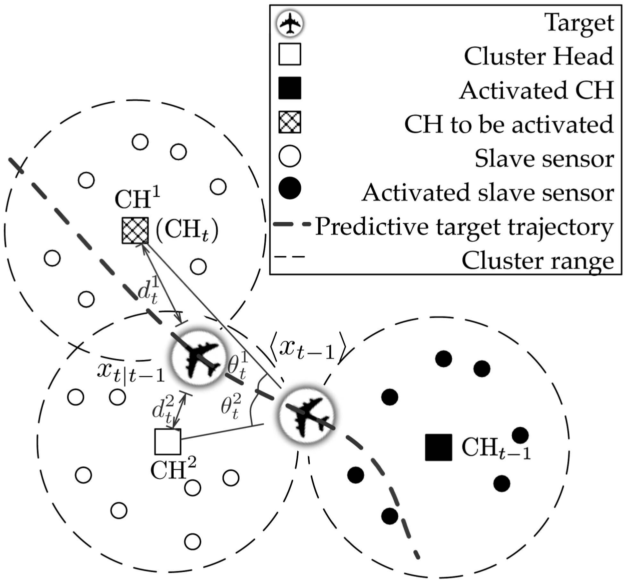

Fig. 8. Prediction-based CH_t activation.

predictive target and the CH is small, which corresponds to a large value of $\cos(\theta_t^j)$, it indicates that the target will move toward this CH. Thus, the rule in (17) combines two kinds of information related to

- the actual distance between the predictive target position $\boldsymbol{x}_{t|t-1}$ and the CH position $\boldsymbol{p}_{\mathrm{CH}}^j$, via the distance d_t^j and
- the future tendency of the target moving manner toward the CH, via the angle θ_t^j.

Furthermore, a handoff operation is triggered to transfer the temporal dependence information $q(\boldsymbol{\mu}_{t-1})$ from the CH_{t-1} to the CH_t. As illustrated in Fig. 8, traditional cluster activation rule activates CH^2 for updating the filtering distribution at time t, since it is the closest CH to the target prediction $\boldsymbol{x}_{t|t-1}$. But according to the tendency, the target is very likely to go out of its vicinity in a short time, causing excessive handoff operations. However, based on the decision rule (17), it is CH^1 that is activated, as the target is most likely detected by it and would stay in its vicinity for a comparative longer period. Taking into consideration the future tendency of the target, the nonmyopic decision rule (17) avoids unnecessary handoff operations compared to the traditional one. Therefore, the target tracking accuracy is ensured and the energy consumption is minimized as well. Due to the variational calculus, communication between CH_{t-1} and CH_t is limited to simply sending the mean and the precision of $q(\boldsymbol{\mu}_{t-1})$. Therefore, the cluster-based VF algorithm outperforms the classical PF algorithm in resource (energy, bandwidth, and memory) saving, as a large number of particles and corresponding weights are maintained and propagated in the latter case. With respect to the tracking accuracy, the VF and the PF algorithms approximate the true state distribution in different ways. When calculating the integral involved in the Bayesian filtering, the PF uses a large amount of particles whereas the VF introduces hidden variables to bypass the difficulty. These random variables introduced by the VF act as links that connect the observations to the unknown parameters via Bayes law. Furthermore, the error propagation problem is dramatically reduced as approximation of the filtering distribution is performed during observation incorporation.

Pseudocode of the cluster-based VF algorithm for target tracking in the BSN is summarized in Algorithm 1.

Algorithm 1. Variational Filtering algorithm

```
Input: $z_t$, $\bar{\boldsymbol{\lambda}}$ , $\bar{\boldsymbol{S}}$, $\bar{n}$, $\boldsymbol{\mu}_0^*$, $\boldsymbol{\lambda}_0^*$
Output: $\langle \boldsymbol{x}_t \rangle$
for $t = 1, 2, \dots,$ do
    $\boldsymbol{\mu}_t^p = \boldsymbol{\mu}_{t-1}^*$, $\boldsymbol{\lambda}_t^p = ({\boldsymbol{\lambda}_{t-1}^*}^{-1} + \bar{\boldsymbol{\lambda}}^{-1})^{-1}$, $q_p(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}_t^p, \boldsymbol{\lambda}_t^p)$;
    Predict $p(\boldsymbol{\alpha}_t | \boldsymbol{z}_{1:t-1})$ according to (15);
    The predicted expectation $\langle \boldsymbol{x}_{t|t-1} \rangle$ is calculated as (16);
    if hand-off then
        Select the new $\mathrm{CH}_t$ according to the decision rule (17);
        Communicate $q(\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}^*, \boldsymbol{\lambda}_{t-1}^*)$ to the new $\mathrm{CH}_t$;
    else
        Replace the storage of particles in the $\mathrm{CH}_t$ by $\boldsymbol{\mu}_{t-1}^*$ and $\boldsymbol{\lambda}_{t-1}^*$;
    end
    Initiate $\boldsymbol{\mu}_t^* = \boldsymbol{\mu}_t^p$, $\boldsymbol{\lambda}_t^* = 2\boldsymbol{\lambda}_t^p$, $n^* = \bar{n} + 1$,
    $\boldsymbol{S}^* = (2{\boldsymbol{\lambda}_t^p}^{-1} + \bar{\boldsymbol{S}}^{-1})^{-1}$;
    Calculate the initial expectations $\langle \boldsymbol{\mu}_t \rangle$ and $\langle \boldsymbol{\lambda}_t \rangle$ as (11);
    while not converge do
        Generate $N$ samples $\{\boldsymbol{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ from $q(\boldsymbol{x}_t)$,
        where $q(\boldsymbol{x}_t) \propto p(z_t | \boldsymbol{x}_t) \mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle)$;
        Compute the expectation $\langle \boldsymbol{x}_t \rangle$ and corresponding precision matrix as (14);
        Update the variational parameters $\boldsymbol{\mu}_t^*$, $\boldsymbol{\lambda}_t^*$, $n^*$, $\boldsymbol{S}^*$ according to (10);
        Re-update the expectations $\langle \boldsymbol{\mu}_t \rangle$ and $\langle \boldsymbol{\lambda}_t \rangle$ by (11);
    end
    Return the target position estimation $\langle \boldsymbol{x}_t \rangle$;
end
```

4 Evaluation and Simulation

The performance of the tracking algorithm can be quantified by the following three criteria:

- Tracking accuracy: evaluated by the Root Mean Square Error (RMSE) between the estimated and the true target trajectory.
- Internode communication: quantified by the amount of bits exchanged during a sampling interval.
- Computation complexity: evaluated by the execution time of the algorithm.

4.1 Initialization

We evaluate the cluster-based VF algorithm on a synthetic example. The purpose of the synthetic example is to establish a baseline performance evaluation. In fact, the deployment of sensors must ensure a high probability of detecting the appearance of a target. In order to generate enough information for target tracking, at least three slave sensors are required to detect the target and to report their observations for further processing. According to the network properties

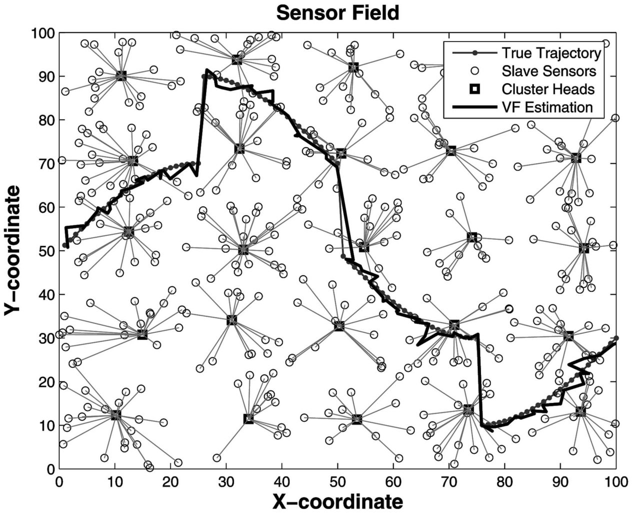

Fig. 9. Cluster-based VF algorithm in BSN.

described above, the probability distribution of the number of slave sensors in any given area A is Poisson with rate $\rho_s A$. Therefore, the probability for any arbitrary point in the field to be sensed by at least three slave sensors [36] is

$$p_s = \sum_{i=3}^{\infty} \frac{e^{-\rho_s \pi r_s^2} \left(\rho_s \pi r_s^2\right)^i}{i!}.$$

Similarly, the deployment of CHs should guarantee that at least one cluster is activated to track the target, the corresponding probability is

$$p_{CH} = \sum_{i=1}^{\infty} \frac{e^{-\rho_{CH} \pi r_{CH}^2} \left(\rho_{CH} \pi r_{CH}^2\right)^i}{i!}.$$

Therefore, in the region under surveillance, 400 slave sensors were randomly deployed in a two-dimensional field of $100 \times 100\ m^2$ ($\rho_s = 0.04\ \mathrm{sensors}/m^2$). The detection radius r_s was identically set to 10 m to ensure that $p_s \approx 1$. All these sensors belonged to 25 sparsely and uniformly positioned CHs. The cluster radius r_{CH} was 15 m for the one-coverage ($p_{CH} \approx 1$) requirement. The shadowing ϵ_y^i and the communication noise ϵ_z^i involved in the BPOM defined by (6) were assumed to be white Gaussian distributed, with variance $\sigma_y^2 = \sigma_z^2 = 0.01$. The parameters involved in the general state evolution model (2) were set as follows:

$$\bar{\boldsymbol{\lambda}} = \begin{bmatrix} 1/100 & 0 \\ 0 & 1/100 \end{bmatrix}, \quad \bar{n} = 1, \quad \bar{\boldsymbol{V}} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

The low state precision $\bar{\lambda}$ and the low degree of freedom $\bar{n}$ allow a general noninformative prior. All the simulations shown in this section were implemented by Matlab version 7.1, using an Intel Pentium D CPU 3.4 GHz, 1.0 GB RAM PC.

4.2 Cluster-Based VF for BSNs

Fig. 9 demonstrates the tracking performance of the proposed cluster-based VF algorithm using the binary proximity observation model. Accordingly, the RMSEs are depicted in Fig. 10. The simulation results confirm that despite of the abrupt changes in the target trajectory, the general state evolution model succeeds in describing the target state evolution.

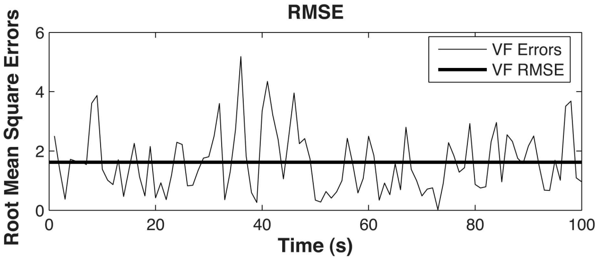

Fig. 10. Root mean square errors.

TABLE 1
Monte Carlo Evaluation of the Cluster Activation Rules

Evaluation	Non-myopic	Conventional
RMSE	1.4473	1.4428
Hand-off times	11.3	16.1
Inter-cluster communication (bits)	3966	8096
Execution time (s)	0.0682	0.0711

Monte Carlo simulations of 100 runs were performed in order to evaluate the performance of the proposed algorithm in terms of the criteria mentioned above. To prove the efficiency of the nonmyopic cluster activation rule defined by (17), the conventional cluster activation strategy, which activates the closest cluster to the prediction, is also evaluated and compared with ours in Table 1. Note that both of them are executed on the same configuration using the VF algorithm despite of different cluster activation rule. The expected qualitative performance of the proposed algorithm has been confirmed. With respect to the RMSE, both of them can ensure the desired tracking accuracy even with the coarse binary observation model, which proves the efficient approximation of the VF algorithm. The estimation precision, reflected by the RMSE, of the conventional strategy is slightly better than that of the nonmyopic one, as the conventional one chooses the closest cluster, which is the most informative. The average handoff occurrences are reduced from 16.1 times in the Re-CM strategy to 11.3 times, due to the nonmyopic selective cluster activation rule. Accordingly, the intercluster communication of the nonmyopic rule among the CHs is minimized compared to that of the conventional one. Finally, compared to the sampling instant (1 second), both the strategies succeed in real-time target tracking.

4.3 Comparison with State-of-the-Art Algorithms

To further demonstrate the efficiency of the proposed cluster-based variational filter (VF) algorithm, we compared it with the decentralized Gaussian particle filter (GPF) [9], the binary particle filter (BPF) [30], [16], and the decentralized version of the SOI-KF proposed in [15]. The BPOM, formulated in (6), was adopted, except for the SOI-KF algorithm. As have been mentioned above, the observation model of the SOI-KF algorithm is much more informative. Slave sensors of the SOI-KF also just transfer 1 bit information, which is not related to the presence or absence

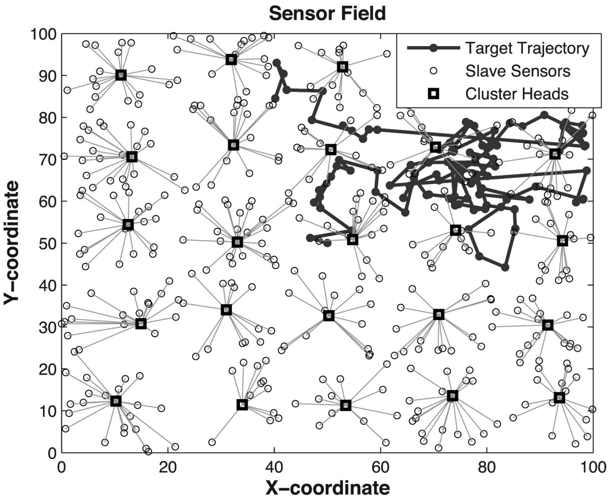

Fig. 11. Target trajectory generated by the RWM model.

of the target, but to the sign of difference between the range estimation and its prediction.

Concerning the state evolution model, instead of the GSEM model used in the VF algorithm, the other considered algorithms adopted the traditional kinematic parameter model:

$$\begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{v}_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{v}_{t-1} \end{pmatrix} + \begin{pmatrix} T_s^2/2 & 0 \\ 0 & T_s^2/2 \\ T_s & 0 \\ 0 & T_s \end{pmatrix} \boldsymbol{u}_t,$$

where $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ denote the position and the velocity of the target at instant t, respectively. The sampling interval is $T_s = 1\mathrm{s}$ and $\boldsymbol{u}_t \in \mathbb{R}^2$ is zero-mean white Gaussian noise. To establish a baseline performance comparison on a relatively difficult problem, the target motion was simulated by a random walk mobility (RWM) model [37], which mimics the erratic movement of a target in extremely unpredictable ways. This characteristic can generate unrealistic movements such as sudden stops and sharp turns. Whereas, it happens to depict the nonlinear and non-Gaussian dynamics in WSN. Due to the instability of wireless communication and detection failure of sensors, the probability of lost trace of the target increases consequently, which is then reflected as sharp turns or sudden stops in the target trajectory. Fig. 11 shows an example of target movement in a 2D field. The target begins its movement at the point $[50, 50]$. At each sampling instant, the target chooses an arbitrary direction between 0 and 2π and a random velocity. The target in Fig. 11 travels for 100 sampling time intervals.

The tracking performances of the VF algorithm are compared with those of the three representative state-of-the-art algorithms in Fig. 12, where the target follows the trajectory of Fig. 11. The tracking precision is represented by the RMSE. One can note that, even with abrupt changes at instant $t = 69$ and $t = 90$ in target trajectory, desired tracking quality is ensured by the VF algorithm with RMSE of 1.851 and maximum error of 5.648. The GPF and the BPF algorithms achieve comparable precisions, demonstrating their effectiveness in the nonlinear context. The RMSE of them is 2.103 and 2.087, respectively. However, at the instant of abrupt changes, the two algorithms fail to track the target; accordingly, the maximum errors are 9.852 for the GPF and 17.420 for the BPF. The precision of the particle filtering depends on the choice of the importance sampling distribution. In the VF algorithm, we also resort to the IS method to estimate the target position $\boldsymbol{x}_t$, where an optimal choice of the sampling distribution is yielded by minimizing the KL divergence. In fact, variational calculus leads to a simple Gaussian distribution whose parameters are iteratively optimized. Unlike the VF algorithm, the temporal dependence of the BPF is a huge amount of particles. The GPF algorithm then approximates the particle distribution by a Gaussian distribution. Similar to the VF algorithm, the GPF also only propagates the mean and the precision matrix of the single Gaussian distribution. However, the Gaussian distribution of GPF is based on the classical PF, where the mean and the precision are generated from the weighted average value and the inverse of the variance of the point-mass posterior distribution, respectively. Therefore, besides the approximation of the true posterior pdf by the particles $\{\boldsymbol{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$, a second approximation is performed in GPF, resulting the problem of error propagation. Consequently, the RMSE of the GPF is bigger than that of the BPF algorithm. The smaller RMSE of the VF in comparison with the two approximation methods confirms once again the effectiveness of the VF algorithm in terms of tracking accuracy. However, because of essential limitation of the KF algorithm in nonlinear environment, the SOI-KF cannot

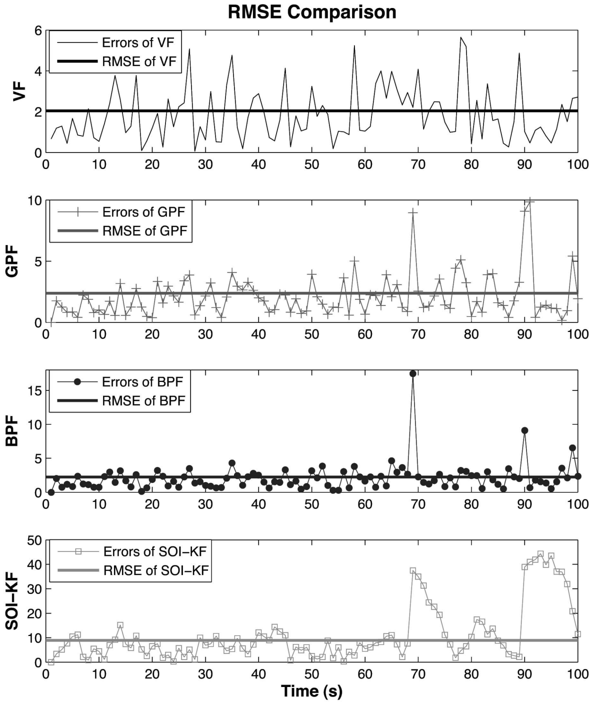

Fig. 12. Tracking performance comparison.

TABLE 2
Monte Carlo Simulation Results of the Algorithms

Comparison	RMSE	Execution time (s)
VF	2.129	0.052
GPF	5.626	0.053
BPF	5.030	0.049
SOI-KF	11.718	0.001

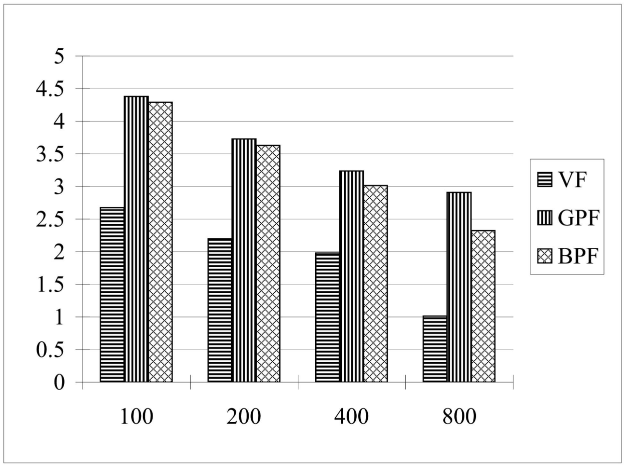

Fig. 13. RMSE versus the number of particles.

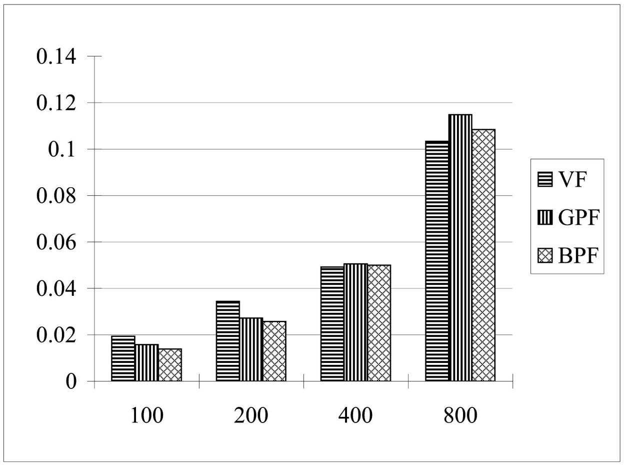

Fig. 14. Computation time versus the number of particles.

TABLE 3
Internode Communication Comparison

Comparison	Inter-node Communication (bits)
VF	$N_{as} + 32 + 64$
SOI-KF	$N_{as} + 32N_s$
BPF	$N_{as} + 32N_p + 16N_w$
GPF	$N_{as} + 32 + 64$

provide accurate tracking even with more informative observations, the RMSE of which is 11.010 and maximum error of which is 44.359.

Further extensive Monte Carlo simulations of 100 runs were performed on the same hardware configuration. Diverse target trajectories were generated at each run using the RWM model to get the universal results. During the Monte Carlo simulations, the GPF and the BPF demonstrated similar tracking performance, where the BPF slightly outperformed the GPF. However, when abrupt changes occurred in the target trajectory, the traditional kinematic model adopted in the GPF and the BPF failed to describe the state evolution, leading to serious estimation errors. Furthermore, such cases always resulted in exceptional interrupt of the execution of the GPF, when calculating the inverse of the particle variance. It clearly appears in Table 2 that the VF algorithm achieves the most accurate tracking performances, while the SOI-KF is the most computationally efficient. In fact, the VF, the BPF, and the GPF algorithms are essentially based on the importance sampling. Their space complexities during a time slot are thus proportional to the number of involved particles. Obviously, the SOI-KF appears as the most space-efficient algorithm, since it only involves simple matrix transformations at every sampling instant. Concerning the storage occupation rate during successive sampling instants, the VF and the GPF algorithms maintain only one Gaussian statistic, whereas the BPF algorithm stores a large number of particles.

Tracking performances of VF, GPF, and BPF versus the number of particles are shown in Fig. 13. Similarly, the computation time of them versus the number of particles is reported in Fig. 14. As can be expected, with the amount of particles increasing, the algorithms demonstrate much more accurate tracking at the cost of a higher computation complexity. In particular, the computation time grows almost proportionally to the increment of the number of particles. It clearly shows that the amount of 200 particles outperforms the others with respect to the trade-off between tracking accuracy and computation complexity. Therefore, 200 particles were used in the simulations demonstrated above.

We shall now consider the internode communication in Table 3. Let N_{as} denote the number of activated sensors that detect the target, N_s the total amount of sensors ($N_s \gg N_{as}$), N_p the number of particles, and N_w the number of corresponding weights ($N_w = N_p$). The intercluster communications of these algorithms were compared when a handoff operation occurred. One can note in Table 3 that the components N_{as} are equal for all the four algorithms, since each activated sensor that detects the target transmits 1 bit of information to the CH. By approximating the filtering distribution with a single Gaussian statistic, the VF and the GPF just have to transfer the expectation and the precision matrix. In the SOI-KF, all sensors need to be updated with the target prediction to make binary decisions, thus, requiring additional $32N_s$ bits. Internode communication of the BPF mainly lies in communicating the particles and their corresponding weights, which is much greater than that of the VF and GPF algorithms.

5 Conclusion

A cluster-based VF algorithm was proposed in the context of WSNs for single target tracking. The use of the general state evolution model to describe the target state with no a priori information on the target motion is more appropriate to practical nonlinear and even non-Gaussian situations. By adopting the VF algorithm, the update and the approximation of filtering distribution are jointly performed, allowing a lossless compression through the sensor network. As the temporal dependence is reduced to only one single Gaussian statistic, the VF algorithm outperforms the classical PF algorithm in terms of resource consumption. Aiming at conserving resource of the network, a nonmyopic CH activation rule is proposed to reduce the occurrence of the handoff operations, further minimizing energy consumption and intercluster communication. Based on the reliable prediction of the VF algorithm, effective clustering is ensured not only in terms of tracking accuracy but also in terms of energy efficiency. Furthermore, the binary proximity observation model quantifies the detected information to a single bit, dramatically reducing the energy consumption of sensors and also the intracluster communication cost.

We are currently working on the clustering protocols to serve different requirements of sensor networks. Besides minimizing the resource consumption over the whole network, an efficient clustering protocol should also maximize the lifetime of WSN by evenly distributing energy consumption over all sensors. The relationship between the prediction accuracy and the cluster activation strategy remains an open issue.

Appendix A

The Proof of the Inference (8)

The filtering distribution $p(\boldsymbol{\alpha}_t|z_{1:t})$, at instant t, could be written with respect to the previous filtering distribution $p(\boldsymbol{\alpha}_{t-1}|z_{1:t-1})$ as follows:

$$\begin{aligned} p(\boldsymbol{\alpha}_t|z_{1:t}) &\propto p(z_t|\boldsymbol{\alpha}_t, z_{1:t-1}) \int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}, z_{1:t-1}) d\boldsymbol{\alpha}_{t-1} \\ &\quad \int p(\boldsymbol{\alpha}_{t-1}|z_{1:t-1}) d\boldsymbol{\alpha}_{t-1} \\ &\propto p(z_t|\boldsymbol{\alpha}_t) \int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}) p(\boldsymbol{\alpha}_{t-1}|z_{1:t-1}) d\boldsymbol{\alpha}_{t-1}. \end{aligned} \tag{A.1}$$

Taking into account the approximate variational filtering distribution $q(\boldsymbol{\alpha}_{t-1})$ at time $t-1$, the above filtering distribution is rewritten as follows:

$$\begin{aligned} \hat{p}(\boldsymbol{\alpha}_t|z_{1:t}) &\propto p(z_t|\boldsymbol{\alpha}_t) \int q(\boldsymbol{\alpha}_{t-1}) d\boldsymbol{\alpha}_{t-1} \\ &\quad \int p(\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\lambda}_{t-1}) d\boldsymbol{\alpha}_{t-1}. \end{aligned} \tag{A.2}$$

The a priori Markovian model GSEM (2) could be further simplified as follows:

$$\begin{aligned} p(\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\lambda}_{t-1}) &= p(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t, \boldsymbol{x}_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\lambda}_{t-1}) \\ &\quad p(\boldsymbol{\mu}_t|\boldsymbol{\lambda}_t, \boldsymbol{x}_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\lambda}_{t-1}) \\ &\quad p(\boldsymbol{\lambda}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\lambda}_{t-1}) \\ &= p(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) p(\boldsymbol{\lambda}_t). \end{aligned}$$

Expression (A.2) can be simplified by noticing that $\int q(\boldsymbol{\alpha}_{t-1}) d\boldsymbol{\alpha}_{t-1} = \prod_i \int q(\alpha^i_{t-1}) d\alpha^i_{t-1} = 1$, and $p(z_t|\boldsymbol{\alpha}_t) = p(z_t|\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) = p(z_t|\boldsymbol{x}_t)$, according to the GSEM (2) and the BPOM (6). Therefore,

$$\begin{aligned} \hat{p}(\boldsymbol{\alpha}_t|\boldsymbol{z}_{1:t}) &\propto p(\boldsymbol{z}_t|\boldsymbol{\alpha}_t) \int q(\boldsymbol{\alpha}_{t-1}) d\boldsymbol{\alpha}_{t-1} \\ &\quad \int p(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) p(\boldsymbol{\lambda}_t) d\boldsymbol{\alpha}_{t-1} \\ &= p(\boldsymbol{z}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t, \boldsymbol{\lambda}_t|\boldsymbol{\mu}_t) \\ &\quad \int p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) q(\boldsymbol{\mu}_{t-1}) d\boldsymbol{\mu}_{t-1} \\ &= p(\boldsymbol{z}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t, \boldsymbol{\lambda}_t|\boldsymbol{\mu}_t) q_p(\boldsymbol{\mu}_t). \end{aligned}$$

Appendix B

Variational Calculus

Appendix B.1 $q(\boldsymbol{\mu}_t)$

Assuming the Gaussianity of the approximate distribution for the mean $\boldsymbol{\mu}_{t-1}$ ($q(\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}^*_{t-1}, \boldsymbol{\lambda}^*_{t-1})$) and taking into account the Gaussian transition of the mean ($p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}})$), we have

$$\begin{aligned} q_p(\boldsymbol{\mu}_t) &= \int p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) q(\boldsymbol{\mu}_{t-1}) d\boldsymbol{\mu}_{t-1} \\ &\sim \mathcal{N}\left(\boldsymbol{\mu}^*_{t-1}, \left({\boldsymbol{\lambda}^*_{t-1}}^{-1} + \bar{\boldsymbol{\lambda}}^{-1}\right)^{-1}\right). \end{aligned}$$

To simplify notations, let $\boldsymbol{\mu}^p_t$ and $\boldsymbol{\lambda}^p_t$ denote, respectively, the mean and the precision of the Gaussian distribution $q_p(\boldsymbol{\mu}_t)$: $q_p(\boldsymbol{\mu}_t) \sim \mathcal{N}(\boldsymbol{\mu}^p_t, \boldsymbol{\lambda}^p_t)$. According to (7), the approximate distribution $q(\boldsymbol{\mu}_t)$ has the following expression:

$$\begin{aligned} q(\boldsymbol{\mu}_t) &\propto \exp\langle \log p(z_{1:t}, \boldsymbol{\alpha}_t)\rangle_{q(\boldsymbol{x}_t)q(\boldsymbol{\lambda}_t)} \\ &\propto \exp\langle \log p(\boldsymbol{\alpha}_t|z_t)\rangle_{q(\boldsymbol{x}_t)q(\boldsymbol{\lambda}_t)} \\ &\propto \exp\langle \log p(z_t|\boldsymbol{x}_t) + \log p(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \\ &\quad + \log p(\boldsymbol{\lambda}_t) + \log q_p(\boldsymbol{\mu}_t)\rangle_{q(\boldsymbol{x}_t)q(\boldsymbol{\lambda}_t)}. \end{aligned} \tag{B.1}$$

For any $q(\boldsymbol{x}_t)$ and $q(\boldsymbol{\lambda}_t)$ integrable function $f(\boldsymbol{x}_t, \boldsymbol{\lambda}_t)$, $\int f(\boldsymbol{x}_t, \boldsymbol{\lambda}_t) dq(\boldsymbol{x}_t) q(\boldsymbol{\lambda}_t)$ is constant with respect to $\boldsymbol{\mu}_t$. Therefore,

$$\begin{aligned} q(\boldsymbol{\mu}_t) &\propto q_p(\boldsymbol{\mu}_t) \exp\langle \log p(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)\rangle_{q(\boldsymbol{x}_t)q(\boldsymbol{\lambda}_t)} \\ &\propto q_p(\boldsymbol{\mu}_t) \exp\left\langle -\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t)\right\rangle \\ &\propto q_p(\boldsymbol{\mu}_t) \exp -\frac{1}{2}\{tr[\langle\boldsymbol{\lambda}_t\rangle\langle(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T(\boldsymbol{x}_t - \boldsymbol{\mu}_t)\rangle]\} \\ &\propto \exp -\frac{1}{2}\left[(\boldsymbol{\mu}_t - \boldsymbol{\mu}^p_t)^T \boldsymbol{\lambda}_t (\boldsymbol{\mu}_t - \boldsymbol{\mu}^p_t) \right. \\ &\quad \left. - 2\boldsymbol{\mu}^T_t\langle\boldsymbol{\lambda}_t\rangle\langle\boldsymbol{x}_t\rangle + {\boldsymbol{\mu}_t}^T\langle\boldsymbol{\lambda}_t\rangle\boldsymbol{\mu}_t\right], \end{aligned} \tag{B.2}$$

yielding a Gaussian distribution $q(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}^*_t, \boldsymbol{\lambda}^*_t)$. Computing the first and second derivatives of the logarithm of $q(\boldsymbol{\mu}_t)$:

$$\begin{aligned} \frac{\partial \log(q(\boldsymbol{\mu}_t))}{\partial \boldsymbol{\mu}_t} &= -\frac{1}{2}\left[2\boldsymbol{\lambda}^p_t(\boldsymbol{\mu}_t - \boldsymbol{\mu}^p_t) \right. \\ &\quad \left. - 2\langle\boldsymbol{\lambda}_t\rangle\langle\boldsymbol{x}_t\rangle + 2\langle\boldsymbol{\lambda}_t\rangle\boldsymbol{\mu}_t\right], \\ \frac{\partial^2 \log(q(\boldsymbol{\mu}_t))}{\partial \boldsymbol{\mu}_t \partial {\boldsymbol{\mu}_t}^T} &= -\boldsymbol{\lambda}^p_t - \langle\boldsymbol{\lambda}_t\rangle, \end{aligned}$$

the precision $\boldsymbol{\lambda}_t^*$ and the mean $\boldsymbol{\mu}_t^*$ of $q(\boldsymbol{\mu}_t)$ are obtained as follows:

$$\begin{cases} \boldsymbol{\lambda}_t^* = & \langle \boldsymbol{\lambda}_t \rangle + \boldsymbol{\lambda}_t^p, \\ \boldsymbol{\mu}_t^* = & \boldsymbol{\lambda}_t^{*-1} \left(\langle \boldsymbol{\lambda}_t \rangle \langle \boldsymbol{x}_t \rangle + \boldsymbol{\lambda}_t^p \boldsymbol{\mu}_t^p \right). \end{cases} \tag{B.3}$$

Appendix B.2 $q(\boldsymbol{\lambda}_t)$

The component of the approximate separable distribution corresponding to $\boldsymbol{\lambda}_t$ can be obtained following the same reasoning as above:

$$\begin{aligned} q(\boldsymbol{\lambda}_t) &\propto \exp\langle \log p(\boldsymbol{\alpha}_t | \boldsymbol{z}_t) \rangle_{q(\boldsymbol{x}_t) q(\boldsymbol{\mu}_t)} \\ &\propto \exp\langle \log p(\boldsymbol{z}_t | \boldsymbol{x}_t) + \log p(\boldsymbol{x}_t | \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \\ &\quad + \log p(\boldsymbol{\lambda}_t) + \log q_p(\boldsymbol{\mu}_t) \rangle \\ &\propto p(\boldsymbol{\lambda}_t) \exp\langle \log p(\boldsymbol{x}_t | \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \rangle \\ &\propto \mathcal{W}_{n_x}(\bar{\boldsymbol{V}}, \bar{n}) |\boldsymbol{\lambda}_t|^{\frac{1}{2}} \\ &\quad \exp -\frac{1}{2} \{ tr[\boldsymbol{\lambda}_t \langle (\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \rangle] \} \\ &\propto |\boldsymbol{\lambda}_t|^{\frac{\bar{n}+1-(n_x+1)}{2}} \exp -\frac{1}{2} \Big\{ tr \Big[\boldsymbol{\lambda}_t \Big(\langle \boldsymbol{x}_t \boldsymbol{x}_t^T \rangle - \langle \boldsymbol{x}_t \rangle \langle \boldsymbol{\mu}_t \rangle^T \\ &\quad - \langle \boldsymbol{\mu}_t \rangle \langle \boldsymbol{x}_t \rangle^T + \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle + \bar{\boldsymbol{V}}^{-1} \Big) \Big] \Big\}, \end{aligned} \tag{B.4}$$

which yields a Wishart distribution $\mathcal{W}_{n_x}(\boldsymbol{V}^*, n^*)$ for the precision matrix $\boldsymbol{\lambda}_t$ with the following parameters:

$$\begin{cases} n^* &= \bar{n} + 1, \\ \boldsymbol{V}^* &= \left(\langle \boldsymbol{x}_t \boldsymbol{x}_t^T \rangle - \langle \boldsymbol{x}_t \rangle \langle \boldsymbol{\mu}_t \rangle^T - \langle \boldsymbol{\mu}_t \rangle \langle \boldsymbol{x}_t \rangle^T \right. \\ & \quad \left. + \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle + \bar{\boldsymbol{V}}^{-1} \right)^{-1}. \end{cases} \tag{B.5}$$

Appendix B.3 $q(\boldsymbol{x}_t)$

Finally, the approximate distribution $q(\boldsymbol{x}_t)$ has the following expression:

$$\begin{aligned} q(\boldsymbol{x}_t) &\propto \exp\langle \log p(\boldsymbol{\alpha}_t | \boldsymbol{z}_t) \rangle_{q(\boldsymbol{\mu}_t) q(\boldsymbol{\lambda}_t)} \\ &\propto \exp\langle \log p(\boldsymbol{z}_t | \boldsymbol{x}_t) + \log p(\boldsymbol{x}_t | \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \\ &\quad + \log p(\boldsymbol{\lambda}_t) + \log q_p(\boldsymbol{\mu}_t) \rangle \\ &\propto p(\boldsymbol{z}_t | \boldsymbol{x}_t) \exp\langle \log p(\boldsymbol{x}_t | \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \rangle \\ &\propto p(\boldsymbol{z}_t | \boldsymbol{x}_t) \exp -\frac{1}{2} \{ tr[\langle \boldsymbol{\lambda}_t \rangle \langle (\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \rangle] \} \\ &\propto p(\boldsymbol{z}_t | \boldsymbol{x}_t) \mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle), \end{aligned} \tag{B.6}$$

which does not have a closed form. Therefore, contrary to the cases of the mean $\boldsymbol{\mu}_t$ and the precision $\boldsymbol{\lambda}_t$, in order to compute the expectations relative to the distribution $q(\boldsymbol{x}_t)$, one has to resort to the importance sampling method, where samples are generated according to the Gaussian $\mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle)$ and then weighted according to the likelihood $p(\boldsymbol{z}_t | \boldsymbol{x}_t)$.

References

[1] M. Tubaishat and S. Madria, "Sensor Networks: An Overview," *IEEE Potentials,* vol. 22, no. 2 pp. 20-23, Apr./May 2003.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Analysis of a Prediction-Based Adaptive Mobility Tracking Algorithm," *Proc. Second Int'l Conf. Broadband Networks,* pp. 809-816, 2005.

[3] M.Y.S. Uddin and M.M. Akbar, "Addressing Techniques in Wireless Sensor Networks: A Short Survey," *Proc. Fourth Int'l Conf. Electrical and Computer Eng.,* 2006.

[4] Y. Wang, G. Attebury, and B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," *IEEE Comm. Surveys and Tutorials,* vol. 8, no. 2, pp. 2-23, 2006.

[5] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, "Particle Filtering for Multi-Target Tracking and Sensor Management," *Proc. Fifth Int'l Conf. Information Fusion,* 2002.

[6] *Optimal Filtering,* B.D.O. Anderson and J.B. Moore, eds. Prentice-Hall, 1979.

[7] V. Hasu and H. Koivo, "Decentralized Kalman Filter in Wireless Sensor Networks—Case Studies," *Proc. Int'l Joint Conf. Computer, Information, and Systems Sciences, and Eng.,* 2005.

[8] S.J. Julier and J.K. Unlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. 11th Int'l Symp. Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II (AeroSense),* 1997.

[9] J.H. Kotecha and P.M. Djuric, "Gaussian Particle Filtering," *IEEE Trans. Signal Processing,* vol. 51, no. 10, pp. 2592-2601, Oct. 2003.

[10] *Sequential Monte Carlo Methods in Practice,* A. Doucet, N. de Freitas, and N. Gordon, eds. Springer, 2001.

[11] J.H. Kotecha and P.M. Djuric, "Gaussian Sum Particle Filtering," *IEEE Trans. Signal Processing,* vol. 51, no. 10, pp. 2602-2612, Oct. 2003.

[12] X. Sheng, Y. Hu, and P. Ramanathan, "Distributed Particle Filter with GMM Approximation for Multiple Targets Localization and Tracking in Wireless Sensor Network," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks,* p. 24, 2005.

[13] A.T. Ihler, J.W. Fisher III, and A.S. Willsky, "Particle Filtering under Communications Constraints," *Proc. IEEE Workshop Statistical Signal Processing,* 2005.

[14] H. Snoussi and C. Richard, "Ensemble Learning Online Filtering in Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Comm. Systems,* 2006.

[15] A. Ribeiro, G.B. Giannakis, and S.I. Roumeliotis, "SOI-KF: Distributed Kalman Filtering with Low-Cost Communications Using the Sign of Innovations," *IEEE Trans. Signal Processing,* vol. 54, no. 12, pp. 4782-4795, Dec. 2006.

[16] P.M. Djurić, M. Vemula, and M. Bugallo, "Tracking with Particle Filtering in Tertiary Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* Mar. 2005.

[17] P.M. Djurić, M. Vemula, and M.F. Bugallo, "Target Tracking by Particle Filtering in Binary Sensor Networks," *IEEE Trans. Signal Processing,* vol. 56, no. 6, pp. 2229-2238, June 2008.

[18] A.S. Chhetri, D. Morrell, and A.P. Suppappola, "Energy Efficient Target Tracking in a Sensor Network Using Non-Myopic Sensor Scheduling," *Proc. Seventh Int'l Conf. Information Fusion,* 2005.

[19] J. Vermaak, N.D. Lawrence, and P. Pérez, "Variational Inference for Visual Tracking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* June 2003.

[20] J. Teng, H. Snoussi, and C. Richard, "Binary Variational Filtering for Target Tracking in Sensor Networks," *Proc. IEEE Workshop Statistical Signal Processing,* 2007.

[21] O.E. Barndorff-Nielsen, "Exponentially Decreasing Distributions for the Logarithm of the Particle Size," *Proc. Royal Soc. London. Series A: Math. and Physical Sciences,* vol. 353, pp. 401-419, 1977.

[22] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. ACM MobiCom,* 2003.

[23] P.M. Djurić, M. Vemula, M. Bugallo, and J. Míguez, "Non-Cooperative Localization of Binary Sensors," *Proc. 13th Workshop Statistical Signal Processing,* 2005.

[24] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a Moving Object with a Binary Sensor Network," *Proc. First Int'l Conf. Embedded Networked Sensor Systems,* pp. 150-161, 2003.

[25] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target Tracking with Binary Proximity Sensors: Fundamental Limits, Minimal Descriptions, and Algorithms," *Proc. Fourth Int'l Conf. Embedded Networked Sensor Systems,* pp. 251-264, 2006.

[26] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On Target Tracking with Binary Proximity Sensors," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks,* p. 40, 2005.

[27] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking Multiple Targets Using Binary Proximity Sensors," *Proc. Sixth Int'l Conf. Information Processing in Sensor Networks,* pp. 529-538, 2007.

[28] H. Lin, J. Rushing, S.J. Graves, S. Tanner, and E. Criswell, "Real Time Target Tracking with Binary Sensor Networks and Parallel Computing," *Proc. IEEE Int'l Conf. Granular Computing,* 2006.

[29] K. Mechitov, S. Sundresh, Y. Kwon, and G. Agha, "Cooperative Tracking with Binary-Detection Sensor Networks," *Proc. First ACM Conf. Embedded Networked Sensor Systems,* I.F. Akyildiz, D. Estrin, D.E. Culler, and M.B. Srivastava, eds., pp. 332-333, 2003.

[30] P.M. Djurić, M. Vemula, and M.F. Bugallo, "Signal Processing by Particle Filtering for Binary Sensor Networks," *Proc. IEEE 11th Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop,* 2004.

[31] A. Ribeiro and G.B. Giannakis, "Bandwidth-Constrained Distributed Estimation for Wireless Sensor Networks—Part I: Gaussian Case," *IEEE Trans. Signal Processing,* vol. 54, pp. 1131-1143, 2006.

[32] A. Ribeiro and G.B. Giannakis, "Bandwidth-Constrained Distributed Estimation for Wireless Sensor Networks—Part II: Unknown Probability Density Function," *IEEE Trans. Signal Processing,* vol. 54, pp. 2784-2796, 2006.

[33] M.J. Beal, "Variational Algorithms for Approximate Bayesian Inference," PhD thesis, Univ. of Cambridge, 2003.

[34] S. Pattem, S. Poduri, and B. Krishnamachari, "Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks," *Proc. Int'l Conf. Information Processing in Sensor Networks,* pp. 32-46, 2003.

[35] T. Onel, C. Ersoy, and H. Delic, "Information Content-Based Sensor Selection for Collaborative Target Tracking," *Proc. 14th European Signal Processing Conf.,* 2006.

[36] H. Yang and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks," *Proc. First IEEE Int'l Workshop Sensor Network Protocols and Applications,* 2003.

[37] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing,* vol. 2, pp. 483-502, 2002.

Jing Teng received the BEng degree in electronic information engineering from the Central South University, China, in 2003, and the PhD degree in systems optimization and security from the University of Technology of Troyes, France, in November 2009. Since 2010, she has been a lecturer at the North China Electric Power University. Her research interests include statistical signal processing and its applications in wireless sensor networks.

Hichem Snoussi received a diploma in electrical engineering from the Ecole Supérieure d'Electricité (Supélec), Gif-sur-Yvette, France, in 2000, and the DEA and PhD degrees in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003, respectively. Since 2005, he has been an associate professor at the University of Technology of Troyes, France. Between 2003 and 2004, he was a postdoctoral researcher at IRCCyN, Institut de Recherches en Communications et Cybernétiques de Nantes. He has spent short periods as a visiting scientist at the Brain Science Institute, RIKEN, Japan, and Olin Neuropsychiatry Research Center at the Institute of Living in the US. Since January 2008, he has been leading the research group "Surveillance" of LM2S Laboratory. He is in charge of the regional research program System Security and Safety (S3) of CPER 2007-2013 and the CapSec platform (wireless embedded sensors for security). He is the principal investigator of an ANR-Blanc project (mv-EMD), a CRCA project (new partnership and new technologies), and a GDR-ISIS young researcher project. He is a partner of many ANR projects, GIS, strategic UTT programs. He obtained the National Doctoral and Research Supervising Award PEDR 2008-2012. He is a member of the IEEE.

Cédric Richard received the Dipl-Ing and MS degrees in 1994 and the PhD degree in 1998 from the University of Technology of Compiègne, France, all in electrical and computer engineering. Since September 2009, he has been a full professor at Fizeau Laboratory (UNS, CNRS UMR 6525, Observatoire de la Côte d'Azur), University of Nice Sophia-Antipolis, France. From 1999 to 2003, he was an associate professor at the University of Technology of Troyes (UTT), France. From 2003 to 2009, he was a full professor at the Institut Charles Delaunay (CNRS FRE 2848) at UTT, and the supervisor of a group consisting of 60 researchers and PhD. In Winter 2009, he was a visiting researcher in the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianòpolis, Brazil. His current research interests include statistical signal processing and machine learning. He is the author of more than 100 papers. He was the general chair of the XXIth francophone conference GRETSI on signal and image processing that was held in Troyes, France, in 2007. Since 2005, he is a member of the board of the federative CNRS Research Group ISIS on Information, Signal, Images, and Vision. He is a member of GRETSI association board and the EURASIP society. He serves as an associate editor of the *IEEE Transactions on Signal Processing* since 2006 and the *EURASIP Signal Processing Magazine* since 2009. In 2009, he was nominated as liaison local officer for EURASIP and a member of the Signal Processing Theory and Methods (SPTM) Technical Committee of the IEEE Signal Processing Society. He and Paul Honeine received the Best Paper Award for "solving the pre-image problem in kernel machines: a direct method" at the 2009 IEEE Workshop on Machine Learning for Signal Processing (IEEE MLSP 2009). He is a senior member of the IEEE.