

# Distributed Variational Filtering for Simultaneous Sensor Localization and Target Tracking in Wireless Sensor Networks

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

**Abstract**—The tracking of a moving target in a wireless sensor network (WSN) requires exact knowledge of sensor positions. However, precise information about sensor locations is not always available. Given the observation that a series of measurements are generated in the sensors when the target moves through the network field, we propose an algorithm that exploits these measurements to simultaneously localize the detecting sensors and track the target (SLAT). The main difficulties that are encountered in this problem are the ambiguity of sensor locations, the unrestricted target moving manner, and the extremely constrained resources in WSNs. Therefore, a general state evolution model is employed to describe the dynamical system with neither prior knowledge of the target moving manner nor precise location information of the sensors. The joint posterior distribution of the parameters of interest is updated online by incorporating the incomplete and inaccurate measurements between the target and each of the sensors into a Bayesian filtering framework. A variational approach is adopted in the framework to approximate the filtering distribution, thus minimizing the intercluster communication and the error propagation. By executing the algorithm on a fully distributed cluster scheme, energy and bandwidth consumption in the network are dramatically reduced, compared with a centralized approach. Experiments on an SLAT problem validate the effectiveness of the proposed algorithm in terms of tracking accuracy, localization precision, energy consumption, and execution time.

**Index Terms**—Bayesian method, filtering algorithm, simultaneous localization and tracking, wireless sensor networks.

Manuscript received December 5, 2011; accepted February 13, 2012. Date of publication March 10, 2012; date of current version June 12, 2012. This work was supported in part by the Fundamental Research Funds for the Central Universities, the China Scholarship Council–French University of Technology Applied Science Group Program, and the CapSec Program funded by the Contrat de Projets Etat-Région Champagne-Ardenne 2007–2013. The review of this paper was coordinated by Prof. A. Boukerche.

J. Teng is with the School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China (e-mail: jing.teng@ncepu.edu.cn).

H. Snoussi is with the University of Technology of Troyes, 10010 Troyes Cedex, France.

C. Richard is with the University of Nice Sophia Antipolis, 06108 Nice Cedex 02, France.

R. Zhou is with the National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China, and also with North China Electric Power University, Beijing 102206, China.

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

Digital Object Identifier 10.1109/TVT.2012.2190631

## I. INTRODUCTION

A WIRELESS sensor network (WSN) is one of the most promising emerging technologies in recent years [1]–[3]. It establishes the parameterization of physical phenomena and enables the processing of the sensed information for inference and estimation [4], [5]. In these applications, the WSN perceives the environment data by sensing different physical properties, e.g., signal strength (for target tracking), pressure, temperature, and humidity. However, the sensed data are meaningless without supplementary sensor location information, particularly when used for tracking or other correlation purposes [6]–[9]. It is generally known that a moving target can be better tracked if the sensors that have detected it are exactly localized. On the other hand, the estimation of these detecting sensor locations can also be refined based on their observations of the moving target [10]. That is, sensor localization and target tracking are complementary to each other. Simultaneous localization and tracking (SLAT) thus combines the two typical problems of WSNs together, leading to a practically relevant solution. A constraint structure is built in SLAT to shrink the uncertainties of sensor positions and to estimate the target temporal state, because each observation between a detecting sensor and the target adds geometric constraints on both of them and leads to improvements in their estimations over time. The attractive solution of SLAT poses no restriction on the moving manner of the mobile target, whose temporal position is estimated in the presence of sensor localization errors. With respect to the sensors, no additional hardware configuration is required, fitting well with the low-cost budget of WSNs. Furthermore, SLAT allows a continuous refinement of sensor localization, even during the tracking phase.

A general situation of SLAT is considered in this paper. The resident location of sensors is randomly distributed around their deployment points due to spatially varying environment factors and deployment errors. After the deployment phase, sensors exchange information with the sensors within their communication ranges  $r_c$ . A set of observation is thus collected and used to prelocalize themselves, improving the coarse *a priori* position information. The SLAT procedure begins when a mobile target  $x$  enters into the WSN, which arbitrarily moves through the environment, with no constraint on its trajectory or velocity. At the sampling instant  $t$ , only the sensors that detect the presence of the target  $x_t$  form an activated cluster  $S_t$  for further signal processing. The temporal observation between the target and each activated sensor is incorporated to

update the target temporal estimation. With respect to the activated sensor, its temporal observation of the target is combined with the static observation set stored during the prelocalization phase together to refine its location estimation. Therefore, the sensors that have detected the target are localized together with the tracking of the target. In particular, sensors that are located in the high-traffic area could be relocated, which shrinks the corresponding sensor localization error and refines the target tracking precision in return.

The proposed algorithm is called distributed variational filtering for simultaneous localization and tracking (DVaSLAT), which inherits many desirable properties from the Bayesian framework [11]. A general state evolution model is employed to describe the target  $x_t$  and the activated sensors  $S_t$  using a joint variable  $X_t$ . Because the algorithm is executed in a distributed cluster scheme, a variational approach is employed to approximate the joint state during the measurement incorporation phase, avoiding description complexity and unnecessary intercluster communication. In addition, the error propagation problem, which is always unavoidable in other approximation methods, is terminated in our case, because the approximation phase is jointly performed with the update phase of the filtering distribution.

The remainder of this paper is organized as follows. In Section II, we summarize the related work. In Section III, the SLAT problem is defined. The DVaSLAT algorithm is described in detail in Section IV. The performance of the proposed DVaSLAT algorithm is evaluated through the simulations in Section V. Section VI concludes this paper.

## II. RELATED WORK

Target tracking is currently the principal research problem in WSNs wherein the particle filter (PF) has been used [11]. Ahmed *et al.* provide a comprehensive analysis on the effect of various design parameters (such as the number of deployed nodes, the number of generated particles, and sampling interval) and calibration parameters (such as the gain, path-loss factor, noise variations, and nonlinearity constant) on the performance of the PF-based tracking system [12]. Their simulation consists of two parts. The first part is performed in the network simulator NS-2, where all the measurements that were recorded by sensor nodes are forwarded to the centrally located base station over multihop. Once all the data are available, the base station runs the PF code offline in MATLAB. On the contrary, the proposed DVaSLAT algorithm is executed online in a distributed cluster-based manner in MATLAB, where the communication in an activated cluster is within one hop. In addition, the simulated target motion model in [12] is linear, whereas we adopt the random walk mobility (RWM) model [13], which puts no constraint on the target velocity or moving direction.

With respect to the sensor localization problem, in this paper, sensor nodes do not need assistance from other positioning systems such as the Global Positioning System (GPS). Therefore, the sensor deployment phase always has a direct influence on the accuracy of localization. Fang *et al.* [14] assume that groups of sensors are deployed at regular de-

ployment points, which are different from the final resident locations of sensors, and the distribution of sensors around these points follows a Gaussian distribution. Because sensors can discover their locations by observing the group memberships of their neighbors, the location discovery problem was modeled as a statistical estimation problem, and the maximum-likelihood estimation method was used in [14] to estimate the location. In [15], the beaconless location discovery scheme in [14] over a nonflat terrain was further modified and extended. We adopt the similar assumption in the prelocalization phase of our algorithm, wherein sensor locations are estimated using PF.

As aforementioned, target-tracking algorithms are often based on the geographical information of sensors provided by a localization algorithm. Souza *et al.* assess the performance of classical target tracking algorithms under the impact of errors introduced by sensor localization algorithms in [16]. However, in their case, sensor localization and target tracking are two separate phases, and the evaluated target maneuver is restricted to a uniform motion. Although the cross-layer design of localization and tracking algorithms remains fairly unexplored in [16], the authors also believe that this design may lead to improved solutions for both problems. To the best of our knowledge, it is the first time that Taylor *et al.* defined the two typical problems of WSNs together as SLAT in [17]. The notation of SLAT resembles the related and well-studied simultaneous localization and mapping (SLAM) problem in the robotics and artificial-intelligence community [18]. However, the target to be located in SLAM is a mobile robot [19], [20], whose control input is known *a priori*. On the contrary, SLAT does not require any constraint on the moving manner of the target. In addition, the environment is typically idealized in SLAM so that it consists of an unknown number of stationary “landmarks,” which cannot detect the robot or other landmarks to localize themselves. In other words, SLAM can only be centralized and executed in the robot, which incrementally estimates its temporal location and the landmarks, from the noisy and incomplete observations between them. On the other hand, in SLAT, both the static observations between neighboring sensors and the temporal observations between the detecting sensors and the target can be incorporated to simultaneously estimate sensor locations and track the target. In addition, the observations that were incorporated into the classical SLAM algorithms, i.e., extended Kalman filter for simultaneous localization and mapping (EKF-SLAM) [20], FastSLAM 1.0 [21], and FastSLAM 2.0 [22], are range-bearing measurements, which require a special antenna configuration and omnidirectional signals to determine the bearing information, which is not practical for low-cost WSNs. Therefore, we adopt the received signal strength indicator (RSSI) technology, which requires no additional hardware configuration [6], [23]. In fact, this paper is most related to the algorithm called LaSLAT, which was proposed in [17]. LaSLAT does not require any constraint on the trajectory of the target, where consecutive measurements between sensors and the target are packed up to update the joint distribution. The application of Laplace’s method approximates the joint posterior distribution with a Gaussian, leading to a reduction of the belief propagation. The results of LaSLAT are presented

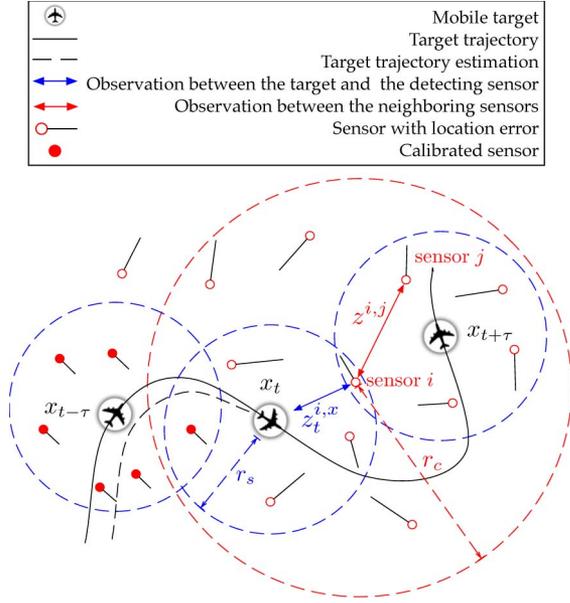


Fig. 1. SLAT scenary.

by a centralized implementation using a network of Cricket sensors. Compared with LaSLAT, our algorithm is executed in a distributed cluster scheme. The variational method is employed instead of the Laplace method to approximate the joint state during the measurement incorporation phase, avoiding description complexity and unnecessary communication. Furthermore, the error propagation problem, which is always unavoidable in other approximation methods, is terminated in our case, because the approximation phase is jointly performed with the update phase of the filtering distribution. In addition, consecutive measurements between the target and sensors are batched to be incorporated into LaSLAT. Strictly speaking, LaSLAT is therefore not a real-time algorithm. Because a larger batch size provides more information, the localization precision and the convergence rate of the LaSLAT algorithm are accordingly improved, which are also justified in [17]. On the contrary, our algorithm is executed online to incorporate the timely observations; thus, the time complexity is one of our utmost concerns. Accordingly, we label our algorithm DVaSLAT.

### III. PROBLEM STATEMENT

In this paper, we assume that sensors are randomly distributed around their deployment points, as shown in Fig. 1, where the lengths of the “node tails” denote the corresponding offsets. This initial assignment results in a node layout that resembles an unfolded and scaled version of the actual deployment, roughly preserving the topological ordering of nodes. The final resident location  $s^i$  of sensor  $i$  is assumed to be Gaussian distributed around its deployment point  $\bar{s}^i$  with precision  $\eta^i$ , i.e.,  $s^i \sim \mathcal{N}(\bar{s}^i, \eta^i)$ . After the deployment, sensor  $i$  exchanges information with its neighboring sensors in the communication range  $r_c$ , denoted by the red dashed circle in Fig. 1. To improve the coarse *a priori* information on the

sensor location, a prelocalization phase is locally launched by incorporating only these measurements, i.e.,

$$p(\hat{s}^i | z^{i,s}) \propto \mathcal{N}(\bar{s}^i, \eta^i) \prod_{\|s^i - s^j\| \leq r_c} p(z^{i,j} | s^i). \quad (1)$$

Thus, according to the prior distribution  $s^i \sim \mathcal{N}(\bar{s}^i, \eta^i)$ , the estimation of sensor  $i$  is refined by incorporating the observation  $z^{i,s} = \{z^{i,j}\}_{\|s^i - s^j\| \leq r_c}$ . After prelocalizing all the sensors, much more precise information on their locations is provided, which is incorporated as the *a priori* information  $\hat{S} = \{\hat{s}^i\}_{i=1}^{N_s}$  for SLAT, where  $N_s$  is the total number of sensors in the WSN.

To minimize energy and bandwidth consumption, the SLAT procedure is distributively executed on a cluster base. Once an intrusion in the WSN is identified, a cluster of sensors  $\mathcal{S}_t$  around the phenomenon of interest is activated. Only the sensors that detect the presence of the target  $x_t$ , i.e.,  $\|s^i - x_t\| \leq r_s$ , form the activated cluster  $\mathcal{S}_t = \{s^i\}_{\|s^i - x_t\| \leq r_s}$ . As shown in Fig. 1, sensors that detect the target  $x_t$  are in the blue dotted circle with the center  $x_t$  and the radius  $r_s$ . The size of a cluster is determined by the relationship between the communication range  $r_c$  and the sensing range  $r_s$  [24]. For localization accuracy and energy efficiency, the communication range is defined as twice the sensing range ( $r_c = 2r_s$ ), which guarantees that only one cluster is formed at each instant and the communication in the activated cluster is within one hop. The activated sensors broadcast their residual energy level in the cluster. The sensor with the maximum residual energy is elected as the cluster head (CH) to take charge of signal processing. The other clustered detecting sensors then transfer their observations to the CH. These observations consist of the following two parts: 1) the temporal observation among the clustering sensors and the target, i.e.,  $z_t^{s,x} = \{z_t^{i,x}\}_{\forall s^i \in \mathcal{S}_t}$ , which is incorporated to update the target temporal estimation, and 2) the static observation that is stored during the prelocalization phase, which is combined with the temporal observation  $Z_t = \{\{z^{i,s}\}_{i=1}^{m_t}, z_t^{s,x}\}$ , where  $m_t$  is the number of activated clustering sensors, to further refine their location estimations. Therefore, estimations of the target and the detecting sensor locations are simultaneously updated in the CH based on these observations.

The SLAT problem can then be defined as follows.

*Definition of the SLAT Problem:* Given a mobile target  $x$  and a WSN with rough location information  $\hat{S}$ , for the clustering sensors that have detected the target  $x$ , we want to simultaneously find their location  $\mathcal{S}_t$  and the temporal position  $x_t$  of the target, where the estimation results are designated as  $\langle \mathcal{S}_t \rangle$  and  $\langle x_t \rangle$ , respectively.

### IV. DISTRIBUTED VARIATIONAL FILTERING FOR SIMULTANEOUS LOCALIZATION AND TRACKING ALGORITHM

The DVaSLAT algorithm adopts the classical Bayesian framework to estimate the unknown state  $X_t = \{x_t, \mathcal{S}_t\}$  over time by using the incoming observations  $Z_t$ . In a probabilistic approach, this procedure means recursively computing the

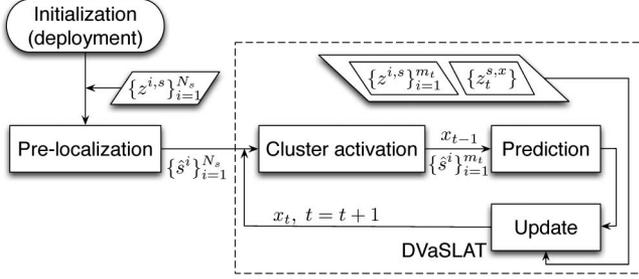


Fig. 2. Flowchart of the SLAT procedure.

posterior distribution  $p(\mathbf{X}_t | \mathbf{Z}_{1:t})$ . A standard two-step Bayesian recursive solution is described as follows.

- Prediction

$$p(\mathbf{X}_t | \mathbf{Z}_{1:t-1}) = \int p(\mathbf{X}_t | \mathbf{X}_{t-1}) p(\mathbf{X}_{t-1} | \mathbf{Z}_{1:t-1}) d\mathbf{X}_{t-1}. \quad (2)$$

- Update

$$p(\mathbf{X}_t | \mathbf{Z}_{1:t}) = \frac{p(\mathbf{Z}_t | \mathbf{X}_t) p(\mathbf{X}_t | \mathbf{Z}_{1:t-1})}{p(\mathbf{Z}_t | \mathbf{Z}_{1:t-1})} \quad (3)$$

where

$$p(\mathbf{Z}_t | \mathbf{Z}_{1:t-1}) = \int p(\mathbf{Z}_t | \mathbf{X}_t) p(\mathbf{X}_t | \mathbf{Z}_{1:t-1}) d\mathbf{X}_t.$$

The Bayesian filtering framework of the DVaSLAT algorithm is composed of the following two distinct phases: 1) prediction and 2) update. The prediction phase uses the state estimate from the previous sampling instant to produce an estimate of the state at the current instant according to (2). In the update phase of (3), measurement information at the current instant is used to refine this prediction to arrive at a new and, hopefully, more accurate state estimate. The computation requires the definition of a state evolution model  $p(\mathbf{X}_t | \mathbf{X}_{1:t-1})$  and an observation model  $p(\mathbf{Z}_t | \mathbf{X}_t)$ . In particular, the state evolution is generally assumed to be a Markov process, i.e.,  $p(\mathbf{X}_t | \mathbf{X}_{1:t-1}) = p(\mathbf{X}_t | \mathbf{X}_{t-1})$ . The SLAT procedure works in a recursive way, which is illustrated by the flowchart in Fig. 2.

#### A. General State Evolution Model

Because the mobile target arbitrarily travels in the sensor field, instead of a traditional kinematic parameter model [25]–[27], we employ a general state evolution model [28]–[30], which is more adaptive to practical situations and has no restriction on the velocity or moving direction of the target. As aforementioned, at instant  $t$ , the joint hidden state to be estimated  $\mathbf{X}_t$  contains the target position  $\mathbf{x}_t$  and a set of activated sensor locations  $\mathbf{S}_t = \{\mathbf{s}_t^i\}_{i=1}^{m_t}$ . Taking sensor  $i$  for example,  $\mathbf{s}_t^i$  is assumed to be a Gaussian variable whose expectation is its latest estimate value  $\hat{\mathbf{s}}^i$ , and the precision matrix  $\eta^i$  indicates its position offset due to the deployment error, estimate uncertainty, or other spatial factors. The target  $\mathbf{x}_t$  is assumed to follow an extended Gaussian model, where the expectation

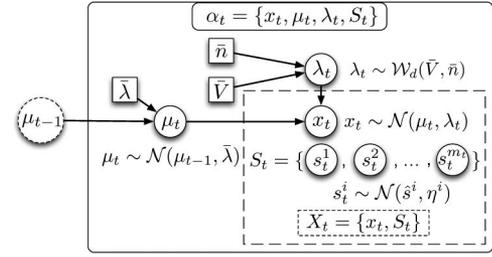


Fig. 3. Graphical model of the general state evolution model. Nodes that are denoted by circles correspond to hidden state variables, whereas nodes that are denoted by squares correspond to parameters of the model.

$\mu_t$  and the precision matrix  $\lambda_t$  are both random, following a Gaussian and a Wishart distribution, respectively, i.e.,

$$\begin{cases} \mathbf{s}_t^i \sim \mathcal{N}(\hat{\mathbf{s}}^i, \eta^i) \quad \forall \mathbf{s}_t^i \in \mathbf{S}_t \\ \mathbf{x}_t \sim \mathcal{N}(\mu_t, \lambda_t) \\ \mu_t \sim \mathcal{N}(\mu_{t-1}, \bar{\lambda}) \\ \lambda_t \sim \mathcal{W}_d(\bar{\mathbf{V}}, \bar{n}). \end{cases} \quad (4)$$

The precision matrix  $\lambda_t$  reflects the uncertainty of the target location estimation at instant  $t$ , which is modeled by a  $d$ -D Wishart distribution ( $d$  is equal to the dimension of the target state, and  $d = 2$  in this paper), with  $\bar{\mathbf{V}}$  and  $\bar{n}$  denoting, respectively, its precision matrix and degree of freedom. Note that  $\bar{\cdot}$  denotes the initial fixed parameter. The general state evolution model is demonstrated in Fig. 3. Assuming a random mean and covariance for the state  $\mathbf{x}_t$  leads to a probability distribution that covers a wide range of tail behaviors, which allows discrete jumps in the target trajectory. In fact, the marginal state distribution is obtained by integrating over the mean and precision matrix as

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}) = \int \int \mathcal{N}(\mathbf{x}_t | \mu_t, \lambda_t) p(\mu_t, \lambda_t | \mathbf{x}_{t-1}) d\mu_t d\lambda_t$$

where the integration with respect to the precision matrix leads to the known class of scale mixture distributions introduced by Barndorff-Nielsen [31]. A low value of the degree of freedom  $\bar{n}$  reflects the heavy tails of the marginal distribution  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ .

#### B. Observation Model

Localization algorithms for WSNs mainly rely on the range measurement, bearing measurement, neighborhood proximity, or hop count method. With regard to the budget of WSNs, the RSSI technology, which has been proposed for hardware-constrained systems [23], is employed in this paper. The RSSI determines the distance between a receiver  $P_r$  and a transmitter  $P_t$  based on the knowledge of a path-loss model as

$$\nu^r(P_t) = \Psi_0 - 10\zeta \log \frac{\|P_r - P_t\|}{d_0}. \quad (5)$$

The denotations in the formulation are, respectively, the reference distance  $d_0$ , the known received signal power  $\Psi_0$  (in decibel-milliwatts) at  $d_0$ , and the known path-loss distance exponent  $\zeta$ , which takes a value in the range [2], [4] ( $\zeta = 2$  for propagation in the free space,  $\zeta = 4$  for relatively lossy environments, and for the case of full specular reflection from

the earth's surface [32]). We can notice in (5) that the received signal power  $\nu^r(P_t)$  is theoretically a one-to-one mapping to the distance  $\|P_r - P_t\|$  that is traveled by the signal. In wireless communication, the multipath reflections, non-line-of-sight conditions, and other shadowing effects always lead to erroneous distance estimates based on the RSSI. Therefore, we introduce a white Gaussian error  $\epsilon_y^i \sim \mathcal{N}(0, \sigma_y^{-2})$  to model the sensed observation  $y$  at sensor  $i$  as

$$\begin{aligned} y_t^{i,x} &= \nu^i(\mathbf{x}_t) + \epsilon_y^i \\ &\sim \mathcal{N}(\nu^i(\mathbf{x}_t), \sigma_y^{-2}) \end{aligned}$$

where

$$\nu^i(\mathbf{x}_t) = \Psi_0 - 10\zeta \log \frac{\|\mathbf{s}^i - \mathbf{x}_t\|}{d_0}.$$

Similarly

$$y_t^{i,j} \sim \mathcal{N}(\nu^i(\mathbf{s}^j), \sigma_y^{-2})$$

where

$$\nu^i(\mathbf{s}^j) = \Psi_0 - 10\zeta \log \frac{\|\mathbf{s}^i - \mathbf{s}^j\|}{d_0}. \quad (6)$$

Due to the noisy wireless links, the signal  $z$  that was received at the CH is assumed to be corrupted by a normally distributed error  $\epsilon_z^i \sim \mathcal{N}(0, \sigma_z^{-2})$ . The realistic measurements that are received at the CH from sensor  $i$  are thus formulated as follows:

$$z_t^{i,x} = \begin{cases} \beta^i y_t^{i,x} + \epsilon_z^i, & \text{if } y_t^{i,x} \geq \gamma_s \\ \epsilon_z^i, & \text{otherwise.} \end{cases}$$

Similarly

$$z_t^{i,j} = \begin{cases} \beta^i y_t^{i,j} + \epsilon_z^i, & \text{if } y_t^{i,j} \geq \gamma_c \\ \epsilon_z^i, & \text{otherwise} \end{cases} \quad (7)$$

where  $\beta^i$  is the attenuation coefficient that is associated with sensor  $i$ . The signal detection threshold  $\gamma_s$  and the communication threshold  $\gamma_c$  correspond to the sensor detection range  $r_s$  and communication range  $r_c$ , respectively, i.e.,  $\gamma_s = \Psi_0 - 10\zeta \log(r_s/d_0)$  and  $\gamma_c = \Psi_0 - 10\zeta \log(r_c/d_0)$ .

The Bayesian filtering framework of SLAT requires the construction of an observation model  $p(\mathbf{Z}_t | \mathbf{X}_t)$ . To track the target  $\mathbf{x}_t$ , the available observations at the activated CH are denoted by  $\mathbf{z}_t^{\mathbf{s}, \mathbf{x}} = \{z_t^{i,x}\}_{i=1}^{m_t}$ . Assuming that the noise samples  $\epsilon_z^i$  are independently distributed, we have

$$\begin{aligned} p(\mathbf{z}_t^{\mathbf{s}, \mathbf{x}} | \mathbf{x}_t) &= \prod_{i=1}^{m_t} \left[ \mathcal{N}(z_t^{i,x} | \beta^i y_t^{i,x}, \sigma_z^{-2}) P(y_t^{i,x} \geq \gamma_s) \right. \\ &\quad \left. + \mathcal{N}(z_t^{i,x} | 0, \sigma_z^{-2}) P(y_t^{i,x} < \gamma_s) \right] \end{aligned} \quad (8)$$

where

$$\begin{aligned} P(y_t^{i,x} \geq \gamma_s) &= \int_{\gamma_s}^{\infty} \mathcal{N}(y_t^{i,x} | \nu^i(\mathbf{x}_t), \sigma_y^{-2}) dy_t^{i,x} \\ P(y_t^{i,x} < \gamma_s) &= 1 - P(y_t^{i,x} \geq \gamma_s). \end{aligned}$$

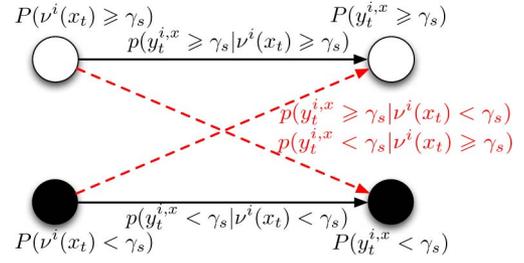


Fig. 4. Probability model of the sensed observation  $y_t^{i,x}$  with false alarms, where the dashed lines with arrows denote the false alarms.

With respect to the cluster sensors, the observations used for localizing sensor  $i$  at instant  $t$  are formulated as

$$\begin{aligned} p(\mathbf{z}_t^i | \mathbf{s}_t^i) &= p(z_t^{i,x} | \mathbf{s}^i, \mathbf{x}_t) p(\mathbf{z}^{i,\mathbf{s}} | \mathbf{s}^i) \\ &= p(z_t^{i,x} | \mathbf{s}^i, \mathbf{x}_t) \prod_{j \neq i}^{m_t-1} p(z^{i,j} | \mathbf{s}^i, \mathbf{s}^j) \\ &= \left[ \mathcal{N}(z_t^{i,x} | \beta^i y_t^{i,x}, \sigma_z^{-2}) P(y_t^{i,x} \geq \gamma_s) \right. \\ &\quad \left. + \mathcal{N}(z_t^{i,x} | 0, \sigma_z^{-2}) P(y_t^{i,x} < \gamma_s) \right] \\ &\quad \times \prod_{j \neq i}^{m_t-1} \left[ \mathcal{N}(z^{i,j} | \beta^i y^{i,j}, \sigma_z^{-2}) P(y^{i,j} \geq \gamma_c) \right. \\ &\quad \left. + \mathcal{N}(z^{i,j} | 0, \sigma_z^{-2}) P(y^{i,j} < \gamma_c) \right]. \end{aligned} \quad (9)$$

In particular,  $\mathbf{z}^{i,\mathbf{s}}$  denotes the observations between the neighboring sensors, which are collected during the prelocalization phase and are stored in sensor  $i$  for further use. At sampling instant  $t$ , the temporal observation of the target  $z_t^{i,x}$  is incorporated together with  $\mathbf{z}^{i,\mathbf{s}}$  to help refine the localization of sensor  $i$ .

One important problem that is introduced by the definition of the observation model is the false alarm. We can notice in (6) that the mapping between  $\nu^i(\mathbf{x}_t)$  and the sensor observation  $y_t^{i,x}$  is not deterministic due to the shadowing effect of  $\epsilon_y^i$ . That is, if  $\nu^i(\mathbf{x}_t)$ , the one-to-one mapping to the true distance  $\|\mathbf{s}^i - \mathbf{x}_t\|$ , is greater than the threshold  $\gamma_s$ , the observed measurement  $y_t^{i,x}$  is not necessarily greater than  $\gamma_s$ . In fact,  $P(y_t^{i,x} \geq \gamma_s)$  can also be formulated by  $P(y_t^{i,x} \geq \gamma_s) = [p(y_t^{i,x} \geq \gamma_s | \nu^i(\mathbf{x}_t) \geq \gamma_s) P(\nu^i(\mathbf{x}_t) \geq \gamma_s) + p(y_t^{i,x} \geq \gamma_s | \nu^i(\mathbf{x}_t) < \gamma_s) P(\nu^i(\mathbf{x}_t) < \gamma_s)]$ , as shown in Fig. 4. According to (8), the probability of false alarm  $p(y_t^{i,x} \geq \gamma_s | \nu^i(\mathbf{x}_t) < \gamma_s)$  has naturally been incorporated during the integral. Similarly, the symmetric probability of false alarm  $p(y_t^{i,x} < \gamma_s | \nu^i(\mathbf{x}_t) \geq \gamma_s)$  is incorporated in the calculation of  $P(y_t^{i,x} < \gamma_s)$ . The same holds true for the calculation of  $P(y^{i,j})$ .

### C. Update

In a distributed context, the filtering distribution of the target  $p(\mathbf{x}_t | \mathbf{Z}_{1:t})$  needs to be transferred for future use, whereas the estimations of the detecting sensors are updated and locally stored. With regard to energy and bandwidth efficiency, a

variational approach simultaneously updates and compresses the filtering distribution of the target to a single Gaussian distribution between successive clusters in a consistent manner [29]. Thus, distributed signal processing is effectively achieved. The following section introduces the details of the variational approach compared with the traditional PF method.

1) *PF Method*: Given the general state evolution model described in Section IV-A, the joint hidden state has been extended to  $\alpha_t = \{x_t, \mu_t, \lambda_t, S_t\}$ . According to the Bayesian filtering framework, the distribution of interest takes the form of a posterior distribution  $p(\alpha_t | Z_{1:t})$ .

The nonlinear and non-Gaussian aspects of the state evolution distributions in (4) lead to intractable integrals when calculating the posterior distribution in (3). Generally, we can resort to a Monte Carlo procedure to approximate the joint posterior distribution  $p(\alpha_{0:t} | Z_{1:t})$  by  $N$  random independent and identically distributed samples, i.e., particles  $\{\alpha_{0:t}^{(k)}\}_{k=1}^N$ . We have

$$\hat{p}_N(\alpha_{0:t} | Z_{1:t}) = \frac{\sum_{k=1}^N w_t^{(k)} \delta_{\alpha_{0:t}^{(k)}}(\alpha_{0:t})}{\sum_{k=1}^N w_t^{(k)}}$$

where  $\delta_{\alpha_{0:t}^{(k)}}(\alpha_{0:t})$  denotes the Dirac delta function of  $\alpha_{0:t}$  located at  $\alpha_{0:t}^{(k)}$ . Based on the same set of particles, the marginal posterior probability  $p(\alpha_t | Z_{1:t})$  can also be approximated as  $\hat{p}_N(\alpha_t | Z_{1:t}) = \sum_{k=1}^N w_t^{(k)} \delta_{\alpha_t^{(k)}}(\alpha_t) / \sum_{k=1}^N w_t^{(k)}$ . In the Bayesian importance sampling (IS) method, the particles  $\{\alpha_{0:t}^{(k)}\}_{k=1}^N$  are sampled according to a proposal distribution  $\pi(\alpha_{0:t} | Z_{1:t})$ , and  $\{w_t^{(k)}\}_{k=1}^N$  are the corresponding importance weights. We have

$$w_t^{(k)} \propto \frac{p(Z_{1:t} | \alpha_{0:t}^{(k)}) p(\alpha_{0:t}^{(k)})}{\pi(\alpha_{0:t}^{(k)} | Z_{1:t})}.$$

PF consists of propagating the trajectories  $\{\alpha_{0:t}^{(k)}\}_{k=1}^N$  in time without modifying the past simulated particles. Consequently, the class of the proposal distributions has the following form:

$$\pi(\alpha_{0:t} | Z_{1:t}) = \pi(\alpha_{0:t-1} | Z_{1:t-1}) \pi(\alpha_t | \alpha_{0:t-1}, Z_{1:t}).$$

The importance weights are then recursively computed in time as

$$w_t^{(k)} \propto w_{t-1}^{(k)} \frac{p(z_t | \alpha_t^{(k)}) p(\alpha_t^{(k)} | \alpha_{0:t-1}^{(k)})}{\pi(\alpha_t^{(k)} | \alpha_{0:t-1}^{(k)}, Z_{1:t})}.$$

The optimal choice of the proposal function is  $p(\alpha_t | \alpha_{t-1}, z_t)$  [33], which conditionally minimizes the variance of the importance weights upon the simulated trajectory  $\alpha_{0:t-1}^{(k)}$  and the observations  $Z_{1:t}$ . However, considering the Markov nonlinear state-space model defined in (4), we can simply adopt the transition prior  $p(\alpha_t | \alpha_{t-1})$  as the proposal distribution, i.e.,

$$\pi(\alpha_t | \alpha_{0:t-1}, Z_{1:t}) = p(\alpha_t | \alpha_{t-1})$$

in which case, the weights are updated according to the likelihood function as

$$w_t^{(k)} \propto w_{t-1}^{(k)} p(z_t | \alpha_t^{(k)}).$$

The PF method suffers from two major drawbacks. First, an efficient implementation requires the ability to sample from  $p(\alpha_t | \alpha_{t-1})$ . Second, propagating such a huge amount of particles and their corresponding weights makes the communication burden of WSN even heavier. In addition, a resampling step is involved in the classical PF to avoid the collapse of the Monte Carlo approximation, which dramatically increases the computational complexity [34].

2) *Variational Approach*: Instead of the point-mass distribution  $\hat{p}_N(\alpha_t | Z_{1:t})$ , a variational approach is employed here to approximate the posterior probability  $p(\alpha_t | Z_{1:t})$  by a separable distribution  $q(\alpha_t)$ , which minimizes the Kullback–Leibler (KL) divergence error as

$$D_{\text{KL}}(q||p) = \int q(\alpha_t) \log \frac{q(\alpha_t)}{p(\alpha_t | Z_{1:t})} (d\alpha_t) \quad (10)$$

where

$$q(\alpha_t) = \prod_i q(\alpha_t^i) = q(x_t) q(\mu_t) q(\lambda_t) q(S_t)$$

$$q(S_t) = \prod_{i=1}^{m_t} q(s_t^i).$$

To minimize the KL divergence subject to the constraint  $\int q(\alpha_t) d\alpha_t = \prod_i \int q(\alpha_t^i) d\alpha_t^i = 1$ , the Lagrange multiplier method is used, i.e.,

$$D_{\text{KL}}(q||p) = \int \prod_i q(\alpha_t^i) \left[ \sum_i \log q(\alpha_t^i) - \log p(\alpha_t | Z_{1:t}) \right] d\alpha_t$$

differentiating it with respect to  $q(\alpha_t^i)$  as

$$\frac{\partial D_{\text{KL}}(q||p)}{\partial q(\alpha_t^i)} = \log q(\alpha_t^i) - \langle \log p(Z_{1:t}, \alpha_t) \rangle_{\prod_{j \neq i} q(\alpha_t^j)} + 1 + \lambda_i$$

where  $\lambda_i$  is a Lagrange multiplier that is introduced to ensure that  $q(\alpha_t^i)$  is normalized. The approximate distribution is thus yielded [35] as follows:

$$q(\alpha_t^i) \propto \exp \langle \log p(Z_{1:t}, \alpha_t) \rangle_{\prod_{j \neq i} q(\alpha_t^j)} \quad (11)$$

where  $\langle \cdot \rangle_{q(\alpha_t^j)}$  denotes the expectation operator relative to the distribution  $q(\alpha_t^j)$ . Therefore, these dependent parameters can jointly and iteratively be updated. According to the variational algorithm, the filtering distribution at time  $t-1$  is approximated by the separable distribution, i.e.,  $\hat{p}(\alpha_{t-1} | Z_{1:t-1}) = q(\alpha_{t-1})$ , the filtering distribution at time  $t$  is thus deduced as [30]

$$\hat{p}(\alpha_t | Z_{1:t}) = \frac{p(z_t | \alpha_t) \int p(\alpha_t | \alpha_{t-1}) q(\alpha_{t-1}) d\alpha_{t-1}}{p(z_t | Z_{1:t-1})}$$

$$\propto p(z_t | x_t, S_t) p(S_t | S_{t-1}) p(x_t | \mu_t, \lambda_t) p(\lambda_t) q_p(\mu_t) \quad (12)$$

with  $q_p(\mu_t) = \int p(\mu_t | \mu_{t-1}) q(\mu_{t-1}) d\mu_{t-1}$ .

Therefore, through a simple integral with respect to  $\boldsymbol{\mu}_{t-1}$ , the filtering distribution  $p(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t})$  can sequentially be updated. Considering the general state evolution model proposed in (4), the evolution of  $\boldsymbol{\mu}_t$  is Gaussian, i.e.,  $p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}})$ . Defining  $q(\boldsymbol{\mu}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_{t-1}^*, \boldsymbol{\lambda}_{t-1}^*)$ ,  $q_p(\boldsymbol{\mu}_t)$  is also Gaussian [30], with the following parameters:

$$q_p(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}_t^p, \boldsymbol{\lambda}_t^p) \quad (13)$$

where

$$\boldsymbol{\mu}_t^p = \boldsymbol{\mu}_{t-1}^* \quad \text{and} \quad \boldsymbol{\lambda}_t^p = \left( \boldsymbol{\lambda}_{t-1}^{*-1} + \bar{\boldsymbol{\lambda}}^{-1} \right)^{-1}.$$

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  $p(\boldsymbol{\alpha}_t|\mathbf{z}_{1:t})$  are jointly performed, yielding a natural and adaptive compression [29]. According to (11) and taking into account (12) and (13), variational calculus leads to closed-form expressions of  $q(\boldsymbol{\mu}_t)$  and  $q(\boldsymbol{\lambda}_t)$  as [30]

$$\begin{cases} q(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}_t^*, \boldsymbol{\lambda}_t^*) \\ q(\boldsymbol{\lambda}_t) = \mathcal{W}_d(\mathbf{V}_t^*, n^*) \end{cases}$$

where the parameters are iteratively updated until convergence according to the following scheme [28]:

$$\begin{cases} \boldsymbol{\mu}_t^* = \boldsymbol{\lambda}_t^{*-1} (\langle \boldsymbol{\lambda}_t \rangle \langle \mathbf{x}_t \rangle + \boldsymbol{\lambda}_t^p \boldsymbol{\mu}_t^p) \\ \boldsymbol{\lambda}_t^* = \langle \boldsymbol{\lambda}_t \rangle + \boldsymbol{\lambda}_t^p \\ n^* = \bar{n} + 1 \\ \mathbf{V}_t^* = \left( \langle \mathbf{x}_t \mathbf{x}_t^T \rangle - \langle \mathbf{x}_t \rangle \langle \boldsymbol{\mu}_t \rangle^T - \langle \boldsymbol{\mu}_t \rangle \langle \mathbf{x}_t \rangle^T + \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle + \bar{\mathbf{V}}^{-1} \right)^{-1}. \end{cases} \quad (14)$$

Note that  $\langle \cdot \rangle$  denotes the expectation relative to the distribution  $q(\cdot)$ . The mean state and the precision matrix distributions  $q(\boldsymbol{\mu}_t)$  and  $q(\boldsymbol{\lambda}_t)$  have closed forms such that their expectations are easily derived as

$$\begin{cases} \langle \boldsymbol{\mu}_t \rangle = \boldsymbol{\mu}_t^* \\ \langle \boldsymbol{\mu}_t \boldsymbol{\mu}_t^T \rangle = \boldsymbol{\lambda}_t^{*-1} + \boldsymbol{\mu}_t^* \boldsymbol{\mu}_t^{*T} \\ \langle \boldsymbol{\lambda}_t \rangle = n^* \mathbf{V}_t^*. \end{cases} \quad (15)$$

However, the target state  $\mathbf{x}_t$  and the activated sensors positions  $\mathbf{S}_t$  do not have closed forms. Combining (11) and (12),  $q(\mathbf{x}_t)$  and  $q(\mathbf{s}_t^i)$  have the following expressions:

$$\begin{cases} q(\mathbf{x}_t) \propto \mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle) \prod_{i=1}^{m_t} \exp \left\langle \log p \left( z_t^{i,\mathbf{x}} | \mathbf{x}_t, \mathbf{s}^i \right) \right\rangle_{q(\mathbf{s}^i)} \\ q(\mathbf{s}_t^i) \propto \mathcal{N}(\hat{\mathbf{s}}^i, \eta^i) \exp \left\langle \log p \left( z_t^{i,\mathbf{x}} | \mathbf{x}_t, \mathbf{s}^i \right) \right\rangle_{q(\mathbf{x}_t)} \\ \quad \times \prod_{j \neq i}^{m_t-1} \exp \left\langle \log p \left( z_t^{i,j} | \mathbf{s}^i, \mathbf{s}^j \right) \right\rangle_{q(\mathbf{s}^j)}. \end{cases} \quad (16)$$

Therefore, the general state evolution model (4) and the observation model (8) and (9) are naturally incorporated to update  $q(\mathbf{x}_t)$  and  $q(\mathbf{s}_t^i)$ . Their distribution forms immediately suggest an IS procedure, where samples are drawn from Gaussian distributions  $\mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle)$  and  $\mathcal{N}(\hat{\mathbf{s}}^i, \eta^i)$ , respectively, and

are weighted according to their likelihoods (taking into account the observation model) as

$$\begin{cases} \mathbf{x}_t^{(k)} \sim \mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle) \\ w_t^{(k)} \propto \prod_{i=1}^{m_t} p \left( z_t^{i,\mathbf{x}} | \mathbf{x}_t^{(k)}, \langle \mathbf{s}_t^i \rangle \right) \\ \mathbf{s}_t^{i,(k)} \sim \mathcal{N}(\hat{\mathbf{s}}^i, \eta^i) \\ w_t^{i,(k)} \propto p \left( z_t^{i,\mathbf{x}} | \langle \mathbf{x}_t \rangle, \mathbf{s}_t^{i,(k)} \right) \prod_{j \neq i}^{m_t-1} p \left( z_t^{i,j} | \mathbf{s}^{i,(k)}, \langle \mathbf{s}^j \rangle \right). \end{cases} \quad (17)$$

Note that the expected observations relative to  $q(\cdot)$ , such as  $\exp(\log p(z_t^{i,\mathbf{x}} | \mathbf{x}_t, \mathbf{s}^i))_{q(\mathbf{s}^i)}$  in (16), are approximated by the observation with respect to the expectations, e.g.,  $p(z_t^{i,\mathbf{x}} | \mathbf{x}_t, \langle \mathbf{s}_t^i \rangle)$ , when computing the particle weights. Accordingly, the expectations relative to  $q(\mathbf{x}_t)$  and  $q(\mathbf{s}_t^i)$  can be approximated by the Monte Carlo method as

$$\begin{cases} \langle \tilde{\mathbf{x}}_t \rangle = \sum_{k=1}^N w_t^{(k)} \mathbf{x}_t^{(k)} \\ \langle \tilde{\mathbf{s}}_t^i \rangle = \sum_{k=1}^N w_t^{i,(k)} \mathbf{s}_t^{i,(k)}. \end{cases} \quad (18)$$

Contrary to the traditional PF, the VF method reduces the temporal dependence to a single Gaussian distribution  $q(\boldsymbol{\mu}_{t-1})$  instead of propagating the huge amount of particles. To estimate the positions of the detecting sensors and the target, new particles are generated at each sampling instant, from the closed Gaussian distributions deduced from  $q(\boldsymbol{\mu}_{t-1})$  using variational calculus, and then, they are sampled according to the temporal observations to calculate the corresponding expectations.

#### D. Prediction

As aforementioned, a standard SLAT solution contains a prediction step and an update step. Aside from the aforementioned update of the filtering distribution  $p(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t})$ , the predictive distribution  $p(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t-1})$  can also efficiently be calculated using the variational approach. In fact, by incorporating the separable approximate distribution  $q(\boldsymbol{\alpha}_{t-1})$  in the place of  $p(\boldsymbol{\alpha}_{t-1}|\mathbf{Z}_{1:t-1})$ , the recursive DVaSLAT algorithm calculates the predictive distribution  $p(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t-1})$  in the following form:

$$\begin{aligned} \hat{p}(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t-1}) &\propto \int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})q(\boldsymbol{\alpha}_{t-1})d\boldsymbol{\alpha}_{t-1} \\ &\propto p(\mathbf{S}_t)p(\mathbf{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)p(\boldsymbol{\lambda}_t)q_p(\boldsymbol{\mu}_t). \end{aligned} \quad (19)$$

The exponential-form solution, which minimizes the KL divergence between the predictive distribution  $p(\boldsymbol{\alpha}_t|\mathbf{Z}_{1:t-1})$  and the separable approximate distribution  $q_{t|t-1}(\boldsymbol{\alpha}_t)$ , yields Gaussian distributions for the predicted expectations and a Wishart distribution for the target precision matrix as

$$\begin{cases} q_{t|t-1}(\mathbf{x}_t) \propto \mathcal{N}(\langle \boldsymbol{\mu}_t \rangle_{q_{t|t-1}}, \langle \boldsymbol{\lambda}_t \rangle_{q_{t|t-1}}) \\ q_{t|t-1}(\boldsymbol{\mu}_t) \propto \mathcal{N}(\boldsymbol{\mu}_{t|t-1}^*, \boldsymbol{\lambda}_{t|t-1}^*) \\ q_{t|t-1}(\boldsymbol{\lambda}_t) \propto \mathcal{W}_d(\mathbf{V}_{t|t-1}^*, n_{t|t-1}^*) \\ q_{t|t-1}(\mathbf{S}_t) \propto \mathcal{N}(\hat{\mathbf{S}}, \boldsymbol{\eta}) \end{cases} \quad (20)$$

where the parameters are updated according to the same iterative schemes in (14) and (15). The target state and the activated sensors are now predicted by the following expressions:

$$\begin{aligned} \langle \mathbf{x}_t \rangle_{q_{t|t-1}} &= \langle \boldsymbol{\mu}_t \rangle_{q_{t|t-1}} \\ \langle \mathbf{x}_t \mathbf{x}_t^T \rangle_{q_{t|t-1}} &= \langle \boldsymbol{\lambda}_t \rangle_{q_{t|t-1}}^{-1} + \langle \boldsymbol{\mu}_t \rangle_{q_{t|t-1}} \langle \boldsymbol{\mu}_t \rangle_{q_{t|t-1}}^T \\ \langle \mathbf{S}_t \rangle_{q_{t|t-1}} &= \langle \hat{\mathbf{S}} \rangle. \end{aligned} \quad (21)$$

The computational cost and the memory requirements are dramatically reduced by the variational approach in the prediction phase compared to the PF method. As previously demonstrated, the expectations involved in the computation of the predictive distribution have closed forms, avoiding the need of Monte Carlo integration.

Therefore, the variational approach predicts and updates the probability distributions at each instant, allowing the covering of the whole state evolution. The pseudocode of the DVaSLAT algorithm is summarized in Algorithm 1.

---

#### Algorithm 1: DVaSLAT algorithm.

---

**Input:**  $\bar{\mathbf{S}}, \bar{\boldsymbol{\lambda}}, \bar{\mathbf{V}}, \bar{n}, \boldsymbol{\mu}_0^*$ , and  $\boldsymbol{\lambda}_0^*$   
**Output:**  $\langle \mathbf{x}_t \rangle$  and  $\langle \mathbf{S}_t \rangle$

- 1 **for**  $i = 1 : N_s$  **do**
- 2   Generate  $N$  samples  $\{\mathbf{s}_0^{i,(k)}\}_{k=1}^N$  from  $\mathcal{N}(\bar{\mathbf{s}}^i, \eta^i)$ ;
- 3   Prelocalize  $\mathbf{s}^i$  according to (1), where the initial estimation  $\hat{\mathbf{s}}^i$  is calculated based on  $\mathbf{z}^{i,\mathbf{s}}$ ;
- 4 **end**
- 5 **for**  $t = 1, 2, \dots$ , **do**
- 6   Cluster activation and CH election;
- 7   Incorporate the *a priori* information:  $\boldsymbol{\mu}_t^p = \boldsymbol{\mu}_{t-1}^*$ ,  $\boldsymbol{\lambda}_t^p = ((\boldsymbol{\lambda}_{t-1}^*)^{-1} + \bar{\boldsymbol{\lambda}}^{-1})^{-1}$ ,  $q_p(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}_t^p, \boldsymbol{\lambda}_t^p)$ , and  $\{\hat{\mathbf{s}}^i\}_{i=1}^{m_t}$ ;
- 8   Predict  $p(\boldsymbol{\alpha}_t | \mathbf{Z}_{1:t-1})$  according to (19);
- 9   Compute the corresponding expectations as (21);
- 10   Initiate  $\boldsymbol{\mu}_t^* = \boldsymbol{\mu}_t^p$ ,  $\boldsymbol{\lambda}_t^* = 2\boldsymbol{\lambda}_t^p$ ,  $n_t^* = \bar{n} + 1$ , and  $\mathbf{V}_t^* = ((2\boldsymbol{\lambda}_t^p)^{-1} + (\bar{\mathbf{V}})^{-1})^{-1}$ ;
- 11   Calculate the initial expectations  $\langle \boldsymbol{\mu}_t \rangle = \boldsymbol{\mu}_t^*$  and  $\langle \boldsymbol{\lambda}_t \rangle = n_t^* \mathbf{V}_t^*$ ;
- 12   **while not converge do**
- 13     Generate  $N$  samples  $\{\mathbf{x}_t^{(k)}\}_{k=1}^N$  from  $\mathcal{N}(\langle \boldsymbol{\mu}_t \rangle, \langle \boldsymbol{\lambda}_t \rangle)$ ;
- 14     Compute the expectation of the target state based on  $\mathbf{z}_t^{\mathbf{s}, \mathbf{x}}$  in (17) and (18);
- 15     **for**  $i = 1 : m_t$  **do**
- 16        $N$  samples  $\{\mathbf{s}_t^{i,(k)}\}_{k=1}^N$  from  $\mathcal{N}(\hat{\mathbf{s}}^i, \eta_i)$ ;
- 17       Update the expectation of sensor  $i$  by incorporating the observations  $\{\mathbf{z}_t^{i,\mathbf{x}}, \mathbf{z}_t^{i,\mathbf{s}}\}$  as (17) and (18);
- 18     **end**
- 19     Update the variational parameters  $\boldsymbol{\mu}_t^*$ ,  $\boldsymbol{\lambda}_t^*$ ,  $n_t^*$ , and  $\mathbf{V}_t^*$  according to (14);
- 20     Reupdate the expectations  $\langle \boldsymbol{\mu}_t \rangle$  and  $\langle \boldsymbol{\lambda}_t \rangle$  by (15);
- 21   **end**
- 22   Communicate  $q(\boldsymbol{\mu}_t) = \mathcal{N}(\boldsymbol{\mu}_t^*, \boldsymbol{\lambda}_t^*)$  for the next instant;
- 23   Return the target position estimation  $\langle \mathbf{x}_t \rangle$  and those of the activated sensors  $\langle \mathbf{S}_t \rangle = \{\langle \mathbf{s}_t^i \rangle\}_{i=1}^{m_t}$ ;
- 24 **end**

---

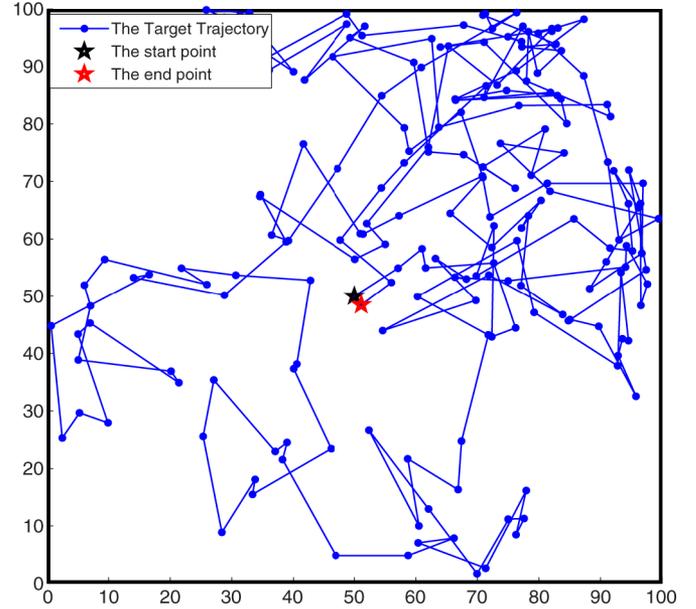


Fig. 5. Target trajectory following an RMW model.

## V. EVALUATION AND SIMULATION

This section evaluates the DVaSLAT algorithm on a synthetic example, which involves the tracking of a mobile target and the localization of sensors. The purpose of the synthetic example is to establish a baseline performance comparison on a relatively difficult problem.

### A. Simulation Configuration

As aforementioned, no constraint is put on the target velocity or moving direction. Therefore, the RWM model [13], which mimics an erratic movement of a target in extremely unpredictable situations, was employed in our simulation to describe the target movement. The target that follows an RWM model moves from its current location by choosing a random direction and a random speed. If the target reaches the boundary of the simulation field, it rebounds with an angle determined by the incoming direction. Fig. 5 shows our configuration of the target motion in the simulation field, where the target moving duration is 200 time slots.

With regard to the sensor deployment step, the coverage is an important issue to be guaranteed. In general, at least three sensors are required to detect the target presence at every sampling instant for the tracking work. If sensors are uniformly distributed over the simulation region with density  $\rho$ , sensor distribution in any given area  $A$  is Poisson with rate  $\rho A$ . Thus, the probability for any arbitrary point to be sensed by at least three sensors is  $p = \sum_{i=3}^{\infty} (\rho \pi r_s^2)^i \exp(-\rho \pi r_s^2) / i!$  [36], where  $r_s$  is the sensing range of sensors. In our simulations, to ensure the 3-coverage condition, 400 sensors were assumed to uniformly be deployed in a 2-D field ( $100 \times 100 \text{ m}^2$ ), and their sensing ranges were identically fixed to 15 m. However, due to the spatially varying environment factors and deployment errors, sensors were, in fact, randomly distributed around their initially setting locations  $\bar{\mathbf{S}} : \mathbf{s}^i \sim \mathcal{N}(\bar{\mathbf{s}}^i, \eta^i)$ , where  $\eta^i$  was identical

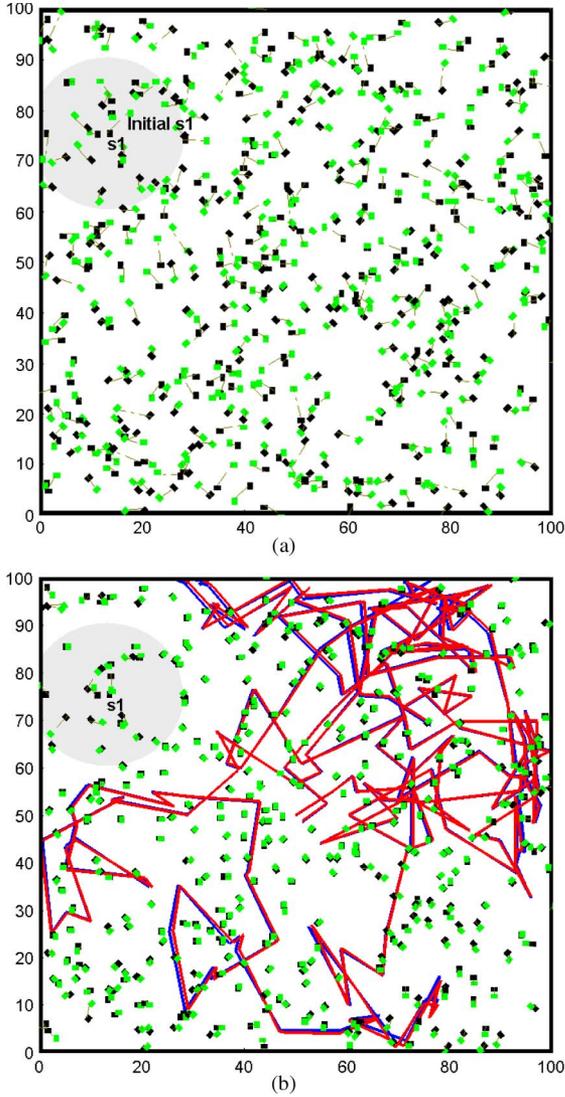


Fig. 6. Simulation field. (a) Initial configuration in the simulation field, where the green rectangles denote the initial known sensor locations, and the black rectangles are their actual locations. The dashed lines denote the deviations between sensor initial known positions and the actual positions. The gray field shows the sensing range of sensor  $s^1$ . (b) Performance of the DVaSLAT algorithm after one trial. The blue line is the trajectory of the target, and the red line is the tracking result of the DVaSLAT algorithm. The green rectangles denote the sensor positions estimated by the DVaSLAT algorithm. Other denotations are the same as in (a).

for all the sensors. The initial configuration is demonstrated in Fig. 6(a).

The parameters involved in the general state evolution model (4) were set as follows:

$$\bar{\mathbf{V}} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \quad \eta_i = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}$$

$$\bar{\lambda} = \begin{bmatrix} 1/900 & 0 \\ 0 & 1/900 \end{bmatrix}, \quad \bar{n} = 10.$$

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 central processing unit, 3.4-GHz, 1.0-GB random access memory personal computer.

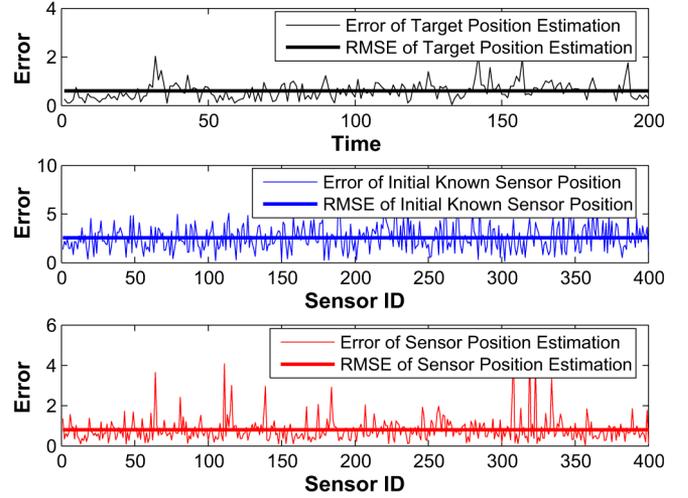


Fig. 7. Target tracking and sensor localization performance of the DVaSLAT algorithm.

TABLE I  
EVALUATION OF THE DVaSLAT ALGORITHM

Evaluation	DVaSLAT
Target Tracking RMSE	0.6047m
Sensor Localization RMSE	0.8042m
Sensor Deployment RMSE	2.5605m
Overall Energy Consumption	CHs: 51.3271 J, Slaves: 84.8403 mJ
Average Energy Consumption	256.6353 mJ/CH, 33.9361 mJ/Slave
Execution time	0.1962s

### B. Evaluation

The performance of the DVaSLAT algorithm using 100 particles is shown in Fig. 6(b) and evaluated by the root mean square error (RMSE) in Fig. 7. The average target tracking error is 0.6047 m, and the average sensor localization error is 0.8042 m against the initial deployment error of 2.5605 m (see Table I). Therefore, in addition to the accurate tracking performance, remarkable refinements for sensor localization are demonstrated. However, because DVaSLAT is executed on a cluster base for energy efficiency, only the sensors that have detected the presence of the target are activated and relocated, which leads to precise localization of the sensors in high-traffic regions. On the contrary, sensors that are far from the target trajectory are left unlocalized, corresponding to the peaks in the bottom plot of Fig. 7.

The energy expenditure during the whole SLAT process is calculated based on the hypothesis that the energy consumed in computation can be neglected relative to the energy consumed in communication. According to the energy consumption model that was proposed in [26] and [37], the energy consumed (in transmissions per bit) is  $E_T = \epsilon_e + \epsilon_a d^3$ , where  $\epsilon_e$  is the energy consumed by the circuit per bit,  $\epsilon_a$  is the energy dissipated (in Joules per bit per cubic meter), and  $d$  is the transmission distance ( $\epsilon_a = 3.5 \times 10^{-3}$  pJ/b/m<sup>3</sup>,  $\epsilon_e = 45$  nJ/b). The energy consumed when receiving data is given by  $E_R = \epsilon_r N$ , where  $\epsilon_r$  denotes the energy expended on receiving 1 b of data ( $\epsilon_r = 135$  nJ/b). Similarly, the energy consumed in detection is defined by  $E_S = \epsilon_s N$ , where  $\epsilon_s$  is the energy expended on sensing 1 b of data ( $\epsilon_s = 50$  nJ/b). We calculate the overall

TABLE II  
EVALUATION OF THE DVASLAT ALGORITHM USING  
DIFFERENT NUMBERS OF PARTICLES

Number of particles	100	200	400	800
RMSE of Target Tracking ( $m$ )	0.5777	0.5139	0.4671	0.3886
RMSE of Sensor Localization ( $m$ )	0.7056	0.6561	0.5863	0.5585
Average execution time ( $s$ )	0.2158	0.2875	0.4589	0.8049

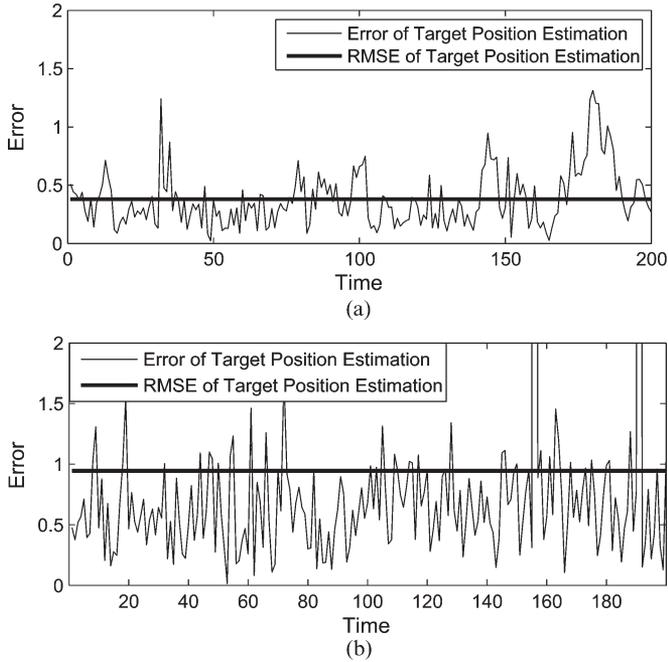


Fig. 8. Tracking performance comparison. (a) DVaSLAT. (b) LaSLAT.

and the average energy consumptions of the CHs and the slave sensors, respectively. The execution time is evaluated by the average time consumed per sampling slot (1 s). As shown in Table I, the average execution time of 0.1962 s guarantees the online implementation of our scheme.

To benchmark the performance of the proposed DVaSLAT algorithm, we run the algorithm with an increasing number of particles, i.e.,  $N = 100, 200, 400, 800$ . Monte Carlo simulations are executed to get a statistical reflection of the behavior of the algorithm. As shown in Table II, the RMSE generally decreases with the increase in the number of particles. However, the performance does not appear to significantly upgrade. The computational complexity is reflected by the average execution time statistics in Table II. The execution time increases with the increasing of the number of particles, exhibiting a proportional trend.

### C. Comparisons With LaSLAT and Traditional PF Algorithms

The performance of our DVaSLAT algorithm is compared with the LaSLAT algorithm [17]. For comparison, we initially set the batch size to one time slot in LaSLAT. The tracking performance of the DVaSLAT algorithm and LaSLAT are evaluated by the RMSE in Fig. 8. We can notice that both algorithms yield accurate estimation of the target position. However, the precise estimation of LaSLAT is achieved at the cost of computation complexity. In fact, with the LaSLAT procedure, the

smaller the batch size is, the less information is incorporated, and consequently, the more Newton–Raphson optimization iterations are needed until the solution is fully converged. Furthermore, the representation of the belief state requires a storage space that is quadratic with respect to the number of activated sensors, because the correlations between the target and activated sensors must be updated with each observation. The centralized implementation is therefore preferable for the LaSLAT algorithm.

Monte Carlo simulations of the LaSLAT algorithm that were executed on different batch sizes are reported in Table III. Note that the execution time of LaSLAT, which is evaluated by the average time consumed per sampling slot, fluctuates during the simulations. The reason for the fluctuation is that the amount of the Newton–Raphson optimization iterations dramatically drops along with the increment of batch size, whereas the time consumed per iteration greatly increases due to the quadratic increment of the belief state to be updated. On the contrary, the execution time for each iteration involved in the DVaSLAT algorithm is constant, because the number of particles and the involved iterations are of constant size. The small execution time of the DVaSLAT algorithm guarantees its real-time implementation in online applications. By changing the batch size of measurements incorporated in the LaSLAT algorithm, we can find that LaSLAT is vulnerable to substantial errors when the batch size is small (see Fig. 9). Because LaSLAT incorporates only range information between the target and activated sensors, the algorithm risks to fall into false local minima. With regard to the precision of sensor localization, Fig. 10 compares the RMSEs of initial sensor deployment and of sensor localization in DVaSLAT. We can notice that DVaSLAT accomplishes a remarkable improvement in the precision of sensor localization. Due to incorporating the measurements between the neighboring sensors, the DVaSLAT algorithm has much more information for localization at every sampling instant, and the risk of local minima is also reduced. On the contrary, LaSLAT demonstrates barely small amelioration in RMSE compared with the initial sensor deployment. The precisions of sensor localization are also compared in Table III with Monte Carlo simulations of the two algorithms. Our DVaSLAT algorithm demonstrates superiority in online applications with excellent tracking accuracy and localization precision.

We also compare our DVaSLAT algorithm with the traditional strategy [16], which localizes all the sensors in the field and then tracks the target based on the sensor location estimations. In our simulation of the traditional strategy, we employ the classical PF algorithm to localize all the sensors in the simulation field and to track the target in the second stage. Monte Carlo simulations were performed on the same configuration to compare the performance of the two strategies, where the number of particles was set to 400. Table IV shows that the DVaSLAT algorithm outperforms the traditional strategy by interdependently and continuously improving estimates of both sensors and target. The tracking accuracy and the localization precision are evaluated by their RMSE. With respect to the execution time, the traditional strategy outperforms the DVaSLAT algorithm, because it terminates the sensor relocalization phase during the target tracking.

TABLE III  
 MONTE CARLO SIMULATIONS OF THE DVASLAT AND LASLAT ALGORITHMS THROUGH 100 TRIALS

	Batch size	Tracking accuracy ( $m$ )	Deployment error ( $m$ )	Localization precision ( $m$ )	Execution time (s)	Iterations
DVaSLAT	1	0.6132	2.4087	0.4044	0.5948	1624
	1	0.9857	2.6050	2.5291	4.0251	9898
	2	13.6683	2.4847	2.4632	1.5048	4762
	4	9.2433	2.5170	2.4208	0.8834	2418
	5	11.0831	2.6578	2.5571	0.7540	1931
LaSLAT	10	26.8761	2.5884	2.4390	0.4550	918
	20	17.5902	2.4942	2.4910	0.3998	494
	25	21.0880	2.5501	2.4839	0.4217	400
	40	27.4372	2.5285	2.4926	0.3625	250
	50	18.9745	2.5847	2.4630	0.5133	192
	100	7.8057	2.4798	2.4667	0.7800	100
	200	5.2353	2.6066	2.5208	1.6574	50

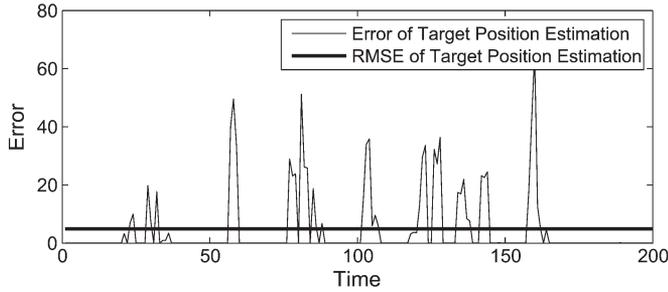


Fig. 9. Example of false local minima in target estimation by LaSLAT, where batch size = 4. As has been analyzed, LaSLAT falls into local minima at some instants due to the lack of information. The Newton–Raphson optimization does not always find the global minima.

 TABLE IV  
 COMPARISON BETWEEN THE TRADITIONAL AND THE DVASLAT ALGORITHMS

Comparison	Traditional	DVaSLAT
Tracking accuracy ( $m$ )	11.5661	0.6132
Localization precision ( $m$ )	0.6027	0.4044
Execution time (s)	0.3251	0.5948

#### D. Cramér–Rao Bound

The Cramér–Rao bound expresses a lower bound on the variance of a deterministic parameter estimator. It is well known that, for an unbiased estimator, this lower bound is the inverse of the Fisher information matrix [32], [38]. To show the advantage of incorporating the target measurement in the localization process, we calculate the Fisher information matrix before and after the passage of the target in the detection field of sensor  $i$  as follows.

- Before the passage of the target

$$I(s^i) = E \left[ \frac{\partial^2 \ln p(z^{i,s} | s^i)}{\partial s^i \partial s^{iT}} \right].$$

- After the passage of the target

$$I(s^i) = E \left[ \frac{\partial^2 \ln \left( p(z_{1:t}^{i,x} | s^i) p(z^{i,s} | s^i) \right)}{\partial s^i \partial s^{iT}} \right].$$

Here,  $z_{1:t}^{i,x}$  are the observations that were collected until the instant  $t$ , when the target  $x$  passes through the detection field of sensor  $s^i$ .

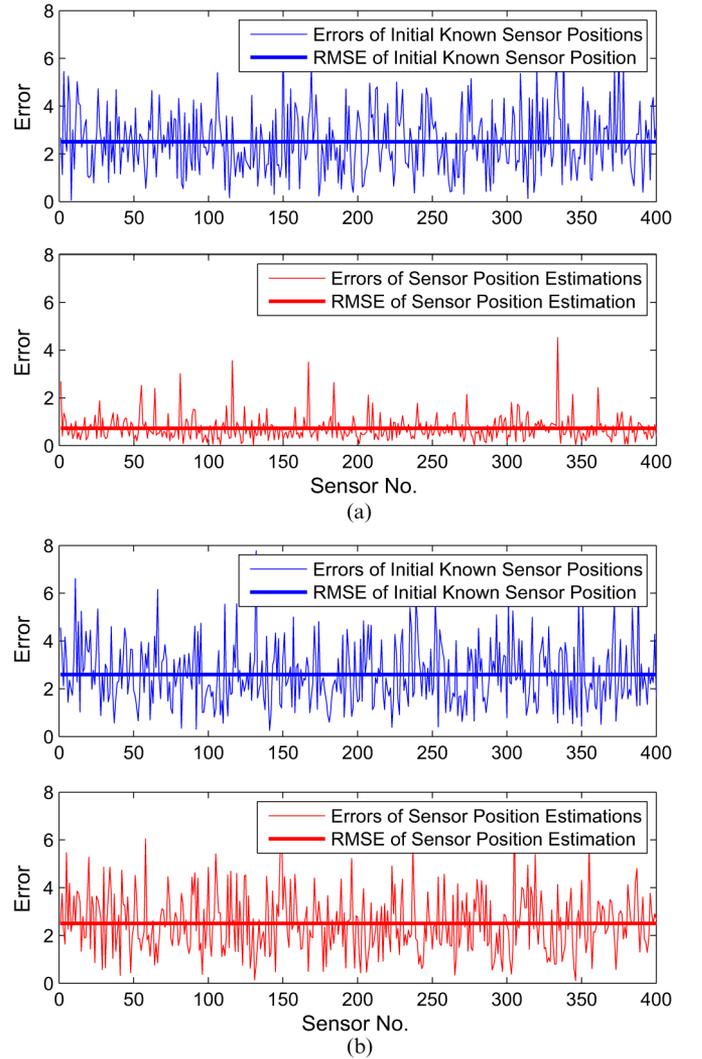


Fig. 10. Localization performance comparison. (a) DVaSLAT. (b) LaSLAT.

The Cramér–Rao bounds of the sensors are illustrated in Fig. 11. The top left part of Fig. 11, shows that the Cramér–Rao bounds before the passage of the target depend on the deployment of sensors (see Fig. 11, bottom right). As expected, the summit values appear in the boundary of the simulation field, where sensors are sparsely located. After the passage of the target, the Cramér–Rao bounds are remarkably reduced, particularly in the regions of high traffic (see Fig. 11, top

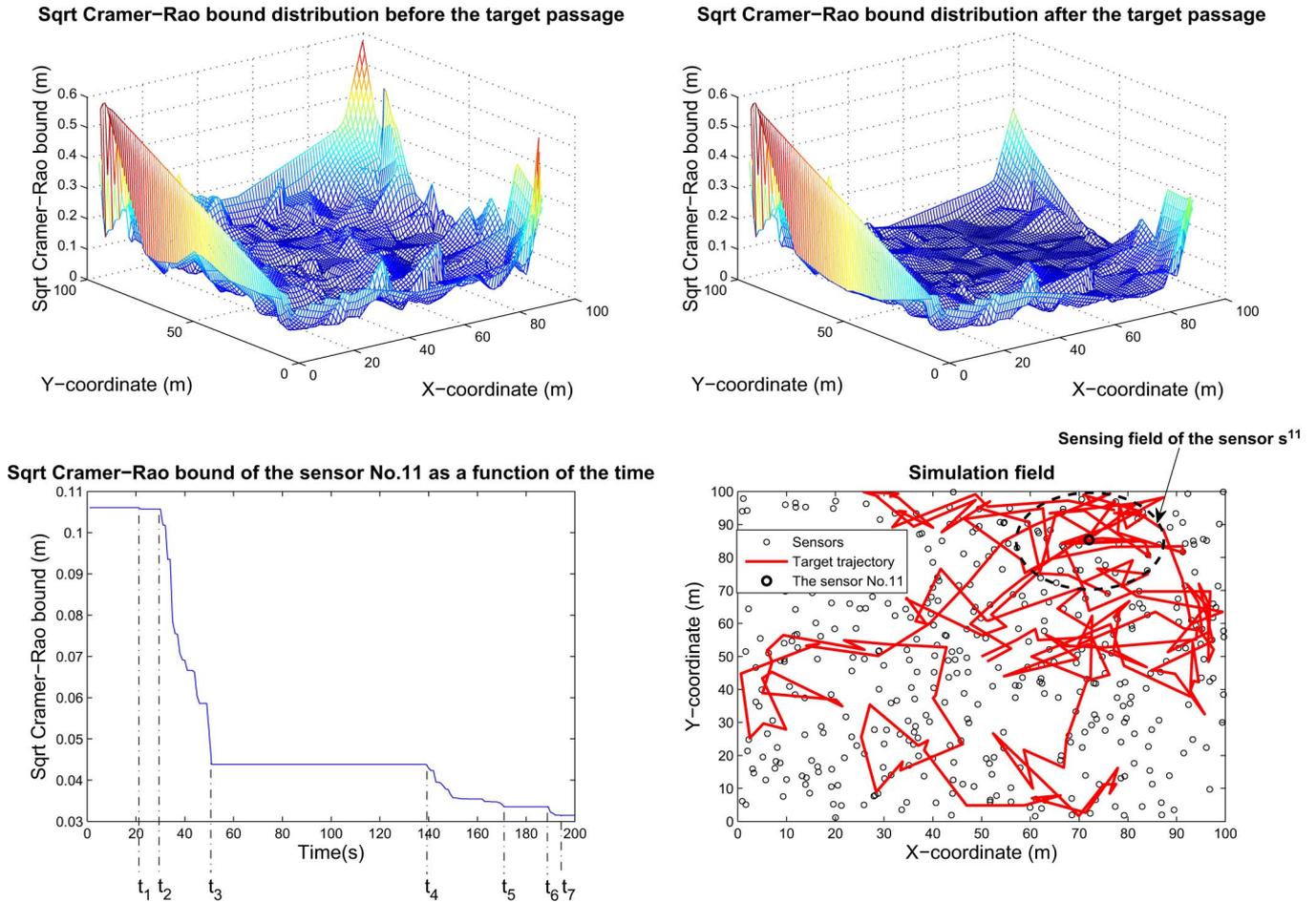


Fig. 11. Cramér–Rao bound of sensor localization.

right). As shown in the bottom right part of Fig. 11, sensor  $s^{11}$  is located in the high-traffic region. In Fig. 11, bottom left, we depict the variation of the Cramér–Rao bound of the sensor  $s^{11}$  as a function of the elapsed time. Before the passage of the target (from  $t = 0$  to  $t = 21$ ), the Cramér–Rao bound of  $s^{11}$  stays at the same value 0.1061, which depends only on the observations from its neighboring sensors. At instant  $t_1 = 22$ , the target is detected by  $s^{11}$ , which results in a slight reduction of the Cramér–Rao bound (from 0.1061 to 0.1057). Until the instant  $t_2 = 31$ , the target continuously appears in the sensing range of  $s^{11}$ . Accordingly, the Cramér–Rao bound demonstrates a persistent decrease between  $t_2 = 31$  and  $t_3 = 51$ . Another passage of the target sensed by  $s^{11}$  appears at the duration between  $t_4 = 140$  and  $t_5 = 171$ . Finally, because  $t_6 = 189$ , the Cramér–Rao bound of  $s^{11}$  reduces until the target goes out of its sensing field at instant  $t_7 = 195$ . The variation tendency of the Cramér–Rao bound confirms the usefulness of using the mobile target trajectory to refine sensor localization.

### VI. CONCLUSION

DVaSLAT has been proposed in the context of WSN. Without any *a priori* information on the target motion, the DVaSLAT algorithm aims at continuously updating and improving the estimates of the activated sensor locations and the target trajectory. Because the target can arbitrarily travel and the location

information of the activated sensors is rather coarse, a general state evolution model has been proposed to describe the hidden state, which is more adaptable to the nonlinear/non-Gaussian situation than other kinematic parameter models. To minimize resource consumption in WSNs, the DVaSLAT algorithm is executed on a fully distributed cluster scheme. That is, only the sensors that have detected the target are activated to form a cluster to process data. The variational method allows an implicit compression of the exchanged statistics between clusters, which greatly decreases the intercluster communication. In conclusion, because the target freely moves in WSNs, a large number of range measurements are generated, which facilitates both the activated sensors localization and the target tracking. As shown in the simulations, estimates of sensors and of the target are interdependently and continuously improved online.

In the simulations, the DVaSLAT algorithm has demonstrated the nice property to be model independent, which we hope to make use in the future. In fact, the DVaSLAT algorithm could be applied in the context of mobile ad hoc networks by making minor changes. Instead of statically storing the observations  $z^{i,s}$  between neighboring sensors, the set of observations  $z_t^{i,s}$  should be updated at each sampling instant for localizing the mobile sensors, whereas modeling the dynamics on the mobile sensors is also necessary. This also inspires us to extend the application of the DVaSLAT algorithm to multitarget tracking in a similar way.

## REFERENCES

- [1] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [2] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.
- [3] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
- [4] A. T. Ihler, J. W. Fisher, III, and A. S. Willsky, "Particle filtering under communications constraints," in *Proc. 13th Workshop Statist. Signal Process.*, 2005, pp. 89–94.
- [5] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proc. 13th Mediterranean Conf. Control Automat.*, 2005, pp. 719–724.
- [6] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 6–12, Dec. 2007.
- [7] X. Li, "Collaborative localization with received signal strength in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3807–3817, Nov. 2007.
- [8] J. Ansari, J. Riihijarvi, and P. Mahonen, "Combining particle filtering with Cricket system for indoor localization and tracking services," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2007, pp. 1–5.
- [9] J. Bruck, J. Gao, and A. Jiang, "Localization and routing in sensor networks by local angle information," in *Proc. 6th ACM Int. Symp. Mobile ad hoc Netw. Comput.*, 2005, pp. 181–192.
- [10] A. Cenedese, G. Ortolan, and M. Bertinato, "Low-density wireless sensor networks for localization and tracking in critical environments," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2951–2962, Jul. 2010.
- [11] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surveys*, vol. 39, no. 3, pp. 1–55, 2007.
- [12] N. Ahmed, M. Rutten, T. Bessell, S. S. Kanhere, N. Gordon, and S. Jha, "Detection and tracking using particle-filter-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1332–1345, Sep. 2010.
- [13] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, pp. 483–502, 2002.
- [14] L. Fang, W. Du, and P. Ning, "A beaconless location discovery scheme for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2005, vol. 30, pp. 33–55.
- [15] S. Krupadanam and H. Fu, "Beaconless location detection in wireless sensor networks for nonflat terrain," *Int. J. Software Eng. Appl.*, vol. 2, no. 3, pp. 55–76, 2008.
- [16] E. L. Souza, E. F. Nakamura, and H. A. De Oliveira, "On the performance of target tracking algorithms using actual localization systems for wireless sensor networks," in *Proc. MSWiM*, 2009, pp. 418–423.
- [17] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 27–33.
- [18] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. 18th IJCAI*, G. Gottlob and T. Walsh, Eds., 2003, pp. 1157–1164.
- [19] H. D. Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.
- [20] T. Bailey and H. D. Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
- [21] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 593–598.
- [22] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. 16th Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, 2003.
- [23] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 81–95.
- [24] W. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 3, pp. 258–271, Jul./Aug. 2004.
- [25] P. M. Djurić, M. Vemula, and M. Bugallo, "Tracking with particle filtering in tertiary wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, Mar. 18–23, 2005, pp. 757–760.
- [26] A. S. Chhetri, D. Morrell, and A. P. Suppappola, "Energy-efficient target tracking in a sensor network using nonmyopic sensor scheduling," in *Proc. 8th Int. Conf. Inf. Fusion*, 2005, p. 8.
- [27] J. Yick, B. Mukherjee, and D. Ghosal, "Analysis of a prediction-based adaptive mobility tracking algorithm," in *Proc. 2nd Int. Conf. Broadband Netw.*, 2005, pp. 809–816.
- [28] J. Vermaak, N. D. Lawrence, and P. Pérez, "Variational inference for visual tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 2003, pp. 773–780.
- [29] H. Snoussi and C. Richard, "Ensemble learning online filtering in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. Syst.*, 2006, pp. 1–5.
- [30] J. Teng, H. Snoussi, and C. Richard, "Decentralized variational filtering for target tracking in binary sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 10, pp. 1465–1477, Oct. 2010.
- [31] O. E. Barndorff-Nielsen, "Exponentially decreasing distributions for the logarithm of the particle size," in *Proc. Roy. Soc. London. Ser. A. Math. Phys. Sci.*, 1977, pp. 401–419.
- [32] P. M. Djurić, M. Vemula, M. Bugallo, and J. Míguez, "Noncooperative localization of binary sensors," in *Proc. 13th Workshop Statist. Signal Process.*, 2005, pp. 1244–1249.
- [33] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
- [34] J. H. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- [35] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.
- [36] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks," in *Proc. IEEE Int. Workshop Sensor Netw. Protocols Appl.*, 2003, pp. 71–81.
- [37] H. Wu and A. A. Abouzeid, "Error robust image transport in wireless sensor networks," in *Proc. 5th Workshop Appl. Services Wireless Netw.*, Paris, France, 2005.
- [38] S. Dulman, P. J. M. Havinga, A. Baggio, and K. Langendoen, "Revisiting the Cramér–Rao bound for localization algorithms," in *Proc. 4th IEEE Int. Conf. Distrib. Comput. Sensor Syst.*, 2008, pp. 1–4.



**Jing Teng** (M'11) received the B.Eng. degree in electronic information engineering from the Central South University, Changsha, China, in 2003 and the Ph.D. degree in systems optimization and security from the University of Technology of Troyes, Troyes, France, in 2009.

Since 2010, she has been a Lecturer with the School of Control and Computer Engineering, North China Electric Power University, Beijing, China. She is the author or a coauthor of more than 10 papers in refereed journals and international conference proceedings and has been serving as a Reviewer of several journals and international conference proceedings. Her research interests include statistical signal processing and its applications in wireless sensor networks.



**Hichem Snoussi** (M'08) received the Diploma degree in electrical engineering from the Ecole Supérieure d'Electricité, Gif-sur-Yvette, France, in 2000 and the D.E.A. and Ph.D. degrees in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003, respectively.

Between 2003 and 2004, he was a Postdoctoral Researcher with the Institut de Recherche en Communications et Cybernétiques de Nantes, Nantes, France. He has spent short periods as a Visiting Scientist with the RIKEN Brain Science Institute,

Tokyo, Japan, and Olin Neuropsychiatry Research Center, Institute of Living, Hartford, CT. Since 2005, he has been an Associate Professor with the University of Technology of Troyes (UTT), Troyes, France, where he has been leading the research group "Surveillance" of the Laboratory of Systems Modeling and Dependability, Institut Charles Delaunay, since January 2008. He is in charge of the regional research program System Security and Safety of the Contrat de Projets Etat-Région Champagne-Ardenne 2007–2013 and the CapSec platform (wireless embedded sensors for security). He is the Principal Investigator of a French National Research Agency (ANR)–Blanc project, a Center for Research in Computing and the Arts project (new partnership and new technologies), and a GDR–Information, Signal, Images, ViSion young researcher project. He is a partner of several ANR projects, GIS, and strategic UTT programs.

Dr. Snoussi received the 2008–2012 National Doctoral Supervision and Research Award.



**Rong Zhou** received the M.E. degree in computer application technology from the North China Electric Power University, Beijing, China, in 2003. She is currently working toward the Ph.D. degree in computer application technology with the Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing.

Since 2003, she has been a Lecturer with the North China Electric Power University. Her research interests include statistical signal processing and small target tracking and detection.



**Cédric Richard** (S'98–M'01–SM'07) received the Dipl.-Ing. and M.S. degrees and the Ph.D. degree in electrical and computer engineering from the University of Technology of Compiègne, Compiègne, France, in 1994 and 1998, respectively.

From 1999 to 2003, he was an Associate Professor with the University of Technology of Troyes (UTT), Troyes, France. From 2003 to 2009, he was a Full Professor with the Institut Charles Delaunay (CNRS FRE 2848), UTT, and the Supervisor of a group that consists of 60 researchers and Ph.D. students.

In the winter of 2009, he was a Visiting Researcher with the Department of Electrical Engineering, Federal University of Santa Catarina, Florianópolis, Brazil. Since September 2009, he has been a Full Professor with the Fizeau Laboratory [Centre National de la Recherche Scientifique Unite Mixte de Recherche (CNRS UMR) 6525, Observatoire de la Côte d'Azur], University of Nice Sophia-Antipolis, Nice, France. He is the author of more than 100 papers. He has been an Associate Editor for the *EURASIP Signal Processing Magazine* since 2009. His research interests include statistical signal processing and machine learning.

Dr. Richard is a Member of the Groupe d'Etudes du Traitement du Signal et des Images Association Board and the European Association for Signal Processing. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING since 2006.