

Fourier Density Approximation for Belief Propagation in Wireless Sensor Networks

Chongning Na, Hui Wang, Dragan Obradovic, and Uwe D. Hanebeck

Abstract—Many distributed inference problems in wireless sensor networks can be represented by probabilistic graphical models, where belief propagation, an iterative message passing algorithm provides a promising solution. In order to make the algorithm efficient and accurate, messages which carry the belief information from one node to the others should be formulated in an appropriate format. This paper presents two belief propagation algorithms where non-linear and non-Gaussian beliefs are approximated by Fourier density approximations, which significantly reduces power consumptions in the belief computation and transmission. We use self-localization in wireless sensor networks as an example to illustrate the performance of this method.

I. INTRODUCTION

ADVANCES in sensor technology and telecommunications make wireless sensor network (WSN) an appropriate solution for a wide variety of applications[1][2]. In a WSN, sensor nodes are spatially distributed to monitor the physical or environmental states. Information can be exchanged through the wireless channel so that the whole network works in a cooperative fashion. Many estimation problems in WSNs can be represented by probabilistic graphical models and solved by belief propagation methods. Belief propagation (BP) is an iterative message passing algorithm in which each node calculates its belief about other nodes and communicates with them to exchange their beliefs about each other. Compact messages that are transmitted between nodes carry the necessary information of the beliefs, based on which the receiver can reconstruct the transmitter's belief about it. For discrete beliefs, messages can be a short vector of probabilities. For continuous beliefs with Gaussian distribution, it is enough to ensemble the mean and variance in the message. However, in many applications, beliefs have non-linear and non-Gaussian distributions so that belief calculation and transmission consumes a lot of power. That limits its application in WSNs which have strong power constraints. Hence, an appropriate representation of beliefs which reduces the complexity while keeping the accuracy is necessary but non-trivial.

Monte Carlo methods can be used where messages contain samples that are drawn from the distribution to represent the beliefs. Gibbs sampling is a popular method in this case. However, this is only possible for sufficiently small

networks.

Authors of [3] used non-parametric BP method where beliefs are represented by Gaussian mixtures. It generalizes particle filtering for inference in non-linear, non-Gaussian time series.

In this paper, we introduce Fourier density approximation (FDA) method to represent the beliefs. Fourier series were first employed to estimate probability densities in [4]. Recently, [5] and [6] ensured the non-negativity of Fourier series by approximating the square root of the density instead of the density itself. The usage of Fourier series in nonlinear Bayesian filtering is also derived in [5] and [6]. Using Fourier density approximation, the belief can be represented sufficiently by only a small number of Fourier coefficients. Hence, the transmission power and time between sensor nodes are significantly saved. Compared to other density representations like Gaussian mixture or Monte Carlo methods, the optimal number of coefficients under a required approximation error with respect to a density distance metric is more efficiently obtained. Furthermore, the sum-product operations in BP algorithms can be more effectively calculated in Fourier domain since some convolution-like integral operations are more easily calculated than in space domain. Since the Fourier series are orthogonal expansions, the coefficients are derived independently and effectively [5]. In practice, this is done by efficient Fast Fourier Transform (FFT).

In this paper, the self-localization in WSNs, a common practice of belief propagation, is used to evaluate the performance of Fourier density approximation. Two Fourier based algorithms are proposed, which are simplified transmission based on Fourier density approximation (ST-FDA) and simplified computation and transmission based on Fourier density approximation (SCT-FDA). ST-FDA reduces the size of the belief message to save radio transmission power, which is a critical factor for WSNs. SCT-FDA further simplifies the sum-product algorithm (SPA) to reduce computation power.

The paper is organized as follows. Section II presents BP as a general approach to the inference problems in WSNs. Fourier density approximation method will be introduced in Section III. Section IV uses a sensor self-calibration example to illustrate the use of Fourier density approximation for BP. ST-FDA and SCT-FDA algorithms are proposed. Their performances will be evaluated through simulation and the results will be shown in Section V. Finally, Section VI concludes the paper.

C. Na, H. Wang and D. Obradovic are with Siemens AG, Corporate Technology, Munich, Germany (phone: +49-89-636-49499; e-mail: {na.chongning.ext, dragan.obradovic, hui.wang.ext}@siemens.com).

U. D. Hanebeck is with Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany (e-mail: uwe.hanebeck@ieee.org).

II. BELIEF PROPAGATION IN WIRELESS SENSOR NETWORKS

A. Probabilistic Model of a Wireless sensor network

Let's consider a WSN with N sensor nodes that are distributed in space. We use x_i to denote the physical state associated with sensor node i and use \mathbf{x} to denote the collection of state variables at all sensor nodes. Each sensor makes a local noisy observation which we denote by y_i . In general, the following assumptions are valid:

- Given the state variables, observations at different nodes are independent, i.e. $p(y_i, y_j | \mathbf{x}) = p(y_i | \mathbf{x})p(y_j | \mathbf{x})$.
- Observation at one node depends only on a subset of state variables, i.e. $p(y_i | \mathbf{x}) = p(y_i | \mathbf{x}_{Pa(y_i)})$ with $\{\mathbf{x}_{Pa(y_i)}\} \subset \{\mathbf{x}\}$.
- Usually, local correlation exists between neighboring nodes. This indicates that the joint probability of state variables can be factorized into a product of local functions which present the correlation among the nodes in neighborhoods, i.e. $p(\mathbf{x}) = \prod_c p(\mathbf{x}_c)$.

Based on these assumptions and using the Bayes rule, the joint distribution of state variables and observations can be factorized in the following form:

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y} | \mathbf{x}) \cdot p(\mathbf{x}) = \prod_{i=1}^N p(y_i | \mathbf{x}) \cdot p(\mathbf{x}) = \prod_{i=1}^N p(y_i | \mathbf{x}_{Pa(y_i)}) \cdot p(\mathbf{x}) = \prod_{i=1}^N p(y_i | \mathbf{x}_{Pa(y_i)}) \cdot \prod_c p(\mathbf{x}_c) \quad (1)$$

The conditional independences encoded in (1) can be presented by a graphical model, e.g. Markov random field [7]. A graphical model consists of a set of vertices which represent the variables. There exists an edge between two vertices which indicates the conditional dependence between them. So the whole graph represents the factorization of a joint distribution of all variables. The relationship between the graphical model and the joint distribution is given by the Hammersley-Clifford theorem [8], that is, a joint probability can be written as a product of potential functions φ which are defined on cliques (sub-graphs that are fully connected). In probabilistic inference in WSNs, we'd like to write this factorization as:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^N \varphi_i(y_i, \mathbf{x}_{c_i}) \quad (2)$$

so that each factor in (2) can be associated with one sensor node. Such a factorization automatically provides the possibility to distribute the computation. Each node processes parts of the total computation and results are eventually disseminated through the communication between nodes.

Each potential function in (2) is obtained from (1). We first assign $p(y_i | \mathbf{x}_{Pa(y_i)})$ as a factor of $\varphi_i(y_i, \mathbf{x}_{c_i})$, then distribute each factors in $\prod_c p(\mathbf{x}_c)$ into one of the potential functions.

In many applications, the distribution of $\prod_c p(\mathbf{x}_c)$ is not unique. For the assignment, we should also take factors such as computational complexity, communication connectivity

and transmission power into consideration. Authors of [9] have introduced a method that first constructs a spanning tree and then assigns factors to the nodes of the tree. Such an assignment eventually results in a junction tree that can be solved by message passing algorithms [10]. In some other applications, the final graphical model is a graph with loops.

B. Belief Propagation in Wireless Sensor Networks

Inference of the variables defined on a graphical model has been intensively studied. For a graph without loops, this can be solved by junction tree algorithm. Exact inference on a graph with loops is generally an N-P hard problem. Approximate methods, such as loopy BP [11] have produced convictive results in many applications. BP is an iterative message passing algorithm in which each node calculates its belief about other nodes and communicates with them to exchange their beliefs about each other. Each node updates its beliefs when it receives messages from other nodes. Updated beliefs will be sent in messages to other nodes. This procedure repeats for a number of iterations or until a defined convergence criterion has been met.

In WSN applications, we are interested in the posterior probability of $p(x_i | \mathbf{y})$ for each state variable x_i . Such an inference problem on graphical models can be solved by using sum-product algorithm, which is a common practice [12].

Having defined the local potentials for each node like in (2), we can write the analytic formula for the belief updating in SPA at each sensor node. We define $m_{ij}^t(\mathbf{x}_{c_i} \cap \mathbf{x}_{c_j})$ to be the message sent from node i to node j in the t^{th} iteration. Having received messages from all neighboring nodes in the t^{th} iteration, node i calculates the message to be sent to node j for the $t+1^{\text{th}}$ iteration by:

$$m_{ij}^{t+1}(\mathbf{x}_{c_i} \cap \mathbf{x}_{c_j}) = \alpha \cdot \int_{\{\mathbf{x}_{c_i}\} \setminus \{\mathbf{x}_{c_j}\}} \varphi_i(y_i, \mathbf{x}_{c_i}) \prod_{k \in N(i) \setminus j} m_{ki}^t(\mathbf{x}_{c_i} \cap \mathbf{x}_{c_k}) \quad (3)$$

where α is a constant value to normalize the message. $N(i)$ denotes the neighbors of node i . At node i , we can also conclude the marginal probability of the variables in $\varphi_i(y_i, \mathbf{x}_{c_i})$. This is done by combining all the incoming messages with its local potential:

$$\hat{p}^t(\mathbf{x}_{c_i}) = \alpha \cdot \varphi_i(y_i, \mathbf{x}_{c_i}) \prod_{k \in N(i)} m_{ki}^t(\mathbf{x}_{c_i} \cap \mathbf{x}_{c_k}) \quad (4)$$

C. Form of the Messages

The computation in SPA is relatively simple if the messages and potential functions involved in (3) are discrete or they are a Gaussian distribution. However, in many cases, the local potential functions have a very complex non-Gaussian distribution and there exist high non-linear relationships between the variables. Discretizing the continuous functions (uniform sampling method) would be too expensive for many inference problems. Other forms of representation of the belief functions are needed.

A particle-based method, called non-parametric BP (NBP) is presented in [3] to solve self-localization problem in WSNs. In NBP, messages are presented by Gaussian particles

which are generated from the belief functions. This method enables the use of SPA. However, calculating products of Gaussian mixtures and generating proper samples is not a trivial task.

The following part introduces a novel implementation of messages in BP using FDA method.

III. FOURIER DENSITY APPROXIMATION

[5] and [6] derived the basic operations using Fourier density approximation. Here some important equations related to BP are briefly described.

A. Definition of Fourier Densities

A d -dimensional density function can be approximated by a d -dimensional Fourier expansion as

$$p(\vec{x}) = \sum_{\vec{k} \in \mathbf{K}} \gamma_{\vec{k}} e^{j\vec{k}^T \vec{x}} = \sum_{\vec{k} \in \mathbf{K}} (\alpha_{\vec{k}} + \beta_{\vec{k}}) e^{j\vec{k}^T \vec{x}}, \quad (5)$$

where $\vec{x} = [x_1, x_2, \dots, x_d]^T \in [-\pi, \pi]^d$ is a multidimensional variable. $\gamma_{\vec{k}} = (\alpha_{\vec{k}} + \beta_{\vec{k}})$ is the coefficients of the Fourier series. $\vec{k} = [\kappa_1, \kappa_2, \dots, \kappa_d]^T \in \mathbf{K}$ is an index vector, where $\mathbf{K} = \{-\kappa_1^o, -\kappa_1^o + 1, \dots, \kappa_1^o\} \times \dots \times \{-\kappa_d^o, -\kappa_d^o + 1, \dots, \kappa_d^o\}$ denotes the set of all valid indices [6].

In practice, the coefficients are obtained by the efficient Fast Fourier Transform (FFT) which has a complexity of $O(n \log n)$ where n denotes the number of sampling points.

B. Fourier Density Product

Given two densities $p^a(\vec{x})$ and $p^b(\vec{x})$, they are represented by the Fourier density approximation as:

$$p^a(\vec{x}) = \sum_{\vec{k} \in \mathbf{K}_a} \gamma_{\vec{k}}^a e^{j\vec{k}^T \vec{x}} \quad (6)$$

and

$$p^b(\vec{x}) = \sum_{\vec{k} \in \mathbf{K}_b} \gamma_{\vec{k}}^b e^{j\vec{k}^T \vec{x}}. \quad (7)$$

Their product can be expressed as:

$$p^c(\vec{x}) = p^a(\vec{x}) p^b(\vec{x}) = \sum_{\vec{k} \in \mathbf{K}_c} \gamma_{\vec{k}}^c e^{j\vec{k}^T \vec{x}}, \quad (8)$$

with

$$\gamma_{\vec{k}}^c = \sum_{\vec{\mu} \in \mathbf{K}_c} \bar{\gamma}_{\vec{\mu}}^a \bar{\gamma}_{\vec{\mu} - \vec{k}}^b, \quad (9)$$

where the bar denotes a valid index

$$\bar{\gamma}_{\vec{\mu}}^{(\cdot)} = \begin{cases} \bar{\gamma}_{\vec{\mu}}^{(\cdot)} & \vec{\mu} \in \mathbf{K}_{(\cdot)} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

of $\gamma_{\vec{k}}^a$ and $\gamma_{\vec{k}}^b$. The order of $p^c(\vec{x})$ is $\prod_{l=1}^d (\kappa_l^{o,a} + \kappa_l^{o,b})$, i.e. as many other approximation approaches like Gaussian mixture, the number of coefficient is significantly higher after production. But we can show later that the coefficient reduction in FDA is much easier than its counterparts.

C. Generalized Convolution Integral

Considering the Fourier densities of

$$p^a(\vec{x}) = \sum_{\vec{\mu} \in \mathbf{K}_{\vec{x}}} \gamma_{\vec{\mu}}^a e^{j\vec{\mu}^T \vec{x}} \quad (11)$$

and

$$p^b(\vec{y}, \vec{x}) = \sum_{\substack{\vec{\mu} \in \mathbf{K}_{\vec{y}}, \\ \vec{k} \in \mathbf{K}_{\vec{x}}}} \gamma_{\vec{\mu}, \vec{k}}^b e^{j\vec{\mu}^T \vec{x} + j\vec{k}^T \vec{y}}, \quad (12)$$

their generalized convolution integral is given by

$$p^c(\vec{y}) = \int p^b(\vec{x}, \vec{y}) p^a(\vec{x}) d\vec{x} = \sum_{\vec{k} \in \mathbf{K}_{\vec{x}}} \gamma_{\vec{k}}^c e^{j\vec{k}^T \vec{y}}, \quad (13)$$

with $\gamma_{\vec{k}}^c = \sum_{\vec{\mu} \in \mathbf{K}_{\vec{y}}} \gamma_{-\vec{\mu}}^a \gamma_{\vec{\mu}, \vec{k}}^b$.

Note that the order of resulting density only depends on the order of function $p^b(\vec{y}, \vec{x})$ not $p^a(\vec{x})$, which limits the computational complexity.

In addition, if the function $p^b(\vec{x}, \vec{y})$ has a form

$$p^b(\vec{x}, \vec{y}) = p^b(\vec{y} - \vec{x}), \quad (14)$$

(13) becomes

$$p^c(\vec{y}) = \int p^b(\vec{x}, \vec{y}) p^a(\vec{x}) d\vec{x} = \int p^b(\vec{y} - \vec{x}) p^a(\vec{x}) d\vec{x} = p^b(\vec{y}) * p^a(\vec{x}) \quad (15)$$

which is actually a convolution. Thus the coefficients of $p^c(\vec{y})$ are the multiplication of the coefficients of $p^b(\vec{y})$ and $p^a(\vec{x})$. In this way, the computation is simplified by replacing a high dimensional function $p^b(\vec{x}, \vec{y})$ with a low-dimensional function $p^b(\vec{y})$.

D. Coefficient Reduction

For many density mixture approximation approaches like Gaussian mixture, Dirac mixture or Monte Carlo methods, the number of coefficients increased exponentially after the product operation. Keeping all coefficients are practically impossible. Determining how many coefficients and which ones are needed is challenging. [13] provides a progressive way to calculate the parameters of mixture densities optimally. But the computational requirement is relatively high.

The coefficient reduction in FDA is relatively more efficient. As well known, the signal power in space domain and Fourier domain are equal. The Fourier coefficients ordered by their squared magnitude reflect the order of their influences to the square error between true density function and its Fourier approximation. Therefore, coefficient reduction in FDA is just deleting the coefficients with minimal squared magnitudes under the required density square error.

E. Ensuring Non-negativity

FDA with reduced coefficients is sometimes negative which brings problem for further calculation. [5] proposed to use the square root of density function instead of density function itself for calculation. In this way, the final approximated density is ensured to be non-negative.

F. Computational Complexity

Table I lists the comparison of computational complexities for density product and generalized convolution between FDA and uniform sampling method where m denotes the number of coefficients used by Fourier density

approximation. n is the number of uniform distributed samples. From this table, we see that the computation power is saved for the generalized convolution given the same number of m and n . By reducing the Fourier coefficients, both operations can be more efficient.

TABLE I
COMPUTATIONAL COMPLEXITY

	FDA	Uniform Sampling
Generalized Convolution	$O(m)$	$O(n^2)$
Product	$O(m \log m)$	$O(n)$

IV. A SENSOR LOCALIZATION EXAMPLE

In this chapter, we will use a sensor localization example to illustrate the BP method we proposed. Sensor localization is obtained by combining absolute positioning information (e.g. GPS) with relative distance information (e.g. time delay or power decay of the signal transmitted between sensors). In this paper, we restudy the self-calibration problem presented in [3] where each sensor has noisy measurements of its distances to neighboring nodes. The problem is formulated as a probabilistic inference problem that can be presented by probabilistic graphical model. BP algorithm is applied to exchange the calibration information between sensor nodes so that each sensor can obtain the MAP estimate of its location. Instead of Gaussian mixtures, FDA will be used to present the messages that are transmitted between nodes. Relative sensor geometry or the absolute sensor positions can be obtained depending on whether extra information about absolute positions is available at certain sensors.

A. System Model

Let's assume that we have a WSN with N sensors distributed in a planar space. The position of sensor i is denoted by \vec{x}_i . The measurement taken at sensor i about its distance to sensor j takes the form:

$$d_{ij} = \|\vec{x}_i - \vec{x}_j\| + v_{ij} \quad (16)$$

where d_{ij} denotes the observation, v_{ij} is additive Gaussian noise with zero mean and standard deviation of σ . $\|\vec{x}_i - \vec{x}_j\|$ calculates the Euclidean distance between two points. d_{ij} is not always available since sensor i does not always detect its neighbor j . We use a binary random variable o_{ij} to indicate whether a distance measurement is available, i.e. $o_{ij} = 1$ when observation is made, $o_{ij} = 0$ otherwise. According to [3], the probability that distance between sensor i and j is available with a probability of:

$$p(o_{ij} = 1 | \vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^p}{R_i^p}\right) \quad (17)$$

Furthermore, each sensor has a prior knowledge about its position, which is given by a prior distribution $p(\vec{x}_i)$. The prior distribution is normally uninformative unless the sensor has obtained its position information from other resources, e.g. GPS signal. In this case, the prior distribution might look

like a Dirac function.

B. Belief Propagation in Sensor Localization

Apparently, the assumptions mentioned in Section II are valid for this model. The joint distribution of the sensor locations $\{\vec{x}_i\}$ and the observations $\{d_{ij}\}$ and $\{o_{ij}\}$ can be factorized as:

$$p(\{\vec{x}_i\}, \{d_{ij}\}, \{o_{ij}\}) = \prod_{(i,j)} p(o_{ij} | \vec{x}_i, \vec{x}_j) \prod_{(i,j): o_{ij}=1} p_v(d_{ij} | \vec{x}_i, \vec{x}_j) \prod_i p(\vec{x}_i) \quad (18)$$

Based on (7), we can define the local potential for sensor i :

$$\phi_i(\vec{x}_1, \dots, \vec{x}_N) = p(\vec{x}_i) \cdot \prod_{j:j \neq i} p(o_{ij} | \vec{x}_i, \vec{x}_j) \prod_{j:j \neq i, o_{ij}=1} p_v(d_{ij} | \vec{x}_i, \vec{x}_j) \quad (19)$$

so that each sensor has now its local potential function.

Distributed inference can be done by using SPA. For sensor location problem, the message updating equation, obtained from (3) and (19), takes the form:

$$m_{ij}^{t+1}(\vec{x}_1, \dots, \vec{x}_N) = \alpha \cdot \phi_i(\vec{x}_1, \dots, \vec{x}_N) \prod_{k \in N(i) \setminus j} m_{ki}^t(\vec{x}_1, \dots, \vec{x}_N) \quad (20)$$

Each message in (20) involves N variables. The presentation of messages and the multiplication of messages will be too complicated that it makes the inference intractable. To simplify the problem, we define a message from node i to node j to be a function that only involves \vec{x}_j . In another word, message from node i to node j only contains a summary of sensor i 's belief on the position of j , position information about other sensor nodes are summed out. Based on this simplification, (20) will be revised to:

$$\begin{aligned} m_{ij}^{t+1}(\vec{x}_j) &= \alpha \cdot \int_{\{\vec{x}_1, \dots, \vec{x}_N\} \setminus \vec{x}_j} \phi_i(\vec{x}_1, \dots, \vec{x}_N) \prod_{k \in N(i) \setminus j} m_{ki}^t(\vec{x}_1, \dots, \vec{x}_N) \\ &= \alpha \cdot \int_{\{\vec{x}_1, \dots, \vec{x}_N\} \setminus \vec{x}_j} \left\{ p(\vec{x}_i) \cdot \prod_{k:k \neq i} p(o_{ik} | \vec{x}_i, \vec{x}_k) \cdot \right. \\ &\quad \left. \prod_{k:k \neq i, o_{ik}=1} p_v(d_{ik} | \vec{x}_i, \vec{x}_k) \prod_{k \in N(i) \setminus j} m_{ki}^t(\vec{x}_i) \right\} \\ &= \alpha \cdot \int_{\vec{x}_i} p(\vec{x}_i) \cdot \phi_{ij}(\vec{x}_i, \vec{x}_j) \prod_{k \in N(i) \setminus j} m_{ki}^t(\vec{x}_i) \end{aligned} \quad (21)$$

where $\phi_{ij}(\vec{x}_i, \vec{x}_j)$ is defined as:

$$\phi_{ij}(\vec{x}_i, \vec{x}_j) = \begin{cases} p(o_{ij} = 1 | \vec{x}_i, \vec{x}_j) \cdot p(d_{ij} | \vec{x}_i, \vec{x}_j) \\ 1 - p(o_{ij} = 1 | \vec{x}_i, \vec{x}_j) \end{cases} \quad (22)$$

The marginal probability of sensor location is given by:

$$\hat{p}_i^{t+1}(\vec{x}_i) = \alpha \cdot p(\vec{x}_i) \prod_{k \in N(i)} m_{ki}^t(\vec{x}_i) \quad (23)$$

Although the complexity of messages has been greatly simplified in (21), calculation in (21) and (23) is still complicated because of its non-linearity and the non-Gaussian distribution. To solve this problem, we use FDA method to approximate the density functions and present the messages as a collection of Fourier components and their coefficients.

C. Algorithm Description

Using FDA and the coefficient reduction method introduced in Section III, the size of the messages are significantly reduced. This has brought benefits in two folds. On one side, it reduces the transmission power. On the other hand, it

reduces the complexity of the SPA with a penalty of computing FFT.

We propose two algorithms. The ST-FDA algorithm, depicted in Table II, uses FDA only to reduce the transmission power. SCT-FDA, depicted in Table III, does all the calculation in the frequency domain thus reduces both the transmission power and the computational complexity.

TABLE II
DESCRIPTION OF ST-FDA ALGORITHM

ST-FDA
1. Discretize the local potential functions.
2. Initialize messages, e.g. a vector of ones.
3. Calculate the outgoing message using (21). Since now the potential functions and the messages are discrete, we replace the integral in (21) with sum. Use FFT to transform the outgoing message into the frequency domain and use coefficient reduction method introduced in Section III to reduce the size of the messages.
4. Once a new message (presented by Fourier coefficients) is received, an IFFT will be used to change the message to the 2D space domain for the SPA.
5. Run SPA for a defined number of iterations.
6. Posterior probability can be calculated by using (23).

TABLE III
DESCRIPTION OF SCT-FDA ALGORITHM

SCT-FDA
1. Discretize the local potential.
2. Initialize messages, e.g. a vector of ones.
3. Use FFT to transform all messages and potential functions to frequency domain. Use coefficient reduction method (Section III D) to reduce the number of Fourier components. All messages stay in frequency domain until the end of the algorithm.
4. The SPA of (21) in the frequency domain is implemented by using (8) and (15). Coefficient reduction is done in each step.
5. Run SPA for a defined number of iterations.
6. Finally, use IFFT to convert the posterior probability from frequency domain into space domain.

V. SIMULATION RESULTS

To verify the performance of the FDA based BP methods, we simulate the BP for self-localization problem in a WSN that is illustrated in Fig. 1.

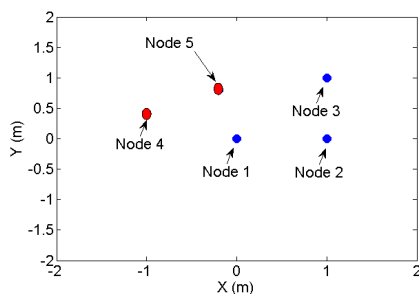


Fig. 1 Sensor Distribution.

The positions of sensor node 1, 2 and 3 are known as (0, 0), (1, 0) and (1, 1) respectively. Unknown sensor nodes 4 and 5 are located at (-1, 0.4) and (-0.2, 0.8). Note that although the

Fourier densities are defined in $[-\pi, \pi]^d$ in (5), the definition in a large area can be also derived by a simple linear mapping. In this paper, we limit the area to $[-\pi, \pi]^2$ for simplicity.

The parameter ρ and R_1 in (17) are set to 2 and 3m respectively. The standard deviation of distance measurements σ in (16) is set to 0.4m. The BP is forced to stop after 7 iterations.

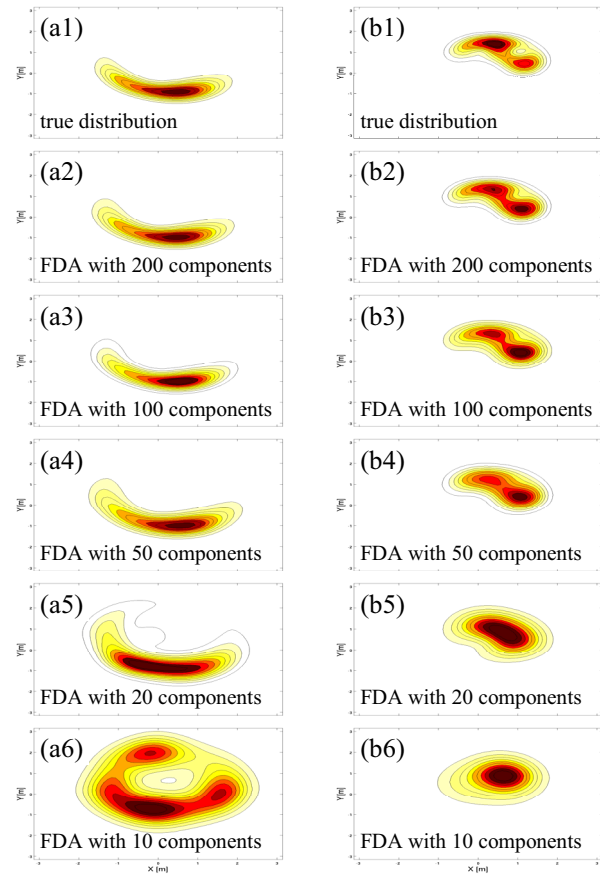


Fig. 2. Comparison of distribution estimates by SCT-FDA with different number of coefficients

Fig. 2 depicts the estimates of posterior distribution of sensor positions at node 4 (Fig. 2 (a2)–(a6)) and node 5 (Fig. 2 (b2)–b(6)) by SCT-FDA using different number of Fourier coefficients to represent a single potential function or a message and compare them with the true result generated by uniform sampling based method (Fig. 2 (a1) and (a2)). The sampling resolution is 65x65 for all experiments. From the results we can see with 100 Fourier coefficients, the approximation is already very close to the true value, whereas too few components can not fully characterize the very non-linear, non-Gaussian distribution.

Fig. 3 shows the estimation results from SCT-FDA algorithm using different sampling resolutions. Sampling resolution of 15x15, 25x25, 35x35, 65x65 are applied to Fig. 3 (a1) to (a4) and Fig. 3 (b1) to (b4) respectively. The sampling resolution determines the precision of the estimate. According to *Nyquist Theorem*, original function can be recovered from its samples only if the sampling rate is greater than twice the maximum frequency of that function. Bad results can be observed from Fig. 3 (b3) and (b4) because the

sampling rate is too low.

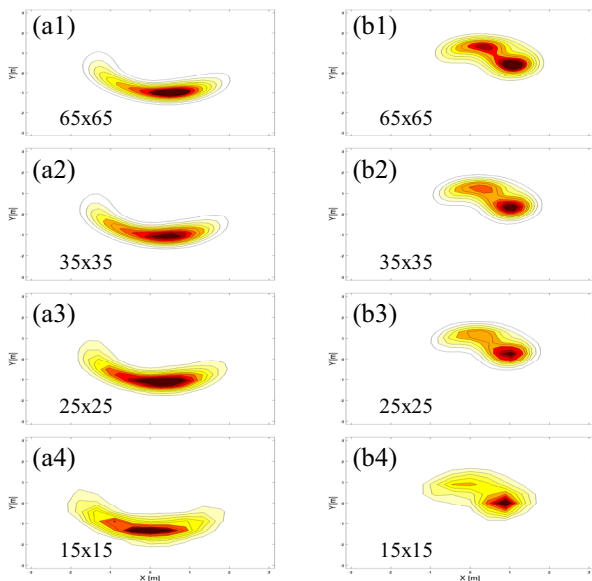


Fig. 3. Comparison of distribution estimate by SCT-FDA from different sample resolutions

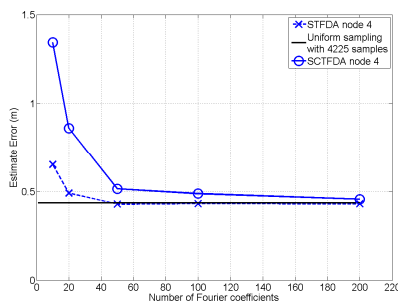


Fig. 4. Comparison of estimation errors in uniform sampling, STFDA and SCT-FDA.

Fig. 4 plots the error of the position estimate of node 4 vs. number of Fourier coefficients curves of STFDA and SCT-FDA algorithms. It can be seen that, increasing number of coefficients results in a better performance for both methods. But compared to the result by uniform sampling which requires 4225 samples to represent a message, a close result is achieved by much less Fourier coefficients. In the simulation, 119 messages are transmitted. If sample resolution is 65x65 and 50 Fourier coefficients are kept, ST-FDA and SCT-FDA methods transmit 23800 components while uniform sampling based method has to transmit 502775 components. Obviously, Fourier density approximation significantly reduces the transmission power.

ST-FDA method outperforms SCT-FDA methods in Fig. 4 because approximation is only made for the transmission in ST-FDA while SCT-FDA method also greatly simplifies the computation by using fewer coefficients in the SPA. Although SCT-FDA loses some accuracy, it saves the computation power and time. Furthermore, note that ST-FDA performs FFT and IFFT at the transmission and reception of each message while in SCT-FDA, FFT is only performed at the beginning and the end of BP, which further reduces the complexity.

VI. CONCLUSIONS AND FUTURE WORK

WSNs can be modeled by graphical models, where BP algorithm provides a promising solution. However, in WSNs, the computational ability and battery life of sensor nodes are limited. The intensive probability density computation and transmission between nodes required by BP make a big problem. This paper presents a method to use Fourier density approximation to represent belief densities. ST-FDA algorithm uses Fourier approximation to compress the complex non-Gaussian densities in order to reduce the radio transmission which is regarded as the most power consuming part in WSNs. Another algorithm SCT-FDA implements the SPA in Fourier domain so that it saves power consumptions not only in transmissions but also in belief calculations.

ST-FDA and SCT-FDA use a fixed number of Fourier coefficients. A more general algorithm with adaptive Fourier coefficient reduction can be investigated. In addition, other density representation like wavelet approximation could also be considered for the same application. The comparison between Fourier approximation and standard Gaussian mixture representation in [3] would also be interesting.

REFERENCES

- [1] H. Gharavi and S. Kumar (Eds.), "Special issue on sensor networks and applications," in *Proceedings of the IEEE*, 91(8), August 2003.
- [2] S. Kumar, F. Zhao, and D. Shepherd (Eds.), "Special issue on collaborative information processing," in *IEEE Signal Processing Magazine*, 19(2), March 2002.
- [3] A. Ihler, J. Fisher, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *Proc. IPSN*, 2004.
- [4] R. Kronmal and M. Tarter, "The estimation of probability densities and cumulatives by Fourier series methods," in *Journal of the American Statistical Association*, vol. 63, no. 323, pp. 925-952, September 1968.
- [5] D. Brunn, F. Sawo and U. D. Hanebeck, "Efficient nonlinear Bayesian estimation based on Fourier densities," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, 2006, Germany.
- [6] D. Brunn, F. Sawo, U. D. Hanebeck, "Nonlinear multidimensional Bayesian estimation with Fourier densities," in *Proceedings of the 2006 IEEE Conference on Decision and Control (CDC 2006)*, pp. 1303-1308, San Diego, California, December, 2006.
- [7] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.
- [8] P. Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems*, pages 19-32. Oxford University Press, Oxford, 1990.
- [9] M. Paskin, C. Guestrin. A robust architecture for distributed inference in sensor networks. Intel Research, Tech. Rep. IRB-TR-03-039, 2004.
- [10] Aji, S. M., and McEliece, R. J. The generalized distributive law. *IEEE Transactions on Information Theory* 46 (March 2000), 325-343.
- [11] K. Murphy, Y. Weiss, and M. Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence 15*, pages 467-475, July 1999
- [12] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor graph and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol.47, no.2, pp.498-518, Feb. 2001.
- [13] U. D. Hanebeck, "Progressive Bayesian estimation for nonlinear discrete-time systems: the measurement step", in *Proceedings of the 2003 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2003)*, pp. 173-178, Tokyo, Japan, July, 2003.