A New Variant of Nonparametric Belief Propagation for Self-Localization

Hadi Noureddine*†, Nicolas Gresset*, Damien Castelain*, Ramesh Pyndiah†
*Mitsubishi Electric R&D Centre Europe, France
†Telecom Bretagne, France

***Abstract*—We consider the problem of relative self-localization of a network of fixed communicating devices that evaluate range measurements between each other. The solution is obtained in two stages: First, a new variant of the Nonparametric Belief Propagation algorithm is used for estimating the beliefs. This variant is based on a Monte-Carlo integration with rejection sampling where a delimited space region is determined for each node in order to reduce the rejection ratio. Then, a new algorithm based on estimation in discrete states space is proposed for solving the flipping ambiguities resulting from the lack of measurements. This solution has the advantage of reducing the amount of communicating particles and the computation cost.**

I. Introduction

Finding the positions of a set of wireless communicating devices has a lot of practical applications, going from the deployment of ad-hoc networks and its related topics, e.g. communication enhancement and location-based routing, toward a variety of location-based services and applications, e.g. military, environmental and health.

The communicating devices may take several forms, such as sensors, femto cells, access points, etc., with indoor or outdoor deployment. They might be subject to several constraints on their size, power consumption and price. Thus, developing GPS-free localization techniques is capital.

If pairs of nodes perform measurements relevant to their relative positions, they can be localized in a coordinate system. Several names have been attributed to this topic in the literature, such as 'network calibration', 'cooperative' and 'self-localization'.

Several centralized and distributed algorithms have been developed in order to solve this localization problem. For centralized algorithms, measurements are collected to a central processor where the overall processing is done. One example of such an algorithm is the ML estimation [1][2], which can be applied if the statistical model of the measurements is known.

In distributed algorithms, all the nodes are involved in the estimation process, and the computation is distributed among them. These algorithms are most useful for large networks. In [3], a node estimates its distance to several reference nodes according to the number of hops of the shortest connection path. The positions are then found by multilateration. In [4][5], successive refinement is processed, where one node refines and sends its estimate to its neighbors at each iteration. The Belief Propagation (BP) algorithm is based on probabilistic graphical models [6][7] where each node calculates the probability density function of its coordinates, based on prior information, measurements and probability densities provided at each iteration by neighboring nodes. This algorithm produces both an estimate of locations and metrics of uncertainties.

In this paper, we are interested in Nonparametric Belief Propagation (NBP)[8], which is a particle-based version of the BP.

In the first phase of our solution, NBP is implemented using a Monte-Carlo integration instead of the stochastic method of [9]. Furthermore, the samples are selected from the beliefs by using rejection sampling. The errors on measurements are supposed to lie in known intervals, which allows for constructing limited space regions for each node and thus reducing the rejection ratio.

In the second phase, we propose an algorithm for mitigating the flipping ambiguities that result from the lack of measurements. This algorithm is based on K-means clustering and estimation in discrete-valued states space, and has the advantage of drastically reducing the computation complexity and the amount of data to be exchanged.

The rest of the paper is organized as follows: In section II, the problem is formulated as a graphical model, and our implemented variant of the NBP is presented. In section III, we present a new method for solving the flipping ambiguities remaining after convergence of the NBP algorithm. Simulation results and conclusions are provided in sections IV and V.

II. Probabilistic graphs application to Self-localization

In this section, we consider the belief propagation applied to self-localization. We assume that we have N fixed nodes scattered in a planar space, and only consider 2D localization. In addition, we only consider relative localization as no node knows its absolute position.

Each node obtains distance measurements with the set of its neighboring nodes, and these measurements are corrupted by an additive error. Nodes are mutually neighbors, and the relationship between the nodes can be described by an undirected graph.

We assume that neighboring nodes share the same observation on their distance. Let $\mathbf{x}_i$ denote the two-dimensional position of node i, and $\tilde{d}_{ij}$ the noisy distance measurement with its neighbor j. The joint a posteriori probability distribu-

tion factorizes as:

$$p\left(\mathbf{x}_1,\cdots,\mathbf{x}_N/\left\{\tilde{d}_{ij}\right\}\right)\propto\prod_{i\in V}\Phi_i(\mathbf{x}_i)\prod_{j\in\Omega(i),i<j}\Psi_{ij}(\mathbf{x}_i,\mathbf{x}_j) \tag{1}$$

where V is the nodes set and $\Omega(i)$ is the set of neighbors of node i. $\Psi_{ij}(\mathbf{x}_i,\mathbf{x}_j)=p(\tilde{d}_{ij}/\mathbf{x}_i,\mathbf{x}_j)$ is a pairwise potential function and $\Phi_i(\mathbf{x}_i)=p_i(\mathbf{x}_i)$ is the a priori probability on the location of node i.

Two approaches are possible for estimating the positions of the nodes:

- Find the joint maximum a posteriori (MAP) of all $\mathbf{x}_i$'s, or in other words, the sequence of states $\{\mathbf{x}_i\}$ maximizing (1). For example, the Max-Product algorithm finds this most likely sequence of states.
- Find the MAP of each $\mathbf{x}_i$ apart. This can be done by a marginalization of (1) so as to obtain the a posteriori probability distribution at each node. For example, the Sum-Product algorithm is a way for evaluating the marginalization.

In the following, the Sum-Product is considered so as to determine the belief of each node for a given position, and is described in the following subsection.

A. Belief Propagation

The previously described model can be qualified as a probabilistic graphical model, in which a node represents a random variable or a parameter to be estimated, and an edge expresses the existence of a probabilistic relationship, or a compatibility, between two nodes.

Belief Propagation (BP) is an iterative message passing algorithm that calculates the posterior marginalization at each node. At the n-th iteration, each node computes its belief by taking the product of its local potential and the incoming messages from its neighbors as follows:

$$\hat{p}^{(n)}(\mathbf{x}_i)\propto\Phi_i(\mathbf{x}_i)\prod_{k\in\Omega(i)}m_{ki}^{(n)}(\mathbf{x}_i) \tag{2}$$

The message from node j to node i, called the update rule, is:

$$m_{ji}^{(n)}(\mathbf{x}_i)\propto\int\Phi_j(\mathbf{x}_j)\Psi_{ij}(\mathbf{x}_i,\mathbf{x}_j)\prod_{k\in\Omega(j)\backslash i}m_{kj}^{(n-1)}(\mathbf{x}_j)d\mathbf{x}_j \tag{3}$$

where $\Omega(j)\backslash i$ is the set of neighbors of j except i. All messages are initialized to an arbitrary value, for example 1.

In the case of graphs without loops, it is known that this algorithm perfectly computes the marginal probability distributions, and the needed number of iterations is equal to the graph diameter. If loops occur in the graph, good approximations of the marginal probability distributions are observed under some conditions [10].

The integral equation (3) can be evaluated when the variables are discrete valued or in the case of Gaussian distributions. When these conditions are not fulfilled, the integral equation rarely has tractable analytic solution and must be replaced by an approximation, such as for the Nonparametric Belief Propagation. As relative positioning is considered here, $\Phi_i(\mathbf{x}_i)$ will be dropped from the equations given above.

B. Nonparametric Belief Propagation

Nonparametric Belief Propagation [8] is based on stochastic methods for propagating kernel-based approximations of the continuous messages. In this algorithm, we propagate a set of values $\{\mathbf{r}_{ji}^{(l)}\}_{l=1}^{M}$ from node j to node i, where $\mathbf{r}_{ji}^{(l)}\sim\Psi_{ij}(\mathbf{x}_i,\mathbf{s}_{ji}^{(l)})$ is a sample taken from Ψ_{ij} for a position sample $\mathbf{s}_{ji}^{(l)}$ of node j. The set of position samples $\{\mathbf{s}_{ji}^{(l)}\}_{l=1}^{M}$ are drawn from the beliefs with an association of weights. The message m_{ji} is then formed by placing identical Gaussian kernels about the points $\{\mathbf{r}_{ji}^{(l)}\}$ which requires an appropriate choice of the kernel covariance matrix. The belief function, computed by taking the product of the incoming messages, becomes a Gaussian mixture with a huge number of components. In the case where the potentials are Gaussian mixtures, [11] proposed to use Monte-Carlo integration for estimating the message equation (3).

In relative positioning, we consider that each node lies in a known limited region of space. This region is obtained by assuming that the measurement error is constrained to a known interval, with a good probability. This allows for the application of rejection sampling in drawing independent samples. Thus, we propose to perform a Monte-Carlo integration of equation (3) without resorting to Gaussian mixtures approximations and kernel covariance matrix choice.

A Monte-Carlo integration of equation (3) yields $\tilde{m}_{ji}$, an approximation of m_{ji}, by drawing M samples $\{\mathbf{s}_{ji}^{l}\}_{l=1}^{M}$ from $p_{ji}^{(n)}$ defined as:

$$p_{ji}^{(n)}(\mathbf{x}_j)\propto\prod_{k\in\Omega(j)\backslash i}m_{kj}^{(n-1)}(\mathbf{x}_j) \tag{4}$$

which can be considered as a probability density function. In general, we can draw the samples from any density function $g_{ji}(\mathbf{x}_j)$ that does not vanish when $p_{ji}^{(n)}$ does not. The message $\tilde{m}_{ji}$ then becomes the weighted mixture:

$$\tilde{m}_{ji}^{(n)}(\mathbf{x}_i)=\frac{1}{\sum_{M}^{k=1}\pi_{ji}^{k}}\sum_{l=1}^{M}\pi_{ji}^{l}\Psi_{ij}(\mathbf{x}_i,\mathbf{s}_{ji}^{l}) \tag{5}$$

where $\pi_{ji}^{l}=p_{ji}^{(n)}(\mathbf{s}_{ji}^{l})/g_{ji}(\mathbf{s}_{ji}^{l})$ is the weight associated to sample $\mathbf{s}_{ji}^{l}$. We choose $g_{ji}(\mathbf{x}_j)$ equal to the belief of node j:

$$g_{ji}^{(n)}(\mathbf{x}_j)\propto\prod_{k\in\Omega(j)}\tilde{m}_{kj}^{(n-1)}(\mathbf{x}_j) \tag{6}$$

This function is the same for all $i\in\Omega(j)$.

When compared to existing relative positioning techniques using NBP [9][12], we propagate the generated samples $\mathbf{s}_{ji}^{l}$ from the node j to all its neighbors and sampling at node j is done only once. Thus, we don't have to sample from the different potentials. Furthermore, we don't have to estimate densities for the relative directions, in order to concentrate the samples in regions of interest, as done by [9] in order to

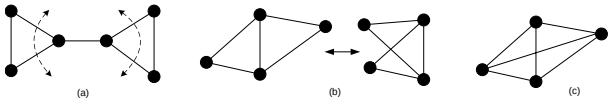

Fig. 1. Three graphs in 2-dimensions. (a) is flexible and can be continuously deformed. (b) is rigid and can have only discontinuous deformations. (c) is globally rigid and cannot be deformed.

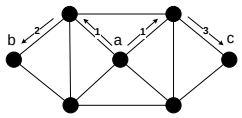

Fig. 2. A rigid network. Message from node (a) has to be sent three times before reaching nodes (b) and (c).

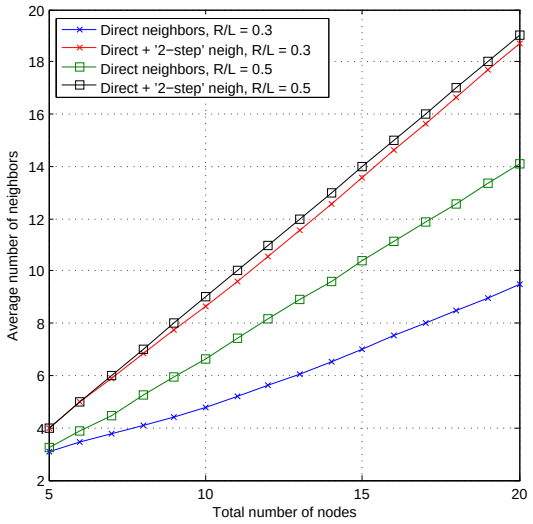

Fig. 3. Average number of neighbors vs Total number of nodes for R/L = 0.3 and R/L = 0.5.

alleviate the fact that potentials do not contain information on the directions.

As a remark, the associated weights $\pi_{ji}^{l} = 1/\tilde{m}_{ij}^{(n-1)}(\mathbf{s}_{ji}^{l})$ could be calculated locally at node i and not propagated.

III. Dealing with ambiguities

The problem of relative localization can be resolved up to congruence, i.e., translation, rotation or reflection of the whole network. Firstly, we define the node 1 as the origin in order to remove this ambiguity. We attribute the coordinates vector $\mathbf{x}_1 = (0,0)^T$ to this node. Secondly, the node 2 is set on the positive half of the x-axis, $\mathbf{x}_2 = (x_2, 0)^T$ and $x_2 > 0$. Finally, the node 3 is set in the half plane with positive y-component, $y_3 > 0$. After verifying these three conditions, the region of the space where each node can lie can be determined based on the hypothesis that measurement errors are bounded.

It is important to understand the conditions under which the problem is solvable. For example, in the case of lack of measurements, the network can be subject to deformation as shown in Fig. 1(a), where any rotation of the two pairs of two points on the extrema around the center points lead to a possible solution. A sufficient condition for obtaining a unique solution is to observe a globally rigid graph of the network [13][14]. In this paper, we only consider rigid graphs, where discontinuous deformations are possible as shown in Fig. 1(b). An efficient algorithm for testing graph rigidity [15], called *The Pebble game*, is implemented in our simulations. This algorithm can also identify all rigid subgraphs and its complexity is at most $O(N^2)$.

A. Flipping Ambiguities

Discontinuous deformations create a kind of ambiguity on the solution that we call flipping ambiguity. It follows that the beliefs of some nodes occur to be multimodal. In Fig. 4, a network of 7 nodes is plotted altogether with the region of each node. Nodes 5 and 7 can be flipped, resulting in 4 possible solutions for node 7. Thus its belief has four modes as is illustrated in Fig. 5.

In this subsection, we present the sate-of-the-art for solving ambiguities, as introduced in [9]. The fact that two nodes are not neighbors gives the additional information that they are probably far one from the other. This information will be exploited for solving ambiguities. We note $P_o(\mathbf{x}_i, \mathbf{x}_j)$ the probability for two nodes to be neighbors one of the other. This probability is a function of the communication channel quality, and mainly of the distance between the two nodes. We do not investigate thoroughly the communication performance and consider the following simplified model for $P_o(\mathbf{x}_i, \mathbf{x}_j)$ [9]:

$$P_o(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-0.5\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / R^2\right) \tag{7}$$

In [9][12], P_o is included in (1) and in the NBP exchanged messages that occur with direct and '2-step' neighbors. The '2-step' neighbors of node i are the set of neighbors of neighbors of i (except i). For the latter, potentials are taken as $1 - P_o$.

One drawback of this method is the complexity and overhead of exchanged messages. Indeed, if the nodes perform a broadcasting, the number of broadcast operations at some nodes should at least be doubled before the messages reach '2-step' neighbors. For example, the message from node a in Fig. 2 has to be broadcast three times before attaining nodes b and c. Furthermore, samples drawing and beliefs computation become more complicated, as the latter are constructed by taking the product of all incoming messages, whether from direct or '2-step' neighbors. Fig. 3 shows the average number neighbors in rigid networks. Nodes are drawn uniformly in an $L \times L$ square, and the connectivity is constructed according to (7) and independently for each pair of nodes. It shows that the number of neighbors is much increased when considering direct and '2-step' ones, especially when R/L is small.

B. A clustering-based disambiguiting algorithm

We propose a new algorithm for solving flipping ambiguities which reduces both the exchanges overhead and computation complexity. It is applied in a second phase after finding the

beliefs with the NBP, during which we considered only direct neighbors. The algorithm is composed of the following steps:

1) We first identify the different beliefs modes. In order to do so, we apply K-means clustering [16] on the samples, which is particularly relevant as the samples tend to be concentrated around the modes. As a remark, the farther we go from the node 1 located on the origin, the higher will be the number of possible flips. We propose to take the number of clusters proportional to the smallest number of hops to node 1. We can also take a constant overestimated number of clusters. Other methods for automatically determining the number of clusters from the samples are described in [17].
2) For each cluster, we retain only the sample that has the maximum belief. For example, in Fig. 6, clustering is done for the samples of node 7, where four clusters are considered.
3) At this point, each node will have a small set of points that include the belief's modes. We apply a discrete version of the BP to find, again, the beliefs of these retained points, with involving the '2-step' neighbors this time.
   - We can use the Sum-Product rules, and in that case the messages are:

$$m_{ji}^{(n)}(\mathbf{s}_i^q) = \sum_{l=1}^{|S_j|} \Psi_{ij}(\mathbf{s}_i^q, \mathbf{s}_j^l) \prod_{k \in \Omega(j) \setminus i \cup \Omega_2(j)} m_{kj}^{(n-1)}(\mathbf{s}_j^l) \tag{8}$$

     where S_i is the set of retained points of node i after clustering, $q = 1, \cdots, |S_i|$ and $\Omega_2(i)$ is the set of '2-step' neighbors of i. We compute $\Psi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = p(\tilde{d}_{ij}/\mathbf{s}_i, \mathbf{s}_j) P_o(\mathbf{s}_i, \mathbf{s}_j)$ for direct neighbors, and $\Psi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = 1 - P_o(\mathbf{s}_i, \mathbf{s}_j)$ for '2-step' ones.
   - If the Max-Product is used instead, the messages are:

$$m_{ji}^{(n)}(\mathbf{s}_i^q) = \max_{l=1}^{|S_j|} \Psi_{ij}(\mathbf{s}_i^q, \mathbf{s}_j^l) \prod_{k \in \Omega(j) \setminus i \cup \Omega_2(j)} m_{kj}^{(n-1)}(\mathbf{s}_j^l) \tag{9}$$

4) The beliefs at node i are computed with:

$$\hat{B}^{(n)}(\mathbf{s}_i^q) = \prod_{k \in \Omega(i) \cup \Omega_2(i)} m_{ki}^{(n)}(\mathbf{s}_i^q) \tag{10}$$

5) The estimated position is taken as the point with the maximum belief:

$$\hat{\mathbf{x}}_i = \text{argmax}_{\mathbf{s}_i^q \in S_i} \hat{B}^{(n)}(\mathbf{s}_i^q) \tag{11}$$

With this algorithm, the '2-step' neighbors are implicated in the message exchange process, but the amount of data contained in the message is much smaller than that of the first phase NBP.

For the network of Fig. 4, the estimated positions are plotted in Fig. 7. The crosses represent the estimates from the NBP without applying the second stage algorithm. Circles represent the estimates after the second stage of disambiguiting where the Max-Product algorithm is applied.

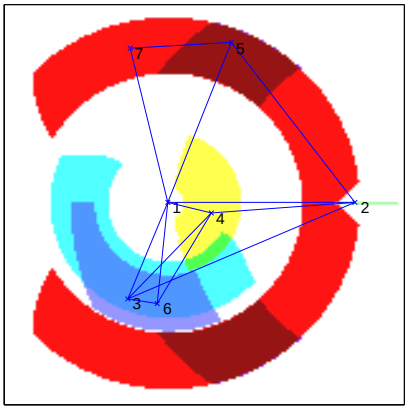

Fig. 4. A network of 7 nodes. The region of each node is represented with a color. Nodes 5 and 7 can be flipped causing an ambiguity.

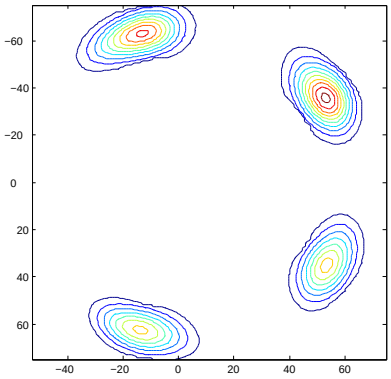

Fig. 5. A contour plot of the belief of node 7 after 4 iterations. It has 4 modes.

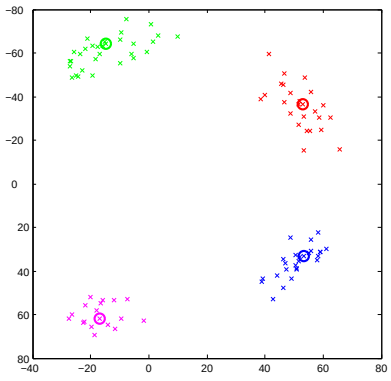

Fig. 6. Clustering of the samples of node 7. A color is attributed to each cluster. These samples are drawn from the belief after 4 iterations.

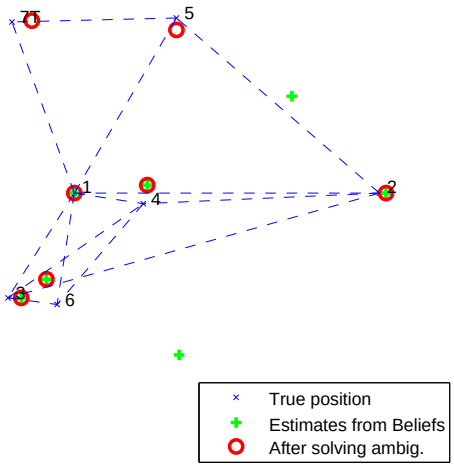

Fig. 7. Estimates of the positions of nodes of Fig. 4, with and without solving ambiguities

IV. Simulations

To measure the performance of the localization algorithm, we use a metric called Global Energy Ratio (GER) [18], given by (12).

$$\text{GER} = \sqrt{\frac{\text{mean}\left(\sum_{i<j} \hat{e}_{ij}^2\right)}{N(N-1)/2}} \tag{12}$$

where $\hat{e}_{ij} = (\hat{d}_{ij} - d_{ij})/d_{ij}$ is the normalized error, d_{ij} is the true distance and $\hat{d}_{ij}$ is the distance in the algorithm's result. This metric measures the performance compared to the true configuration topological properties, by taking into account all the distances, whether measured or not. The method developed in the previous section is compared to the ML estimate. In order to make a fair comparison, the density function $P_o(\mathbf{x}_i, \mathbf{x}_j)$ is included in the joint probability distribution. We also include $1 - P_o(\mathbf{x}_i, \mathbf{x}_j)$ for the '2-step' neighbors.

The *Pebble Game* algorithm is implemented to identify the rigid graphs, for which the localization is done. The measurement errors follow a truncated Gaussian distribution, with variance σ , and interval $[-a, a]$. The region of space where each node can exist is determined according to the measurements.

Note that truncated Gaussian potentials will cause that the belief given by (6) will have a support different than that of the density function (4). To circumvent this problem, we make a relaxation of the potentials by taking them as Gaussian during the application of the NBP and discrete BP algorithms.

In Fig. 8, the GER is plotted vs the iteration number for rigid networks of 10 nodes. The error interval is $[-7; 7]$ and the variance is 3. The connectivity is established according to (7) with $R/L = 0.2$ and independently for each pair of nodes and $L \times L$ is the total considered area. It shows that the GER is much better in the cases where disambiguiting is applied. It also shows that the Max-Product algorithm performs better than the Sum-Product.

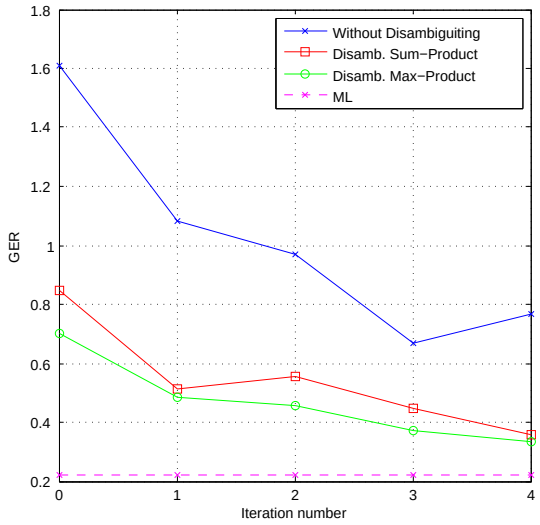

Fig. 8. GER vs Iteration number for a network of 10 nodes, R/L = 0.2 , error interval = [-7;7] and $\sigma = 3$.

In Fig. 9, the average rejection ratio for rigid networks of 10 nodes is plotted. It shows that this ratio increases with iteration number. This can be justified by the fact that beliefs become more tightened around their modes as the nodes gather more information about their locations. Here, the samples are drawn uniformly from the determined regions before applying the rejection test.

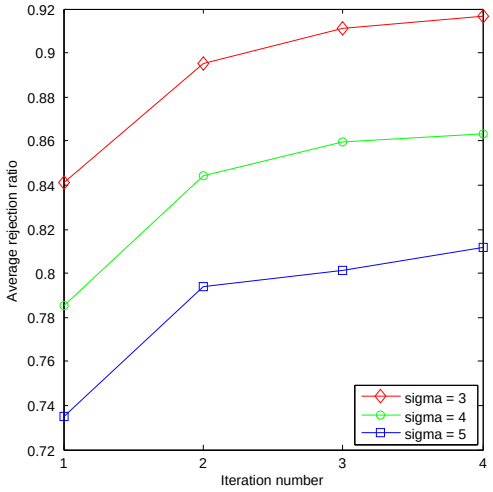

Fig. 9. Rejection ratio as a function of the number of iterations. The error is taken to be a truncated Gaussian with interval [-10;10] and variance σ.

V. Conclusions

In this paper, we presented the problem of sensor networks localization with a use of a graphical model. We also presented nonparametric belief propagation (NBP), and applied a new variant of this method, which is based on a Monte-Carlo estimation of the propagated messages. We used rejection sampling to draw samples from the nodes' beliefs. These

samples are taken from determined regions of the space where the nodes can exist. Other sampling methods can be investigated, such as the Metropolis-Hastings method [19], which is a Markov Chain Monte-Carlo method (MCMC), and the Ziggurat method [20]. If the rejection ratio is very high, rejection sampling can be applied to some of the samples. Then these points are considered as kernels centers, and the remaining samples are drawn from the probability density represented by the kernels.

Due to the fact that rigid graphs are subject to discontinuous deformations, ambiguities on nodes positions may exist. To deal with them, we proposed an algorithm, based on K-means clustering, which reduces both the communication and computation cost. The estimation result is much improved with this algorithm and the result is good in comparison to the ML.

VI. Acknowledgement

The work presented in this paper has been performed in the framework of the ICT-248894 WHERE2 project funded by the European Union.

References

[1] N. Patwari, A.O. Hero III, M. Perkins, N.S. Correal, and R.J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137 – 2148.

[2] R. Moses, D. Krishnamurthy, and R. Patterson. A Self-Localization Method for Wireless Sensor Networks. *Eurasip journal on applied signal processing*, 2003(4):348–358.

[3] D. Niculescu and B.; Nath. Ad hoc positioning system (APS). In *IEEE Global Telecommunications Conference, 2001. GLOBECOM '01*, volume 5, pages 2926 – 2931, 2001.

[4] A. Savvides, H. Park, and M.B. Srivastava. The Bits and Flops of the N-hop Multilateration Primitive For Node Localization Problems. pages 112–121, 2002.

[5] B. Denis and N. Daniele. NLOS ranging error mitigation in a distributed positioning algorithm for indoor UWB ad-hoc networks. In *International Workshop on Wireless Ad-Hoc Networks, 2004*, pages 356 – 360, 2004.

[6] C. Bishop. *Pattern Recognition And Machine Learning*. Springer-Verlag New York Inc., 2007.

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* . Morgan Kaufmann, 1988.

[8] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric Belief Propagation. In *IEEE CVPR 2003*, volume 1, pages 605–612, June 2003.

[9] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. *IEEE Journal of Selected Areas in Communication*, 2005.

[10] Y. Weiss. Belief Propagation and Revision in Networks with Loops. Technical report, MIT, Cambridge, MA, USA.

[11] M. Isard. PAMPAS: real-valued graphical models for computer vision. In *IEEE CVPR 2003*, volume 1, pages 613–620, June 2003.

[12] V. Savic and S. Zazo. Nonparametric Boxed Belief Propagation for Localization in Wireless Sensor Networks. In *IEEE Third International Conference on Sensor Technologies and Applications, 2009. SENSORCOMM '09*, pages 520–525, 2009.

[13] B. Hendrickson. Conditions for unique graph realizations. *SIAM journal on computing*, 21(1):65–84, 1992.

[14] B. Jackson. Notes on the Rigidity of Graphs, 2007.

[15] D. Jacobs and B. Hendrickson. An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game. *JOURNAL OF COMPUTATIONAL PHYSICS*, 137:346–365, 1997.

[16] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[17] S. Ray and R. Turi. Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation, 1999.

[18] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. AnchorFree Distributed Localization in Sensor Networks. Technical report, MIT Laboratory for Computer Science, 2003.

[19] I. Murray. Note on Rejection sampling and exact sampling with the Metropolised Independence Sampler. 2004.

[20] Marsaglia G. and W.W. Tsang. The Ziggurat Method for Generating Random Variables. *Journal of Statistical Software*, 2000.