Selection of Relevant Particles in Nonparametric Distributed Message Passing for Cooperative Localization

Rico Mendrzik, Jan Lewandowsky, Gerhard Bauch
Institute of Communications
Hamburg University of Technology
Email: {rico.mendrzik, jan.lewandowsky, bauch}@tuhh.de

Abstract—Distributed nonparametric belief propagation can be employed for cooperative localization. Its natural asset of providing a measure of uncertainty of position estimates makes distributed nonparametric belief propagation a powerful localization technique. However, distributed nonparametric belief propagation requires the exchange of arbitrarily complex messages over communication-constrained links. To cope with these communication constraints, parametric approximations (e.g. Gaussian mixtures) of these messages have evolved to become the de facto standard. These message approximations suffer from a limited representational power for highly nonstandard distributions, which results in limited localization accuracy. We propose two novel particle-based message approximations that select important particles according to mutual information shared with a relevant random variable. The first approach is based on the Information Bottleneck algorithm, while the second scheme is based on partial mutual information. We compare both particle-based methods through simulations, demonstrating an improvement over the parametrized approach in terms of localization error with only moderate increase in communications.

I. INTRODUCTION
Location awareness has the potential to give rise to a diverse set of (unforeseen) applications in current and future wireless networks [1], including next-generation cellular services [2], [3], sensor networks [4], [5], robotics [6], [7], and many others. Belief propagation can be deployed to facilitate localization. Oftentimes for continuous random variables, the integrals required for message calculation cannot be solved in closed form. Nonparametric belief propagation (NBP) is a particle-based approximation of exact belief propagation, which circumvents the calculation of the cumbersome integrals by Monte-Carlo integration [8].

NBP can naturally be implemented in a distributed manner, which splits the computational burden over the entire network [9]. Moreover, cooperation among agents (nodes with location uncertainty) can significantly increase the accuracy and availability of location estimates [10]. As opposed to messages from anchors (nodes with precise location information) to agents, messages from agents to other agents (inter-agent messages) have an enormous size, because they entail the uncertainty of the agent’s location [11]. This uncertainty is in general nonstandard and thus hard to represent by families of distributions (e.g. exponential family).

In distributed processing, the cost for communication is in general significantly higher than the cost for computation [11] or the systems have to obey communication constraints. In either way, it is worthwhile to invest some effort in computing (lossy) compressed messages in order to reduce the amount of data to be transmitted. In other words, one is willing to spend computational resources to trade inference quality for compression of messages [12].

Parametrized message approximations have become the de facto standard in literature. Many researchers have investigated the effects of parametrized message approximations considering different types of distributions, number of components, and ways to estimate the parameters [13]–[15]. Ihler et al. showed in [13] that only a few Gaussian mixture components are necessary to represent messages with an acceptable distortion, and adding more components does not reduce the distortion notably. Nevertheless, the representational power of parametric approaches is limited, when the exact message has a non-standard distribution. Furthermore, parametric methods lack a universal application, since distribution types should be well matched to applications [16]. These facts necessitate alternative message approximations with higher representational power.

We propose a novel technique to compress messages by selecting important particles out of the set of all particles. Our decision criterion for particle selection depends on the mutual information (MI) of the position of an agent (with distribution according to the particles) shared with a self-defined relevant random variable. In this paper, we propose an approach based on the Information Bottleneck (IB) method and an approach based on partial mutual information (PMI). The Information Bottleneck-based approach clusters the particles according to their importance. The clusters are generated such that the mutual information between the relevant and compressed random variable are maximized, while the MI of the position of the agent shared with the compressed random variable (compression information) is minimized [17]. The partial mutual information-based approach sorts all particles according to their contribution to the mutual information of the relevant random variable and the position of the agent. The particles with the highest contribution to the mutual information are selected until a reasonable part of mutual information is captured. The threshold enables a flexible tradeoff between compression and communication. Both proposed methods outperform the parametrized approach in terms of localization accuracy, when compared to the parametrized approach.

We investigate the degree of compression that can be achieved with both proposed methods and analyze localization accuracy. Our results show that in situations with high uncertainty (typically present in the first iteration of NBP) only
limited compression is achieved, while for all subsequent iterations, the communications increase only moderately compared to the parametrized approach. Both of our proposed schemes automatically adapt the required amount of communication (which can be upper bounded by a tunable knot) depending on the agent's uncertainty.

**Notation:** Random variables are denoted in italic, realizations of random vectors are denoted in boldface and lower case (e.g. $x$), $I(X;Y)$ denotes the mutual information of $X$ shared with $Y$, $(\cdot)^T$ the transpose, $\| \cdot \|_2$ the Euclidean norm, and $|S|$ the cardinality of the set $S$.

## II. Cooperative Localization

### A. System Model

We briefly introduce the model for positioning with its corresponding terminologies. The following list contains the major variables of concern:

- $N$: Number of agents,
- $M$: Number of anchors,
- $N_s$: Number of particles,
- $x_i$: Position of node $i$,
- $z_{j\rightarrow i}$: Noisy distance estimate of $i$ w.r.t. $j$,
- $e_{j\rightarrow i}$: Noise distortion of the measurement of $i$ w.r.t. $j$,
- $S_{i\rightarrow j}$: Set of cooperating nodes in the communication range of node $i$ (neighborhood).

Suppose that each node $i$ has obtained (noisy) distance measurements w.r.t. all nodes $j \in S_{i\rightarrow i}$:

$$z_{j\rightarrow i} = d(x_i, x_j, e_{j\rightarrow i}) = \| x_i - x_j \|_2 + e_{j\rightarrow i},$$

(1)

where $e_{j\rightarrow i}$ is distributed according to $p_e(e_{j\rightarrow i})$. Without loss of generality, we consider only time-of-arrival-based (TOA) line-of-sight (LOS) measurements. Thus, $p_e$ is distributed according to a Gaussian distribution with zero mean and variance $\sigma^2$, i.e. $p_e \sim N(0, \sigma^2)$. We emphasize that the proposed message compression methods of this paper can be applied regardless of the channel condition, i.e. LOS and NLOS condition are supported. Without loss of generality, we assume that the indices $j = 1, \ldots, N$ represent agents and $j = N + 1, \ldots, N + M$ are anchors. Positioning is said to be cooperative, if at least one $j \in S_{i\rightarrow i}$ $\leq N$ for at least one agent $i$. Note that in general the measurements $z_{j\rightarrow i}$ and $z_{i\rightarrow j}$ are not identical.

### B. Bayesian Inference

In Bayesian estimation, we treat the position of an agent $i$, $x_i$, as a realization of a random variable with a priori distribution $p_{X_i}(x)$. Our goal is then to determine the maximum a posteriori (MAP) estimate of the positions of all agents given all measurements:

$$\hat{x}_{MAP} = \arg \max_x p_{X|z}(x|z) = \arg \max_x p_{X|x}(x)p(x|z)$$

(2)

where $x = [x_1^T, x_2^T, \ldots, x_N^T]^T$ is a vector containing the positions of all agents and $z$ is the vector of all observations (measurements), $z_{j\rightarrow i}$, for $j \in S_{i\rightarrow i}, \forall i$.

We are, however, more interested in calculating the marginal MAP distributions (often referred to as marginal posteriors) for mainly two reasons:

1. the (global) MAP estimate is fairly difficult to compute and
2. its calculation cannot be implemented in a distributed manner.

Belief propagation can provide a computational efficient way to calculate the marginal posteriors in a distributed manner. If we assume independent measurements between nodes, the a posteriori distribution, $p_{X|x}(x|z)$, can be factorized as follows:

$$p_{X|x}(x|z) = \prod_{i=1}^{N+M} p_{X_i}(x_i) \prod_{j \in S_{i\rightarrow i}} p(z_{j\rightarrow i} | x_i, x_j)$$

(3)

### C. Distributed Nonparametric Belief Propagation

The factorization in (3) can be visualized by a factor graph. Factor graphs are bipartite graphs, which contain variable and factor nodes. They graphically encode the (in)dependence of random variables and thus facilitate the analysis of convoluted probabilistic networks. In our case, belief propagation can be applied to the factor graph in order to obtain the marginal posteriors. For the sake of brevity, we will neither address the construction of factor graphs nor belief propagation fundamentals. Excellent tutorials on both topics can be found in literature (e.g. [8], [18]).

In general, the marginal posteriors are calculated as follows:

$$p(x_i|z) = \int_{x_{\neq i}} p(x_1, x_2, \ldots, x_N|z) \, dx_{\neq i},$$

(4)

where $\int_{x_{\neq i}}(\cdot) \, dx_{\neq i}$ denotes the integral over all variables except for $x_i$. Belief propagation on cyclic factor graphs approximates the marginal posteriors iteratively by passing messages along the edges (connections between variable and factor nodes) of the graph. In our example, messages exchanged between node $j$ and $i$ at iteration $l$ have the following general type:

$$\mu_{j\rightarrow i}^{(l)}(x_i) \propto \int p(z_{j\rightarrow i} | x_i, x_j)b_{X_j}^{(l-1)}(x_j) \, dx_j,$$

(5)

where $b_{X_j}^{(l-1)}(x_j)$ denotes the approximate marginal posterior of node $j$ at iteration $l - 1$. Node $i$'s belief at iteration $l$ is computed as the product of its belief from the previous iteration and all incoming messages, i.e.

$$b_{X_i}^{(l)}(x_i) \propto b_{X_i}^{(l-1)}(x_i) \prod_{j \in S_{i\rightarrow i}} \mu_{j\rightarrow i}^{(l)}(x_i).$$

(6)

The integral in (5) can seldom be solved in closed form. This requires an enhanced variant of belief propagation, so called nonparametric belief propagation (NBP). NBP approximates the integral in (5) by means of Monte Carlo integration. Let us assume that we are given a particle representation of agent $j$'s belief at iteration $l - 1$ with $N_s$ particles, $R_{X_j}(b_{X_j}^{(l-1)}(x_j)) = \{ (x_{j}^{(k,l-1)}, w_{j}^{(k,l-1)}) \}_{k=1}^{N_s}$, then we can approximate the message from $j$ to $i$ as follows:

$$\mu_{j\rightarrow i}^{(l)}(x_i) \approx \sum_{k=1}^{N_s} p(z_{j\rightarrow i} | x_i, x_j^{(k)})w_{j}^{(k)}.$$

(7)

## III. Message Representation

With a naive implementation of NBP, node $i$ would compute $|S_{i\rightarrow i}|$ distinct messages and transmit them to each node $j \in S_{i\rightarrow i}$ individually. However, this would result in an unnecessarily high amount of communication. If, instead of transmitting $|S_{i\rightarrow i}|$ distinct messages, node $i$ would solely
Our goal is to obtain a Gaussian mixture model (GMM). We will focus on the transmission of the parameters of the components of some particles, which would require the transmission of (dimension + 1) \cdot N_x single or double precision numbers for every single message.

Parametric message approximations are frequently used to reduce communications, since these methods solely require the transmission of the parameters of the components of some standard distributions. The most commonly chosen distribution is a Gaussian mixture model (GMM). We will focus on GMM as benchmark compression scheme in this paper. A maximum likelihood (ML) estimate of the parameters of a GMM can be obtained by the Expectation Maximization (EM) algorithm. However, there are many other methods to estimate the parameters of the mixture components, like gradient-based parameter optimization, k-means algorithms, etc. For the sake of brevity, we will not explain these schemes in detail and will use EM to fit the parameters of the GMM to the belief.

Instead of using a parametrized approximation of the belief, our goal is to obtain a concentrated particle representation, \(\mathcal{R}_N(b_{X_j}(\cdot))\), of each agent’s belief in order to reduce communications. This concentrated particle representation only consists of important particles, a subset of the original particles, with adjusted weights. One can associate the selection procedure with draining the original particles in a colander, which is structured in a way that only important particles can pass its perforated bottom. The particle selection concept is illustrated in Fig. 1.

Fig. 1: Visualization of the generation of the concentrated particle representation - small particles correspond to high levels of importance.

transmit its belief, \(b_{X_j}^{(l-1)}(x_i)\), the receiving nodes, \(j \in S_{\rightarrow}\), could calculate the messages \(\mu^{(l)}_{i \rightarrow j}(x_j)\) on their own [1]. This reduces the amount of communications significantly, when compared to the naive implementation.

Yet, the transmission of an entire particle representation is prohibitive. Accurate particle representations contain \(N_x = 10^3 \text{ up to } 10^4\) particles, which would require the transmission of \((\text{dimension} + 1) \cdot N_x\) single or double precision numbers for every single message.

However, there are many other methods to estimate the parameters of the mixture components, like gradient-based parameter optimization, k-means algorithms, etc. For the sake of brevity, we will not explain these schemes in detail and will use EM to fit the parameters of the GMM to the belief.

Instead of using a parametrized approximation of the belief, our goal is to obtain a concentrated particle representation, \(\mathcal{R}_N(b_{X_j}(\cdot))\), of each agent’s belief in order to reduce communications. This concentrated particle representation only consists of important particles, a subset of the original particles, with adjusted weights. One can associate the selection procedure with draining the original particles in a colander, which is structured in a way that only important particles can pass its perforated bottom. The particle selection concept is illustrated in Fig. 1.

IV. PARTICLE SELECTION

This section contains the main contributions of this work. First, we will review some important concepts from information theory. We will then turn to our definition of the relevance. Finally, we will describe, how we use the Information Bottleneck and partial mutual information to generate a concentrated particle representation.

A. Background on Information Theory

Consider two random variables, \(X\) and \(Y\), with marginal distribution \(p_X(x)\) and \(p_Y(y)\), respectively. Let \(p_{X,Y}(x,y)\) denote their joint distribution. Mutual information is then calculated as the relative entropy between the joint distribution and the product of the marginals, i.e.

\[
I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log \left( \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \right) .
\]

We consider the concept of partial mutual information from [19]. Unlike [19], where quantization is considered, we are interested in the contribution of a single particle of the sum \(x \in X\) to the mutual information \(I(X;Y)\). Thus we define

\[
I_Y(x) = \sum_{y \in Y} p_{X,Y}(x,y) \log \left( \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \right) .
\]

B. Relevant Random Variable

We regard the particle representation of the belief of agent \(i\), \(\mathcal{R}_{N_x}(b_{X_i}(\cdot))\), as a random variable, \(X_i\), with the set of events \(X_i = \{x^{(k)}\}_{k=1}^N\) and associated probabilities \(\Pr(x^{(k)}) = w_i^{(k)}\).

We define a relevant random variable, \(Y\), which we will later use to determine the importance particles. It is intuitive to assume that the variable has two states: “relevant” and “irrelevant”, i.e. \(Y = \{\text{relevant, irrelevant}\}\). We interpret the relevance of particles depending on their weight. The conditional probability that \(Y\) takes the state “relevant” given a particle, \(x^{(k)}\), is defined as

\[
p_{Y|X_i}(y = \text{relevant}|x^{(k)}) = \frac{w_i^{(k)}}{\max_k \{w_i^{(k)}\}} .
\]

Consequently, the conditional probability that \(Y\) takes the state “irrelevant” given a particle, \(x^{(k)}\), yields

\[
p_{Y|X_i}(y = \text{irrelevant}|x^{(k)}) = 1 - \frac{w_i^{(k)}}{\max_k \{w_i^{(k)}\}} .
\]

Let us consider the extreme case that \(x^{(l)}\) is the particle with the highest weight, i.e. \(w^{(l)} = \max_k \{w_i^{(k)}\}\), then the probability that the \(Y\) takes the value relevant is \(p(y = \text{relevant}|x^{(l)}) = 1\). Hence, the particle with the highest weight always results in the state “relevant”.

According to the chain rule, the joint distribution is then computed by

\[
p_{X,Y}(x_i, y) = p_{Y|X_i}(y|x_i)p_{X_i}(x_i) .
\]

The joint distribution, \(p_{X,Y}(x_i, y)\), will be crucial for the calculation of the concentrated particle representation with both proposed approaches (IB-based and PMI-based).
The IB method requires the joint distribution, $p_{X,Y}(x,y)$, as input and computes the conditional probability density, $p_{T|X,Y}(t|x,y)$. The density $p_{T|X}(t|x_i)$ is the optimal (lossy) compression, which could be used to encode the observed random variable, $X_i$. Suppose that $p_{T|X}(t|x_i)$ is known at all receiver nodes, $j \in S_{Nat}$, node $i$ would have to broadcast only the representative of each cluster. All nodes $j \in S_{Nat}$ could then decode the representative and reconstruct the original particle representation with a given loss of relevant information.

However, $p_{T|X}(t|x_i)$ is not known at the receivers. More importantly, it changes in every iteration, which makes its transmission impractical (the communication overhead would be significantly larger than the naive transmission of the entire particle representation).

Anyhow, we can still utilize the resulting clustering for message compression. That might seem absurd in the first place. However, recall that our initial goal is to select important particles from the set of all particles. The clusters can be sorted according to their center of gravity. After sorting, the clusters are arranged in an ascending manner according to our definition of relevance, i.e., the importance of $t_i < t_j \forall j \neq i$. Thus, we are tempted to choose only the particles, which reside in the most important clusters. An efficient algorithm which only optimizes over hard partitionings, i.e., $p_{T|X}(t|x_i) \in \{0,1\}$, is called sequential Information Bottleneck algorithm (sIB) [20].

The arrangement of the clusters guided the design of the algorithm for the IB-based particle selection. It turns out that the first cluster, $t_1$, (cluster with the least importance) contains many (irrelevant) particles. Hence, the IB-based algorithm selects the particles which fall into the least important clusters $t_i$, $i = 2, ..., |T|$. We determine the number of clusters, $|T|$, required to closely approximate $X_i$ iteratively, i.e., we increment the number of clusters iteratively until the mutual information $I(X_i;T)$ changes only negligibly, if we add an additional cluster. Here, negligibly means that the change is below a given threshold $\delta_I$. The threshold, $\delta_I$, can be used to trade compression for communications. The IB-based particle selection scheme is shown in Algorithm 1.

The IB-based particle selection is a powerful tool to generate a concentrated particle representation. Yet, there is still some room for improvement, when we consider the fact that the IB generates a cluster structure, $p_{T|X}(t|x_i)$, for optimal lossy compression, which is not used. We have already discussed earlier that this structure needs to be known at the receivers to be able to decode it. In fact, we are only selecting the particles according to the importance of their clusters. Thus, we waste a lot of computational resources to compute an optimal compression mapping, which we hardly make use of in the end. Thus, we are motivated to look for a cheaper alternative, which also provides the possibility to flexibly trade compression for communications.

**Algorithm 1 IB-based Particle Selection**

**Precondition:** $\mathcal{R}_{N_s}(b_{X_i}()) = \left\{ (w_i^{(k)}, x_i^{(k)}) \right\}_{k=1}^{N_s}, \delta_I$  

1. Compute the conditional density according to (10) and (11)  
2. Compute the joint distribution according to (12)  
3. Initialize: $I(Y;T) = 0$, $|T| = 1$  
4. while $I(Y;T)^{|T|} - I(Y;T)^{|T|-1} \geq \delta_I$ do  
5. Increase the number of clusters  
6. Calculate clustering $I(Y;T)^{|T|}, p_{T|X}(t|x_i)) = \text{sIB}(p_{X,Y}(x_i,y), |T|)$  
7. end while  
8. for $k = 1$ to $N_s$, do  
9. if $x_i^{(k)} \in t_i, i = 2, ..., |T|$ then  
10. Select the relevant particles $X_i^{(k)} = x_i^{(k)}$  
11. Assign weight $w_i^{(k)} = w_i^{(k)}$  
12. end if  
13. end for  
14. Normalize the weights $w_i^{(k)} = w_i^{(k)}/\sum_{k=0}^{N_s} w_i^{(k)}$, $\forall k$  
15. Return the concentrated particle representation $\mathcal{R}_{K}(b_{X_i}())$

**D. Sorted Partial Mutual Information**

Our goal is still to tackle the problem of message compression in an information theoretic way, but with reduced complexity. The concept of partial mutual information is very helpful in this case. Let us consider the PMI taken over the relevant random variable, $I_X(Y)$. The contribution of particle $k$ to the mutual information $I_{X,Y}(X;Y)$ is the PMI of the
The PMI-based approach consists of two stages: a pre-processing and a selection stage. In the first (pre-processing) stage, the PMI $I_Y(x_i)$ is computed and the particles are sorted according to their contribution to mutual information $I_{X_i,Y}(X_i,Y)$. Without loss of generality, we assume that the order is chosen to be descending. In the second (selection) stage, the particles with the largest PMI (first in the row) are selected to the set of important particles and their weight is copied successively until $\gamma I_{X_i,Y}(X_i,Y)$ is captured. We want to stress that particles are not sorted according to their weight, but they are rather sorted according to their PMI. Sorting particles with respect to their weight, sorting particles with respect to PMI gives in general a different order than sorting particles with respect to their weight. Subsequently, the weights of the important particles are normalized. The sorting particles with respect to their weight, sorting particles with respect to PMI gives in general a different order than sorting particles with respect to PMI. Sorting particles with respect to PMI gives in general a different order than sorting particles with respect to PMI.

Algorithm 2: PMI-based Particle Selection

**Precondition:** $\mathcal{R}_{N_s}(b|X_i) = \left\{ \left( w_{i}^{(k)} \right), X_i^{(k)} \right\}_{k=1}^{N_s}$, $\gamma$

1. Compute the conditional density according to (10) and (11)
2. Compute the joint distribution according to (12)
3. Compute the PMI $I_Y(X_i)$
   $$I_Y(X_i) = \sum_{y \in Y} p_{X_i,Y}(x_i,y) \log \left( \frac{p_{X_i,Y}(x_i,y)}{p_{X_i}(x_i)p_Y(y)} \right).$$
4. Sort the particles according to their PMI in a descending order
   $$I_{Y,sort} = \left\{ \left( x_i^{(k)} \right), \left( \tilde{w}_i^{(k)} \right) \right\}^{N_s}_{k=1} = sort(I_Y(x_i), \left\{ \left( x_i^{(k)} \right), \left( \tilde{w}_i^{(k)} \right) \right\}^{N_s}_{k=1})$$
5. Initialize parameters for the loop
6. while $\sum_{k=1}^{l} I_{Y,sort}(\tilde{x}_i^{(k)}) < \gamma$ do
7.    Add particle to the set of important particles
8.    Increment the number of particles
   $$l = l + 1$$
   end while
10. Normalize the weights
    $$\tilde{w}_i^{(k)} = \frac{w_i^{(k)}}{\sum_{k=1}^{N_s} w_i^{(k)}}, \forall k$$
11. Return the concentrated particle representation
    $$\mathcal{R}_{\tilde{K}}(b|X_i)$$

Localization performance is evaluated according to the outage probability. An agent is said to be in outage when its error $\epsilon_i = \|x_i - x_{est}\| > \epsilon_{th}$. The outage probability is given by
$$P_{out}(\epsilon_{th}) = \mathbb{E}(\mathbb{I}(\epsilon_i > \epsilon_{th})),$$ (13)
where $\mathbb{I}(\cdot)$ denotes the indicator function. The outage probability has been averaged over 100 Monte-Carlo trials.

In the tested scenario, convergence is achieved after 4 iterations.

It can be seen in Fig. 3 that the parametrized model yields the highest outage probability. The best achievable performance is indicated by the uncompensated scheme, in which the entire particle representation is transmitted. The sIB shows some improvement in terms of outage probability. Among the three given message compression schemes, the PMI-based approach achieves the lowest outage probability.

A fair comparison of the schemes also requires the analysis of the communication costs. Only inter-agent messages are considered. Here, the clear winner is the parametrized approach. It solely requires the transmission of $N = N_{components} \cdot \left( \text{1 + dimension + dimension}^2 \right) = 35$ single or double precision numbers. When compared to the transmission of the entire particle representation ($10^3$ particles with 3 components each), the cost saving at each iteration yields 98.8%. The second iteration is the first cooperative iteration, i.e., it is the first iteration, in which inter-agent messages have to be compressed. From Fig. 4 one can gather that the proposed compression schemes show significantly lower communication savings at the second iteration, when compared to the parametrized approach. After the first iteration, the belief of many agents contains high uncertainty, because most agents

\[1\] By convergence, we mean that the outage curve no longer changes noticeably for subsequent iterations.

**E. Reconstruction and Complexity**

The reconstruction of a particle representation with $N_s$ particles from the concentrated particle representation is identical for both proposed methods. For that purpose, we employ resampling, i.e., we sample from the concentrated particle representation until we have $N_s$ samples. Sampling selects a particle, $\tilde{x}_i^{(k)}$, with probability $Pr \{ \tilde{x}_i^{(k)} \} = \tilde{w}_i^{(k)}$. On average, each particle, $X_i$, appears $N_s \tilde{w}_i^{(k)}$ times in the reconstructed particle representation.

Finally, we would like to discuss the complexity of the proposed compression schemes briefly. For the PMI-based approach it is easy to give an expression for the complexity, because it is dominated by the sorting operation. Thus, the complexity is in the order of $O(N_s \log(N_s))$. Yet, it is difficult to come up with a similar expression for the sIB-based approach. A practical complexity bound is also provided in [20], bounding the complexity by $O(l|Y||X||T|)$, where $l$ is a nondeterministic iteration counter in the sIB algorithm. Practically occurring small $l$ (typically $l|T| \ll |X|^2$) result in approximately linear complexity in $X$. Empirical observations show that despite the efficiency of the sIB algorithm, the sIB approach has considerably higher complexity than the partial mutual information approach for practically relevant values of $N_s$.

**V. NUMERICAL RESULTS**

We consider localization in a $100 \times 100 \text{ m}^2$ plane with 13 anchors and 100 agents. Both anchors and agents have a communication range of 20 m. Every node can perform range measurements with nodes within the communication range. Anchors are distributed such that every point within the considered area is covered by at least one anchor, while agents are distributed randomly. The number of particles is $N_i = 10^5$. All links are assumed to be line-of-sight (LOS). Initially, none of the agents has any information about their position. Thus, agents do not broadcast their beliefs at the first iteration. The number of mixture components of the parametrized GMM is $N_{GMM} = 5$. Thresholds for the proposed schemes are as follows: $\delta_{TH} = 0.1$ (sIB) and $\gamma = 0.99$ (PMI).
have only one anchor in their sight. To prevent the dismissal of (potentially important) samples, the proposed methods naturally and autonomously reduce the degree of message compression in situations with high uncertainty. Hence, the sIB-based and PMI-based approach only achieve a reduction in communications of 50% and 40%, respectively. The required amount of communication rapidly converges (within one iteration) to 94% and 88%, respectively. When compared to the parametrized approach, the increase in communications in only moderate, while the outage probability is reduced notably.

VI. CONCLUSION

We proposed and analyzed a new class of message compression techniques for distributed NBP, namely particle-based approximations. We have investigated the sIB-based and the PMI-based approach, which select particles according to their mutual information shared with a relevant random variable. We showed that the outage probability can be reduced, when compared to parametrized message approximations (the de facto standard in literature). Particle-based message approximations can substantially reduce communications. However, they require a somewhat higher amount of communication than their parametrized counterpart. When particle-based message compression is compared with parametrized compression, particle-based message compression enables the reduction of the outage probability, if a slight increase in communications can be accepted.

REFERENCES