A Network Traffic Reduction Method for Cooperative Positioning

Kallol Das, Henk Wymeersch
Department of Signals and Systems
Chalmers University of Technology, Gothenburg, Sweden
Email: kallol@student.chalmers.se, henkw@chalmers.se

***Abstract*—Cooperative positioning is suitable for applications where conventional non-cooperative positioning fails due to lack of connectivity with a sufficient number of reference nodes. In a dense network, as the number of cooperating devices increases, the number of packet exchanges also increases proportionally. This causes network congestion and increases packet collisions. In order to maintain the quality of positioning, we need to reduce the packet loss probability due to collisions. This can achieved by reducing the broadcast of unnecessary information. We show that by intelligently suppressing the transmission of selected nodes, the overall network traffic can be reduced without degrading the positioning performance significantly.**

I. Introduction

Wireless positioning has been providing location-based services in many applications, such as navigation [1], emergency rescue operations [2], sensor networks [3], health care monitoring [4], vehicular communications [5], etc. Different positioning techniques are currently available, among them the performance of range-based positioning depends on the distance estimation accuracy and coverage by reference nodes. Conventional range-based positioning may fail in some applications, such as indoor positioning, due to the lack of connectivity with a sufficient number of reference nodes. In these cases, cooperative positioning can improve the positioning performance by exchanging positional information between devices [3], [6].

Although cooperative positioning can provide improved positioning performance in GPS challenged conditions, it suffers from increased network traffic, as all of the cooperating devices repeatedly broadcast packets containing their positional information. If some packets are lost due to collisions, the overall positioning performance will also be degraded [7], [8]. By reducing unnecessary packet broadcasts over the network, the packet collision probability can be reduced. In our previous work [9] on non-Bayesian positioning, we showed that by blocking the broadcasts of the nodes that do not have reliable estimates, network traffic can be reduced without degrading positioning performance. While network traffic reduction and collision avoidance are traditionally higher layer issues [10], [11], our work in [9] operates across the physical layer and the medium access layer. In this paper, we extend our previous work by developing a network traffic reduction method for *Bayesian* positioning algorithms. We show that by blocking the broadcast of selected nodes for which all neighbors have satisfactory positional information, a significant reduction in network traffic can be obtained.

The remainder of this paper is arranged as follows. In Section II, we describe our model and assumptions. In Section III, we briefly describe the Bayesian positioning algorithm from [6]. In Section IV, our proposed method is explained. Results from simulations are presented in Section V. In Section VI, we draw our conclusions with possible future extensions of this work.

II. Problem formulation

Let us consider a wireless network with two types of static nodes: anchors, which know their positions, and agents, whose positions have to be estimated. In distributed networks, agents iteratively update their position estimates. The anchor nodes act as reference points for positioning. In a cooperative context, the position update phase of the agents depends on agent-to-anchor range measurements as well as agent-to-agent measurements.

We denote by $\mathbf{x}_i$ the position of node i in the network and by $\hat{\mathbf{x}}_i$ the corresponding estimated position. $S_{\rightarrow i}$ is the set of nodes from which node i can receive signals. Based on a ranging protocol (e.g., time of arrival (TOA), time differences of arrival (TDOA), receive signal strength (RSS) etc.) with node $j \in \mathcal{S}_{\rightarrow i}$, node i can estimate the distance $\hat{d}_{j\rightarrow i} = \|\mathbf{x}_i - \mathbf{x}_j\| + n_{j\rightarrow i}$, where $n_{j\rightarrow i}$ is the ranging noise. We assume $n_{j\rightarrow i} \sim \mathcal{N}\left(0, \sigma^2_{j\rightarrow i}\right)$ [6]. The goal of node i is to estimate its own position. Ideally, the positioning process should require low complexity and communication overhead per node as well as low latency.

In a dense cooperative network, the number of neighbors per agent will be large. In this case, every user (agent) broadcasts its positional information in every iteration, which increases the network traffic. This makes the implementation of practical cooperative positioning challenging.

III. Sum Product Algorithm over a Wireless Network (SPAWN)

Before describing SPAWN, we will introduce the basic idea of factor graphs and the sum-product algorithm.

A. Factor Graphs

Factor graphs are a graphical representation of factorization of a function [12], [13]. Consider a network comprising one

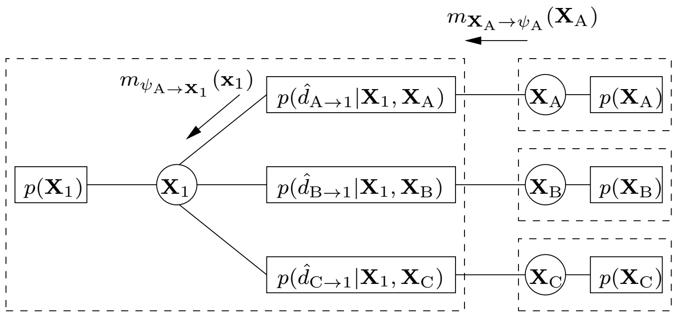
$m_{\mathbf{X}_A \to \psi_A}(\mathbf{X}_A)$
$m_{\psi_A \to \mathbf{x}_1}(\mathbf{x}_1)$
$p(\hat{d}_{A\to 1}|\mathbf{X}_1, \mathbf{X}_A)$
$p(\hat{d}_{B\to 1}|\mathbf{X}_1, \mathbf{X}_B)$
$p(\hat{d}_{C\to 1}|\mathbf{X}_1, \mathbf{X}_C)$
$p(\mathbf{X}_1)$
$\mathbf{X}_1$
$\mathbf{X}_A$
$\mathbf{X}_B$
$\mathbf{X}_C$
$p(\mathbf{X}_A)$
$p(\mathbf{X}_B)$
$p(\mathbf{X}_C)$

Figure 1. Net factor graph and its message passing. Let ψ_A be a shorthand for $p(\hat{d}_{A\to 1}|\mathbf{x}_1, \mathbf{x}_A)$.

agent at position $\mathbf{x}_1$ and three anchors at positions $\mathbf{x}_A$, $\mathbf{x}_B$, $\mathbf{x}_C$ as shown in Figure 1. Assume that the agent has distance estimates available w.r.t. the three anchors $(\hat{d}_{A\to 1}, \hat{d}_{B\to 1}, \hat{d}_{C\to 1})$. Then the joint a posteriori distribution is given by

$$\begin{aligned} &p\left(\mathbf{x}_1, \mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \hat{d}_{A\to 1}, \hat{d}_{B\to 1}, \hat{d}_{C\to 1}\right) \\ &\propto p(\mathbf{x}_1)p(\mathbf{x}_A)p(\mathbf{x}_B)p(\mathbf{x}_C)p(\hat{d}_{A\to 1}|\mathbf{x}_1, \mathbf{x}_A) \\ &\quad \times p(\hat{d}_{C\to 1}|\mathbf{x}_1, \mathbf{x}_C)p(\hat{d}_{C\to 1}|\mathbf{x}_1, \mathbf{x}_C). \end{aligned} \quad (1)$$

The factor graph corresponding to this distribution is shown in the same figure. This factor graph contains variable vertices (one for every variable) and factor vertices (one for every factor). Edges connect a variable vertex with a factor vertex when the corresponding variable appears as an argument in the corresponding factor.

B. Sum-Product Algorithm

The sum-product algorithm is a message passing algorithm on a factor graph that is used to compute marginal distributions of the original distribution. Messages are passed along the edges between variable vertices and factor vertices in both directions. We denote the message from variable vertex X to factor vertex f by $m_{X\to f}(x)$ and the message from factor vertex f to variable vertex X by $m_{f\to X}(x)$. Assume variables X, Y, Z appear in $f(\cdot)$, then the message $m_{f\to X}(x)$ is given by

$$m_{f\to X}(x) = \int f(x,y,z) m_{Y\to f}(y) m_{Z\to f}(z) \mathrm{d}y\mathrm{d}z. \quad (2)$$

Similarly, assume X appears in factors f, g, h, then the message from $m_{X\to f}(x)$ is given by the product of the incoming messages:

$$m_{X\to f}(x) = m_{g\to X}(x) \times m_{h\to X}(x). \quad (3)$$

Finally, the marginal distribution of X is given by

$$b_X(x) \propto m_{f\to X}(x) \times m_{X\to f}(x),$$

where $\propto$ denotes equality up to a normalization constant. We call $b_X(\cdot)$ the *belief*.

For example, the message from factor $p(\hat{d}_{A\to 1}|\mathbf{x}_1, \mathbf{x}_A)$ (abbreviated by ψ_A in Figure 1) is given by

$$m_{\psi_A\to \mathbf{x}_1}(\mathbf{x}_1) = \int p(\hat{d}_{A\to 1}|\mathbf{x}_1, \mathbf{x}_A) m_{\mathbf{x}_A\to \psi_A}(\mathbf{x}_A)\mathrm{d}\mathbf{x}_A,$$

while the reverse message is given by

$$m_{\mathbf{x}_1\to \psi_A}(\mathbf{x}_1) = m_{\psi_B\to \mathbf{x}_1}(\mathbf{x}_1) \times m_{\psi_C\to \mathbf{x}_1}(\mathbf{x}_1) \times p(\mathbf{x}_1).$$

The marginal a posteriori distribution of $\mathbf{X}_1$ is given by

$$\begin{aligned} b_{\mathbf{x}_1}(\mathbf{x}_1) &\propto m_{\psi_A\to \mathbf{x}_1}(\mathbf{x}_1) \times m_{\mathbf{x}_1\to \psi_A}(\mathbf{x}_1) \\ &= p(\mathbf{x}_1) \times \prod_{\alpha\in\{A,B,C\}} \int p(\hat{d}_{\alpha\to 1}|\mathbf{x}_1, \mathbf{x}_\alpha) p(\mathbf{x}_\alpha)\mathrm{d}\mathbf{x}_\alpha, \end{aligned}$$

which is of course the correct marginal.

C. Distributed Positioning Algorithm

We briefly describe the Sum-product algorithm over a wireless network (SPAWN). SPAWN maps a factor graph onto the network topology and develops a distributed message passing scheme [6]. For example, for the factor graph in Figure 1, we can associate a sub-graph with every device in the network, marked in dashed rectangles. Messages are either computed within a sub-graph, i.e., internal to a devices, or exchanged between sub-graphs, i.e. sent as packets between devices. This concept naturally extends to large networks with many agents, leading to a distributed algorithm, summarized in Algorithm

Algorithm 1 SPAWN (iteration k, agent i).

```
1: receive $b^{(k)}_{\mathbf{X}_j}(\cdot)$ from neighbors $j \in S_{\to i}$
2: check whether the belief is converged
3: convert $b^{(k)}_{\mathbf{X}_j}(\cdot)$ to a distribution over $\mathbf{X}_i$ using (3)
       $m_{\mathbf{X}_j\to\mathbf{X}_i}(\mathbf{x}_i) \propto \int p(\hat{d}_{j\to i}|\mathbf{x}_i, \mathbf{x}_j) b^{(k)}_{\mathbf{X}_j}(\mathbf{x}_j)\mathrm{d}\mathbf{x}_j$
4: compute new message using (3)
       $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i) \propto p(\mathbf{x}_i) \prod_{j\in S_{\to i}} m_{\mathbf{X}_j\to\mathbf{X}_i}(\mathbf{x}_i)$
5: decide if block the broadcast
6: broadcast $b^{(k)}_{\mathbf{X}_i}(\cdot)$ if not blocked
```

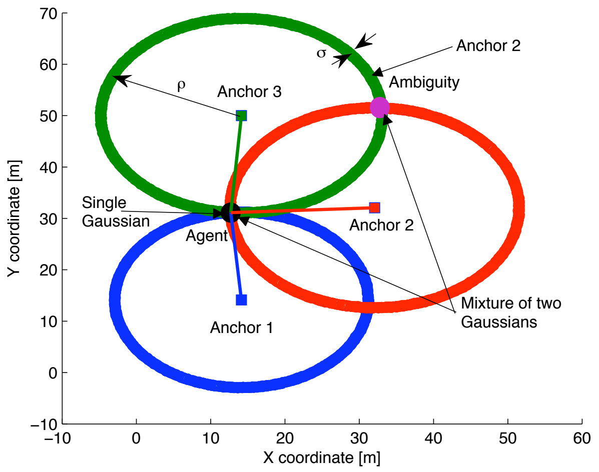

Figure 2. Different typical distributions and their approximations.

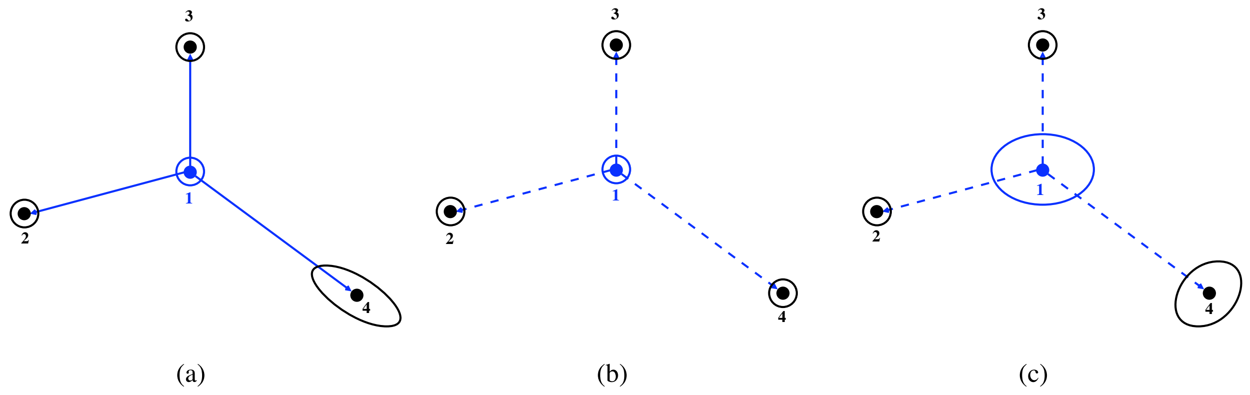

Figure 3. Demonstration of network traffic reduction method in 3 different scenarios.

1. This algorithm is executed in parallel by every agent in the network until the beliefs have converged.

Messages in SPAWN can be represented using non-parametric [14] or parametric [15] representations. We will make no a priori assumptions regarding the message representations.

IV. Proposed method

A. Message approximation

Considering Figure 2, when an agent communicates with one anchor, its belief has ring-shape. This type of belief is not informative for neighbors, so we will approximate it with a single, broad Gaussian distribution with the same covariance. When an agent communicates with two anchors, it will lie on the intersection of two rings. Hence, its belief can be approximated by a mixture of two Gaussian distributions. When an agent communicates with three or more anchors, its belief can be approximated with a single Gaussian. While a belief can have many other shapes, we will approximate them in the same manner: we first determine the number of components of the belief $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ of agent $\mathbf{x}_i$ at iteration k, denoted by $N^{(k)}_{c,i} \in \{1,2\}$. For every component, we then determine the mean and the covariance matrix. For simplicity and robustness, we further only consider the covariance matrix with the largest trace. Finally, we approximate all beliefs at iteration k by a mixture of two Gaussians as

$$b^{(k)}_{\mathbf{X}_i}(\cdot) \approx \frac{1}{2}\mathcal{N}\left(\boldsymbol{\mu}^{(k)}_{1,i}, \boldsymbol{\Sigma}^{(k)}_i\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{\mu}^{(k)}_{2,i}, \boldsymbol{\Sigma}^{(k)}_i\right), \quad (4)$$

where the means are equal when $N^{(k)}_{c,i} = 1$.

We note that (i) this approximation is used to decide whether to block a broadcast of an agent or not and to decide whether the agents are converged or not, while the messages computed and propagated in SPAWN remain unaffected; (ii) our approximation relatively simple, but, as we will see, leads to excellent results.

B. Description of the method

We will first demonstrate the proposed network traffic reduction method using an example shown in Figure 3 where agent 1 has three neighbors (agent 2, 3 and 4). Here, the dots represent the position estimates of the agents at a certain iteration and the circles represent the uncertainty regions of the corresponding beliefs. In Figure 3-(a), agent 1 finds one of its neighbors (agent 4) has unsatisfactory positional information (i.e., a broad belief). Hence, agent 1 will broadcast its positional information to its neighbors. On the other hand, agent 1 will block its broadcast in Figure 3-(b), as it finds that all of its neighbors have satisfactory positional information. Finally, in Figure 3-(c), agent 1 itself has unreliable estimate, so it should block its broadcast.

We will assume that an agent j will stop updating its belief when it is well-localized, i.e., trace $\left(\boldsymbol{\Sigma}^{(k-1)}_j\right) < \gamma_{\text{stop}}$ and $N^{(k-1)}_{c,j} = 1$ for some preset value of γ_{stop}. Every agent i will block its broadcast at iteration k when one of the following conditions is met:

- **Condition 1**: trace $\left(\boldsymbol{\Sigma}^{(k)}_i\right) \geq \gamma$
- **Condition 2**: $\forall j \in S_{\rightarrow i}$: trace $\left(\boldsymbol{\Sigma}^{(k-1)}_j\right) < \gamma_{\text{stop}}$ and $N^{(k-1)}_{c,j} = 1$

The first condition simply reflects the fact that the agent will not broadcast unreliable information. The second condition reflects that an agent should not broadcast when all its neighbors were already well localized at the previous iteration. Both thresholds (γ_{stop} and γ, expressed in m^2) depend on the ranging model and the performance requirements.

When *all* neighbors of agent i satisfy the stopping threshold (condition 2), the broadcasts of agent i will be ignored by all neighbors. Hence, those broadcasts are unnecessary. We can thus develop a smart network traffic reduction scheme. We will denote by *Modified SPAWN* the SPAWN algorithm with our proposed traffic reduction method. Note that this scheme suffers from a hidden node problem: when an agent is not aware a neighbor is present (due to packet loss, packet blocking, or asymmetric links), it may decide to block the broadcast too early.

V. Numerical Results

A. Simulation Parameters

We have simulated random networks in a 100 m×100 m map with 13 systematically placed anchors and 100 randomly

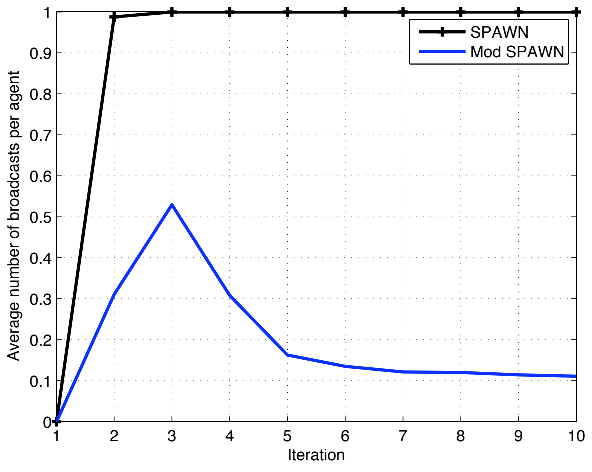

Figure 4. Comparison of average number of broadcasts per agent of SPAWN and modified SPAWN.

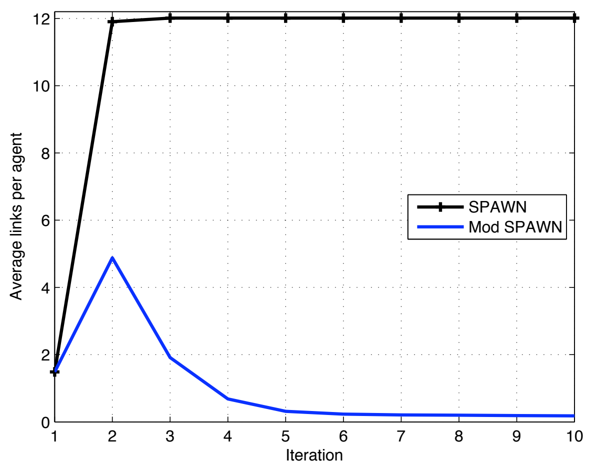

Figure 5. Comparison of number of used links of SPAWN and modified SPAWN.

placed agents having 20 m communication range and 10 cm ranging noise standard deviation to evaluate the performance of our proposed method. In our simulation we have to set several parameters. The stopping threshold γ_{stop} is directly related to positioning accuracy. To achieve high accuracy we have to allow more iterations. On the other hand if we relax our accuracy requirements most of the agents will converge after 5-6 iterations and the overall process will be faster. After the first iteration of the positioning phase (non-cooperative) very few agents (those who have connections with three or more anchors) can achieve satisfactory position estimates (i.e., these agents achieve concentrated beliefs) and the others have relatively bad estimates (i.e., those agents have broad beliefs). The blocking threshold γ should be chosen such a way that only the agents who are satisfied with their estimates should broadcast their positional information. Both thresholds are dependent on the network geometry. For the final results we set $\gamma_{\text{stop}} = (0.28\,\text{m})^2$, which is on the order of the ranging noise variance, and $\gamma = 0.2\,\text{m}^2 = (0.45\,\text{m})^2 > \gamma_{\text{stop}}$. We observe that changes in γ around this value do not affect positioning accuracy or average number of transmission significantly, but too conservative blocking (i.e., higher value of γ) causes increased network traffic.

B. Simulation Discussion

1) Traffic analysis: The main benefit of our proposed scheme is that it can not only block the broadcasts of unreliable information but also unnecessary information. This significantly reduces the overall number of packet broadcasts in the network. The average number of broadcasts in the network is shown in Figure 4. In this figure, an average number of broadcasts of 0.3 means that on average 30% of the agents in the network broadcast their positional information at that particular iteration. In the beginning of the iterative algorithm very few nodes have good positional information. By applying modified SPAWN we can block the broadcast of the unreliable nodes (i.e., Condition 1 is met), which results in a low number of broadcasts per agent. After few iterations most of the nodes have been well-localized and start broadcasting, then modified SPAWN start to reduce the traffic further as more and more agents find that all of their neighbors have concentrated beliefs (i.e., Condition 2 is met).

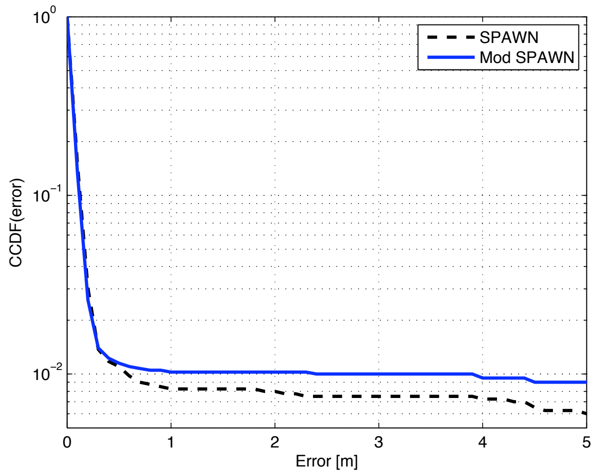

Figure 6. Positioning performance comparison of SPAWN and modified SPAWN after 10 iterations.

2) Complexity analysis: Another benefit of modified SPAWN is to reduce the computational complexity of message multiplication. As the overall transmission will be less and less as iterations progress, agents will receive information from fewer and fewer neighbors. This helps to reduce the amount of information to fuse, which in turn make the information fusion faster. We can see from Figure 5 that with modified SPAWN the average number of used links can be significantly reduced. During simulations, we observed that modified SPAWN can execute roughly 10.5 times faster than normal SPAWN.

3) Performance analysis: Finally, we will evaluate positioning performance of our proposed method by showing the complementary cumulative distribution function (CCDF) of the positioning error, i.e., the probability that the positioning error exceeds a certain value, after 10 iterations. The CCDFs of SPAWN and modified SPAWN are shown in Figure 6. We observe that the CCDF of modified SPAWN almost follows the CCDF of SPAWN. Hence, modified SPAWN can still maintain excellent positioning performance, with less packet broadcasts, and less complexity.

VI. Conclusions and Future Work

In this paper, we have evaluated a network traffic reduction method for cooperative positioning in dense networks, while still maintaining good performance. All broadcast blocking decisions are distributed and based on positional information of the agents. By applying our proposed scheme to SPAWN we have found that modified SPAWN (i) can reduce the network traffic both in the first few iterations, but also when agents start obtaining concentrated beliefs, without any significant performance loss; (ii) can reduce the number of links that are using for the message multiplication. These advantages of the proposed scheme (distributed nature, reduced network traffic and complexity, while maintain good positioning performance) make it promising for large-scale dense networks. Future work includes extending the proposed scheme to account for NLOS propagation and testbed implementation.

References

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[2] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proceedings of the 8th International Conference on Advanced Robotics (ICAR)*, 1997, pp. 193–200.

[3] N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[4] C.-C. Lin, M.-J. Chiu, C.-C. Hsiao, R.-G. Lee, and Y.-S. Tsai, "Wireless health care service system for elderly with dementia," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 696–704, 2006.

[5] A. Festag, H. Fußler, H. Hartenstein, A. Sarma, and R. Schmitz, "Fleetnet: Bringing car-to-car communication into the real world," in *Proceedings of the 11th World Congress on ITS*, vol. 4, no. L15, 2004, p. 16.

[6] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427 –450, 2009.

[7] C. Mensing and J. J. Nielsen, "Centralized cooperative positioning and tracking with realistic communications constraints," in *Proceedings of the 7th Workshop on Positioning Navigation and Communication (WPNC)*, 2010, pp. 215–223.

[8] S. Severi, G. Abreu, G. Destino, and D. Dardari, "Efficient and accurate localization in multihop networks," in *2009 Conference Record of the 43rd Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 1071 –1076.

[9] K. Das and H. Wymeersch, "Censored cooperative positioning for dense wireless networks," in *Proceedings of 2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops)*, 2010, pp. 262–266.

[10] T. Stathopoulos, R. Kapur, D. Estrin, J. Heidemann, and L. Zhang, "Application-based collision avoidance in wireless sensor networks," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 506 – 514.

[11] C. Jandaeng, W. Suntiamontut, and N. Elz, "Throughput improvement of collision avoidance in wireless sensor networks," in *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–5.

[12] H. Wymeersch, *Iterative receiver design*. Cambridge University Press, 2007.

[13] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[14] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Topics in Communications*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[15] L. Wenjing, "Message representation and updates for cooperative positioning," Master's thesis, Department of Signals and Systems, Chalmers University of Technology, 2010.