Bayesian Outlier Detection in Location-aware Wireless Networks

Yi Li, Dapeng Liu, Henk Wymeersch
Department of Signals and Systems
Chalmers University of Technology, Gothenburg, Sweden
Email: {yili,dapengl}@student.chalmers.se, henkw@chalmers.se

***Abstract*—Location-aware networks are a rapidly growing area of research with a wide range of applications. The accuracy of localization depends on the reliability of the information exchanged between devices in the network. In practice, devices may fail or maliciously inject false position information into the network. This paper aims to design and test algorithms to verify the location consistency in wireless networks. We propose a new method based on factor graphs. This method is flexible, easily extendible to cooperative networks, and leads to significant performance improvements compared to existing techniques that are based on linear programming.**

I. Introduction

Location-aware technologies [1], [2], have applications in many areas, including military tracking [3], public safety [4], and asset tracking [5]. The main goal of location-aware networks is to estimate the position of a target communication entity (referred to as an agent) in a wireless network, based on the information provided by other navigational devices as well as position-dependent measurements. Other navigational devices includes reference nodes (referred to as anchors) and possibly other agents. Different types of measurement techniques have been developed, including distance estimation from received signal strength or time-of-arrival, or angle estimation from multiple antennas [2]. Our focus will be on a scenario where an agent collects distance estimates with respect to three or more anchors, and then determines its position through trilateration.

When one of the anchors malfunctions or intentionally injects false information in the network, the positioning algorithm typically fails, in the sense that it provides incorrect location information [6], [7]. A small network, consisting of one agent and three anchors, of which one is providing a severely negatively biased range estimate, is shown in Figure 1. While the agent's position estimate will be erroneous, it is possible to detect the malfunctioning anchor. In a cooperative network [8], this effect is further exacerbated, as the agent in question will in turn mislead other agents. For that reason, *outlier detection* is of great importance in location-aware networks, especially for safety-critical applications. Outlier detection has received a great deal of attention in different contexts (see [9], [10], [11] and references therein). In the context of localization, we can distinguish between dealing with non-line-of-sight (NLOS) situations [12], [13], [14], where the focus is on improving accuracy, or with general errors [7], [15], where the focus is on guaranteeing availability.

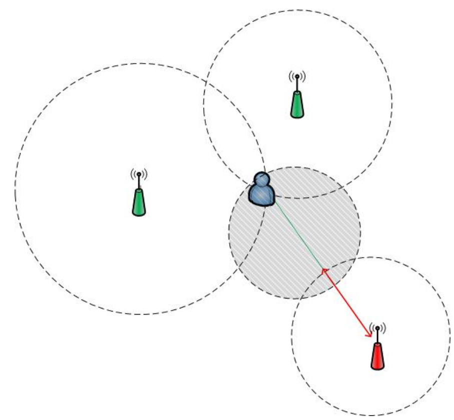

Figure 1. An example of a network with one agent and three anchors. One malfunctioning anchor causes a large ranging error, resulting in large localization errors.

Methods include classical and Bayesian hypothesis testing, linear programming (LP) optimization, and ad-hoc approaches. Little has been done to extend these works to cooperative settings.

In this paper, we develop a Bayesian outlier detection method based on graphical models. Such models have been used for localization [8] and can thus be easily integrated with the proposed method. In essence, we consider the outlier event as a random variable that will be included in the graphical model. Moreover, when multiple agents are present and within communication range, our method can easily enable cooperation among agents to detect outliers. We compare our method with the LP method from [15]. For cooperative networks, we further perform an extrinsic information transfer (EXIT) analysis, which is helpful to visualize the progress of cooperation.

The remainder of this paper is organized as follows. In Section II, we describe the system model and formulate the problem. The linear programming and factor graph methods are described in Section III. Numerical results of the different algorithms are compared and analyzed in Section IV. In Section V, we present our conclusions. Our main conclusion is that the factor graph method using cooperation can achieve near-optimal localization performance, though at a significant complexity cost.

978-1-4577-0450-5/11/$26.00 ©2011 IEEE

II. System Model

A. Observation Model

We consider a network with a single agent and N anchors. We denote the position of the agent by $\mathbf{a} \in \mathbb{R}^2$. The position of the n-th anchor is written as $\mathbf{x}_n$. The agent estimates the distance $d_n = \|\mathbf{x}_n - \mathbf{a}\|$ with respect to every anchor, leading to estimates $\hat{d}_n$. Anchors are in one of two states, $s_n \in \{0,1\}$, where $s_n = 0$ means that the anchor is functioning properly, while $s_n = 1$ indicates an anchor malfunction. We will assume that for functioning anchors the distance estimation error has a zero-mean Gaussian distribution:

$$p\left(\hat{d}_n | s_n = 0, \mathbf{a}, \mathbf{x}_n\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\hat{d}_n - \|\mathbf{x}_n - \mathbf{a}\|\right)^2}{2\sigma^2}\right), \quad (1)$$

while for malfunction anchors, distance estimates are drawn uniformly between 0 and the communication range, $R_{\max}$:

$$p\left(\hat{d}_n | s_n = 1, \mathbf{a}, \mathbf{x}_n\right) = \begin{cases} 1/R_{\max} & \|\mathbf{x}_n - \mathbf{a}\| < R_{\max} \\ 0 & \text{else} \end{cases} \quad (2)$$

B. Localization

1) Least-Squares Method: The least-squares (LS) estimate of $\mathbf{a}$, assuming N functional anchors, is given by

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \sum_{n=1}^{N} \left(\hat{d}_n - \|\mathbf{x}_n - \mathbf{a}\|\right)^2. \quad (3)$$

We note that when all anchors are functioning properly, and the ranging noise is zero-mean Gaussian with a constant variance, the LS estimate is equal to the maximum likelihood (ML) estimate. Moreover, when in addition $\mathbf{a}$ is considered a random variable with a uniform a priori distribution, the LS estimate coincides with the maximum a posteriori (MAP) estimate.

When the agent is somehow informed about which anchors are malfunctioning, it can compute a better LS estimate

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \sum_{n: s_n = 0} \left(\hat{d}_n - \|\mathbf{x}_n - \mathbf{a}\|\right)^2, \quad (4)$$

where the summation is only over those anchors that provide a good distance estimate.

2) Linearized Least-Squares Method: The linearized LS method is derived as follows. Assuming reliable distance estimates, then

$$\hat{d}_j^2 \approx \|\mathbf{a}\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{a}^T \mathbf{x}_j.$$

Collecting the distance estimates in $\hat{\mathbf{d}}^2 = [\hat{d}_1^2, \ldots, \hat{d}_N^2]^T$, then we can introduce $\mathbf{y} = \left[\|\mathbf{x}_1\|^2, \ldots, \|\mathbf{x}_N\|^2\right]^T - \hat{\mathbf{d}}^2$, $\mathbf{z} = [\mathbf{a}^T, -\|\mathbf{a}\|^2]^T$, and $\mathbf{A} = \left[2\left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\right]^T, \mathbf{1}_N\right]$, where $\mathbf{1}_N$ is an $N \times 1$ vector consisting of N ones. This allows us to write $\mathbf{A}\mathbf{z} = \mathbf{y}$. Then $\mathbf{z}$ can be found as

$$\hat{\mathbf{z}} = \mathbf{A}^\dagger \mathbf{y}, \quad (5)$$

where $(\cdot)^\dagger$ denotes the pseudo-inverse. The estimate of $\mathbf{a}$ is given as the first two components of $\hat{\mathbf{z}}$.

When the agent is somehow informed about which anchors are malfunctioning, it can compute a better estimate, by removing rows in $\mathbf{A}$ and $\mathbf{y}$ corresponding to malfunctioning anchors, leading to $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{y}}$. The final estimate of $\mathbf{z}$ is then

$$\hat{\mathbf{z}} = \tilde{\mathbf{A}}^\dagger \tilde{\mathbf{y}}. \quad (6)$$

III. Outlier Detection

In this section, we describe two outlier detection methods. The first method is based on linear programming, inspired by [15]. The second method is based on graphical models, and will be described in a non-cooperative and cooperative context.

A. Linear Programming Method

We linearize the positioning problem as in Section II-B2. Assuming error-free distance estimates, then $\mathbf{A}\mathbf{z} = \mathbf{y}$. Due to outliers, this relation becomes $\mathbf{y} = \mathbf{A}\mathbf{z} + \mathbf{e}$, where $\mathbf{e}$ is a vector of errors due to outliers. When there are few outliers, $\mathbf{e}$ is a sparse vector. An estimate of $\mathbf{e}$ can be found by solving the following linear program

$$\min_{\mathbf{z}} \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_1, \quad (7)$$

where $\|\cdot\|_1$ is the ℓ_1-norm. From the solution $\mathbf{z}^*$, we find $\mathbf{e}$ as $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{A}\mathbf{z}^*$. In practice, the distance estimates $\hat{d}_j$ are not perfect, so the relation becomes $\mathbf{y} = \mathbf{A}\mathbf{z} + \mathbf{e} + \mathbf{n}$. Following [15], solving the LP leads to an estimate of $\widehat{\mathbf{e} + \mathbf{n}} = \mathbf{y} - \mathbf{A}\mathbf{z}^*$. Let $\tilde{\mathbf{y}} = \mathbf{y} - \widehat{\mathbf{e} + \mathbf{n}}$. An estimate of $\mathbf{z}$ is then $\hat{\mathbf{z}} = \mathbf{A}^\dagger \tilde{\mathbf{y}}$. The estimate of $\mathbf{a}$ is again given as the first two components of $\hat{\mathbf{z}}$. An alternative approach consists of solving the LP, removing the row in $\mathbf{A}$ and $\mathbf{y}$ corresponding to the largest outlier (i.e., $\max_i |e_i|$), then solving the LP again, until three anchors are left.

B. Factor Graph Method Without Cooperation

1) Factor Graphs and the Sum-Product Algorithm: A factor graph [16] represents a factorization of a distribution as a bipartite graph: it consists of variable vertices (one per variable, drawn as circles), factor vertices (one per factor, drawn as rectangles), and edges (when the variable associated with the adjacent variable vertex appears in the factor associated with the adjacent factor vertex). In conjunction with message passing algorithms, factor graphs enable efficient computation of marginal a posteriori distributions [17]. The marginal a posteriori distributions of the variables can be computed by performing the sum-product algorithm (SPA) on the factor graph. The SPA consists of two message passing rules. Given a variable (say x) that appears in function $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$. The message from variable vertex x to factor vertex f is a real-valued, non-negative function over x, given by

$$\mu_{x \to f}(x) = \mu_{g \to x}(x) \times \mu_{h \to x}(x). \quad (8)$$

Given a factor $f(x, y, z)$, the message from factor vertex f to variable vertex x is a non-negative function over x, given by

$$\mu_{f \to x}(x) = \sum_{y,z} f(x, y, z) \times \mu_{y \to f}(y) \times \mu_{z \to f}(z). \quad (9)$$

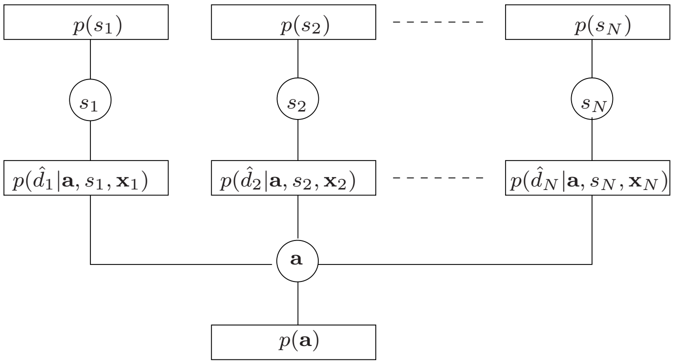

Figure 2. Factor graph for non-cooperative positioning corresponding to the factorization $p(\mathbf{a}) \prod_{n=1}^{N} p(\hat{d}_n|\mathbf{a}, s_n, \mathbf{x}_n)p(s_n)$.

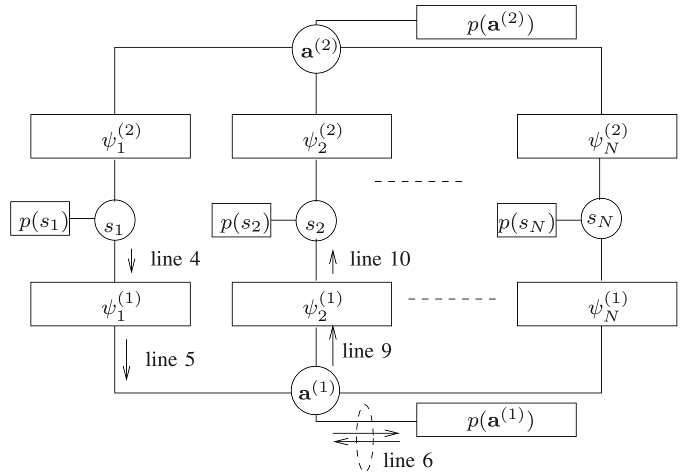

Figure 3. Factor graph for cooperative positioning where $\psi_n^{(i)}(\mathbf{a}^{(i)}, s_n) = p(\hat{d}_n^{(i)}|\mathbf{a}^{(i)}, s_n, \mathbf{x}_n)$.

Finally, the marginal a posteriori distribution of x is given by, up to a normalization constant $b(x) \propto \mu_{f \to x}(x) \times \mu_{x \to f}(x)$, where $f(\cdot)$ is any function in which x appears as a variable. The notation $b(\cdot)$ refers to the so-called *belief* of variable x.

2) Outlier detection : We consider an a posteriori distribution that factorizes as follows:

$$p(\mathbf{a}, \mathbf{s}|\hat{\mathbf{d}}, \mathbf{X}) \propto p(\hat{\mathbf{d}}|\mathbf{a}, \mathbf{s}, \mathbf{X})p(\mathbf{a})p(\mathbf{s}) \quad (10)$$

$$= p(\mathbf{a}) \prod_{n=1}^{N} p(\hat{d}_n|\mathbf{a}, s_n, \mathbf{x}_n)p(s_n) \quad (11)$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{s} = [s_1, \ldots, s_N]$. The distribution $p(\hat{d}_n|\mathbf{a}, s_n, \mathbf{x}_n)$ is provided in (1)–(2). The factor graph corresponding to this factorization is shown in Figure 2. Abbreviating $p(\hat{d}_n|\mathbf{a}, s_n, \mathbf{x}_n)$ by ψ_n, then the message from factor vertex ψ_n to variable vertex $\mathbf{a}$ is, according to (9), given by

$$\mu_{\psi_n \to \mathbf{a}}(\mathbf{a}) = \sum_{s_n \in \{0,1\}} p(s_n)p(\hat{d}_n|\mathbf{a}, s_n, \mathbf{x}_n). \quad (12)$$

Finally, the marginal a posteriori distribution of $\mathbf{a}$ is

$$p(\mathbf{a}|\hat{\mathbf{d}}, \mathbf{X}) = \prod_{n=1}^{N} \mu_{\psi_n \to \mathbf{a}}(\mathbf{a})p(\mathbf{a}) \quad (13)$$

$$\doteq b(\mathbf{a}). \quad (14)$$

Based on this marginal distribution, the estimated position can be found as the mode or mean of $p(\mathbf{a}|\hat{\mathbf{d}}, \mathbf{X})$.

C. Factor Graph Method with Cooperation

Here we will extend the above marginalization problem to a cooperative scenario. We will focus on a scenario with two agents and N anchors, though this scenario is easily extended to more general settings. The two agents (with positions $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$) can range with all N anchors, leading to distance estimates $\hat{d}_n^{(m)}$, $m \in \{1, 2\}$, $n \in \{1, 2, \ldots, N\}$. Furthermore, the agents can exchange information. Contrary to other forms of cooperative positioning [8], we do not require that agents estimate distances among each-other. The factorization of the joint a posteriori distribution is given by

$$p(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{s}|\hat{\mathbf{d}}^{(1)}, \hat{\mathbf{d}}^{(2)}, \mathbf{X}) \propto \quad (15)$$
$$p(\mathbf{a}^{(1)})p(\mathbf{a}^{(2)}) \prod_{n=1}^{N} \left[p(s_n) \prod_{m \in \{1,2\}} p(\hat{d}_n^{(m)}|\mathbf{a}^{(m)}, s_n, \mathbf{x}_n) \right].$$

The corresponding factor graph is shown in Figure 3. The message passing algorithm now operates as shown in Algorithm 1. The algorithm simply follows the SPA. The messages $\mu_{\psi_n^{(m)} \to s_n}(s_n)$ are exchanged[1] between agents and can be sent as packets over the wireless link. They represent the confidence agent m has that anchor n is malfunctioning. Note that lines 5–12 in Algorithm 1 are computed locally by agent m. The only interaction between agents is through the messages $\mu_{\psi_n^{(m)} \to s_n}(s_n)$, in lines 4 and 12.

IV. Numerical Results

A. Simulation Setup

We consider a scenario with $N = 10$ anchors and two agents, randomly distributed in an environment of size 50 distance units by 50 distance units (d.u.). We set $R_{\max} = 200$ d.u., and $\sigma = 1$ d.u. Numerical optimization and integration was carried out on a grid of resolution 0.5 d.u. Malfunctioning anchors are generated according to a Bernoulli distribution with parameter $p \in [0, 1]$. In other words, out of the N anchors

[1]The messages in line 12 of Algorithm 1 can efficiently represented as log-likelihood ratios (LLRs). Let

$$\lambda_n^{(m)} = \log \frac{\mu_{\psi_n^{(m)} \to s_n}(1)}{\mu_{\psi_n^{(m)} \to s_n}(0)}, \quad (16)$$

then line 12 of Algorithm 1 can be replaced by: broadcast $\boldsymbol{\lambda}^{(m)} = [\lambda_1^{(m)}, \ldots, \lambda_N^{(m)}]$. Line 4 of Algorithm 1 should then be replaced by

$$\mu_{s_n \to \psi_n^{(m)}}(s_n) \propto \frac{e^{s_n \sum_{k \neq m} \lambda_n^{(k)}}}{1 + e^{\sum_{k \neq m} \lambda_n^{(k)}}} p(s_n). \quad (17)$$

Algorithm 1 Cooperative outlier detection.

initialize $\mu_{\psi_n^{(m)} \to s_n}(s_n) = 1/2, \, \forall n, m$
for $l = 1$ to L **do** {iteration index}
  for $m = 1$ to M **do** {agent index}
    compute message $\mu_{s_n \to \psi_n^{(m)}}(s_n)$

$$\mu_{s_n \to \psi_n^{(m)}}(s_n) \propto \prod_{k \neq m} \mu_{\psi_n^{(k)} \to s_n}(s_n) p(s_n)$$

    compute message $\mu_{\psi_n^{(m)} \to \mathbf{a}^{(m)}}(\mathbf{a}^{(m)})$

$$\mu_{\psi_n^{(m)} \to \mathbf{a}^{(m)}}(\mathbf{a}^{(m)}) \propto \sum_{s_n \in \{0,1\}} \mu_{s_n \to \psi_n^{(m)}}(s_n) p(\hat{d}_n | \mathbf{a}, s_n, \mathbf{x}_n)$$

    compute current belief

$$b(\mathbf{a}^{(m)}) \propto \prod_{n=1}^{N} \mu_{\psi_n^{(m)} \to \mathbf{a}^{(m)}}(\mathbf{a}^{(m)}) p(\mathbf{a}^{(m)})$$

    compute estimate of $\mathbf{a}^{(m)}$

$$\hat{\mathbf{a}}^{(m)} = \arg\max_{\mathbf{a}^{(m)}} b(\mathbf{a}^{(m)})$$

    for $n = 1$ to N **do** {anchor index}
      compute message $\mu_{\mathbf{a}^{(m)} \to \psi_n^{(m)}}(\mathbf{a}^{(m)})$

$$\mu_{\mathbf{a}^{(m)} \to \psi_n^{(m)}}(\mathbf{a}^{(m)}) \propto \frac{b(\mathbf{a}^{(m)})}{\mu_{\psi_n^{(m)} \to \mathbf{a}^{(m)}}(\mathbf{a}^{(m)})}$$

      compute message $\mu_{\psi_n^{(m)} \to s_n}(s_n)$

$$\mu_{\psi_n^{(m)} \to s_n}(s_n) \propto \int p(\hat{d}_n^{(m)} | \mathbf{a}^{(m)}, s_n, \mathbf{x}_n) \mu_{\mathbf{a}^{(m)} \to \psi_n^{(m)}}(\mathbf{a}^{(m)}) \mathrm{d}\mathbf{a}^{(m)}$$

    end for
    broadcast $\left\{ \mu_{\psi_n^{(m)} \to s_n}(s_n) \right\}_{n=1}^{N}$ to all other agents, $s_n \in \{0, 1\}$
  end for
end for

an average of pN anchors will be malfunctioning. A malfunctioning anchor will generate distance estimates according to (2). The remaining anchors will generate distance estimates according to (1). For every data point in the figures, 1000 random networks were generated. We collect the localization errors $\|\hat{\mathbf{a}} - \mathbf{a}\|$.

B. Localization Performance

We first investigate linearized LS positioning. The positioning results, in terms of the average localization error as a function of p, are shown in Figure 4. We observe that using all anchors leads to severe degradation when $p > 0$, even for small p. This motivates the need for performing outlier detection. When only the reliable anchors are taken into account, the bottom-most curve indicates that for $p \leq 0.3$, only small degradations are incurred. Applying the LP-based method from [15] results in improved performance for $p \leq 0.6$, but degradations are still large for $p \geq 0.2$. Performing an iterative LP, removing the worst anchor at every iteration until 3 anchors remain results in the curve labeled "best 3 anchors". We see that for $p \in [0, 0.1]$, this method yields excellent performance. Observe that for $p = 0$, a small degradation is visible compared to the other methods, because not all the anchors are taken into consideration. We conclude that the best performance is obtained from the iterative LP method, but small degradations are obtained only for small values of p.

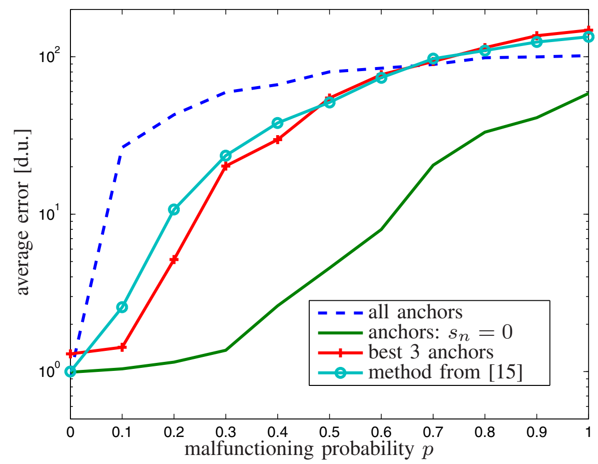

Figure 4. Positioning performance for linear LS positioning as a function the malfunctioning probability.

We now move on the non-linear LS positioning. The positioning results as a function of p are shown in Figure 5. As before, fusing information from all the anchors leads to severe degradations when $p > 0$, while using only those anchors for which $s_n = 0$ incurs negligible degradations for $p \leq 0.3$. We now consider three variations of the factor graph (FG) method: non-cooperative with known p (labeled "FG-non-coop p known"), non-cooperative with assumed $p = 0.5$ (labeled "FG-non-coop $p = 0.5$"), and cooperative among $M = 2$ agents with known p (labeled "FG-coop p known"). The method FG-non-coop with known p yields near-optimal performance for $p \in [0, 0.2]$. When the agent does not know p and replaces it with $p = 0.5$, small degradations are incurred. When two agents cooperate by exchanging reliability information of the $N = 10$ anchors, additional performance gains can be achieved: the method FG-coop p known achieves near-optimal performance for $p \in [0, 0.5] \cup [0.8, 1]$. This is in stark contrast with the methods in Figure 4, where good performance was only obtained for $p \leq 0.1$.

C. EXIT Chart Analysis

To gain additional insight in the behavior of the impact of cooperation, we have performed an EXIT chart analysis. Such extrinsic information transfer charts were originally

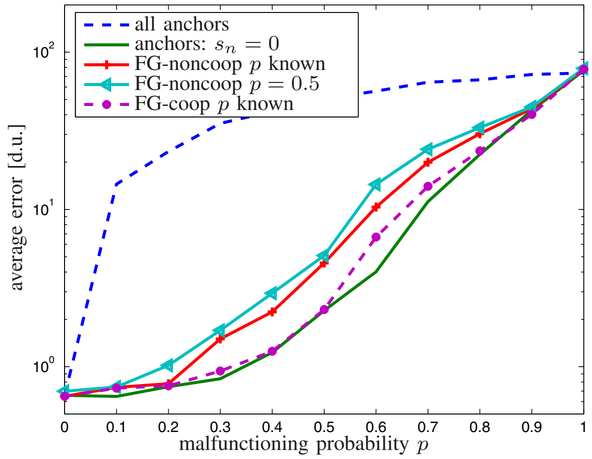

Figure 5. Positioning performance for non-linear LS positioning as a function the malfunctioning probability.

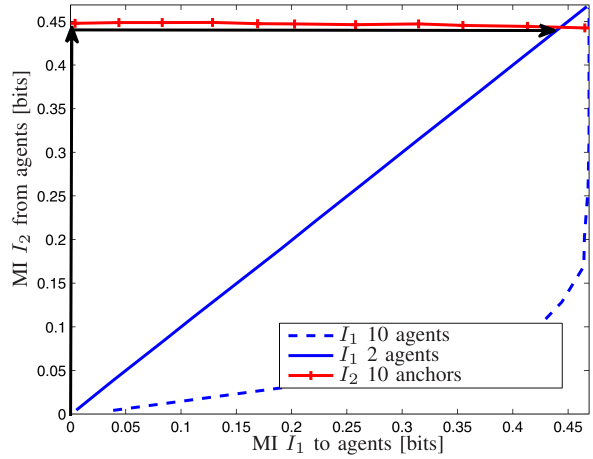

Figure 6. EXIT chart for $p = 0.1$. Arrows indicate iterative processing for a scenario with 10 anchors and 2 agents.

introduced in the coding community to understand iterative decoding [18]. As a detailed description of EXIT charts is beyond the scope of this paper, we limit ourselves to a short description: we represent the messages $\mu_{s_n \to \psi_n^{(m)}}(s_n)$ and $\mu_{\psi_n^{(m)} \to s_n}(s_n)$ in log-likelihood ratio (LLR) format, say $\lambda_n^{(m)}$ and $\gamma_n^{(m)}$, respectively. These LLRs, which are scalars, are then considered as random variables. We can then numerically compute the mutual information (MI) between the LLRs and the state of the anchors $s_n \in \{0, 1\}$. For example, the MI between $\lambda_n^{(m)}$ and s_n is given by

$$I_1 = \sum_{s_n \in \{0,1\}} p(s_n) \int_{-\infty}^{+\infty} p(\lambda_n^{(m)}|s_n) \log_2 \frac{p(\lambda_n^{(m)}|s_n)}{p(\lambda_n^{(m)})} \mathrm{d}\lambda_n^{(m)}, \tag{18}$$

which is independent of m. The MI between $\gamma_n^{(m)}$ and s_n is denoted by I_2, and is computed in a similar fashion. Since the values of $\lambda_n^{(m)}$ depend on $\gamma_n^{(m)}$ and vice versa, we can consider I_1 as a function of I_2, and I_2 as a function of I_1. To determine I_2 as a function of I_1, we provide an agent with artificial messages $\mu_{s_n \to \psi_n^{(m)}}(s_n)$ in LLR format, which have a preset mutual information I_1 with the variables s_n. We then compute the MI I_2 of the outgoing messages $\mu_{\psi_n^{(m)} \to s_n}(s_n)$ (also in LLR format) with the same variables s_n. By changing the input mutual information I_1, we can generate a curve of output MI I_2. Similarly, to determine I_1 as a function of I_2, we provide artificial messages $\mu_{\psi_n^{(m)} \to s_n}(s_n)$ (in LLR format) with a certain MI I_2 with s_n to the variable vertex s_n. We then compute the MI I_1 between the variable s_n and the outgoing message $\mu_{s_n \to \psi_n^{(m)}}(s_n)$ (in LLR format). By changing the input mutual information I_2, we can generate a curve of output MI I_1. These two curves form the EXIT chart. The EXIT chart depends on p, the number of agents, and the number of anchors.

Figure 6 shows the EXIT chart for $p = 0.1$. First of all, we note that the MI is limited to 0.469 (expressed in bits), corresponding to the entropy of a Bernoulli random variable with success probability $p = 0.1$. Secondly, we see that irrespective of the quality of information provided to the agents, I_2 is roughly constant to 0.45 bits. When iterating, the MI saturates to 0.45 bits (indicated by the black arrows, resulting in an intersection point). From Figure 5, we expect little gain from cooperation among two agents. The EXIT chart confirms this.

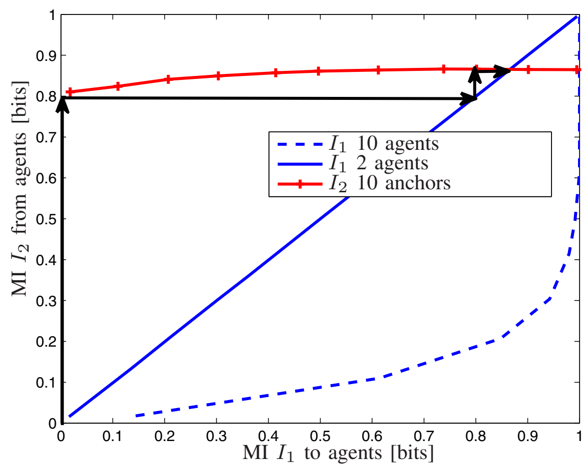

Figure 7. EXIT chart for $p = 0.5$. Arrows indicate iterative processing for a scenario with 10 anchors and 2 agents.

Figure 7 shows similar results for $p = 0.5$. Note the MI is limited to 1 bit. Contrary to Figure 6, we now observe that as more reliable information is provided to an agent, the MI I_2 increases from 0.8 bits to a maximum 0.86 bits. Hence, iterations will be beneficial. When more agents cooperate, the improvement will be faster. These findings corroborate our results from Figure 5.

V. Conclusions

Ranging outliers in location-aware wireless networks can lead to catastrophic localization failures. We have proposed a novel method of dealing with outliers induced by malfunctioning or malicious anchors. Based on probabilistic graphical models, we propose a message passing algorithm that enables multiple devices to cooperate in identifying points of failure. In comparison to existing methods that offer good performance for a small amount of outliers, our novel methods lead to near-optimal performance for up to 30% outliers. We show how simple cooperation between agents leads to increased performance, both through simulations and through an EXIT chart analysis. Future work includes the extension to colluding anchors, as well as a reduced-complexity version of our message passing algorithms.

Acknowledgment

This research is supported, in part, by the Swedish Research Council (VR), under Grant No. 2010-5889.

References

[1] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 45–57, ACM, 2004.

[2] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, pp. 41–53, July 2005.

[3] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, pp. 1247–1256, Aug. 2003.

[4] S. J. Ingram, D. Harmer, and M. Quinlan, "Ultrawideband indoor positioning systems and their use in emergencies," in *Position Location and Navigation Symposium*, pp. 706–715, Apr. 2004.

[5] R. J. Fontana and S. J. Gunderson, "Ultra-wideband precision asset location system," in *Proc. of IEEE Conf. on Ultra Wideband Syst. and Technol. (UWBST)*, vol. 21, pp. 147–150, May.

[6] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *Oceanic Engineering, IEEE Journal of*, vol. 31, no. 4, pp. 949 –958, 2006.

[7] E.D. Kaplan and C.J. Hegarty, ed., *GPS: Principles and Applications*. Artech House, 2006.

[8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, pp. 427–450, Mar. 2009.

[9] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[10] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 159 –170, 2010.

[11] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek, "Distributed anomaly detection in wireless sensor networks," in *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*, pp. 1–5, IEEE, 2007.

[12] S. Venkatesh and R. Buehrer, "NLOS mitigation using linear programming in ultrawideband location-aware networks," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 5, pp. 3182 –3198, 2007.

[13] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, pp. 1026–1035, Sept. 2010.

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

[15] J. Picard and A. Weiss, "Accurate Geolocation in the Presence of Outliers using Linear Programming," in *17th European Signal Processing Conference (EUSIPCO, Glasgow, Scotland*, 2009.

[16] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[17] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006.

[18] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *Communications, IEEE Transactions on*, vol. 49, no. 10, pp. 1727–1737, 2002.