INDOOR SENSOR NODE POSITIONING USING UWB RANGE MEASUREMENTS

Mohammad Reza Gholami, Erik G. Ström, and Mats Rydström

Dept. of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg, SWEDEN
phone: +46 31 772 1000, fax: +46 31 772 1748, email: {moreza, estrom, d98mats}@chalmers.se
web: www.chalmers.se/s2

ABSTRACT

The performance of several algorithms for positioning a single sensor node based on distance estimates to it from a number of nodes at known positions (anchor nodes) is compared when the distance estimates are obtained from a measurement campaign. The distance estimates are based on time-of-arrival measurements done by ultrawideband devices in an indoor office environment. The compared algorithms are based on nonlinear least squares, least squares and total least squares after data preprocessing, and projection methods. No algorithm is uniformly best; however, least squares and total least squares after data preprocessing show higher mean squared errors in almost all cases, while the nonlinear least squares and projection methods have similar performance; projection methods performs slightly better.

1. INTRODUCTION

Position information of the nodes that make up a wireless sensor network is required in most, if not all, applications. Preferably, the positioning should be carried out by the network itself to avoid a cumbersome manual node deployment.

We will here consider the problem of positioning a single node using range (distance) estimates to a number of nodes at known positions (so-called anchor nodes or reference nodes). In general, the range estimates can be based on many different types of measurements, e.g., received signal strength (RSS) or, as the case is in this papers, on time of arrival (TOA) [1]. The quality of the positioning depends on the quality of the range measurements, the geometry of the network, and the performance of the positioning algorithm. In particular, it is important that any assumptions on the range estimates posed by the positioning algorithms are satisfied to a reasonable degree. For example, a maximum likelihood approach requires knowledge of the joint PDF of the range estimates. In complex environments, e.g., indoor scenarios, the PDF might not be readily available, and we have to settle for other methods, such as least squares methods.

The well-known nonlinear least squares (NLS) method will therefore be used to benchmark a number of more novel algorithms that offer either lower computational complexity, robustness against positive bias in the distance estimates (which tends to occur in non line-of-sight situations), or better performance compared to standard NLS. Details about the algorithms are found in Sec. 3.

This work was supported in part by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715) and in part by the Swedish Research Council (contract no. 2007-6363)

The range estimates used to evaluate the algorithms come from a recent ultrawideband (UWB) measurement campaign planned and carried out under the auspices of NEWCOM++, an EU FP7 Network of Excellence (see Sec. 6 for details). UWB technology has the potential to deliver very accurate range estimates and thereby enabling accurate positioning [2]. However, it is not clear how to best use the range estimates, and this paper aims to shed some light on this.

This paper is organized as follows: in Sec. 2 the range measurement scenario is explained, the considered positioning algorithms are introduced in Sec. 3, and their performance using the measured ranges are compared in Sec. 4, and conclusions are drawn in Sec. 5.

2. RANGE MEASUREMENTS

The measurement campaign was performed on the second floor of the Department of Electronics, Information and Systems at the Cesena campus of the University of Bologna, Italy. The sensor node positions, numbered from 1 to 20, are indicated with red dots in the floor plan in Fig.1. The measurement area was equipped with typical office furniture like tables, chairs, etc. During measurements, no moving objects were present between the nodes and the nodes remained at fixed positions. The nodes were placed at the same height (1.13 meters), and the range measurements can therefore be used by 2D position algorithms.

Two Timedomain PulseON 220 UWB nodes [3] were used to produce range estimates between any pair of distinct node positions. A total of 1000 range measurements for each node position pair were made by the devices using their built-in, proprietary TOA-based estimator. Unfortunately, no details about the range estimator are publicly available.

The range estimate $\hat{d}_{i,j}$ between node i and j is simply modeled as

$$\hat{d}_{i,j} = d_j(\theta_i) + w_{i,j}, \qquad i, j \in \{1, \ldots, 20\} \tag{1}$$

where $w_{i,j}$ is the ranging error, $d_j(\theta)$ is the distance from node j to θ, i.e.,

$$d_j(\theta) = \|\theta - \theta_j\|. \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm and $\theta_k = [x_k \quad y_k]^T$ is the coordinates of node k, see Fig.1. In this paper, we consider the ranging error $w_{i,j}$ as a random variable with unknown distribution. Due to connectivity issues, not all $\hat{d}_{i,j}$ are available (see Fig. 2).

We can use the measurement data to evaluate the performance of positioning algorithms by considering all nodes

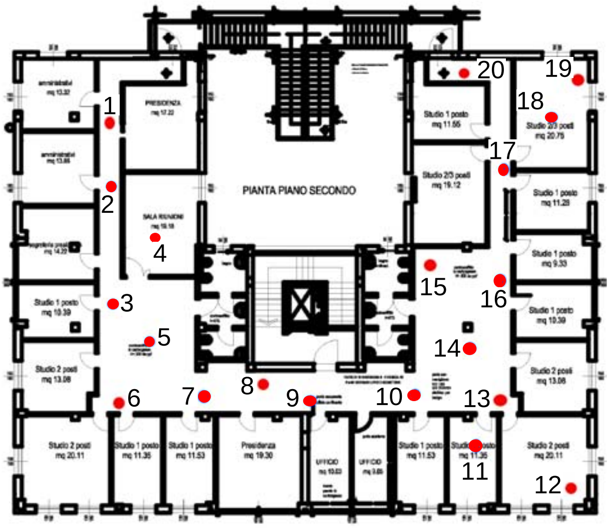

Figure 1: The measurement environment where the red points indicate the sensor node positions.

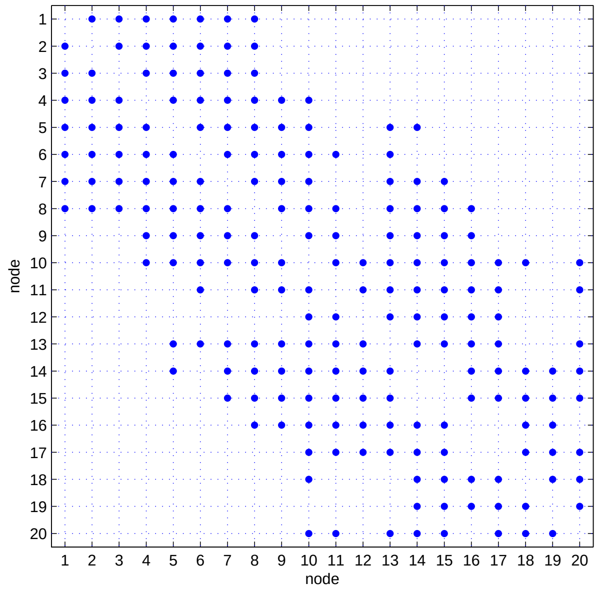

Figure 2: Connectivity matrix: the markers show which nodes were connected during the measurements and, consequently, for which node pairs that distance estimates exist.

that is connected to node i, for $i = 1,2,\dots,20$, as anchor nodes. Since we know the position of the nodes, we can easily compute the positioning error and form statistics, e.g., cumulative distribution function (CDF) or mean squared value (MSE) of the error.

3. POSITIONING ALGORITHMS

In this section, some previously proposed methods are briefly reviewed and four partly new algorithms are introduced: nonlinear least squares using differential evolution (Sec. 3.2), projection onto rings (Sec. 3.4), least squares (Sec. 3.5), and total least squares (Sec. 3.6). In the following, the index i is used for the unknown node and the set of anchor nodes used to position node i is denoted by $\mathscr{C}_i$. We make $\mathscr{C}_i$ as large as possible by including all nodes that are connected to node i into $\mathscr{C}_i$ (which implies that $\hat{d}_{i,j}$ exists if and only if $j \in \mathscr{C}_i$), and we let N_i be the number of elements in $\mathscr{C}_i$. As an example, we deduce from the 20th row in Fig. 2 that $\mathscr{C}_{20} = \{10,11,13,14,15,17,18,19\}$ and $N_{20} = 8$.

3.1 Nonlinear least squares

The nonlinear least squares (NLS) position estimate based on the ranging measurement (1) can be found as the solution to the minimization problem

$$\hat{\theta}_i = \arg\min_{\theta} \sum_{j \in \mathscr{C}_i} \|\hat{d}_{i,j} - d_j(\theta)\|^2. \tag{3}$$

The signal model (2) is nonlinear in the unknown parameters, which implies that the objective function in (3) normally suffers from local minima and saddle points. There is, in general, no analytical solution to (3) and we are forced to use numerical methods. We note, that if $w_{i,j}$ are uncorrelated, identically distributed, zero-mean Gaussian random variables for all $j \in \mathscr{C}_i$, the NLS estimate is also the maximum likelihood estimate [4].

In this paper we will compute the NLS estimate using two different methods: the MATLAB routine `lsqnonlin` [5] and the differential evolution method described below; both search methods are initialized randomly in the deployment area (see Fig. 1).

3.2 Differential Evolution

Differential evolution (DE) is a global optimization method, which was introduced by Storn and Price [6]. DE can be thought of as a random search of the objective function, which, in our case, is the NLS objective function used to estimate θ_i, i.e., $J_i(\theta) = \sum_{j\in\mathscr{C}_i} \|\hat{d}_{i,j} - d_j(\theta)\|^2$. To implement the search, DE maintains a population of N_p candidate parameter vectors, $\hat{\theta}_i^k(n)$ for $n = 1,2,\dots,N_p$, where $k = 0,1,\dots,K$ is the generation index. A new generation is formed from the previous generation in a partially random fashion, as detailed below, and the final estimate $\hat{\theta}_i$ is best candidate vector found in any generation.

The variant of DE considered here (DE/rand/1/bin, see [6]) is implemented as follows.

1. Initialize first generation: select randomly $\hat{\theta}_i^0(n)$ for $n = 1,2,\dots,N_P$.
2. Form new generations: for $k = 0,1,\dots,K$, form the $k+1$ generation from the kth generation as
   (a) Mutate: for $n = 1,2,\dots,N_P$ form
   $$\mathbf{v}(n) = \hat{\theta}_i^k(r_1) + \left[\hat{\theta}_i^k(r_2) - \hat{\theta}_i^k(r_3)\right] F,$$
   where r_1, r_2, r_3 are distinct integers drawn uniformly from the set $\{1,2,\dots,N_p\}$.
   (b) Crossover: for $n = 1,2,\dots,N_P$, (i) let $\mathbf{u}(n) = \mathbf{v}(n)$ and (ii) replace, with probability p_{CR}, each coordinate of $\mathbf{u}(n)$ with the corresponding coordinate of $\hat{\theta}_i^k(n)$
   (c) Select: for $n = 1,2,\dots,N_P$,
   $$\hat{\theta}_i^{k+1}(n) = \begin{cases} \hat{\theta}_i^k(n), & \text{if } J_i(\hat{\theta}_i^k(n)) \le J_i(\mathbf{u}(n)) \\ \mathbf{u}(n), & \text{otherwise} \end{cases}$$

We note from the description above that the tuning parameters of DE is the number of candidate vectors in a generation, N_P, the step size F, the crossover probability, p_{CR}, and the number of new generations K. The constraints are that $N_P \geq 4$ and $F \in [0,2]$. Some practical advice on how to set the tuning parameters can be found at `www.icsi.berkeley.edu/~storn/code.html`.

3.3 Projection onto convex sets

It is clear that the minimum of each term in the cost function in (3) is obtained when $\hat{d}_{i,j} = d_j(\theta)$. Now, suppose we define the discs $\mathscr{D}_{i,j}$ as

$$\mathscr{D}_{i,j} = \{\theta \in \mathbb{R}^2 : d_j(\theta) \leq \hat{d}_{i,j}\}, \qquad j \in \mathscr{C}_i, \tag{4}$$

it then is reasonable to define an estimate of θ_i as a point in the intersection $\mathscr{D}_i$ of the discs $\mathscr{D}_{i,j}$,

$$\hat{\theta}_i \in \mathscr{D}_i = \bigcap_{j \in \mathscr{C}_i} \mathscr{D}_{i,j}. \tag{5}$$

A method called projection on convex sets (POCS) can be used to compute an estimate of the form (5), and was proposed for the positioning problem by Blatt and Hero in [7]. If the intersection is the empty set (which can occur due to measurement noise), the POCS estimate will be any point that minimizes the sum of the distance to the discs,

$$\hat{\theta}_i = \arg\min_{\theta} \sum_{j \in \mathscr{C}_i} \|\theta - \mathscr{P}_{\mathscr{D}_{i,j}}(\theta)\|, \tag{6}$$

where $\mathscr{P}_{\mathscr{D}_{i,j}}(\theta)$ is the orthogonal projection of θ onto $\mathscr{D}_{i,j}$. POCS, and several variations of it, have been studied in [1,7,8]. In these papers, it is concluded that POCS (as described above) has problems when the unknown node is outside the convex hull of the anchor nodes (the problem was alleviated to some extent in [8]). On the other hand, POCS is quite robust against overestimated range estimates (which may occur in non-line-of-sight environments) as long as the unknown node is inside the convex hull of the anchor nodes.

3.4 Projection onto rings

In the case when the measurement noise in (1) is small, we can often improve POCS by replacing the disc $\mathscr{D}_{i,j}$ with a ring (or, more formally, an annulus) defined as

$$\mathscr{R}_{i,j} = \{\theta \in \mathbb{R}^2 : \hat{d}_{i,j} - \varepsilon_l \leq d_j(\theta) \leq \hat{d}_{i,j} + \varepsilon_u\}, \quad j \in \mathscr{C}_i \tag{7}$$

where $\varepsilon_l + \varepsilon_u$ determines the width of the ring. The width is a tuning parameter of the resulting algorithm; it is reasonable to make the width dependent on the ranging error statistics in (1). Since we do not assume any knowledge of the error statistics, we simply choose $\varepsilon_l = \varepsilon_r = 0$ in the numerical results presented in Sec. 4.

The projection onto rings (POR) positioning algorithm is an iterative algorithm. The estimate after the kth iteration is denoted as $\hat{\theta}_i^k$, and is computed as

1. Initialize: assign $\hat{\theta}_i^0$ to an arbitrary value.
2. Iterate: for $k > 0$, let $\hat{\theta}_i^{k+1} = \hat{\theta}_i^k + \lambda_k[\mathscr{P}_{\mathscr{R}_{i,j(k)}}(\hat{\theta}_i^k) - \hat{\theta}_i^k]$.
3. Terminate after convergence or after a fixed number of iterations

Here, $j(k)$ is a periodic function that cycles through all elements in the set $\mathscr{C}_i$ and $\{\lambda_k\}_{k\geq 1}$ is a sequence of relaxation parameters. The relaxation parameters can be tuned to enhance performance and convergence rate. No such optimization is attempted here; we simply choose

$$\lambda_k = \begin{cases} 1, & k \leq k_0 \\ \left\lceil \frac{k-k_0+1}{N_i} \right\rceil^{-1}, & k > k_0 \end{cases} \tag{8}$$

where $\lceil x \rceil$ is the smallest integer that is greater or equal to x. Finally, the projection onto the ring $\mathscr{R}_{i,j}$ can be written as $\mathscr{P}_{\mathscr{R}_{i,i}}(\theta) = \theta + r(\theta_j - \theta)/d_j(\theta)$, where

$$r = \begin{cases} d_j(\theta) - (\hat{d}_{i,j} - \varepsilon_l), & d_j(\theta) < \hat{d}_{i,j} - \varepsilon_l \\ d_j(\theta) - (\hat{d}_{i,j} + \varepsilon_u), & d_j(\theta) > \hat{d}_{i,j} + \varepsilon_u \\ 0, & \text{otherwise} \end{cases}.$$

3.5 Least squares

In this section, we formulate a linear LS algorithm similar to the acoustic source localization algorithm in [9]. To form a linear least squares problems, we need to find a signal model that is linear in the unknown parameters. One approach is to consider pairs of distance estimates as follows. We can form $M_i = N_i(N_i - 1)/2$ distinct 2-element subsets (pairs) of $\mathscr{C}_i$. For the mth pair, $\{j,k\}$, it is easily seen from (2) that

$$b_m(\theta) = [d_k^2(\theta) - \|\theta_k\|^2] - [d_j^2(\theta) - \|\theta_j\|^2] = \underbrace{2(\theta_j - \theta_k)^T}_{=\mathbf{a}_m^T}\theta, \tag{9}$$

which is seen to be a linear function of θ. We couple this signal model with the measurement

$$\hat{b}_m = (\hat{d}_{i,k}^2 - \|\theta_k\|^2) - (\hat{d}_{i,j}^2 - \|\theta_j\|^2). \tag{10}$$

An estimate of θ_i can now be obtained by fitting the signal model (9) to the measurements (10). To this end, we form

$$\mathbf{b}_i(\theta) = [b_1(\theta) \quad \cdots \quad b_{M_i}(\theta)]^T = [\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_{M_i}]^T\theta = \mathbf{A}_i\theta$$

and compute the estimate as

$$\hat{\theta}_i = \arg\min_{\theta} \|\hat{\mathbf{b}}_i - \mathbf{b}_i(\theta)\| = \mathbf{A}_i^{\dagger}\hat{\mathbf{b}}_i, \tag{11}$$

where $\hat{\mathbf{b}}_i = \begin{bmatrix}\hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{M_i}\end{bmatrix}^T$ and $\mathbf{A}_i^{\dagger}$ is the left-hand pseudoinverse of $\mathbf{A}_i$. Assuming that $\mathbf{A}_i$ has full column rank, $\mathbf{A}_i^{\dagger} = (\mathbf{A}_i^T\mathbf{A}_i)^{-1}\mathbf{A}_i^T$.

3.6 Total least square approach

The anchor nodes are normally assumed to be at fixed and known positions. However, in practice, there are uncertainties in anchor node positions due to imperfect deployment, movement of anchor nodes, etc. One approach to take these uncertainties into account is to apply a total least squares framework [10].

To explain how this can be done for the problem at hand, we start by rewriting the least squares problem in (11) in an equivalent form: the least squares estimate $\hat{\theta}_i$ is a solution to $\mathbf{A}_i\theta = \tilde{\mathbf{b}}_i$, where $\tilde{\mathbf{b}}_i$ is found as

$$\tilde{\mathbf{b}}_i = \arg\min_{\mathbf{b}' \in \mathbb{R}^{M_i}} \|\hat{\mathbf{b}}_i - \mathbf{b}'\| \qquad \text{subject to } \mathbf{b}' \in \text{range}(\mathbf{A}_i) \tag{12}$$

Hence, we can think of $\tilde{\mathbf{b}}_i$ as the smallest perturbation to the measurement $\hat{\mathbf{b}}_i$ such that $\tilde{\mathbf{b}}_i$ is consistent with the signal model (i.e., that $\tilde{\mathbf{b}}_i$ is a linear combination of the columns of $\mathbf{A}_i$).

Now, if we allow for uncertainties in the anchor node positions, the $\mathbf{A}_i$ matrix contains uncertainties, just as $\hat{\mathbf{b}}_i$ does. It is then logical to look for small perturbations to both $\mathbf{A}_i$ and $\hat{\mathbf{b}}_i$ such that the signal model is satisfied. The total least squares (TLS) estimate is a solution to $\tilde{\mathbf{A}}_i\theta = \tilde{\mathbf{b}}_i$, where $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{b}}_i$ are found as

$$\begin{aligned} \begin{bmatrix}\tilde{\mathbf{A}}_i & \tilde{\mathbf{b}}_i\end{bmatrix} &= \arg \min_{[\mathbf{A}' \;\; \mathbf{b}'] \in \mathbb{R}^{M_i \times 3}} \left\| \begin{bmatrix}\mathbf{A}_i & \hat{\mathbf{b}}_i\end{bmatrix} - \begin{bmatrix}\mathbf{A}' & \mathbf{b}'\end{bmatrix} \right\|_F \\ &\quad \text{subject to } \mathbf{b}' \in \operatorname{range}(\mathbf{A}') \end{aligned} \tag{13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

As shown in [10], the TLS estimate can be computed as

$$\hat{\theta}_i = (\mathbf{A}_i^T\mathbf{A}_i - \sigma^2\mathbf{I}_2)^{-1}\mathbf{A}_i^T\hat{\mathbf{b}}_i, \tag{14}$$

where σ is the smallest singular value of $\begin{bmatrix}\mathbf{A}_i & \hat{\mathbf{b}}_i\end{bmatrix}$ and $\mathbf{I}_2$ is the 2×2 identity matrix.

4. NUMERICAL RESULTS

In this section, the performance of the six algorithms described in Sec. 3 are compared when applied to the measurement data described in Sec. 2. Performance is measured in terms of mean-squared value and cumulative distribution function (CDF) of the position error $\|\hat{\theta}_i - \theta_i\|$.

Numerical solutions to the nonlinear least squares problem in (3) are found using the MATLAB routine `lsqnonlin` [5] using default parameters (denoted NLS-M in the plots) and differential evolution (NLS-DE) of type DE/rand/1/bin with $N_P = 300$, $F = 0.6$, $p_{CR} = 0.5$, and $K = 0.6$. Differential evolution simulations were done with the MATLAB routine `devec3` downloaded from `www.icsi.berkeley.edu/~storn/code.html`. Both algorithms were initialized randomly in the deployment area.

Projection onto convex sets (POCS) and projection onto rings (POR) use the relaxation parameters as defined in (8) with $k_0 = N_i$, where N_i is the number of anchor nodes. The maximum number of iterations for both methods was $K = 20N_i$. For POR, we set $\varepsilon_l = \varepsilon_u = 0$.

The least squares (LS) and total least squares (TLS) algorithms are based on measurement data after the preprocessing as described in (10). Both methods produce their estimates analytically without any tuning parameters.

Fig. 3a shows the root mean-squared error (RMSE) for all algorithms and nodes (a total of 1000 estimates were done for each node and algorithm). As seen, the RMSE fluctuates and no algorithm is uniformly best. However, both LS and TLS show relatively bad performance; perhaps this is due to the nonlinear preprocessing needed by these algorithms, see (10). We also note that TLS is slightly better than LS. The NLS-M, NLS-DE, and POR algorithms have almost identical RMSE performance, except for nodes 12, 16, and 20, when POR is slightly worse than the others. POCS has the best RMSE for most nodes. However, POCS has problems when the unknown node is outside the convex hull of the anchor node, which is the case for nodes 1, 6, 12, 19, and 20.

The position error CDFs give more insight into the performance of the algorithms. In the following, we will discuss the CDFs for nodes 1, 12, and 19, to point out some interesting features. Coincidentally, the choice of nodes are for cases when POCS does not have the best RMSE; for this reason, the selection of CDFs are a bit unfair to the POCS method.

Fig. 3b shows the error CDFs for node 1. We note that the CDFs for the NLS approaches are quite similar, but different from the POCS and POR CDFs. However, all three methods show very similar RMSEs (see Fig. 3a). This serves as a reminder that the RMSE is not always a complete indicator of performance. We also note that for node 1, the LS and TLS methods suffer from frequent relatively large errors and that the TLS CDF is always to the left of the LS CDF.

The CDFs for node 12 in Fig. 3c show very similar performance for the NLS approaches and the POR, LS, and TLS approaches; NLS is better than POR, LS, and TLS. The POCS method has frequent and large errors, explaining its large RMSE. This relative poor POCS performance is expected since node 12 lies outside the convex hull of its anchor nodes (nodes 10, 11, and 13–17), see Figs. 1 and 2.

For node 19, we see from Fig. 3d that the NLS CDFs are more different compared to the case for node 1 or node 12. We note that the NLS-M algorithm tends to converge to worse solutions than the NLS-DE approach in about 20% of the cases. The RMSE for LS and TLS is very high (see Fig. 3a). We see from the CDFs that is due to infrequent very large errors (and not due to frequent large errors).

5. CONCLUSIONS

Several positioning algorithms have been compared in terms of the resulting mean-square error (MSE) or position error CDF. The algorithms attempt to position a single node given distance estimates to a number of nodes at known positions (anchor nodes). The distance estimates were obtained from an indoor measurement campaign employing ultrawideband devices with built-in time-of-arrival ranging capabilities.

From the experimental data presented in Sec. 4, it can be concluded that the preprocessing (10) followed by least squares (11) or total least squares (14) has higher MSE than the other algorithms for all nodes but one (node 12). The POCS method described in Sec. 3.3 and [1, 8] has low MSE for most nodes. However, POCS performs relatively bad for nodes that are outside the convex hull of the anchor nodes; a conclusion supported by earlier findings in [1, 8]. The NLS approaches (Secs. 3.1 and 3.2) performs similarly, worse than POCS, and slightly better than POR (Sec. 3.4).

Position error CDFs can lead to other rankings of the algorithms. In fact, dissimilar CDFs that produced the (essentially) same MSE were encountered. Hence, it is important to be well aware of the application requirements on the positioning error when choosing positioning algorithm.

6. ACKNOWLEDGMENTS

The authors want to acknowledge the critical contribution to the paper made by their colleagues that together with us planned and carried out the range measurements: Davide Dardari (Università di Bologna, UNIBO), Francesco Sottile (Istituto Superiore Mario Boella), Javier Arribas (Centre Tecnològic de Telecomunicacions de Catalunya), Andrea Conti (Università degli Studi di Ferrara), Marco Rosetti (UNIBO), Valerio Casadei (UNIBO), Sergio Contino (UNIBO), Denis Fusconi (UNIBO), and Nicolò Decarli (UNIBO).

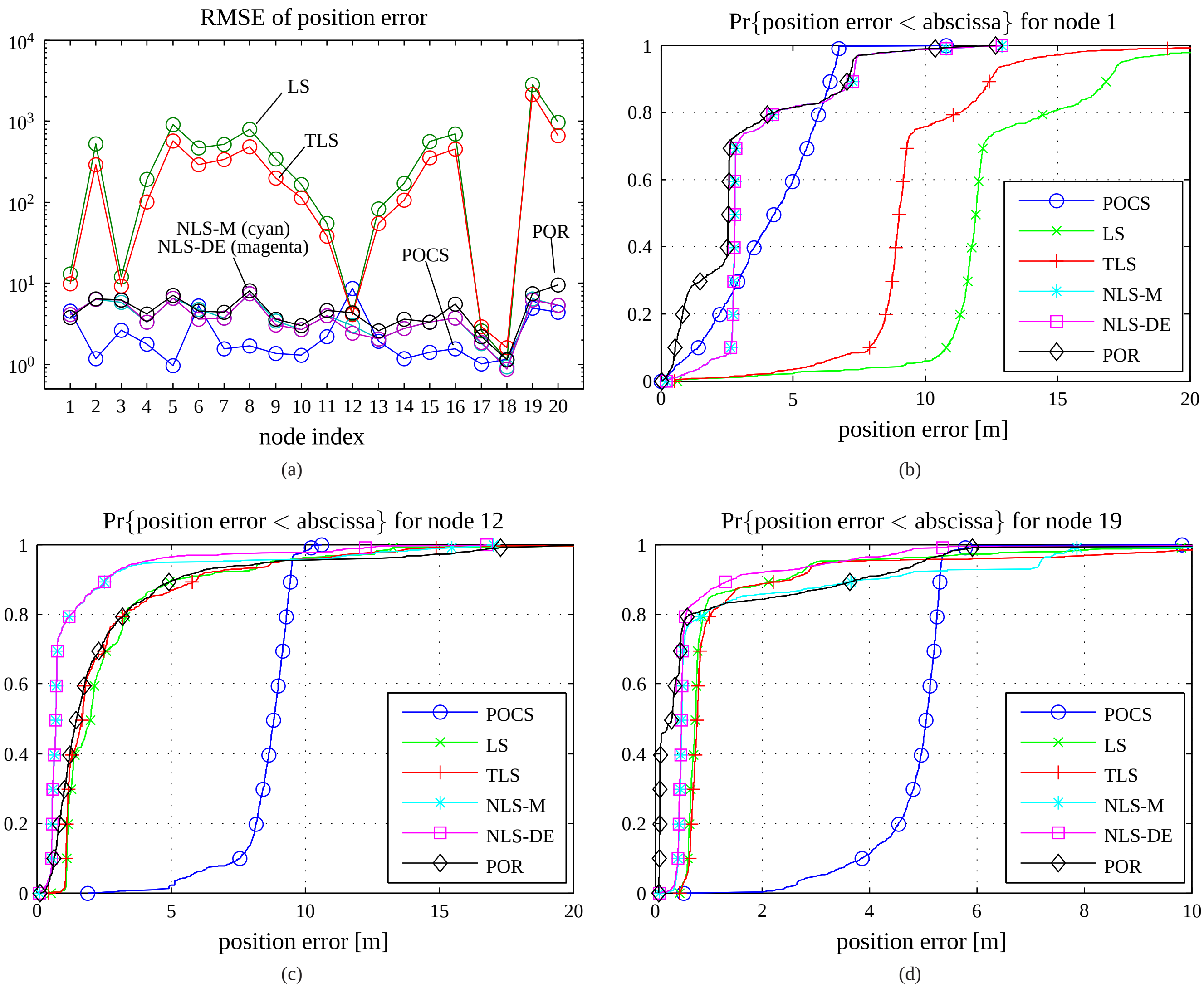

Figure 3: Numerical results; (a) shows the RMSE of different algorithms for different nodes; (b)–(d) shows the position error CDFs for node 1, 12, and 19, respectively.

REFERENCES

[1] M. Rydström, "Algorithms and models for positioning and scheduling in wireless sensor networks," Ph.D. dissertation, Chalmers University of Technology, 2008.

[2] A. Conti, D. Dardari, and M. Win, "Experimental results on cooperative UWB based positioning systems," in *IEEE Int. Conf. on Ultra-Wideband*, vol. 1, Sept. 2008, pp. 191–195.

[3] Time Domain Corporation, Jun. 2009. [Online]. Available: http://www.timedomain.com/

[4] S. Kay, *Fundemental of Statistical Processing*. Prentice-Hall, 1993.

[5] The Mathworks Inc., 2009, `lsqnonlin` documentation. [Online]. Available: http://tinyurl.com/md78bg

[6] R. Storn and K. Price, "Differential Evolution–A simple and efficient adaptive scheme for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, pp. 341–359, 1997.

[7] A. O. Hero and D. Blatt, "Sensor network source localization via projection onto convex sets (POCS)," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 3, Mar. 2005, pp. 689–692.

[8] M. Rydström, E. G. Ström, and A. Svensson, "Robust sensor network positioning based on projections onto circular and hyperbolic convex sets," in *Proc. IEEE Signals Proc. Adv. in Wireless Comm.*, Jun. 2006, pp. 1–5.

[9] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Trans. on Sig. Proc.*, vol. 56, no. 1, pp. 365–377, Jan. 2008.

[10] S. V. Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, 1991.