Béjar and Zazo *EURASIP Journal on Advances in Signal Processing* 2012, **2012**:95
http://asp.eurasipjournals.com/content/2012/1/95

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

A practical approach for outdoors distributed target localization in wireless sensor networks

Benjamín Béjar* and Santiago Zazo

Abstract

Wireless sensor networks are posed as the new communication paradigm where the use of small, low-complexity, and low-power devices is preferred over costly centralized systems. The spectra of potential applications of sensor networks is very wide, ranging from monitoring, surveillance, and localization, among others. Localization is a key application in sensor networks and the use of simple, efficient, and distributed algorithms is of paramount practical importance. Combining convex optimization tools with consensus algorithms we propose a distributed localization algorithm for scenarios where received signal strength indicator readings are used. We approach the localization problem by formulating an alternative problem that uses distance estimates locally computed at each node. The formulated problem is solved by a relaxed version using semidefinite relaxation technique. Conditions under which the relaxed problem yields to the same solution as the original problem are given and a distributed consensus-based implementation of the algorithm is proposed based on an augmented Lagrangian approach and primal-dual decomposition methods. Although suboptimal, the proposed approach is very suitable for its implementation in real sensor networks, i.e., it is scalable, robust against node failures and requires only local communication among neighboring nodes. Simulation results show that running an additional local search around the found solution can yield performance close to the maximum likelihood estimate.

1 Introduction

The deployment of a large number of scattered sensors in a certain area constitutes a very powerful tool for sensing and retrieving information from the environment (e.g., temperature, humidity, motion). The main features of wireless sensor networks (WSN) are that of a large number of low-cost nodes with limited computational and power resources. WSNs must also be scalable and robust against changes in topology (i.e., node failure or addition of new nodes), as well as energy efficient. These are the major design issues in WSNs that make the development of simple and efficient algorithms a challenging problem. These limitations also make centralized approaches not very suitable for being used in WSNs. Localization is a key task (often mandatory) in many applications [1] and therefore, distributed localization algorithms are of high practical importance.

There exist different measurement sources that can be fused in order to get an estimate of the target's position [1,2], like time-of-arrival (TOA), time-difference of arrival (TDOA), angle of arrival (AOA) or received signal strength indicator (RSSI). In this article, we focus on single antenna nodes without tight synchronization abilities, which leads us to the use of RSSI measurements for the localization task. One of the main challenges when using RSSI measurements is that the mapping between the measurement and target's position is nonlinear and hence, finding a suitable solution becomes more challenging. Some approaches to deal with nonlinearities are based on particle filtering principles [3]. In the context of WSN particle filtering approaches have also been proposed for localization and tracking using RSSI measurements [4-8]. In general, particle filtering approaches have shown very good performance when dealing with RSSI measurements but they are centralized and suffer from a high computational cost and hence, their applicability in a real scenario is questionable.

A recent approach based on convex optimization concepts has been proposed in [9,10] for the node localization problem. In [9] a semidefinite relaxation approach is used to cast the localization problem into a

* Correspondence: min@gaps.ssr.upm.es
Universidad Politécnica de Madrid, Dpto. Señales, Sistemas y Radiocomunicaciones, E.T.S.I. de Telecomunicación C-303, 28040, Madrid, Spain

semidefinite program (SDP) that can be solved efficiently via interior point methods, see [11] and references therein. The obtained position estimate through SDP is then further refined via an iterative algorithm. Although the proposed method provide near optimal results (i.e., close to the Cramer-Rao bound) they are centralized so their application to WSN may be limited. The problem of source localization using energy measurements has also been treated in [12], where a distributed algorithm based on projections onto convex sets is presented. The algorithm is shown to asymptotically approach the maximum likelihood (ML) estimate as the number of nodes increases when the target lies in the convex hull defined by the node's coordinates. In [13], an alternative approach is presented that can handle variations in the path-loss exponent. However, in both approaches no restrictions are imposed in the communication among nodes. In real applications this will cause rapid battery depletion if far away nodes are to communicate. Further, in both approaches the estimation is performed only by a subset of nodes that are selected according to their received signal to noise ratio. The main drawback is that such subset must be known to every node in the network. In a real scenario, the required signaling and routing overhead necessary for node coordination may limit their application.

In this contribution, we propose a distributed algorithm for localization in WSN's by fusing RSSI measurements. We approach the ML estimation problem by solving a simplified and more tractable problem which allows the use of convex optimization tools for its distributed solution. More precisely, we use an augmented Lagrangian approach with a primal-dual decomposition [14,15]. The developed approach offers an advantage over centralized approaches as it is scalable, robust against changes in network's topology and requires only local communication among neighboring nodes. These are key properties very desirable in the context of WSN's.

The article is organized as follows: Section 2 introduces the localization problem and the underlying propagation model. In Section 3, we present the localization approach based on RSSI readings and its distributed implementation is presented in Section 4. Simulations are provided in Section 6, while some comments and concluding remarks are given in Section 7.

Notation

Bold lower- and upper-case letters denote vectors and matrices, respectively. For vector quantities the operator $||\cdot||$ denotes Euclidean norm while for matrices it refers to the Frobenius norm. The symbol $\mathbf{0}$ denotes a matrix of appropriate dimensions whose entries are all zeros. The symbol $\mathbf{I}$ is used to denote the identity matrix of appropriate dimensions. The optimal value of a variable $\mathbf{x}$ in an optimization problem is denoted by $\mathbf{x}^*$. The symbol $\mathbb{R}^n$ is used to denote the set of real n-dimensional vectors while $\mathbb{S}_+^n$ is used to denote the set of symmetric, $n \times n$ positive semi-definite matrices.

2 Problem formulation and definitions

Consider a wireless sensor network, as the one depicted in Figure 1, consisting of M nodes randomly deployed on a certain area (in the same x-y plane). Nodes are static and able to communicate with adjacent nodes that lie within a given range for communications. Nodes are aware of their own location but not aware of the location of any other element in the network. Assume the presence of a target node that emits beacon frames that can be heard by all nodes in the network. The goal is to determine the location of the target node in the x-y plane.

For getting estimates of the target position, nodes employ RSSI measurements. The use of RSSI readings is of practical convenience when working with real hardware as they do not need tight synchronization requirements. We assume that the RSSI follows a linear relationship with the received power (we assume they are equal). Let denote r_m as the received power at node m. A common assumption, see [2] and references therein, is that the received power follows a lognormal distribution with a distance-dependent mean as

$$r_m = p_m - 10\alpha_m \log_{10}\left(\frac{d_m}{d_0}\right) + n_m, \tag{1}$$

where p_m is the received power (in dB) at reference distance d_0, α_m is the path-loss exponent, d_m is the true

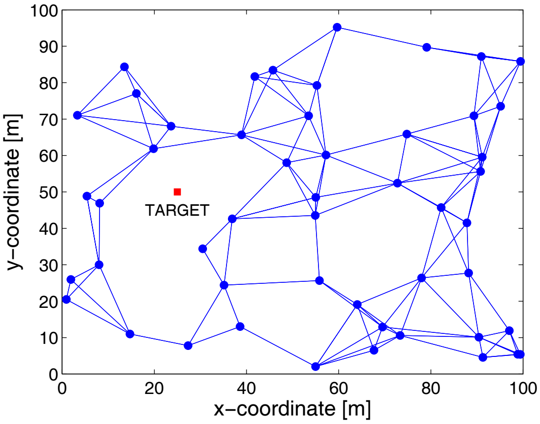

Figure 1 Wireless sensor network. Schematic representation of the network used for the simulations. Blue circles represent the nodes and the link between them indicates that those nodes can communicate. The target is represented as a red square.

distance between the target and the mth node and $n_m \sim \mathcal{N}(0, \sigma_m^2)$ is a Gaussian random variable of zero mean and variance σ_m^2. The received power r_m will be used to get an estimate of the true target position.

3 Localization strategies

In this section, we present different localization strategies using the received power (in dB) at the nodes. We first consider the (centralized) ML estimate and then we propose to use a suboptimal strategy based on local distance estimate at each node. We show that the proposed localization strategy can be implemented in a fully distributed way by only local communication among neighboring nodes.

3.1 ML estimation

Consider the presence of a centralized unit that gathers all measurements coming from the nodes. Let r = $[r_1, \ldots, r_M]^{\mathrm{T}}$ be a vector whose components are the different measurements taken by each sensor and denote $\mathbf{x} = [x_t \; y_t]^{\mathrm{T}} \in \mathbb{R}^2$ as the target's position. The true distance between the target and the mth sensor can be then expressed as

$$d_m = ||\mathbf{x} - \mathbf{c}_m||, \tag{2}$$

where $\mathbf{c}_m = [x_m \; y_m]^{\mathrm{T}} \in \mathbb{R}^2$ are the coordinates of node m with $m = 1, \ldots, M$. The vector of measurements $\mathbf{r}$ can now be written as

$$\mathbf{r} = \begin{bmatrix} p_1 - 10\alpha_1 \log_{10}\left(\frac{||\mathbf{x} - \mathbf{c}_1||}{d_0}\right) \\ \vdots \\ p_M - 10\alpha_M \log_{10}\left(\frac{||\mathbf{x} - \mathbf{c}_M||}{d_0}\right) \end{bmatrix} + \begin{bmatrix} n_1 \\ \vdots \\ n_M \end{bmatrix} = \boldsymbol{\mu}(\mathbf{x}) + \mathbf{n} \tag{3}$$

where the vector $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sum)$ is jointly Gaussian with zero mean and covariance $\boldsymbol{\Sigma}$. It is easy to see that $\mathbf{r}$ will follow a Gaussian distribution with mean $\boldsymbol{\mu}(\mathbf{x})$ and covariance $\boldsymbol{\Sigma}$, that is

$$p(\mathbf{r}; \mathbf{x}) = \frac{1}{(2\pi)^{M/2}\sqrt{\det \sum}} \exp\left[-1/2(\mathbf{r} - \boldsymbol{\mu}(\mathbf{x}))^{\mathrm{T}} {\sum}^{-1} (\mathbf{r} - \boldsymbol{\mu}(\mathbf{x}))\right] \tag{4}$$

where $p(\mathbf{r} ; \mathbf{x})$ is the probability density function of $\mathbf{r}$ with parameter $\mathbf{x}$. The ML estimate of the target position is then

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\max_{\mathbf{x}} p(\mathbf{r}; \mathbf{x}), \tag{5}$$

which is equivalent to maximizing the log of $p(\mathbf{r} ; \mathbf{x})$. Neglecting all terms that do not depend on $\mathbf{x}$ it is easy to see that

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x}} \boldsymbol{\mu}(\mathbf{x})^{\mathrm{T}} {\sum}^{-1} \boldsymbol{\mu}(x) - 2\mathbf{r}^{\mathrm{T}} {\sum}^{-1} \boldsymbol{\mu}(\mathbf{x}). \tag{6}$$

The objective to be minimized in (6) is not convex and therefore, finding the global optimum is not an easy task. In Figure 2 (left), we have an illustration of how the objective in (6) looks like for a network of 20 nodes over a normalized square area. It is clear that the function is not convex and that several local minima and saddle point may exist. Instead of dealing directly with the ML estimate we propose to use a suboptimal approach that offers a reasonable good performance and that allows for its distributed computation.

3.2 Practical approach

In this section, we propose to estimate the target's position based on local distance estimates computed at each node. The use of local distance estimates allows the derivation of simple and distributed estimators of the target's position. We have that, for the propagation model (1), the ML estimate of the distance between the mth node and the target is given by

$$\hat{d}_m = d_0 10 \left(\frac{p_m - r_m}{10\alpha_m}\right). \tag{7}$$

Taking the square at both sides of (2) and further developing, it is easy to see that the following equation must be satisfied

$$\begin{aligned} d_1^2 &= \mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{c}_1^{\mathrm{T}}\mathbf{x} + ||\mathbf{c}_1||^2 \\ &\vdots \\ d_M^2 &= \mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{c}_M^{\mathrm{T}}\mathbf{x} + ||\mathbf{c}_M||^2 \end{aligned} \tag{8}$$

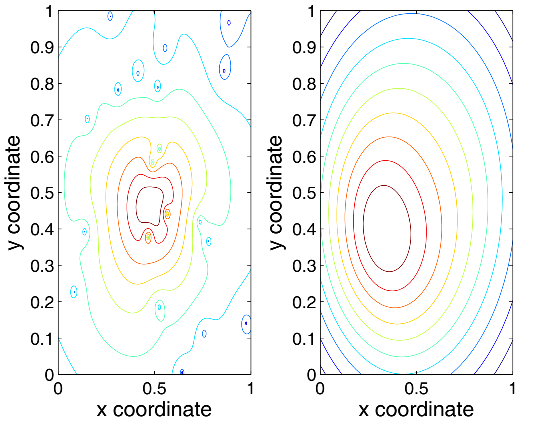

Figure 2 Contour plot of the objective function. This figure represents, on the left, the contour plot of the log of the original ML cost function in (6) while on the right side we have the modified (suboptimal) cost function in (11).

Rearranging terms we can express (8) in a more compact form as

$$\begin{bmatrix} d_1^2 - ||\mathbf{c}_1||^2 \\ \vdots \\ d_M^2 - ||\mathbf{c}_M||^2 \end{bmatrix} = (\mathbf{x}^{\mathrm{T}}\mathbf{x}) \cdot \mathbf{1} - 2 \underbrace{\begin{bmatrix} \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_M^{\mathrm{T}} \end{bmatrix}}_{\mathbf{C}} \mathbf{x}, \tag{9}$$

where $\mathbf{1}$ is a $M \times 1$ vector of all ones. However, we do not have the actual distances to the target but a noisy version of them as per (7). Define the vector $\mathbf{b} = \left[||\mathbf{c}_1||^2 - \hat{d}_1^2, \ldots, ||\mathbf{c}_M||^2 - \hat{d}_M^2\right]^{\mathrm{T}}$ and the vector-valued cost function $\mathbf{f}(\mathbf{x})$: $\mathbb{R}^2 \mapsto \mathbb{R}^M$ as

$$\mathbf{f}(\mathbf{x}) = (\mathbf{x}^{\mathrm{T}}\mathbf{x}) \cdot \mathbf{1} - 2\mathbf{C}\mathbf{x} + \mathbf{b}. \tag{10}$$

We can then get an estimate of the target's position by minimizing the norm of (10). In order to incorporate robustness and make the localization task more applicable to realistic scenarios we propose to use a weighted version of the cost function (10). In a WSN it may happen that some of the nodes exhibit a bias in their measurements due to the presence of obstacles. Additionally, nodes do not have precise information about their own locations instead, some errors may be present. The incorporation of weights will mitigate the effects of biased nodes and uncertainties in nodes' positions. So we compute an estimate $\hat{\mathbf{x}}$ of the true target's position $\mathbf{x}$ as the solution to the following non-linear (weighted) least-squares problem

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \left\| \mathbf{D}\mathbf{f}(\mathbf{x}) \right\|, \tag{11}$$

where $\mathbf{D} = \text{diag}\ (\gamma_1, \ldots, \gamma_M)$ is a diagonal weighting matrix with $\gamma_m \geq 0$ for all $m = 1, \ldots, M$. A proper choice for the weights would be inversely proportional to the variance of the measurements. As we are assuming the log-normal model for the measurements it is well known that the variance of the ML estimate (7) is proportional to the square of the true distance to be estimated [2,16]. With this consideration in mind we may choose to weight our measurements inversely proportional to the measured distance, that is $\gamma_m = 1/\hat{d}_m$.

This problem has been studied in [17,18] where a distributed version of the Gauss-Newton method can be used for its solution. In this study, we present a more flexible approach that uses concepts from convex optimization theory. The proposed approach has better convergence properties and also makes it straightforward to include additional constraints to the problem that may prevent it from instabilities. In order to proceed let's write problem (11) as the following equivalent problem

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \textstyle\sum_{m=1}^{M} \gamma_m (\mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)^2 \tag{12}$$

Note that, although (11) and (12) are equivalent problems (i.e., with the same solution), they are different because in the latter case we are minimizing the squared norm of $\mathbf{Df}(\mathbf{x})$. The minimization of the squared norm is motivated by the fact that it allows a simple distributed implementation as it can be guessed from the structure of (12). The use of the objective in (12) is well motivated by the fact that we get a smoother surface at the cost of introducing some bias with respect to the ML solution (see Figure 2 (right)). However, if the bias is small, we may still get to the ML estimate by performing a local search around the solution of (12). However, in order to use convex optimization methods we need problem (12) to be convex. Unfortunately, the objective function is not convex because we are adding the squares of quadratic convex but not necessarily positive functions [11]. It would be interesting to exploit some hidden convexity of the problem so that convex optimization methods can be applied.

A possible approach to make the problem convex is to use semidefinite relaxation technique. Let $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathbf{T}}$ and note that $\text{Tr}\ (\mathbf{X}) = ||\mathbf{x}||^2$, where $\text{Tr}(\cdot)$ is the trace operator. We can rewrite the problem as

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{x}}{\text{minimize}} \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)^2 \\ &\text{subject to } \mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{T}} \end{aligned} \tag{13}$$

We now have that the objective is convex as it is the composition of an affine function of $\mathbf{X}$ and $\mathbf{x}$ with a convex function [11]. However, the above problem is still non-convex due to the non-linear constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{T}}$. We can then relax the equality constraint by replacing it with a semidefinite constraint. As a result we end up with the following (convex) SDP

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{x}}{\text{minimize}} \quad \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)^2 \\ &\text{subject to } \mathbf{X} - \mathbf{x}\mathbf{x}^{\mathrm{T}} \succcurlyeq 0 \\ &\qquad\qquad\quad \mathbf{X} \in \mathbb{S}_+^2 \end{aligned} \tag{14}$$

where $\mathbb{S}_+^2$ is the set of 2×2 symmetric positive semidefinite matrices. As we are allowing for a larger feasible set, the optimal value of problem (14) would provide a lower bound on the optimal value of the original problem (12). However, if the optimal solution $\mathbf{X}^*$ of (14) is of rank-one, we have that the semidefinite relaxation approach is not a relaxation at all and the found solution $\mathbf{x}^*$ of (14) is also optimal for (12).

It would be interesting to give conditions under which (14) provides a rank-one solution so that the obtained solution is optimal for the original problem, too. To that end let define the matrix $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} 2\sum_m \gamma_m x_m & 2\sum_m \gamma_m y_m & -\sum_m \gamma_m \\ 2\sum_m \gamma_m x_m^2 & 2\sum_m \gamma_m y_m x_m & -\sum_m \gamma_m x_m \\ 2\sum_m \gamma_m y_m x_m & 2\sum_m \gamma_m y_m^2 & -\sum_m \gamma_m y_m \end{bmatrix} \quad (15)$$

and let the vector $\boldsymbol{\delta}$ be given by

$$\boldsymbol{\delta} = \begin{bmatrix} \sum_m \gamma_m b_m \\ \sum_m \gamma_m b_m x_m \\ \sum_m \gamma_m b_m y_m \end{bmatrix} \quad (16)$$

We then define the following feasibility problem

$$\begin{array}{ll} \text{find} & \{\mathbf{z}, t\} \\ \text{subject to} & \|\mathbf{z}\|^2 \le t \\ & \mathbf{A}\begin{bmatrix} \mathbf{z} \\ t \end{bmatrix} = \boldsymbol{\delta} \end{array} \quad (17)$$

with variables $\mathbf{z} \in \mathbb{R}^2$ and $t \in \mathbb{R}_+$. The above problem is convex since it belongs to the class of second-order cone program (SOCP) [11]. Based on the feasibility problem (17) we can state the following result:

Proposition 1. *Assume problem (12) has at least one strictly feasible point. If problem (17) is not feasible, then the optimal solution* $\mathbf{x}^*$ *of the semidefinite relaxed problem (14) is also optimal for the original problem (12).*

Proof. See the Appendix. □

Corollary 2. *If matrix* $\mathbf{A}$ *is singular then, the solution* $\mathbf{X}^*$ *of (14) is of rank one with* $\mathbf{X}^* = \mathbf{x}^*\mathbf{x}^{*\mathrm{T}}$ *and* $\mathbf{x}^*$ *is also optimal for (12).*

Proof. It follows directly from Proposition 1. If $\mathbf{A}$ is singular then, problem (17) is infeasible (because matrix $\mathbf{A}$ is not invertible) so that the relaxed problem is not a relaxation at all. □

It is worth to mention that the feasibility problem (17) can be easily checked without the requirement of an optimization solver. If matrix $\mathbf{A}$ is singular then, the problem is infeasible and we are done. If however, matrix $\mathbf{A}$ is full-rank, we compute $\mathbf{A}^{-1}\boldsymbol{\delta}$ (which is unique) and check whether it satisfies the second order cone (SOC) constraint $||\mathbf{z}||^2 \le t$. If the constraint is not met then we conclude that the problem is infeasible.

4 Distributed algorithm

One of the main advantages that offers the considered approach is that it allows for a distributed implementation. We assume that nodes communicate with their one-hop neighbors as dictated by the communication graph $G(V, E)$, where V is the set of vertices and E is the set of edges of the graph. By only local communication exchange, nodes can agree to compute some desired (global) quantity using consensus algorithms [19].

Let's proceed by reformulating problem (14) into a SDP with a single matrix variable $\mathbf{Z} \in \mathbb{S}_+^3$. For that purpose, let us formulate the following problem

$$\begin{array}{ll} \underset{\mathbf{Z}}{\text{minimize}} & \sum_{m=1}^{M} \gamma_m \left(\mathrm{Tr}(\mathbf{M}_m\mathbf{Z}) + b_m - 1\right)^2 \\ \text{subject to} & \mathbf{Z}(3,3) = 1 \\ & \mathbf{Z} \in \mathbb{S}_+^3 \end{array} \quad (18)$$

where $\mathbf{M}_m = \mathbf{I} - 2\begin{bmatrix} 0 & 0 \\ \mathbf{c}_m^{\mathrm{T}} & 0 \end{bmatrix}$ and $\mathbf{I}$ is the identity matrix. With the above problem definition we have the following equivalence:

Lemma 1. *The two problems (14) and (18) are equivalent. Further, if we denote* $\mathbf{Z}^*$ *as the optimal solution to (18), then the optimal solution* $\mathbf{x}^*$ *of (14) is given by* $\mathbf{x}^* = [\mathbf{Z}^*(3,1), Z^*(3,2)]^{\mathrm{T}}$.

Proof. See the Appendix. □

Now that we have established the equivalence between problems (14) and (18) through Lemma 1 we show how to solve it in a distributed way. For that purpose we use the optimization framework for consensus-networked systems as proposed in [15] that uses an augmented Lagrangian approach [14]. The framework in [15] generalizes the previous work of [20] as it can also handle convex but not necessarily strictly convex objective functions. The augmented Lagrangian method adds a quadratic penalty term to the objective function that is zero at the optimal solution. The resulting problem is then equivalent to the original problem as both of them end up with the same solution. Augmented Lagrangian methods are also attractive because they offer better convergence properties than standard primal-dual decomposition methods. A detailed treatment about augmented Lagrangian methods and their properties can be found in [14].

In order to derive a distributed solution consider first the introduction of M new variables and a global consensus constraint into the problem as

$$\begin{array}{ll} \underset{\mathbf{Z},\{\mathbf{Z}_m\}}{\text{minimize}} & \sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{M}_m\mathbf{Z}_m) + b_m - 1)^2 \\ \text{subject to} & \mathbf{Z}(3,3) = 1 \\ & \mathbf{Z}_m = \mathbf{Z} \\ & \mathbf{Z} \in \mathbb{S}_+^3 \end{array} \quad (19)$$

The problem is now separable in the objective function (as it is the sum of M terms, each one dependent of one node) but we still have the coupling "consensus" constraint $\mathbf{Z}_m = \mathbf{Z}$. However, we do not need to impose that all nodes agree on the same quantity instead, we can only force nodes to agree with their one-hop neighbors. Let $\mathcal{N}_m$ be the set of neighbors of node m, we can then reformulate the problem as

$$\begin{aligned} \underset{\{\mathbf{Z}_m\}}{\text{minimize}} \quad & \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{M}_m \mathbf{Z}_m) + b_m - 1)^2 \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m = \mathbf{Z}_j, j \in \mathcal{N}_m \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{20}$$

The two problems (19) and (20) are equivalent provided that the underlying graph is strongly connected [20]. We can now use the developed framework in [15] to derive an augmented Lagrangian method for the distributed solution of (20). Consider then, the introduction of the additional variables $\mathbf{W}_{m,j} \in \mathbb{S}_+^3$ and formulate the equivalent problem

$$\begin{aligned} \underset{\{\mathbf{Z}_m\},\{\mathbf{W}_{mj}\}}{\text{minimize}} \quad & \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{M}_m \mathbf{Z}_m) + b_m - 1)^2 \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m = \mathbf{W}_{m,j},\ j \in \mathcal{N}_m \\ & \mathbf{Z}_j = \mathbf{W}_{m,j},\ j \in \mathcal{N}_m \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{21}$$

The penalized problem [14] can be written as

$$\begin{aligned} \underset{\{\mathbf{Z}_m\},\{\mathbf{W}_{mj}\}}{\text{minimize}} \quad & \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{M}_m \mathbf{Z}_m) + b_m - 1)^2 \\ & + \tfrac{c}{2} \textstyle\sum_{m=1}^{M} \sum_{j \in \mathcal{N}_m} (||\mathbf{Z}_m - \mathbf{W}_{m,j}||^2 + ||\mathbf{Z}_j - \mathbf{W}_{m,j}||^2) \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m = \mathbf{W}_{m,j},\ j \in \mathcal{N}_m \\ & \mathbf{Z}_m = \mathbf{W}_{m,j},\ j \in \mathcal{N}_m \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{22}$$

where c >0 is a constant that controls the penalization of the disagreement among neighbors. In general, c could be a sequence as long as it is non-decreasing. The choice of c has a direct impact on the rate of convergence of the distributed algorithm [14]. There is no general rule to choose c and its value will vary depending on the problem at hand. It becomes clear from the formulation of problem (22) that the penalty term is zero at the optimum and, therefore the optimal solution to (22) is also optimal for (20). We can now find a solution of (22) by solving its dual problem. By relaxing the consensus constraint we form the partial augmented Lagrangian L_c as

$$\begin{aligned} L_c(\{\mathbf{\Gamma}_{m,j}\},\ \{\mathbf{\Phi}_{m,j}\},\ \{\mathbf{Z}_m\},\ \{\mathbf{W}_{m,j}\}) = & \textstyle\sum_{m=1}^{M} \gamma_m (\text{Tr}(\mathbf{M}_m \mathbf{Z}_m) + b_m - 1)^2 \\ & + \textstyle\sum_{m=1}^{M} \sum_{j \in \mathcal{N}_m} \text{Tr}(\mathbf{\Gamma}_{m,j}(\mathbf{Z}_m - \mathbf{W}_{m,j})) \\ & + \textstyle\sum_{m=1}^{M} \sum_{j \in \mathcal{N}_m} \text{Tr}(\mathbf{\Phi}_{m,j}(\mathbf{Z}_j - \mathbf{W}_{m,j})) \\ & + \tfrac{c}{2} \textstyle\sum_{m=1}^{M} \sum_{j \in \mathcal{N}_m} (||\mathbf{Z}_m - \mathbf{W}_{m,j}||^2 + ||\mathbf{Z}_j - \mathbf{W}_{m,j}||^2) \end{aligned} \tag{23}$$

where $\mathbf{\Gamma}_{m,\,j}$ and $\mathbf{\Phi}_{m,\,j}$ are the Lagrange multipliers. We then have that the dual problem is given by

$$\begin{aligned} \underset{\{\mathbf{\Gamma}_{m,j}\},\,\{\mathbf{\Phi}_{m,j}\}}{\text{maximize}} \ \underset{\{\mathbf{Z}_m\},\{\mathbf{W}_{m,j}\}}{\inf} \ & L_c(\{\mathbf{\Gamma}_{m,j}\},\ \{\mathbf{\Phi}_{m,j}\},\ \{\mathbf{Z}_m\},\ \{\mathbf{W}_{m,j}\}) \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{24}$$

The problem is now separable and strictly convex which allows its solution by alternating the minimization over $\mathbf{Z}_m$ and $\mathbf{W}_{m,\,j}$ as

$$\begin{aligned} \mathbf{Z}_m^{(k+1)} = \underset{\mathbf{Z}m}{\text{minimize}} \quad & L_c\left(\{\mathbf{\Gamma}_{m,j}^{(k)}\}, \{\mathbf{\Phi}_{m,j}^{(k)}\}, \{\mathbf{Z}_m\}, \{\mathbf{W}_{m,j}^{(k)}\}\right) \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{25}$$

$$\mathbf{W}_{m,j}^{(k+1)} = \underset{\mathbf{W}_{m,j}}{\text{minimize}} \quad L_c\left(\{\mathbf{\Gamma}_{m,j}^{(k)}\}, \{\mathbf{\Phi}_{m,j}^{(k)}\}, \{\mathbf{Z}_m^{(k+1)}\}, \{\mathbf{W}_{m,j}\}\right) \tag{26}$$

and then performing an update of the Lagrange multipliers using a subgradient step

$$\mathbf{\Gamma}_{m,j}^{(k+1)} = \mathbf{\Gamma}_{m,j}^{(k)} + c(\mathbf{Z}_m - \mathbf{W}_{m,j}) \tag{27}$$

$$\mathbf{\Phi}_{m,j}^{(k+1)} = \mathbf{\Phi}_{m,j}^{(k)} + c(\mathbf{Z}_j - \mathbf{W}_{m,j}) \tag{28}$$

where the superscript $^{(k)}$ denotes the kth iteration. Following the same steps as in [15], it can be shown that

$$\mathbf{\Gamma}_{m,j}^{(k+1)} = -\mathbf{\Phi}_{m,j}^{(k+1)} \tag{29}$$

$$\mathbf{W}_{m,j}^{(k+1)} = \tfrac{1}{2}(\mathbf{Z}_m^{(k+1)} + \mathbf{Z}_j^{(k+1)}) \tag{30}$$

Let $\mathbf{\Delta}_{m,\,j} = \mathbf{\Gamma}_{m,\,j} = -\mathbf{\Phi}_{m,\,j}$ and define $\mathbf{\Psi}_m = \mathbf{\Delta}_{m,\,j} - \mathbf{\Delta}_{j,\,m}$. Based on the above results it can be easily shown that the solution to problem (24) is obtained in a distributed way by alternating between the following two updates

$$\begin{aligned} \mathbf{Z}_m^{(k+1)} = \underset{\mathbf{Z}_m}{\text{minimize}} \ & \gamma_m (\text{Tr}(\mathbf{M}_m \mathbf{Z}_m) + b_m - 1)^2 \\ & + \text{Tr}\ (\mathbf{\Psi}_m \mathbf{Z}_m) + c \textstyle\sum_{j \in \mathcal{N}_m} ||\mathbf{Z}_m - \tfrac{1}{2}(\mathbf{Z}_m^{(k)} - \mathbf{Z}_j^{(k)})||^2 \\ \text{subject to}\; & \mathbf{Z}_m(3,3) = 1 \\ & \mathbf{Z}_m \in \mathbb{S}_+^3 \end{aligned} \tag{31}$$

and

$$\mathbf{\Psi}_m^{(k+1)} = \mathbf{\Psi}_m^{(k)} + c \sum_{j \in \mathcal{N}_m} \left(\mathbf{Z}_m^{(k+1)} - \mathbf{Z}_j^{(k+1)}\right) \tag{32}$$

with $\mathbf{\Psi}_m^{(0)} = \mathbf{0}$ and $m = 1, \ldots, M$.

The network will then operate as follows: At the beginning of the kth iteration, each node locally solves (31). Then, nodes broadcast the computed estimates $\mathbf{Z}_m^{(k+1)}$ to their neighbors. With the local estimates of the corresponding neighbors at hand, each node updates its multipliers as in (32). The process is repeated until all nodes converge to the same solution which, in turn would be the same as in the centralized case.

5 Approaching the ML estimate

So far, we have shown how to solve (11) by formulating the relaxed problem (14). We have also provided conditions under which the solutions to (11) and (14) coincide. Further, the solution can be computed in a distributed fashion using convex optimization tools. However, the performance of the followed approach in (11) is below that of the ML estimate (6). In order to come closer to the ML solution we could perform an additional local search that improves the obtained estimate through the solution of (14). The idea is to run a distributed optimization routine, taking the solution of (14) as the starting point, to solve for (6). If the previously computed estimate by solving (14) is close to the ML estimate we may converge to it by optimizing in the neighborhood of the solution of (14), otherwise we will converge to a local optima but still improving performance.

Observe that the ML estimate (6) can be cast into a non-linear least-squares problem of the form of (11). Assuming that Σ is positive definite, we can write the ML estimation problem (6) as the following unconstrained optimization problem

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x}}{\text{minimize}} \quad ||\mathbf{f}_{\mathrm{ML}}(\mathbf{x})||^2, \tag{33}$$

where $\mathbf{f}_{\mathrm{ML}}(\mathbf{x}) = \mathbf{S}\ (\mathbf{r} - \boldsymbol{\mu}(\mathbf{x}))$ and $\mathbf{S}$ is the Cholesky factorization of the inverse covariance matrix, i.e., $\mathbf{S}^{\mathrm{T}}\mathbf{S} = \Sigma^{-1}$. A local minimum of the above non-linear least-squares problem (33) can be found using an iterative descent algorithm like the Gauss-Newton method [11,21]. The standard (centralized) Gauss-Newton procedure is given in Algorithm 1, where $\mathbf{h}_{\mathrm{gn}}$ represents the descent direction (i.e., direction that reduces the value of the cost function) and $\mathbf{J}^{(k)} = \mathbf{J}(\mathbf{x}^{(k)})$ with $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{M\times 2}$ representing the Jacobian matrix of $\mathbf{f}_{\mathrm{ML}}(\mathbf{x}) = [f_1^{\mathrm{ML}}(\mathbf{x}),\ \ldots,\ f_M^{\mathrm{ML}}(\mathbf{x})]^{\mathrm{T}}$ whose entries are given by

$$[\mathbf{J}(\mathbf{x})]_{ij} = \frac{\partial f_i^{\mathrm{ML}}}{\partial x_j}(\mathbf{x}), \quad i = 1, \ldots, M, j = 1, 2. \tag{34}$$

Algorithm 1 Gauss-Newton method

```
$\mathbf{x}^{(0)} \Re \mathbf{x}_0$ $k = 0$ {Initialization}
while ! found &$k < k_{\max}$ do
    $\mathbf{h}_{\mathrm{gn}} \Re - (\mathbf{J}^{(k)\mathrm{T}} \mathbf{J}^{(k)})^{-1} \mathbf{J}^{(k)\mathrm{T}} \mathbf{f}_{\mathrm{ML}} (\mathbf{x}^{(k)})$ {Descent direction}
    if $\nexists \mathbf{h}_{\mathrm{gn}}$ then
        found = true
    end if
    $\mathbf{x}^{(k+1)} \Re \mathbf{x}^{(k)} + \mathbf{h}_{\mathrm{gn}}$ {Update}
    $k \Re k + 1$
end while
```

If the covariance matrix Σ has no special structure, then the problem (33) requires a central entity that gathers all the information coming from the nodes in order to solve it. However, it is reasonable to assume independence of the noise processes among the nodes so that Σ has a diagonal structure, say $\sum = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_M^2)$. In that case, matrix $\mathbf{S} = \mathrm{diag}(1/\sigma_1, \ldots, 1/\sigma_M)$ is also diagonal and we can exploit the problem structure in order to find a distributed implementation of the Gauss-Newton procedure given in Algorithm 1. Note that for finding a distributed implementation of Algorithm 1, it suffices to find a way to compute the descent search direction $\mathbf{h}_{\mathrm{gn}}$ is a distributed fashion. For such purpose, let first note that $\mathbf{J}(\mathbf{x})$ has a block-wise structure given by

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{-10\alpha_1}{\sigma_1 ||\mathbf{x}-\mathbf{c}_1||^2}(\mathbf{x} - \mathbf{c}_1)^{\mathrm{T}} \\ \vdots \\ \frac{-10\alpha_M}{\sigma_M ||\mathbf{x}-\mathbf{c}_M||^2}(\mathbf{x} - \mathbf{c}_M)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1(\mathbf{x}) \\ \vdots \\ \mathbf{J}_M(\mathbf{x}) \end{bmatrix}. \tag{35}$$

Based on the block-wise structure of matrix $\mathbf{J}(\mathbf{x})$ it is easy to note that

$$\mathbf{J}(\mathbf{x})^{\mathrm{T}}\mathbf{J}(\mathbf{x}) = \sum_m \mathbf{J}_m(\mathbf{x})^{\mathrm{T}}\mathbf{J}_m(\mathbf{x}) \tag{36}$$

$$\mathbf{J}(\mathbf{x})^{\mathrm{T}}\mathbf{f}_{\mathrm{ML}}(\mathbf{x}) = \sum_m \mathbf{J}(\mathbf{x})^{\mathrm{T}} f_m^{\mathrm{ML}}(\mathbf{x}) \tag{37}$$

and therefore, the above quantities can be computed in a distributed fashion by means of average consensus [19]. Once we have computed the products (36) and (37), it is straightforward to compute the descent search direction $\mathbf{h}_{\mathrm{gn}}$.

Based on these observations we propose here a fully distributed algorithm, shown as Algorithm 2, which asymptotically approaches the same result as in the centralized case using only local information and the exchange of low-volume intermediate results within each node's 1-hop neighborhood. We immediately note that the Steps 3-6 and 11-12 can all be performed locally by each node. The only communication occurs in the Steps 8 and 9 via standard average consensus Algorithms [19]. Seeing how $\boldsymbol{\Delta}_n^{(k)} \in \mathbf{R}^{2\times 2}$ and is symmetric, and $\gamma_n^{(k)} \in \mathbb{R}^2$, we conclude that each consensus round requires a broadcast of only five real values.

Algorithm 2 Distributed Gauss-Newton localization

```
1: $\hat{\mathbf{x}}^{(0)} \Re$ same initial value $\forall m \in M$
2: for $k = 0$ to $K - 1$ do
3:     $\mathbf{J}_m^{(k)} \leftarrow \frac{-10\alpha_m}{\sigma_m \left\| \hat{\mathbf{x}}^{(k)} - \mathbf{c}_n \right\|^2} (\hat{\mathbf{x}}^{(k)} - \mathbf{c}_n)^{\mathrm{T}}$
```

```
$f_m^{\mathrm{ML}}(\hat{\mathbf{x}}^{(k)}) \leftarrow \sigma_m^{-1}(r_m - p_m + 5\alpha_m \log_{10}\left\|\hat{\mathbf{x}}^{(k)} - \mathbf{c}_n\right\|^2 + 10\log_{10} d_0)$
$\boldsymbol{\Delta}_m^{(k)} \leftarrow \mathbf{J}_m^{(k)\mathrm{T}}\mathbf{J}_m^{(k)}$
$\gamma_m^{(k)} \leftarrow \mathbf{J}_m^{(k)\mathrm{T}} f_m^{\mathrm{ML}}(\hat{\mathbf{x}}^{(k)})$
begin consensus
  $\boldsymbol{\Delta}_*^{(k)} \leftarrow \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{\Delta}_m^{(k)} = \frac{1}{M}\mathbf{J}^{(k)\mathrm{T}}\mathbf{J}^{(k)}$
  $\gamma_*^{(k)} \leftarrow \frac{1}{M}\sum_{m=1}^{M}\gamma_m^{(k)} = \frac{1}{M}\mathbf{J}^{(k)\mathrm{T}}\mathbf{f}_{\mathrm{ML}}(\hat{\mathbf{x}}^{(k)})$
end consensus
$\mathbf{h}^{(k)} \leftarrow \boldsymbol{\Delta}_*^{(k)^{-1}}\gamma_*^{(k)} = (\mathbf{J}^{(k)\mathrm{T}}\mathbf{J}^{(k)})^{-1}\mathbf{J}^{(k)\mathrm{T}}\mathbf{f}_{\mathrm{ML}}(\hat{\mathbf{x}}^{(k)})$
$\hat{\mathbf{x}}^{(k+1)} \leftarrow \hat{\mathbf{x}}^{(k)} + \mathbf{h}^{(k)}$
end for
```

6 Numerical simulations

In this section, we provide several numerical examples in order to evaluate the performance of the proposed approach. For the simulations we consider a network of randomly deployed nodes over an area of 100 × 100 squared meters. We have used the same propagation model for all the nodes with reference power p_m = -40 dB at reference distance d_0 = 1 m and path-loss exponent $\alpha_m = 2$ for $m = 1, \ldots, M$. We further assume that the noise processes are independent and identically distributed with $n_m \sim \mathcal{N}(0, \sigma_{\mathrm{dB}}^2)$ for all m.

In Figure 3, we have simulated the distributed localization task using a network of 50 nodes (see Figure 1) with a randomly located target. We have performed distributed estimation of the target's position by the alternating between (31) and (32) with penalty parameter c = 0.05. We have plotted in Figure 3 the error between each node's local estimate and the centralized solution of the problem. As it can be appreciated, the distributed algorithm converges to the optimal centralized solution as the number of iterations increases.

We have also simulated the localization task over the same network of Figure 1. For the propagation model the measurement noise variance has been set to $\sigma_{\mathrm{dB}}^2 = 9$. We have evaluated the performance of the proposed approach with and without weights (labeled SDP and wSDP, respectively) over 1000 random target locations (test points). We compare our approach with Multilateration localization approach [2]. As shown in [2], if one node is chosen as a reference, then the localization problem can be linearized. Therefore, it allows its distributed implementation by means of consensus since it can be cast into a least-squares problem where each node locally contributes to the global cost function. Note, however that nodes must first agree on a common reference. For the simulations, we have chosen the node with the closest distance estimate to be the reference. We also provide the performance of the ML estimate for comparison purposes. The empirical cumulative distribution function (CDF) of the localization error is represented in Figure 4. As it can be observed in Figure 4, the performance of the proposed scheme outperforms that of Multilateration. Further, we observe that the use of weights improves the localization accuracy of the algorithm so that we come closer to the ML estimate.

The performance of the algorithm has also been tested for different values of the measurement noise variance. The results are displayed in Figure 5 where the average error over 1000 locations is depicted as a function of the measurement noise standard deviation. Again, we can appreciate the performance improvement of the proposed approach compared to multilateration as can be seen in Figure 5. We have also displayed the results

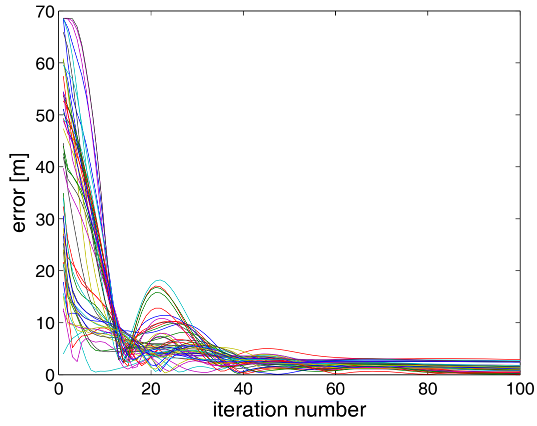

Figure 3 Error versus iteration. Norm of the difference between the local (distributed) and the centralized estimate as a function of the iteration number.

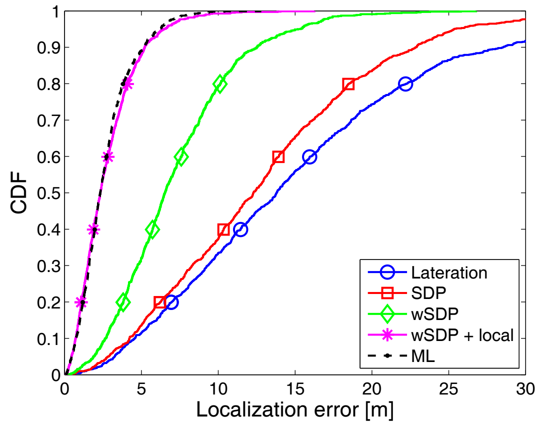

Figure 4 Empirical CDF of the localization error. This figure represents the empirical CDF of the localization error for the considered methods based on 1000 realizations.

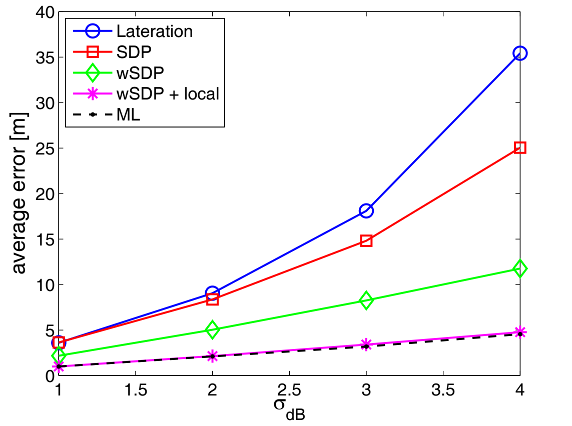

Figure 5 Error versus noise. Average localization error as a function of the measurement noise standard deviation for the considered methods. The average is taken over 1000 different realizations.

when combining the proposed distributed localization approach with a local search (wSDP+local). The local search is performed in a distributed fashion following the steps in Algorithm 2. As it can be observed the results of such combination provide close to ML performance. This implies that our method is capable of providing good estimates that could be used to run a local solver in order to come close to the ML estimate. Although not guaranteed to converge to the ML solution, the local search can only provide better estimates (in the ML sense).

7 Conclusions

We have presented a distributed localization approach over sensor networks using consensus and convex optimization. An alternative problem to the ML position estimation problem has been proposed based on local ML distance estimates at each node. In order to circumvent the non-convexity of the problem, semidefinite relaxation technique has been employed and conditions that guarantee zero gap between the relaxed and the original problem have been given. A distributed algorithm based on an augmented Lagrangian approach using primal-dual decompositions have been proposed and it has been shown to converge to the centralized solution. The approach is suitable for its real implementation in WSN as it is scalable, robust against changes in topology and energy efficient by the use of only local broadcast-type communication among nodes. Another interesting property of the proposed algorithm is that it allows the introduction of additional convex constraints to the localization problem in a straightforward manner.

The proposed algorithm is intended to be usable in real networks and its suitability in terms of accuracy would be determined by the application at hand. However, if higher accuracy is required, we could run an additional optimization step around the found solution. We have verified by means of simulations that the combination of our suboptimal method with a local search provides a localization error close to the ML estimate.

It is worth to mention that the proposed approach has a direct application to distributed tracking in WSN's as well. The tracking procedure would be based on the jointly estimated target's position. As all nodes share the same estimate, they could use that estimate to locally run a tracking filter in order to follow the movement of the target.

Appendix 1: Proof of Proposition 1

To show the validity of Proposition 1, consider first the Lagrangian of (14) which is given by

$$L(\mathbf{X},\ \mathbf{x}) = \sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)^2 - \mathrm{Tr}(\boldsymbol{\Psi}(\mathbf{X} - \mathbf{x}\mathbf{x}^{\mathrm{T}})) \quad (38)$$

with $\boldsymbol{\Psi} \succcurlyeq \mathbf{0}$ being the Lagrange multipliers. Since problem (14) is convex and there exists, by assumption, at least a strictly feasible point, Slater's constraint qualifications are satisfied and therefore, strong duality holds. Moreover, from duality theory we have that, at the optimum, the derivative of the Lagrangian with respect to **X** and **x** must be zero, that is

$$\nabla_{\mathbf{X}} L(\mathbf{X},\ \mathbf{x}) = 2\sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)\mathbf{I} - \boldsymbol{\Psi} = \mathbf{0} \quad (39)$$

and

$$\nabla_{\mathbf{x}} L(\mathbf{X},\ \mathbf{x}) = -4\sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)\mathbf{c}_m - 2\boldsymbol{\Psi}\boldsymbol{x} = \mathbf{0} \quad (40)$$

From Equation (39) it becomes clear that $\boldsymbol{\Psi}$ must be a diagonal matrix. This fact, together with the complementary slackness condition

$$\mathrm{Tr}(\boldsymbol{\Psi}(\mathbf{X} - \mathbf{x}\mathbf{x}^{\mathrm{T}})) = \mathbf{0} \quad (41)$$

implies that the off-diagonal elements of **X** must equal those of $\mathbf{x}\mathbf{x}^{\mathrm{T}}$. However, this does not necessarily mean that **X** is of rank one. From the complementary slackness condition (41) we have that **X** will equal $\mathbf{x}\mathbf{x}^{\mathrm{T}}$ whenever the constraint is active (i.e., $\boldsymbol{\Psi} \neq \mathbf{0}$). So by finding under which conditions $\boldsymbol{\Psi} \neq \mathbf{0}$ we will find the conditions that guarantee that the solution of (14) coincides with the solution of the original problem (12). For that purpose, if we set $\boldsymbol{\Psi} = \mathbf{0}$ we have from (39) and (40) that

$$\sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m) = 0 \tag{42}$$

$$\sum_{m=1}^{M} \gamma_m(\mathrm{Tr}(\mathbf{X}) - 2\mathbf{c}_m^{\mathrm{T}}\mathbf{x} + b_m)\mathbf{c}_m = \mathbf{0} \tag{43}$$

Let $t = \mathrm{Tr}(\mathbf{X})$ and $\mathbf{z} = [z_1, z_2]^{\mathrm{T}} = \mathbf{x}$, keeping in mind that $\mathbf{c}_m = [x_m, y_m]$, we can rewrite the above equations as

$$2\left(\sum_m \gamma_m x_m\right) z_1 + 2\left(\sum_m \gamma_m y_m\right) z_2 - \left(\sum_m \gamma_m\right) t = \sum_m \gamma_m b_m \tag{44}$$

$$2\left(\sum_m \gamma_m x_m^2\right) z_1 + 2\left(\sum_m \gamma_m y_m x_m\right) z_2 - \left(\sum_m \gamma_m x_m\right) t = \sum_m \gamma_m b_m x_m \tag{45}$$

$$2\left(\sum_m \gamma_m x_m y_m\right) z_1 + 2\left(\sum_m \gamma_m y_m^2\right) z_2 - \left(\sum_m \gamma_m y_m\right) t = \sum_m \gamma_m b_m y_m \tag{46}$$

which can be expressed in a compact way as

$$\mathbf{A}\begin{bmatrix}\mathbf{z}\\ t\end{bmatrix} = \boldsymbol{\delta} \tag{47}$$

Therefore, $\boldsymbol{\Psi}$ will be equal to $\mathbf{0}$ only if (47) has a solution where $\mathbf{x} = \mathbf{z}$ and $\mathrm{Tr}(\mathbf{X}) = t$. Additionally, we have that for the solution to be a feasible point it must be satisfied that $\mathrm{Tr}(\mathbf{X} - \mathbf{x}\mathbf{x}^{\mathrm{T}}) \geq 0$ which implies that $\mathrm{Tr}(\mathbf{X}) \geq ||\mathbf{x}||^2$ or, equivalently $||\mathbf{z}||^2 \leq t$. This implies that if problem (17) is infeasible, then $\boldsymbol{\Psi} \neq 0$ and hence, $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{T}}$ so that the solution of the relaxed problem (14) coincides with that of the original problem (12), which proves Proposition 1.

■

Appendix 2: Proof of Lemma 1

By the Schur's complement we have that

$$\mathbf{X} - \mathbf{x}\mathbf{x}^{\mathrm{T}} \succcurlyeq 0 \Leftrightarrow \begin{Bmatrix}\mathbf{X} & \mathbf{x}\\ \mathbf{x}^{\mathrm{T}} & 1\end{Bmatrix} \succcurlyeq 0 \tag{48}$$

Let introduce a new variable $\mathbf{Z} = \begin{bmatrix}\mathbf{X} & \mathbf{x}\\ \mathbf{x}^{\mathrm{T}} & 1\end{bmatrix}$. We then have that $\mathrm{Tr}(\mathbf{X}) = \mathrm{Tr}(\mathbf{Z}) - 1$ and that

$$\mathbf{c}_m^{\mathrm{T}}\mathbf{x} = [0^{\mathrm{T}}1]\mathbf{Z}\begin{bmatrix}\mathbf{c}_m\\ 0\end{bmatrix} = \mathrm{Tr}\left(\mathbf{Z}\begin{bmatrix}0 & 0\\ \mathbf{c}_m^{\mathrm{T}} & 0\end{bmatrix}\right) \tag{49}$$

By rearranging terms and the previous conditions on $\mathbf{Z}$ we end up with problem (18) and the equivalence is established. The equivalence between the two solutions follows directly from the definition of $\mathbf{Z}$.

■

Acknowledgements
The authors would like to thank the anonymous reviewers for their useful comments and suggestions that lead to a significant improvement in the clarity of exposition and motivation of the present study. This study was supported in part by the Spanish Ministry of Science and Innovation under the grant TEC2009-14219-C03-01; El Consejo Social de la UPM; the Spanish Ministry of Science and Innovation in the program CONSOLIDER-INGENIO 2010 under the grant CSD2008-00010 COMONSENS; the European Commission under the grant FP7-ICT-2009-4-248894-WHERE-2; the European Commission under the grant FP7-ICT-223994-N4C and the Spanish Ministry of Science and Innovation under the complementary action grant TEC 2008-04644-E; Spanish Ministry of Science and Innovation under the grant TEC2010-21217-C02-02-CR4HFDVL.

Authors' contributions
BB proposed and developed the solution to the problem, derived the distributed algorithm for its computation and run the numerical simulations. SZ conceived the study and helped in the draft of the manuscript. All authors read and approved the manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 15 May 2011 Accepted: 1 May 2012 Published: 1 May 2012

References

1. AH Sayed, A Tarighat, N Khajehnouri, Network-based wireless location. IEEE Signal Process Mag. **22**(4), 24–40 (2005)
2. N Patwari, JN Ash, S Kyperountas, AO Hero, RL Moses, NS Correal, Locating the nodes. IEEE Signal Process Mag. **22**(4), 54–68 (2005)
3. MS Arulampalam, S Maskell, N Gordon, T Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process. **50**(2), 174–188 (2002). 29. doi:10.1109/78.978374
4. GV Zàruba, M Huber, FA Kamangar, I Chlamtac, Indoor location tracking using RSSI readings from a single Wi-Fi access point. Wirel Netw. **13**(2), 221–235 (2007). doi:10.1007/s11276-006-5064-1
5. Y Tai, Y Bo, Collaborative target tracking in wireless sensor network, in *Proc 9th Int Conf Electronic Measurement & Instruments ICEMI'09, 16-19 Aug 2009; Beijing*, 2-1005–2-1010 (2009)
6. H Wu, G Tian, B Huang, Multi-robot collaborative localization methods based on Wireless Sensor Network, in *Proc IEEE Int Conf Automation and Logistics ICAL, 1-3 Sep 2008; Qindao*, 2053–2058 (2008)
7. H Ren, MQH Meng, Power adaptive localization algorithm for wireless sensor networks using particle filter. IEEE Trans Veh Technol. **58**(5), 2498–2508 (2009)
8. F Aounallah, R Amara, MTH Alouane, Particle filtering based on sign of innovation for tracking a jump Markovian motion in a binary WSN, in *Proc Third Int Conf Sensor Technologies and Applications SENSORCOMM'09, 18-23 Jun 2009; Athens*, 252–255 (2009)
9. KWK Lui, WK Ma, HC So, FKW Chan, Semi-definite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed. IEEE Trans Signal Process. **57**(2), 752–763 (2009)
10. Z Luo, AMC So, Y Ye, S Zhang, Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process Mag. **27**(3), 20–34 (2010)
11. S Boyd, L Vandenberghe, *Convex Optimization*, (Cambridge University Press, New York, 2004)
12. D Blatt, AO Hero, Energy-based sensor network source localization via projection onto convex sets. IEEE Trans Signal Process. **54**(9), 3614–3619 (2006)
13. Q Shi, C He, A new incremental optimization algorithm for ML-based source localization in sensor networks. IEEE Signal Process Mag. **15**, 45–48 (2008)

14. DP Bertsekas, *Constrained optimization and Lagrange Multiplier Methods*, (Academic Press, New York, 1982)
15. J Li, E Elhamifar, IJ Wang, R Vidal, Consensus with robustness to outliers via distributed optimization, in *Proc 49th IEEE Conf Decision and Control (CDC), 15-17 Dec 2010; Atlanta*, 2111–2117 (2010)
16. N Patwari, IAO Hero, M Perkins, NS Correal, RJ O'Dea, Relative location estimation in wireless sensor networks. IEEE Trans Signal Process. **51**(8), 2137–2148 (2003). doi:10.1109/TSP.2003.814469
17. B Bejar, P Belanovic, S Zazo, Distributed Gauss-Newton method for localization in Ad-Hoc networks, in *43rd Asilomar Conference on Signals, Systems, and Computers, 7-10 Nov 2010; Pacific Grove*, 1452–1454 (2010)
18. B Bejar, P Belanovic, S Zazo, Distributed consensus-based tracking in wireless sensor networks: a practical approach, in *Proceedings of the European Signal Processing Conference, EUSIPCO, 29 Aug - 1 Sep 2011; Barcelona*, 2019–2023 (2011)
19. R Olfati-Saber, R Murray, Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans Autom Control. **49**(9), 1520–1533 (2004). doi:10.1109/TAC.2004.834113
20. MG Rabbat, RD Nowak, JA Bucklew, Generalized consensus computation in networked systems with erasure links, in *Proc IEEE 6th Workshop Signal Process Adv Wirel Commun, 5 - 8 Jun 2005; New York*, 1088–1092
21. K Madsen, HB Nielsen, O Tingleff, in *Methods for Non-Linear Least Squares Problems*, 2004) (Richard Petersens Plads, Kgs. Lyngby http://www2.imm.dtu.dk/pubdb/p.php?660

doi:10.1186/1687-6180-2012-95
Cite this article as: Béjar and Zazo: **A practical approach for outdoors distributed target localization in wireless sensor networks.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:95.