Abstract—Energy efficiency, scalability and robustness are key features of Ad-hoc and Wireless Sensor Networks and the use of decentralized algorithms is of practical importance in such scenarios. A method for node localization is proposed by solving a nonlinear least-squares problem in a distributed fashion. For that purpose we propose a Gauss-Newton algorithm with embedded consensus that requires only local communication and converges to the centralized version.

I. INTRODUCTION

A. Motivation

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. This is a key feature of Wireless Sensor Networks (WSN), which are composed of nodes with limited computational and power resources. WSNs must 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 of great practical importance. The spectra of potential applications of these networks is very wide, ranging from monitoring, surveillance, and localization, among others [1], [2]. Localization is a key task (often mandatory) in many applications [1] and therefore, distributed localization algorithms are of high importance.

B. Related work

A great variety of methods exists for acquiring the position of a target node by fusing Time-of-Arrival (TOA), Time-difference of Arrival (TDOA), Angle of Arrival (AOA) or Received Signal Strength Indicator (RSSI) measurements [1]. In this paper we focus on single antenna nodes without tight synchronization abilities, which leads us to the use of RSSI measurements for the localization task. The position of the target node can be determined by solving a nonlinear Least-Squares (LS) problem using well-known optimization methods like the Gauss-Newton method. However the standard Gauss-Newton method is a centralized approach and can not be used directly in a decentralized WSN. We propose a way of distributing this procedure by means of consensus algorithms [3]. Recently, consensus algorithms have been successfully applied to distribute many other well-known algorithms such as Kalman filtering [4].

C. Contributions

In this contribution we use consensus to derive a distributed version of the Gauss-Newton method by exploiting the inherent block-wise structure of the problem in the same spirit as in [5] for principal component analysis. The proposed algorithm solves the nonlinear LS problem of localization by only local communication between neighboring nodes (1-hop). Further, the algorithm is robust and scalable as nodes do not require any knowledge about the network topology or size. In the simulations we show the performance of the proposed algorithm compared to the centralized nonlinear LS solution, as well as a widely used recursive linearized LS algorithm [1]. We also examine the inherent energy-performance trade-off of the proposed approach.

II. PROBLEM FORMULATION AND DEFINITIONS

Consider an ad-hoc network (such as a wireless sensor network) composed of \( N \) nodes. Each node \( n \) communicates only within its 1-hop neighborhood \( \mathcal{N}_n \) and has no knowledge of the network. Explicitly, none of the nodes know \( N \) or the topology matrix (Laplacian) \( \mathbf{L} \). A quasi-static target (its location in the plane \( \mathbf{x} = [x_t, y_t]^T \) can be assumed not to change during the localization process) is sensed by the nodes via some undirected means (e.g. microphone or radio receiver). Hence, each node \( n \) obtains a vector of \( s_n \) noisy distance estimates \( d_n \). The geometry of the problem implies that, for each node, the following equation must be satisfied

\[
d_n^2 = (x_t - x_n)^2 + (y_t - y_n)^2 ,
\]

where \( (x_n, y_n) \) are the coordinates of the \( n^{th} \) node and \( d_n \) is the true distance between the \( n^{th} \) node and the target. Consider the \( s_n \) noisy distance estimates of the \( n^{th} \) and define the function

\[
f_n(\mathbf{x}) = (\mathbf{x}^T \mathbf{x}) \mathbf{1}_n - 2 (\mathbf{a}_n^T \otimes \mathbf{1}_n) \mathbf{x} + \mathbf{b}_n ,
\]
where \( \hat{b}_n = (x_n^2 - y_n^2)1_n - \hat{d}_n \), \( 1_n \) is a \( s_n \times 1 \) vector of all ones, \( a_n = [x_n y_n]^T \) and \( \otimes \) denotes the Kronecker product.

Let the total number of samples in the network be \( S = \sum_{n=1}^{N} s_n \). Considering all data samples together we can form the vector-valued cost function \( f(x) \) of size \( S \) as

\[
f(x) = \left[ f_1(x)^T \ldots f_S(x)^T \right]^T
\]

\[
= (x^T x) \begin{bmatrix} 1 & \vdots & 1_N \end{bmatrix} - 2 \begin{bmatrix} a_N^T \otimes 1 & \vdots & a_N^T \otimes 1_N \end{bmatrix} x + \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix}
\]

\[
= (x^T x)1 - 2Ax + \hat{b}
\]  

(3)

where each row \( s \) of \( A \) holds \([x_s y_s] \), the coordinates of the anchor providing the distance estimate of the \( s \)th equation and \( 1 \) is a \( S \times 1 \) vector of all ones.

In order to have a least-squares estimate of the target position \( x \), the norm of \( f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}^S \) has to be minimized, giving the optimization problem

\[
\hat{x} = \arg \min_{x} \|f(x)\|_2 = \arg \min_{x} f(x)^T f(x).
\]  

(4)

### III. ALGORITHM

The optimization problem (4) represents a nonlinear least-squares problem that can be solved using an iterative descent algorithm like the Gauss-Newton method [6], but since it requires access to the entire contents of \( A \) and \( b \) simultaneously, in an ad-hoc network it could only be implemented in a central location such as a fusion center. The standard (centralized) Gauss-Newton procedure is given in Algorithm 1.

#### Algorithm 1 Gauss-Newton method

1: \( \hat{x}^{(0)} \leftarrow x_0 \), \( k = 0 \) \{Initialization\}
2: \hspace{1em} while \( \neg \text{found} \) \& \( k < k_{\text{max}} \) do
3: \hspace{2em} \( h_{gn} \leftarrow -(J^{(k)}T J^{(k)})^{-1} J^{(k)}T f(x^{(k)}) \) \{Descent direction\}
4: \hspace{2em} if \( \nexists \ h_{gn} \) then  
5: \hspace{3em} \text{found} = true
6: \hspace{2em} end if
7: \hspace{2em} \( x^{(k+1)} \leftarrow x^{(k)} + h_{gn} \) \{Update\}
8: \hspace{2em} \( k \leftarrow k + 1 \)
9: \hspace{1em} end while

where \( h_{gn} \) represents the descent direction (i.e. direction that reduces the value of the cost function) and \( J^{(k)} = J(x^{(k)}) \) with \( J(z) \in \mathbb{R}^{S \times 2} \) representing the Jacobian matrix of \( f(z) \) whose entries are given by

\[
J(z)_{ij} = \frac{\partial f_i(z)}{\partial z_j}, i = 1, \ldots, S, j = 1, 2
\]  

(5)

where \( z = [z_1 \ z_2]^T \) is a two dimensional vector.

From Algorithm 2 it is easy to realize that for distributing the Gauss-Newton algorithm it suffices with computing the search direction in a distributed way. For such purpose, first note the block-wise structure of \( J(z) \)

\[
J(z) = \begin{bmatrix} 2(z_1 - x_1)1_1 & 2(z_2 - y_1)1_1 \\
\vdots & \vdots \\
2(z_1 - x_N)1_N & 2(z_2 - y_N)1_N \end{bmatrix}
\]

(6)

Exploiting the block-wise structure of matrix \( J(z) \) it is easy to note that the search direction of the Gauss-Newton method given in Algorithm 1 can be computed in a distributed way by means of average consensus.

Based on this observation we propose here a fully distributed algorithm, shown as Algorithm 2, which asymptotically approaches the same result using only local information and the exchange of low-volume intermediate results within each node’s 1-hop neighborhood. We immediately note that

#### Algorithm 2 Distributed Gauss-Newton localization

1: \( \hat{x}^{(0)} \leftarrow \text{same initial value } \forall n \in N \)
2: \hspace{1em} for \( k = 0 \) to \( K - 1 \) do
3: \hspace{2em} \( J_n^{(k)} \leftarrow 2 \left[ \hat{x}_n^{(k)} - x_n y_n \right] \)
4: \hspace{2em} \( f_n(\hat{x}_n^{(k)}) \leftarrow \hat{x}_n^{(k)}T x_n - 2 a_n^T \hat{x}_n^{(k)} + x_n^2 + y_n^2 - \frac{d_n^2}{s_n} \)
5: \hspace{2em} \( \Delta_n^{(k)} \leftarrow J_n^{(k)}T J_n^{(k)} \)
6: \hspace{2em} \( \gamma_n^{(k)} \leftarrow \Delta_n^{(k)} \)
7: \hspace{2em} consensus  
8: \hspace{3em} \( \Delta_n^{(k)} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \Delta_n^{(k)} = \frac{1}{N} J^{(k)}T J^{(k)} \)
9: \hspace{3em} \( \gamma_n^{(k)} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \gamma_n^{(k)} = \frac{1}{N} J^{(k)}T f(\hat{x}^{(k)}) \)
10: \hspace{1em} end consensus
11: \hspace{2em} \( h_n^{(k)} \leftarrow \Delta_n^{(k)} \)
12: \hspace{2em} \( \hat{x}^{(k+1)} \leftarrow \hat{x}^{(k)} + h_n^{(k)} \)
13: \hspace{1em} end for

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 [3]. Seeing how \( \Delta_n^{(k)} \in \mathbb{R}^{2 \times 2} \) and is symmetric, and \( \gamma_n^{(k)} \in \mathbb{R}^{2} \), we conclude that each consensus round requires a broadcast of only 5 real values.

### IV. PERFORMANCE

In this section we demonstrate the performance of the proposed algorithm compared to the centralized version via simulations. We also include the performance of the linearized LS solution for comparison [1]. For the simulations we have employed distance estimates using RSSI measurements and assuming the lognormal model \( RSSI_{dB} = P_0 - 10n_p \log_{10}(d/d_0) + X \) with parameters \( P_0 = 0 \text{ dBm}, \ d_0 = 1 \text{ m}, n_p = 2 \) and \( \sigma_{dB} = 2 \), where \( X \sim \mathcal{N}(0, \sigma_{dB}^2) \)
is a Gaussian measurement error term. For the simulations nodes have been randomly deployed into a squared area of 20 × 20 m², each node having a degree (number of neighbors) of 3. All nodes are assumed to start with the initial value $\hat{x}(0) = [0, 0]^T$. In Figure 1 we show a simulation where five anchor nodes (filled circles) try to locate one target node (star). In this particular example five consensus rounds were used in each Gauss-Newton iteration. In the figure we can observe the evolution of the position estimate of each of the anchor nodes as the algorithm iterates. We note that all the anchors follow trajectories close to the centralized solution (red line). With an increased number of consensus rounds, they all converge ever more closely to the same (centralized) solution.

Hence, in Figure 2 we show the average localization error as a function of the number of consensus rounds. Two curves are shown: for the case a network of six (dashed line) and ten (solid line) nodes. For comparison, we also show the average error of the centralized Gauss-Newton and the centralized LS (linearized) algorithms. As stated earlier, the distributed approach converges to the centralized one as the number of consensus rounds increases. It is worth mentioning that the number of consensus rounds depends strongly on how close we start from the true target position.

If some prior information (history) is available on the target position, convergence to the centralized solution will be much faster. It is also important to say that the convergence time will depend on the number of elements on the network, assuming a fixed node degree, as more time will be needed to average as the number of nodes increases. Another important fact is that both nonlinear LS solutions outperform the linearized approximation.

V. CONCLUSIONS

We have presented a distributed consensus-based Gauss-Newton method for target localization that asymptotically converges to the centralized solution as the number of consensus rounds increases. We have also given a comparison with the linearized LS version showing the superior performance of the nonlinear approach. Our future steps include the extension to the tracking problem, which will also allow us a better initialization step, through the use of the tracking history, in order to speed up convergence.

REFERENCES