Cooperative Localization from Imprecise Range-Only Measurements: A Non-Convex Distributed Approach

Ian McInerney, Xu Ma and Nicola Elia

Abstract— **This paper presents a distributed method to locate a target object using multi-agent systems with only knowledge of the agent position and distance between them and the target. The problem is formulated as a non-convex quadratically constrained program, which is then solved using an optimization dynamics approach. The method presented can be applied to an arbitrary undirected network, and only requires agents communicating their estimate of the target's position and their calculated dual variables. The proposed method is derived from the Range-Based Least-Squares method, and becomes the Maximum Likelihood Estimator for this problem under Gaussian noise. We present the convergence results and also numerical simulations of this method.**

I. Introduction

Localizing an object is one of the basic problems in sensor networks, and can be done using a variety of measurements. The most basic types are

- Trilateration - Localizing using only distance measurements
- Triangulation - Localizing using bearing measurements.

To accomplish localization using the triangulation method, the sensing agent must have a method of measuring the direction of the incoming signal from the target object. This can be accomplished by having phased-array antennas (for RF-based positioning), or a microphone array (for audio-based positioning). Then additional signal processing steps must be done to compute the sensed direction from the measurements before sending it into the localization algorithm. In the trilateration method, the sensing agent measures the distance between itself and the target object. This can be accomplished using multiple methods including received-signal strength measurements, propagation time measurements, and various others [1].

We focus on the problem of localizing the target object using only the distance measurements. This problem has been studied extensively in the literature [1]–[14], and can be done using several different methods.

In this work, we propose a novel method for solving the localization problem in a distributed sensor network. This method is based upon a range-based least squares approach, which is the Maximum Likelihood Estimator formulation for this problem.

Ian McInerney, Xu Ma and Nicola Elia are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA.
Emails: {imciner2, maxu, nelia}@iastate.edu
This work was partially supported by NSF grants CNS-1239319 and CCF-1320643, and AFOSR grant FA 9550-15-1-0119.

This paper is structured as follows: in Section II a brief overview of the existing methods and their limitations are provided. In Section III we derive the optimization problem we will solve, and provide a discussion on its relation to the Maximum Likelihood Estimation problem. In Section IV we derive the optimization dynamics for the proposed method, and present its convergence results. Finally, in Section V we provide examples of the proposed method running against existing methods.

Notation: Let N be the total number of agents. Let e_i be the column vector containing a 1 in the ith component and zeros otherwise. Define the vectors $\hat{p} = [\,\hat{x},\, \hat{y},\, \hat{z}\,]$ to be the target object's position in $\mathbf{R}^3$ and $p = [\,x,\, y,\, z\,]$ to be the sensing agent's position in $\mathbf{R}^3$. The subscript i on a varaible indicates that it belongs to the ith node, e.g., p_i is the position of the ith node. The hat on a variable means it is an estimate of the target variable, e.g. $\hat{x}$ is the estimated x location. The symbol $\|\bullet\|$ means the 2-norm of a vector.

II. Existing Methods

Several methods for estimating a target's location based upon range measurements exist, and can be grouped into three distinct categories: 1) Kalman Filters, 2) Range-Based Least Squares, 3) Squared-Range-Based Least Squares.

A. Kalman Filters

A popular method for estimating a target object's position is a Kalman filter. This method is able to combine the sensor measurements, filter the sensor noise, and estimate the object's state vector given an internal model for the target object. The Kalman filter is the preferred approach when the target object is not stationary, e.g., moving along a trajectory, since the filter includes a physics model of the object's motion. In localization applications, there are three main models used

- Position (P)
- Position and Velocity (PV)
- Position, Velocity and Acceleration (PVA).

The Kalman filter was first used as a centralized estimator, where one node will aggregate all information and then perform the calculations to estimate the state and covariance matrices. A good overview of the three types of centralized Kalman filters (P, PV, and PVA) used in localization can be found in [13]. Recent work has focused around creating distributed algorithms for the Kalman filter, such as that in [3]. In that work, the Kalman filter is reduced to a local estimator on each agent (using information from its neighboring agents), and then the agent's local estimates are transmitted to its neighbors to be combined to update the local estimate again.

B. Least Squares Estimation

The most natural way of framing the target localization problem when the object is stationary is to use the Euclidean distance between the agents and the target object, then find the location that satisfies all those distance relations. The Euclidean distance relation between the position of node i and the target object is

$$\|\hat{p} - p_i\| = r_i \tag{1}$$

The set of all these equations will be a system of non-linear equations with 3 unknowns and i constraints. If $i > 3$, there may not be an exact solution to this system, since it becomes overdetermined. To overcome this a non-linear least-squares problem can be formulated. This formulation is called the *Range-Based Least Squares* (R-LS) estimation, and takes the form of

$$\underset{\hat{x},\hat{y},\hat{z}}{\text{minimize}} \quad \sum_{i=1}^{N} \left(r_i - \|\hat{p} - p_i\|\right)^2 \tag{2}$$

This formulation is non-convex, and therefore finding the global solution is a very difficult task. One method to make this problem computationally tractable is to linearize (1) before placing it into the R-LS formulation. This makes the problem similar to an ordinary least-squares problem that can be solved using existing methods [1], [2], [11], [12].

In practical applications, the distance measurements r_i will be noisy measurements of the actual distance. If the assumption is made that the noise of each distance measurement is independent-identically distributed Gaussian with zero-mean and variance σ_i, the R-LS formulation becomes a Maximum Likelihood Estimator (MLE) for the target object's location.

In practice, this method is not used in the form given in (2). Instead, (2) is reformulated as a semidefinite program. To do this, define $g_i = \|\hat{p} - p_i\|$. Then create the matrices

$$X = \begin{bmatrix} \hat{p} \\ 1 \end{bmatrix} \begin{bmatrix} \hat{p}^T & 1 \end{bmatrix}, G = \begin{bmatrix} \hat{g} \\ 1 \end{bmatrix} \begin{bmatrix} \hat{g}^T & 1 \end{bmatrix}, C_i = \begin{bmatrix} I & -p_i \\ -p_i^T & \|p_i\|^2 \end{bmatrix}$$

The definition of the semidefinite program for the R-LS program is then

$$\begin{aligned} \underset{X,G}{\text{minimize}} \quad & \sum_{i=1}^{N} G_{ii} - 2r_i G_{N+1,i} + r_i^2 \\ \text{subject to} \quad & G_{ii} = \textbf{Tr}(C_i X) \qquad \forall i = 1, 2, \ldots, N \\ & G \succeq 0, X \succeq 0 \\ & G_{N+1,N+1} = X_{4,4} = 1 \\ & \textbf{Rank}(X) = \textbf{Rank}(G) = 1 \end{aligned} \tag{3}$$

Note that in (3), the constraints on the rank of X and G are non-convex, so again this problem is computationally difficult. Many methods in the literature (e.g. [4]–[7]) relax this constraint by removing it. This method is called the Semidefinite Relaxation method (SDR method).

Theoretical results in [4] show that the matrix G in the SDR method will always be rank 1, but there exist examples where the rank of X is greater than 1 (see Example 1 in [4]). In those cases, the SDR is no longer an exact relaxation of the original problem and can only provide an approximate optimizer.

C. Squared-Range-Based Least Squares

Due to the non-convexity of the formulation in (2), algorithms trying to solve it exactly are computationally difficult. To overcome this, another problem formulation exists, called the *Squared-Range-Based Least Squares* (SR-LS), and takes the form of

$$\underset{\hat{x},\hat{y},\hat{z}}{\text{minimize}} \quad \sum_{i=1}^{N} \left(r_i^2 - \|\hat{p} - p_i\|^2\right)^2 \tag{4}$$

While it is a non-convex problem, prior results have shown that it falls into the category of generalized trust region subproblems, see [4] and references therein. Those subproblems allow for conditions of optimality to be derived, and the creation of a numerical algorithm that finds the global optimizer.

Note that this problem formulation, while similar to the R-LS formulation, is no longer an MLE for the target object's position.

III. Problem Formulation

In this section, we pose the optimization problem we will use to solve the localization problem over a multi-agent network.

A. Problem Formulation

We start with the problem formulation for the R-LS method, given in (2). By rewriting it, we create the following

$$\begin{aligned} \underset{\hat{x},\hat{y},\hat{z},a_i}{\text{minimize}} \quad & \sum_{i=1}^{N} a_i^2 \\ \text{subject to} \quad & \|\hat{p} - p_i\|^2 = (a_i - r_i)^2 \qquad \forall i = 1, 2, \ldots, N \end{aligned}$$

where a_i has the physical meaning of being the distance between the estimated target location and the measured target location. This relation can be seen in Figure 1.

In this work, we will be using several assumptions about the number of agents and also how they are located in the search space.

Assumption 1: The number of agents should be greater than the dimension of the space being searched.

Assumption 2: The agents are distributed in the search space such that they do not form a lower dimensional space (e.g., for a 3D search space the agents are not all co-planar).

In a multi-agent system, the communication links between the agents can be modeled as a graph with edges where a link exists. In this work, we make the following assumption

Assumption 3: The communications network is a simple, connected, undirected graph.

We can then create a matrix $L \in \mathbf{R}^{N \times N}$ with the ith diagonal entry representing the number of agents talking with agent i and an (i, j) entry of -1 if a communications link exists between agents i and j and 0 otherwise. The

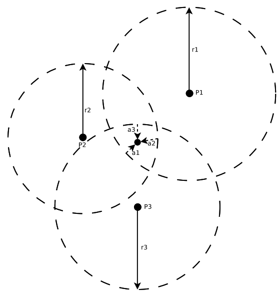


Fig. 1: An object location problem in $\mathbf{R}^2$, shown with error in measurements. The target object is shown in the center, with 3 agents surrounding.

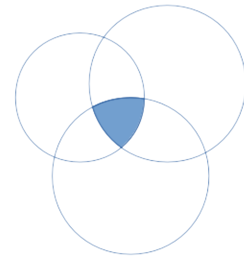


Fig. 2: An example in $\mathbf{R}^2$ showing the set of optimal points (shaded region) when using the convex relaxation and agents with overlapping measurements.

matirx L is also known as the graph Laplacian matrix of the communications network.

Let the vectors $\hat{x} = \left[\, \hat{x}_1 \; \hat{x}_2 \; \cdots \; \hat{x}_N \,\right]^T$, $\hat{y} = \left[\, \hat{y}_1 \; \hat{y}_2 \; \cdots \; \hat{y}_N \,\right]^T$ and $\hat{z} = \left[\, \hat{z}_1 \; \hat{z}_2 \; \cdots \; \hat{z}_N \,\right]^T$ be vectors containing the estimate produced for the target's x, y, and z positions by the ith agent.

The communications network can be represented as three additional constraints that will force consensus between the nodes, namely $L\hat{x} = 0$, $L\hat{y} = 0$ and $L\hat{z} = 0$. This creates the optimization problem

$$\begin{aligned} \underset{\hat{x}_i,\hat{y}_i,\hat{z}_i,a_i}{\text{minimize}} \quad & \sum_{i=1}^{N} a_i^2 && \text{(5a)} \\ \text{subject to} \quad & L\hat{x} = 0 && \text{(5b)} \\ & L\hat{y} = 0 && \text{(5c)} \\ & L\hat{z} = 0 && \text{(5d)} \\ & \|\hat{p}_i - p_i\|^2 = (a_i - r_i)^2 \quad \forall i = 1, 2, \ldots, N && \text{(5e)} \end{aligned}$$

Starting from (5), we insert penalty terms into the cost function (similar to those for the augmented Lagrangian) to allow for faster convergence of the agent's estimates and to allow for convergence guarantees in Section IV. The complete optimization problem we use is therefore

$$\begin{aligned} \underset{\hat{x}_i,\hat{y}_i,\hat{z}_i,a_i}{\text{minimize}} \quad & \left\{ \sum_{i=1}^{N} a_i^2 + \left(\|\hat{p}_i - p_i\|^2 - (a_i - r_i)^2 \right)^2 \right. \\ & \left. \quad + k_1 \hat{x}^T L \hat{x} + k_2 \hat{y}^T L \hat{y} + k_3 \hat{z}^T L \hat{z} \right\} && \text{(6a)} \\ \text{subject to} \quad & L\hat{x} = 0 && \text{(6b)} \\ & L\hat{y} = 0 && \text{(6c)} \\ & L\hat{z} = 0 && \text{(6d)} \\ & \|\hat{p}_i - p_i\|^2 = (a_i - r_i)^2 \quad \forall i = 1, 2, \ldots, N && \text{(6e)} \end{aligned}$$

The objective function in (6a) seeks to minimize the 2-norm of the deviation from measured distance. This norm presents the most natural formulation of the problem, but if desired other norms could be used in the cost function instead.

B. Convex Relaxation

In this work, we began with the non-convex problem given in (5). This problem has a convex objective function, but has a non-convex constraint (5e). For this problem a convex relaxation is trivial, simply turn the non-convex equality constraint into the inequality constraint

$$\|\hat{p}_i - p_i\|^2 \leq (a_i - r_i)^2$$

There is a problem with this relaxation though. When a_i gets minimized, its lowest value will make that constraint become

$$\|\hat{p}_i - p_i\|^2 \leq r_i^2$$

which means that the radius of the ball created around the agent will never be smaller than the sensed distance r_i. This means if agents have measurements that create overlapping regions, there exists a set of optimal points that all share the same optimal cost $\left(\sum_{i=1}^{N} a_i^2 = 0\right)$. This is illustrated for $\mathbf{R}^2$ in Figure 2.

For this reason, we do not use the convex relaxation and instead develop methods to solve the non-convex QCQP given in (5).

C. Similarity with Maximum Likelihood Estimation

If the measurements of the distance have additive noise associated with them, the localization problem becomes a statistical estimation problem that can be formulated using Maximum Likelihood (ML) estimation. In ML estimation, the objective is to find the model parameter that maximizes the probability that the observed value is returned from the model [15].

Assume a zero-mean Gaussian noise distribution with each distance measurement having noise independent of each other and identical variance σ. Then the ML problem for localization can be represented as an optimization problem [7], given in (7)

$$\begin{aligned} \underset{\hat{x},\hat{y},\hat{z},a_i}{\text{minimize}} \quad & \sum_{i=1}^{N} \frac{1}{\sigma} a_i^2 && \text{(7a)} \\ \text{subject to} \quad & \|\hat{p} - p_i\|^2 = (a_i - r_i)^2 \quad \forall i = 1, 2, \ldots, N && \text{(7b)} \end{aligned}$$

***Theorem** 1:* The problem formulation given in (6) is the ML estimator for the localization problem.

Proof: At optimality, the constraints of the opimization problem will be satisfied. That means specifically that (6b), (6c) and (6d) are true, which when coupled with the structure of the Laplacian matrix implies

$$\begin{aligned}\hat{x}_1 &= \hat{x}_2 = \cdots = \hat{x}_N \\ \hat{y}_1 &= \hat{y}_2 = \cdots = \hat{y}_N \\ \hat{z}_1 &= \hat{z}_2 = \cdots = \hat{z}_N\end{aligned} \tag{8}$$

From those equalities, the equality constraints given in (6e) will be the same as those in (7b). Additionally, when the relations in (8) are substituted into the cost function (6a), the following is true

$$L\hat{x} = 0, \quad L\hat{y} = 0, \quad L\hat{z} = 0, \quad \|\hat{p}_i - p_i\|^2 - (a_i - r_i)^2 = 0$$

With those terms zero, the optimal cost will be determined solely by the $\sum_{i=1}^{N} a_i$ term. This makes the optimal cost function (6a) the same as (7a). Since the constraints are identical at optimality and the cost for (6) and (7) differ only by a scaling, they are solving the same problem. Therefore the problem given in (6) is a ML estimator for the target localization problem under the given assumptions. ■

IV. Solver Formulation

In this section, we derive a continuous-time algorithm that will solve the optimization problem posed in (6).

A. Reformulation of Problem

We define the Lagrangian function for (6) using the dual variables μ, α, β, and γ, along with expanding the norm functions. Note that this is the fully augmented Lagrangian of (5) due to our added terms in the cost function.

$$\begin{aligned} L(\hat{x}, \hat{y}, \hat{z}, a, \mu, \alpha, \beta, \gamma) &= k_1 \hat{x}^T L \hat{x} + k_2 \hat{y}^T L \hat{y} + k_3 \hat{z}^T L \hat{z} \\ &+ \alpha^T L \hat{x} + \beta^T L \hat{y} + \gamma^T L \hat{z} \\ &+ \sum_{i=1}^{N} \Big\{ a_i^2 + \left[(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + (\hat{z}_i - z_i)^2 - (a_i - r_i)^2 \right]^2 \\ &+ \mu_i \left[(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + (\hat{z}_i - z_i)^2 - (a_i - r_i)^2 \right] \Big\} \end{aligned} \tag{9}$$

and create the associated optimization problem

$$\max_{\mu,\alpha,\beta,\gamma} \; \min_{\hat{x},\hat{y},\hat{z},a_i} \; L(\hat{x}, \hat{y}, \hat{z}, a, \mu, \alpha, \beta, \gamma) \tag{10}$$

B. Proposed Method

We propose the following dynamical system to solve the problem in (10), and consequently the problem in (6).

$$\dot{a}_i = 2\left(2\dot{\mu}_i + \mu_i\right)(a_i - r_i) - 2a_i \tag{11a}$$

$$\dot{\hat{x}}_i = -2\left(2\dot{\mu}_i + \mu_i\right)(\hat{x}_i - x_i) - e_i^T L \alpha - 2k_1 e_i^T L \hat{x} \tag{11b}$$

$$\dot{\hat{y}}_i = -2\left(2\dot{\mu}_i + \mu_i\right)(\hat{y}_i - y_i) - e_i^T L \beta - 2k_2 e_i^T L \hat{y} \tag{11c}$$

$$\dot{\hat{z}}_i = -2\left(2\dot{\mu}_i + \mu_i\right)(\hat{z}_i - z_i) - e_i^T L \gamma - 2k_3 e_i^T L \hat{z} \tag{11d}$$

$$\dot{\mu}_i = (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + (\hat{z}_i - z_i)^2 - (a_i - r_i)^2 \tag{11e}$$

$$\dot{\alpha}_i = e_i^T L \hat{x} \tag{11f}$$

$$\dot{\beta}_i = e_i^T L \hat{y} \tag{11g}$$

$$\dot{\gamma}_i = e_i^T L \hat{z} \tag{11h}$$

We next derive some theoretical results about the proposed dynamical system.

***Theorem** 2:* The optimal solution to the problem given in (10) occurs at an equilibrium point of the dynamical system in (11).

Proof: Any optimal solution for the problem in (10) must satisfy the KKT conditions, the first of which is that the derivative of the Lagrangian with respect to the primal variables must be zero at the optimal point

$$\nabla_x L(x^\star, \lambda^\star) = 0$$

It can be seen by inspection that equations (11a) through (11d) are equivalent to the derivatives of (9) with respect to the primal variables.

The equilibrium points of (11) occur when the derivative terms equal zero, e.g., $\dot{\hat{x}} = 0$, $\dot{\hat{y}} = 0$, $\dot{\hat{z}} = 0$. Therefore when (11) reaches equilibrium, it is equivalent to the Lagrangian in (9) having its primal derivatives equal to zero.

This implies that the optimal solution to (10) occurs at an equilibrium point of (11). ■

***Lemma** 1:* For large enough k_1, k_2 and k_3, the Hessian of the augmented Lagrangian (9) with respect to the primal variables is positive definite at the equilibrium point.

Proof: The positive semidefiniteness of the Hessian can be easily verified at the equilibrium point, and hence its proof is omitted here due to the limited space. ■

Next we will use Lemma 1 to derive a stability analysis for the equilibrium point of dynamics (11).

***Theorem** 3:* Suppose the primal optimum $x^\star$ to problem (6) is regular.[1] There exist values for $k_1 > 0$, $k_2 > 0$, and $k_3 > 0$ such that if dynamics (11) starts in a neighborhood around the optimal point $(x^\star, \lambda^\star)$, then its trajectories will converge asymptotically to $(x^\star, \lambda^\star)$.

Proof: We only present below a sketch of the convergence proof due to the limitation of space. Readers may refer to [16] for a more generalized convergence analysis.

To study the local stability of the equilibrium $(x^\star, \lambda^\star)$ of dynamics (11), we can linearize (11) around $(x^\star, \lambda^\star)$ and get a Jacobian in the following form

$$J(x^\star, \lambda^\star) = \begin{bmatrix} -H^\star & -B^\star \\ (B^\star)^T & 0 \end{bmatrix}$$

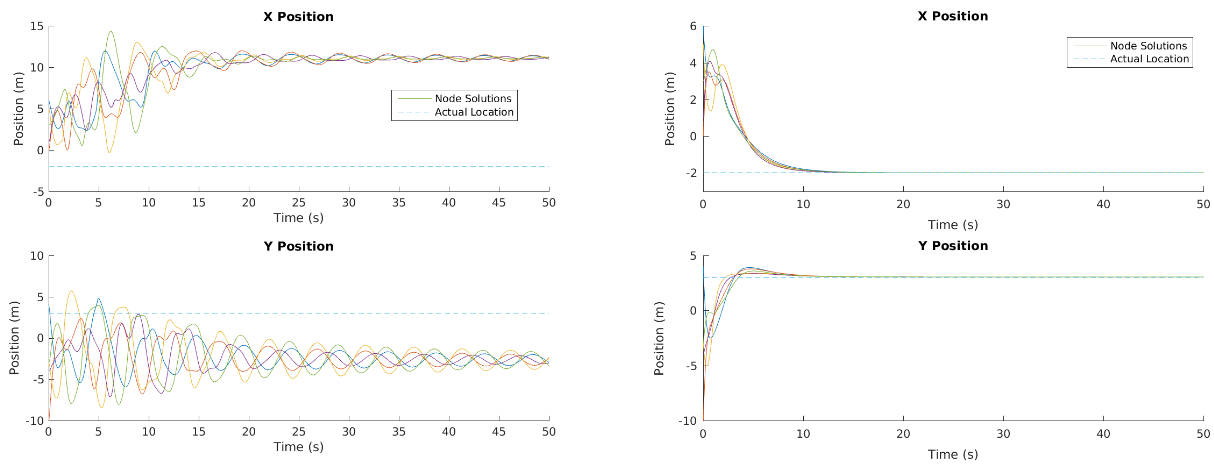
where $H^\star$ is the Hessian of Lagrangian (9) with respect to the primal variable and is evaluated at the equilibrium point $(x^\star, \lambda^\star)$. By Lemma 1 we know that $-H^\star \prec 0$.

Furthermore, by the definition of regularity of $x^\star$ we can also show that $B^\star$ has full column rank. Combining this with $-H^\star \prec 0$ we can show that $J(x^\star, \lambda^\star)$ is Hurwitz, i.e., all its eigenvalues have negative real parts. Therefore, we conclude that equilibrium $(x^\star, \lambda^\star)$ is locally asymptotically stable. ■

C. Distributed Implementation

The dynamical system given in (11) can be implemented on a distributed system where each agent has knowledge of

[1] Given an optimization minimize $f(x)$ subject to $h_i(x) = 0$, for $i = 1, \cdots, m$, a feasible point $\bar{x}$ is said to be *regular* if all gradients $\nabla h_i(\bar{x})$ are linearly independent. Regularity is quite a mild constraint qualification assumption that we can make for an optimization problem.



(a) $k_1 = 0.1, k_2 = 0.1, k_3 = 0.1$ (b) $k_1 = 1, k_2 = 1, k_3 = 1$

Fig. 3: Trajectories of the proposed algorithm with varying k parameter values when solving Example 1.

its own position and distance to the target. Every agent will independently integrate all 8 differential equations in (11) every time step, sharing only the estimated target position and associated dual variables with their neighbors. The variables to be shared include

$$(\hat{x}_i, \hat{y}_i, \hat{z}_i, \alpha_i, \beta_i, \gamma_i)$$

Note that the measured distance r_i, node position p_i and the computed a_i and μ_i do not need to be shared with the neighbors and remain local to the agent. This can add a level of agent privacy to the algorithm, since the agent's position can remain secret and the algorithm still functions.

V. Numerical Examples

In this section, we present several numerical examples demonstrating the proposed solver and its utility against current methods.

In each of these examples, the initial location estimate $\hat{p_i}$ for each agent was placed at the agent's location, and the initial error a_i was set to the measured distance.

Example 1: Based upon an example proposed in [4], place $N = 5$ agents in $\mathbf{R}^2$ at points $p_1 = [6, 4]$, $p_2 = [0, -10]$, $p_3 = [5, -3]$, $p_4 = [1, -4]$, and $p_5 = [3, -3]$, and have them attempt to locate a target object at $\hat{x} = [-2, 3]$ using only distance measurements. The distance measurements have additive Gaussian noise with $\mu = 0$ and $\sigma = 0.1$.

The SDR method of localization (when solved using the CVX toolset [17]) returns an X matrix that is not rank one, meaning the result of the relaxation is not equal to the actual solution [4]. By doing a rank-one approximation using the largest eigenvalue of X, the method returns an estimate $\hat{x} = [-1.3806,\ 2.7741]$.

Running the algorithm proposed in Section IV-B on this problem with $k_1 = 1, k_2 = 1, k_3 = 1$ and a communications network in a ring configuration with Laplacian

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix} \tag{12}$$

produces the result shown in Figure 3b. These nodes converge to an estimate of $\hat{x} = [-1.9907,\ 3.0474]$ with all $k = 1$ since the Hessian at the equilibrium is positive definite.

This example also allows for the investigation of the convergence criteria as the constants k_1, k_2, and k_3 are changed. Figure 3 shows two constant values all $k = 0.1$ in Figure 3a and all $k = 1$ in Figure 3b. The algorithm converges to the solution when $k = 1$, while $k = 0.1$ causes the algorithm to diverge to a different point.

TABLE I: Mean squared error comparison for Example 2.

σ	Proposed	Unconstrained	SDR	GTRS
10^{-3}	$\mathbf{1.01{\times}10^{-6}}$	$4.01{\times}10^{-6}$	$5.92{\times}10^{-6}$	$1.51{\times}10^{-6}$
10^{-2}	$\mathbf{1.26{\times}10^{-4}}$	$4.74{\times}10^{-4}$	$3.29{\times}10^{-4}$	1.72×10^{-4}
10^{-1}	$\mathbf{1.31{\times}10^{-2}}$	$5.61{\times}10^{-2}$	$2.47{\times}10^{-2}$	$2.51{\times}10^{-2}$
1	**1.20**	5.99	4.45	3.62

Example 2: As further comparison, the proposed method was compared against three other established methods

1) Semidefinite relaxation of R-LS (SDR) [5], [6]
2) Unconstrained SR-LS (Unconstrained) [18]
3) Generalized trust region subproblems of SR-LS (GTRS) [4].

This example was done using a Monte Carlo simulation where each method was run 100 times on a problem with 5 agents scattered in a grid. The agent positions were randomly generated from a uniform distribution in $\mathbf{R}^2$ in the range $[-10, 10] \times [-10, 10]$. The radii had zero-mean Gaussian noise added on, with standard deviation as listed in Table

I. The proposed solver was run with the Laplacian given in (12), and $k_1 = k_2 = k_3 = 20000$. In every trial, the Hessian at equilibrium was checked, and all trials resulted in a positive definite Hessian.

The mean squared error of the resulting position (e.g., $\|\hat{p} - p\|$) for each of the methods is reported in Table I. The entries in bold represent the method that performed best for that radii noise level. As seen in the results, the proposed method performed better in all the runs, but had fairly close performance with the GTRS method.

VI. Conclusion

This paper presented a method that can localize an object using only distance measurements. It accomplishes this by creating an optimization system to solve a non-convex problem derived from the R-LS method, maintaining the ML estimator qualities of the R-LS method in the process. The optimization system was shown to have local convergence to the optimal least-squares solution based upon choice of scaling factors k. Its convergence to the proper solution was demonstrated on a known counter-example to the SDR method, and the Monte Carlo simulations have shown this method to produce more accurate results than several other methods while operating on a network of distributed agents.

References

[1] K. Cheung, H. So, W.-K. Ma, and Y. Chan, "A Constrained Least Squares Approach to Mobile Positioning: Algorithms and Optimality," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 20858, pp. 1–23, 2006.

[2] A. Ahmed, Hongchi Shi, and Yi Shang, "SHARP: A New Approach to Relative Localization in Wireless Sensor Networks," in *25th IEEE International Conference on Distributed Computing Systems Workshops*. IEEE, 2005, pp. 892–898.

[3] P. Alriksson and A. Rantzer, "Experimental Evaluation of a Distributed Kalman Filter Algorithm," in *46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5499–5504.

[4] A. Beck, P. Stoica, and J. Li, "Exact and Approximate Solutions of Source Localization Problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.

[5] P. Biswas and Y. Ye, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. Berkeley, California, USA: ACM, 2004, pp. 46–54.

[6] P. Biswas, T.-C. Liang, T.-C. Wang, and Y. Ye, "Semidefinite Programming Based Algorithms for Sensor Network Localization," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 188–220, 2006.

[7] K. W. Cheung, W. K. Ma, and H. C. So, "Accurate approximation algorithm for toa-based maximum likelihood mobile location using semidefinite programming," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2004.

[8] J. A. Jiang, X. Y. Zheng, Y. F. Chen, C. H. Wang, P. T. Chen, C. L. Chuang, and C. P. Chen, "A Distributed RSS-Based Localization Using a Dynamic Circle Expanding Mechanism," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3754–3766, 2013.

[9] A. Khalajmehrabadi, D. Akopian, and D. Pack, "Range-based Trilateration Using Multipurpose Cost Function Optimization with Lagrangian Multipliers," in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, apr 2016, pp. 118–121.

[10] Z. Li, T. Braun, and D. C. Dimitrova, "A Passive WiFi Source Localization System Based on Fine-grained Power-based Trilateration," *Proceedings of the WoWMoM 2015: A World of Wireless Mobile and Multimedia Networks*, 2015.

[11] F. Reichenbach, A. Born, D. Timmermann, and R. Bill, "A Distributed Linear Least Squares Method for Precise Localization with Low Complexity in Wireless Sensor Networks," in *Distributed Computing in Sensor Systems Second IEEE International Conference, DCOSS 2006, San Francisco, CA, USA, June 18-20, 2006. Proceedings*, 2006, pp. 514–528.

[12] A. Savvides, C.-C. Han, and M. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *Proceeding MobiCom '01 Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 166–179.

[13] A. Shareef, "Localization Using Extended Kalman Filters in Wireless Sensor Networks," in *Kalman Filter: Recent Advances and Applications*, V. Moreno and A. Pigazo, Eds., 2009, pp. 297–320.

[14] C. H. Yu, S. H. Min, and J. W. Choi, "Sensor Localization-based Distributed Target Tracking Filter in Underwater Sensor Networks," in *Society of Instrument and Control Engineers of Japan (SICE), 2015 54th Annual Conference of the*, vol. 1. IEEE, 2015, pp. 1–18.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univesity Press, 2004.

[16] X. Ma and N. Elia, "Seeking Saddle-Points for Non-Convex QCQPs via the Distributed Optimization Dynamics Approach," in *Proceedings of the 22nd International Symposium on Mathematical Theory of Networks and Systems*. Minneapolis, MN, USA: University of Minnesota Digital Conservancy, 2016.

[17] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, 2014.

[18] J. Smith and J. Abel, "The spherical interpolation method of source localization," *IEEE Journal of Oceanic Engineering*, vol. 12, no. 1, pp. 246–252, 1987.