Hybrid Localization in Underwater Environment

Filipa Marreiros Malveiro Valdeira
filipa.valdeira@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

November 2018

Abstract

Countless systems require localization methods to accomplish their proposed tasks. While GNSS solutions offer great performance, for large networks this may not be affordable, as it depends on expensive instrumentation. Besides, environmental constraints can even prevent such approaches, as it is the case in indoor or underwater settings. In this work, the latter is used as a testing case for the development of a localization algorithm, suitable for generic mobile networks of agents with range and bearing measurements between them. Existing approaches often depend on parameter tuning, on the accuracy of initialization or their performance may depend on the patterns of motion, thus limiting freedom during missions. Our algorithm deals with these drawbacks, standing on the Maximum Likelihood position estimator for a generic network. The resulting optimization problem is non-convex, so that a convex relaxation is proposed. To serve positioning to mobile agents, two additional horizon-based versions are developed accounting for velocity measurements at each agent. To solve the convex optimization problem, a distributed gradient-based method is provided. This constitutes an advantage over other centralized approaches, which usually exhibit high latency for large networks and present a single point of failure. Additionally, the algorithm estimates all required parameters, and effectively becomes parameter-free. We obtain a parameter-free, outlier-robust and trajectory-agnostic algorithm, with nearly constant positioning error regardless of the trajectories of agents and anchors, achieving better or comparable performance to state-of-the-art methods. Furthermore, the method is distributed, convex and does not require any particular anchor configuration.

Keywords: Localization, Convex Optimization, AUV, Maximum Likelihood Estimation, Convex Relaxation, Ranges and Bearings

1. Introduction

While localization systems often rely on GNSS solutions to accomplish their tasks, some environmental constraints prevent their use and call for different methods. Under this context, we find the case of Autonomous Underwater Vehicles (AUVs), where the high conductivity in water decreases radio wave propagation, hence preventing the use of satellite positioning. Advancements in technology have made communication between these vehicles possible, so that agents may share information to obtain a better localization accuracy. This is called a *cooperative* approach, which is also a trend in other GNSS deprived setting, such as indoors environments. Besides, as networks grow in size, *distributed* approaches (as opposed to centralized) are often desirable. Even though they do not make use of all available information, they involve less communication overhead and latency and do not present a single point of failure. Therefore, we target the case of generic agent networks, able to cooperate with each other to accomplish self-localization tasks in a distributed manner.

A common approach for position estimation is the use of Bayesian Filters, such as the Extended Kalman Filter (EKF), which is very often employed for localization tasks [1, 2]. In spite of its popularity, EKF introduces linearization errors (as non-linear systems must be linearized) and requires correct initialization to avoid convergence problems, which may not be available. In an attempt to overcome this challenges, we adopt a different perspective, which is widely used in the context of sensor networks. Under this framework, the network is usually represented as a graph, where vehicles/sensors are the vertices and measurements relative to each other are edges connecting the vertices.

Regardless of the context and instrumentation used, the acquired measurements will not be exact, that is, they will have an associated error component. This means that an equation system will very likely never provide have a solution, since measurements contradict each other and, generally, no set of positions satisfies all constraints simultaneously. Therefore, a common approach is to formulate this as an optimization problem, where a cost function

must be minimized.

However, in general, formulations of network localization produce *non-convex* optimization problems. Consequently, when a minimum is found for the cost function, it cannot be guaranteed that it is the global and that the optimal value has been found. Considering this, it is possible to either preserve the non-convex function and focus on correct initialization or relax the original formulation into a convex one. Our method falls into the latter category.

1.1. Related work

Related approaches differ in problem formulation, convex relaxation employed and assumptions on network configuration or available measurements. When the error is assumed to follow a known distribution, it is possible to use *Maximum Likelihood* (ML) formulations. They rely on the Maximum Likelihood Estimator (MLE) which is, for large sets of data and high signal to noise ratio, asymptotically unbiased and asymptotically efficient. This means that if the error does follow the considered distribution, MLE will be the optimal estimator, thus making it an attractive approach and our choice in the current work.

A popular relaxation is Semidefinte Programming (SDP) [3], where the original formulation is relaxed to allow for solutions in higher dimensional spaces, originating a standard optimization problem with available solvers. However, this method becomes intractable for larger networks and requires particular network characteristics to obtain good performance. A less computationally expensive approach is Second-Order Cone Programming (SOCP), proposed in [4]. However, this relaxation is shown to be weaker and only able to localize sensors in the convex hull of the anchors, which can be a limitation when the area of actuation is unknown a priori. Another alternative is found in [5], where the authors reformulate the problem to obtain a polynomial and relax it with a common tool for this case, the Sum of Squares (SOS). The main motivation is the possibility to find multiple solutions and counteract the problem of SDP and SOCP, which usually return the analytic centre of the relaxed solution set, but this is often a computationally costly method.

1.1.1 Related work in hybrid localization

While most approaches focus on range only measurements, we consider the hybrid case of both distance and bearing measurements between nodes.

An ML framework is used in [6] to compare different distributions of angle measurements, where ranges are often available. The formulation is closely related to the one used here, although derived for a 2D case and for target location. The non-convexity is not relaxed and a Least Squares approach is used for initialization, after which the optimization is carried out with gradient descent. Naturally, this is subject to local minima as it uses a non-convex cost function and authors note that under higher noise levels some points are very poorly estimated. Authors in [7] study the specific case of AUVs and assume a bounded error model both for ranges and bearing measurements in 2D. These bounds are then converted to linear constraints which, when intersected, produce a convex polyhedron. It is economically advantageous as it does not require any nodes with expensive equipment, but the achieved accuracy is yet slightly inferior to one obtained with heterogeneous cooperative models. In [8], an SDP formulation is extended to accommodate angle measurements, but the number of constraints easily becomes intractable. The proposed solution is to randomly select some of the constraints, but this still constitutes a problem for larger networks. Besides, the anchor placement remains a problem, since targets outside of their convex hull will tend to be positioned inside. An hybrid approach with SDP relaxation is also found in [9], but with an ML formulation, which is very similar to the one in this work. However, we employ a different relaxation, which should prevent the mentioned disadvantages of SDP approaches. The authors also modify the ML and approximate it with a sum of squares, which is not the case in our method.

2. Problem Formulation

In light of previous considerations, we present a localization algorithm for a network of a variable and undefined number of AUVs. Anchors are assumed to be vehicles with access to an accurate positioning system, either by staying on the surface and befitting from GNSS signals or by remaining fixed at a known position. Besides, each vehicle is assumed to have a modem providing noisy distance measurements between itself and other vehicles within a certain range. Some AUVs are also equipped with a vector sensor producing bearing measurements between vehicles. This angular information is assumed to be already in a common frame of reference, which could be obtained recurring to compass measurements, so that all angles are with respect to North direction, for instance. It is further considered that if vehicle "A" has measurements with respect to vehicle "B", then the reverse is also true. All vehicles are mobile and can follow any trajectory, including anchors.

The main goal is, then, to determine the position of each sensor of a network, during a certain time interval. For this purpose the network is represented as an undirected connected graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$, where each different graph cor-

responds to a given time instant. The vertices $\mathcal{V} = \{1, ..., n\}$ correspond to the vehicles, while the edges $i \sim j \in \mathcal{E}^x(t)$ indicate the existence of a measurement ot type x between node i and j, at time t. Particularly, $\mathcal{E}^d(t)$ refers to distance measurements and $\mathcal{E}^u(t)$ to angle measurements between two nodes.

The known sensor positions required for localization (anchors) are indicated by the set $\mathcal{A} = \{1, ..., m\}$. Furthermore, the subset $\mathcal{A}_i^x(t) \subset \mathcal{A}(t)$ contains the anchors relative to which node i has an available measurement of type x at time t. Particularly, $\mathcal{A}_i^d$ refers to distance measurements and $\mathcal{A}_i^u$ to angle measurements.

Considering localization in $\mathbb{R}^p$ space, relevant cases rely on $p = 2$ for a bi-dimensional problem and $p = 3$ for a tri-dimensional one. The position of sensor i at time t is indicated as $x_i(t) \in \mathbb{R}^p$, while anchor positions are designated as $a_k(t) \in \mathbb{R}^p$.

Each node has available some measurements with respect to its neighbours, nodes or anchors within a certain distance. Noisy distance measurements at time t are represented as $d_{ij}(t) = d_{ji}(t)$ if they exist between two nodes (i and j) or as $r_{ik}(t)$ if they occur between a node i and anchor k.

Noisy angle measurements between node i and j are represented as a unit-norm vector in the correspondent direction $u_{ij}(t)$. Similarly, noisy angle measurements between node i and anchor k are taken as the unit-norm vector $q_i(t)$.

The problem is to estimate the unknown sensor positions $x(t) = \{x_i(t) : i \in \mathcal{V}\}$ given measurements $\{d_{ij}(t) : i \sim j \in \mathcal{E}^d(t)\} \bigcup \{r_{ik}(t) : i \in \mathcal{V}, k \in \mathcal{A}_i^d(t)\} \bigcup \{u_{ij}(t) : i \sim j \in \mathcal{E}^u(t)\} \bigcup \{q_{ik}(t) : i \in \mathcal{V}, k \in \mathcal{A}_i^u(t)\}$.

A common assumption is to model distance noise with a Gaussian distribution of zero mean and variances σ_{ij}^2 and ς_{ik}^2, for node-node and node-anchor edges respectively. Bearing noise is better suited by a von Mises-Fisher distribution, specifically developed for directional data, with mean direction zero and concentration parameter κ_{ij} and λ_{ik}, for node-node and node-anchor edges. With the additional assumption of independent and identically distributed noise, the maximum likelihood estimator is given by the optimization problem

$$\underset{x}{\text{minimize}} \quad f_{\text{dist}}(x(t)) + f_{\text{ang}}(x(t)) \quad , \tag{1}$$

where the functions are given as

$$f_{\text{dist}}(x(t)) = \sum_{i \sim j \in \mathcal{E}^d(t)} \frac{1}{2\sigma_{ij}^2} \big(\|x_i(t) - x_j(t)\| - d_{ij}(t)\big)^2 + \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{A}_i^d(t)} \frac{1}{2\varsigma_{ik}^2} \big(\|x_i(t) - a_k(t)\| - r_{ik}(t)\big)^2 \tag{2}$$

and

$$f_{\text{ang}}(x(t)) = - \sum_{i \sim j \in \mathcal{E}^u(t)} \left(\kappa_{ij} u_{ij}(t)^T \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \right) - \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{A}_i^u(t)} \left(\lambda_{ik} q_{ik}^T(t) \frac{x_i(t) - a_k(t)}{\|x_i(t) - a_k(t)\|} \right) \tag{3}$$

3. Convex Relaxation

Problem 1 is non-convex in both the distance and angle terms. Non-convexity in distance terms will be relaxed according to the approach in [10], where a similar formulation is considered without the angle measurements. It should be noted that, under this section, we drop the time dependency notation for clarity purposes. First, a new variable y_{ij} is introduced to obtain the following equivalent formulation

$$(\|x_i - x_j\| - d_{ij})^2 = \inf_{\|y_{ij}\| = d_{ij}} \|x_i - x_j - y_{ij}\|^2. \tag{4}$$

The constraint is then relaxed as $\|y_{ij}\| \leq d_{ij}$, with the same reasoning applied to node-anchor edges.

However, the problem is still not convex due to the angular terms. In this work, we propose to relax the angle terms with their proxies from the distance term minimization. Therefore, the denominator $\|x_i - x_j\|$ is approximated by d_{ij} because this is exactly the purpose of the distance terms in the cost function. With a similar reasoning, $x_i - x_j$ is approximated by y_{ij}. Summing up, the nodes' angular terms become $\frac{x_i - x_j}{\|x_i - x_j\|} \approx \frac{y_{ij}}{d_{ij}}$ and respectively for anchors $\frac{x_i - a_k}{\|x_i - a_k\|} \approx \frac{w_{ik}}{r_{ik}}$. The final expression for this convex approximation is then

$$\begin{aligned} \underset{x,y,w}{\text{minimize}} \quad & f_{\text{dist}}(x, y, w) + f_{\text{ang}}(y, w) \\ \text{subject to} \quad & \|y_{ij}\| \leq d_{ij}, \|w_{ik}\| \leq r_{ik} \end{aligned} \quad , \tag{5}$$

where

$$f_{\text{dist}}(x, y, w) = \sum_{i \sim j \in \mathcal{E}^d} \frac{1}{2\sigma_{ij}^2} \|x_i - x_j - y_{ij}\|^2 + \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{A}_i^d} \frac{1}{2\varsigma_{ik}^2} \|x_i - a_k - r_{ik}\|^2 \tag{6}$$

and

$$f_{\text{ang}}(y, w) = - \sum_{i \sim j \in \mathcal{E}_u} \left(\kappa_{ij} u_{ij}^T \frac{y_{ij}}{d_{ij}} \right) - \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{A}_i^d} \left(\lambda_{ik} q_{ik}^T \frac{w_{ik}}{r_{ik}} \right) \quad . \tag{7}$$

We note that the angle terms will bias y_{ij} towards the Cartesian product of spheres. So, not only are we adding additional directional information but also potentially tightening the relaxation gap.

3.1. Simulation Results for Static Networks

We compare the performance of Formulation (5), with the hybrid MLE method relaxed with SDP as presented in [9], designated as SPHL. Simulations are conducted with localizable networks of 10 nodes, of which 6 are unknown and 4 are anchors with known position. They are deployed over an are of $50m^2$, with a minimum separation of $2m$. Given the SDP vulnerability to network configuration, we consider one case where all nodes (unknown and anchors) are randomly deployed and another where anchors are fixed at the corners of a squares, with all unknown nodes inside their convex hull, as observed in Figure 1.

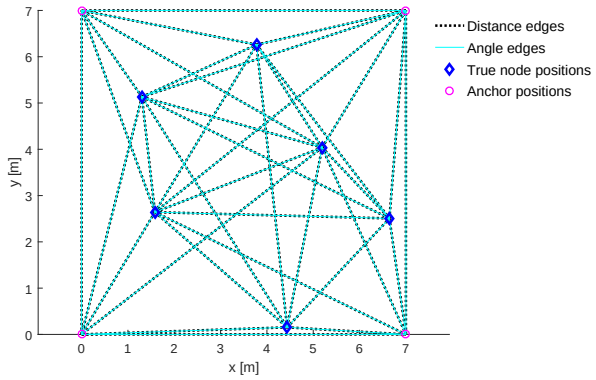

Figure 1: Example of a favourable network configuration, where four anchors are fixed at the corners of a square, with all unknown nodes deployed inside their convex hull.

We also consider all edges to have both ranges and bearing measurements, but this is not a limitation of the method. The disk radius is set to create a highly connected network, with an average of 80% of the total possible edges, as these were the conditions under which SPHL was tested. We introduce noisy distance measurements with a standard deviation (STD) of 0.2m. Regarding bearing measurements it is possible to relate κ with an equivalent σ_{eq} as [11]

$$\sigma_{\text{eq}} = \sqrt{-2\ln\left(1 - \frac{1}{2\kappa} - \frac{1}{8\kappa^2} - \frac{1}{8\kappa^3}\right)}. \quad (8)$$

We consider κ to be 800, which is equivalent to a 2° standard deviation. The metric performance is the Mean Positioning Error, the Euclidean distance between estimated and true positions, averaged over the number of nodes and a number of Monte Carlo (MC) trials. Using only favourable configurations (as in Figure 1) and testing 10 different random positions for unknown nodes, we obtain the results in Figure 2. In this case, both algorithms present a constant error over different configurations and our method consistently outperforms SPHL. To understand the influence of anchor positioning in performance, we use networks with the same number of anchors and unknown nodes as before, except this time all nodes are randomly placed and four anchors randomly chosen amongst them. With the same levels of noise and disk radius, we obtain averaged results for 10 different configuration depicted in Figure 3. The degradation in performance of SPHL is

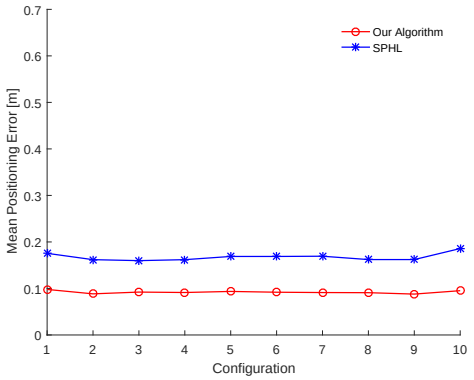

Figure 2: When anchors are uniformly placed on the network boundary, both algorithms present a constant error over different configurations, with our method outperforming SPHL. The resultant MPE is the average of 100 MC trials, where distance measurements noise has a STD of $0.2m$ and bearing noise an equivalent STD of 2°.

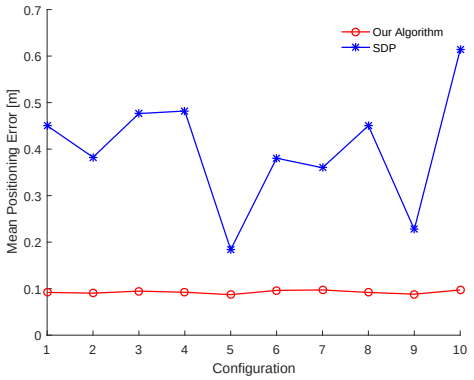

Figure 3: For variable anchor positioning, our algorithm has a constant and better performance than the SPHL, which highly depends on the network configuration. The resultant MPE is the average of 100 MC trials, where distance measurements noise has an STD of $0.2m$ and bearing noise an equivalent STD of 2°.

clear and expected as it includes an SDP relaxation, which is highly susceptible to anchor placement. On the contrary, our relaxation maintains the performance obtained in the first scenario.

4. Dynamic Formulation

While the previous formulation is adequate for static networks, when vehicles perform a trajectory, it is beneficial to take into account different consecutive positions to obtain a better estimate. Therefore, we introduce a time window and consider that each vehicle has access to its own velocity $\beta_i(t)$, regarding node i at time t.

In order to model velocity measurements, we decompose them into a unit vector $v_i(t)$ and magnitude $V_i(t)$. Under this format, we can look at $V_i(t)\Delta T$, the distance travelled by one node during a time interval ΔT, as the distance between two points $x_i(\tau)$ and $x_i(\tau-1)$. Besides, the unit vector $v_i(t)$ can be thought of as the bearing between those same nodes, $x_i(\tau)$ and $x_i(\tau-1)$, usually denoted as heading. Essentially, this means that we have distance and bearing measurements just as in the previous formulation, except that they exist between the same node at different time instants. So, we can model them in a similar way, with $V_i(t)\Delta T$ following a Gaussian distribution with variance σ_i^2 and $v_i(t)$ following a von Mises-Fisher distribution with concentration parameter κ_i.

Therefore, we obtain a formulation with similar structure to Problem 1, which can be relaxed in the same manner. Including the time window, the final problem is given as

$$\begin{aligned} \underset{x,y,w,s}{\text{minimize}} \quad & f_{\text{dist}}(x(t),y(t)) + f_{\text{ang}}(x(t),w(t)) \\ & + f_{\text{vel}}(x(t),s(t)) \\ \text{subject to} \quad & \|y_{ij}(t)\| \leq d_{ij}(t), \quad \|w_{ik}(t)\| \leq r_{ik}(t), \\ & \|s_i(t)\| \leq V_i(t)\Delta T \end{aligned} \quad , \tag{9}$$

where

$$\begin{aligned} f_{\text{vel}}(x(t),s(t)) = & \\ & \sum_{i\in\mathcal{V}} \sum_{\tau=t-T_0+1}^{t} \frac{1}{2\sigma_i^2} \left(\|x_i(\tau) - x_i(\tau-1) - s_i(\tau)\|\right)^2 \\ & - \sum_{i\in\mathcal{V}} \sum_{\tau=t-T_0+1}^{t} \kappa_i \left(v_i(\tau)^T \frac{s_i(\tau)}{V_i(\tau)\Delta T} \right) \end{aligned} \quad , \tag{10}$$

$$\begin{aligned} f_{\text{dist}}(x(t),y(t)) = & \\ & \sum_{i\sim j\in\mathcal{E}_d} \sum_{\tau=t-T_0}^{t} \frac{1}{2\sigma_{ij}^2} \left(\|x_i(\tau) - x_j(\tau) - y_{ij}(\tau)\|\right)^2 \\ & \sum_{i\in\mathcal{V}} \sum_{k\in\mathcal{A}_i^d} \sum_{\tau=t-T_0}^{t} \frac{1}{2\varsigma_{ik}^2} \left(\|x_i(\tau) - a_k(\tau) - w_{ik}(\tau)\|\right)^2 \end{aligned} \tag{11}$$

and

$$\begin{aligned} f_{\text{ang}}(x(t),w(t)) = & \\ & - \sum_{i\sim j\in\mathcal{E}_u} \sum_{\tau=t-T_0}^{t} \left(\kappa_{ij} u_{ij}(\tau)^T \frac{y_{ij}(\tau)}{d_{ij}(\tau)} \right) \\ & - \sum_{i\in\mathcal{V}} \sum_{k\in\mathcal{A}_i^u} \sum_{\tau=t-T_0}^{t} \left(\kappa_{ik} q_{ik}(\tau)^T \frac{w_{ik}(\tau)}{r_{ik}(\tau)} \right). \end{aligned} \tag{12}$$

4.1. Reformulation

Considering $x(\tau) = \{x_i(\tau)\}_{i\in\mathcal{V}}$, $x = \{x(\tau)\}_{t-T_0\leq\tau\leq t}$, $y(\tau) = \{y_{ij}(\tau)\}_{i\sim j}$ and $y = \{y(\tau)\}_{t-T_0\leq\tau\leq t}$. Finally, let Σ_N be the diagonal matrix of $\frac{1}{\sigma_{ij}}$. It is now possible to write the first term of (11) as

$$f_{\text{dist,nodes}}(x,y) = \frac{1}{2}\|\Sigma_N A x - \Sigma_N y\|^2 \quad , \tag{13}$$

where matrix A is the Kronecker product of identity matrix of dimension T_0 with the result from the Kronecker product of C, the arc-node incidence matrix of the network, with identity matrix of dimension p, that is

$$A = I_{T_0} \otimes (C \otimes I_p) = (I_{T_0} \otimes C) \otimes I_p. \tag{14}$$

Essentially, this extends matrix C along the dimension of the problem (p) and time window (T_0), assuming the set of edges remains constant over T_0.

In the same way, taking $s(\tau) = \{s_i(\tau)\}_{i\in\mathcal{V}}$, $s = \{s(\tau)\}_{t-T_0\leq\tau\leq t}$ and Σ_V to be the diagonal matrix of $\frac{1}{\sigma_i}$, the first term of (10) may be written as

$$f_{\text{dist,vel}}(x,s) = \frac{1}{2}\|\Sigma_V N x - \Sigma_V s\|^2 \tag{15}$$

where $N = C_{\text{vel}} \otimes I_p$. However, it should be clearly noted that, in this formulation, the set of edges has changed, hence the different designation C_{vel}. Each node at time t has an edge with its position at $t-1$ and $t+1$, except for the first and last instants of the time window. The last distance term, concerning anchor-node measurements, may be reformulated as

$$f_{\text{dist,anchors}}(x,w) = \frac{1}{2}\|\Sigma_A E x - \Sigma_A \alpha - \Sigma_A w\|^2 \tag{16}$$

where the concatenated vectors are composed as $\alpha_i(\tau) = \{a_{ik}(\tau)\}_{k\in\mathcal{A}_i}$, $\alpha(\tau) = \{\alpha_i(\tau)\}_{i\in\mathcal{V}}$, $\alpha =$

$\{\alpha(\tau)\}_{t-T_0\leq\tau\leq t}$ and $w_i(\tau)=\{w_{ik}(\tau)\}_{k\in\mathcal{A}_i}$, $w(\tau)=\{w_i(\tau)\}_{i\in\mathcal{V}}$, $w=\{w(\tau)\}_{t-T_0\leq\tau\leq t}$. E is a selector matrix, indicating which node has a measurement relative to each anchor and Σ_A the diagonal matrix of $\frac{1}{\varsigma_{ik}}$.

Finally, the angle measurements are grouped as $\tilde{u}_{ij}(\tau)=\frac{\kappa_{ij}u_{ij}(\tau)}{d_{ij}(\tau)}$, $\tilde{q}_{ik}(\tau)=\frac{\lambda_{ik}q_{ik}(\tau)}{r_{ik}(\tau)}$ and $\tilde{v}_i(\tau)=\frac{\kappa_i v_i(\tau)}{V_i(\tau)\Delta T}$. Following the same reasoning as before, $u(\tau)=\{\tilde{u}_{ij}(\tau)\}_{i\sim j}$, $v(\tau)=\{\tilde{v}_i(\tau)\}_{i\in\mathcal{V}}$ and $q_i(\tau)=(\tilde{q}_{ik}(\tau))_{k\in\mathcal{A}_i}$, $q(\tau)=\{q_i(\tau)\}_{i\in\mathcal{V}}$. Then, concatenating along the time window , $u=\{u(\tau)\}_{t-T_0\leq\tau\leq t}$, $v=\{v(\tau)\}_{t-T_0\leq\tau\leq t}$ and $q=\{q(\tau)\}_{t-T_0\leq\tau\leq t}$. It follows that

$$f_{ang}(y,w,s)=-u^Ty-q^Tw-v^Ts. \tag{17}$$

Considering the definitions in (13), (15), (16) and (17), Problem (9) is reformulated as

$$\begin{aligned}\underset{x,y,w,s}{\text{minimize}}\quad & f_{\text{dist,nodes}}(x,y)+f_{\text{dist,anchors}}(x,w)+\\ & f_{\text{dist,vel}}(x,s)+f_{ang}(x,y,w)\\ \text{subject to}\quad & \|y_{ij}(t)\|\leq d_{ij}(t),\quad \|w_{ik}(t)\|\leq r_{ik}(t),\\ & \|s_i(t)\|\leq V_i(t)\Delta T\end{aligned} \tag{18}$$

Introducing a variable $z=(x,y,w,s)$ and defining $\mathcal{Z}=\{z:\|y_{ij}(t)\|\leq d_{ij}(t), i\sim j\in\mathcal{V};\|w_{ik}(t)\|\leq r_{ik}(t), i\in\mathcal{V}, k\in\mathcal{A}_i;\|s_i(t)\|\leq V_i(t)\Delta T, i\in\mathcal{V}\}$, Problem (18) can be written in a quadratic form as

$$\begin{aligned}\underset{z}{\text{minimize}}\quad & \frac{1}{2}z^TMz-b^Tz\\ \text{subject to}\quad & z\in\mathcal{Z}\end{aligned} \tag{19}$$

where $M=M_1+M_2+M_3$, with

$$M_1=\begin{bmatrix}A^T\Sigma_N\\-\Sigma_NI\\0\\0\end{bmatrix}\begin{bmatrix}\Sigma_NA & -\Sigma_NI & 0 & 0\end{bmatrix} \tag{20}$$

$$M_2=\begin{bmatrix}E^T\Sigma_A\\0\\-\Sigma_AI\\0\end{bmatrix}\begin{bmatrix}\Sigma_AE & 0 & -\Sigma_AI & 0\end{bmatrix} \tag{21}$$

$$M_3=\begin{bmatrix}N^T\Sigma_V\\0\\0\\-\Sigma_VI\end{bmatrix}\begin{bmatrix}\Sigma_VN & 0 & 0 & -\Sigma_VI\end{bmatrix} \tag{22}$$

and $b=b_1+b_2$, with

$$b_1=\begin{bmatrix}E^T\Sigma_A\\0\\-\Sigma_AI\\0\end{bmatrix}\Sigma_A\alpha \tag{23}$$

and

$$b_2=\begin{bmatrix}0\\u\\q\\v\end{bmatrix}. \tag{24}$$

5. Distributed Implementation

5.1. FISTA method

The method chosen to minimize Problem (19) is the *Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)*, found in [12], which is an extension of the gradient method. The problem is defined as

$$\underset{x}{\text{minimize}}\, f(x)=g(x)+h(x) \tag{25}$$

where $g(x):\mathbb{R}^n\to\mathbb{R}$ is a smooth convex function, continuously differentiable with Lipschitz continuous gradient L and $h(x)$ is a closed and convex function with an inexpensive prox_{th} operator. Under these assumptions, FISTA's solution for Problem (25) is given iteratively as

$$\begin{aligned}y&=x^{(k-1)}+\frac{k-2}{k+1}\Big(x^{(k-1)}-x^{(k-2)}\Big)\\ x^{(k)}&=\text{prox}_{t_kh}(y-t_k\nabla g(y))\end{aligned} \tag{26}$$

This method presents a convergence rate $f(x^{(k)})-f^*$ (where $f^*$is the optimal value) of $\mathcal{O}(1/k^2)$, for a fixed step size of $t_k=1/L$.

5.2. Implementation

For Problem (19) to be solved with FISTA, it is necessary to define $g(x)$ and $h(x)$, along with prox_h and t_k. So, $g(x)$ it the quadratic cost function $\frac{1}{2}z^TMz-b^Tz$. It is noted that a function is said to have a Lipschitz Continuous Gradient if the following condition is true

$$\|\nabla f(x)-\nabla f(y)\|\leq L\|x-y\| \tag{27}$$

and quadratic functions have a Lipschitz continuous gradient. Besides, the required gradient $\nabla g()$ is straightforward, given as $\nabla g()=Mz-b$.

Term $h(x)$ corresponds to the indicator function $I_{\mathcal{Z}}(z)$, defined as

$$I_{\mathcal{Z}}(z)=\begin{cases}0, z\in\mathcal{Z}\\+\infty, z\notin\mathcal{Z}\end{cases} \tag{28}$$

and to obtain prox_h of an indicator function, we refer to [13]. When $h(x)$ is indicator function of closed convex set $\mathcal{Z}$, then prox_h is the projection of z on set $\mathcal{Z}$, $P_{\mathcal{Z}}(x)$, defined as

$$P_{\mathcal{Z}}(z)=\underset{x\in\mathcal{Z}}{\text{argmin}}\,\|x-z\|^2, \tag{29}$$

which for our constraints with general structure $\{z:\|z\|\leq d\}$ is implemented as

$$P_{\mathcal{Z}}(z)=\begin{cases}z, & \text{if}\quad \|z\|\leq d\\ \frac{z}{\|z\|}d, & \text{if}\quad \|z\|>d\end{cases}. \tag{30}$$

Finally, it is necessary to obtain a value for $1/L$, where L is the Lipschitz constant of function $\frac{1}{2}z^TMz-b^Tz$. Resorting to [10], we obtain

an upper-bound for L as

$$L \leq \frac{1}{\sigma_N^2} 2\delta_{\max} + \frac{1}{\sigma_A^2} \max_{i \in \mathcal{V}} |\mathcal{A}_i| + \frac{1}{\sigma_V^2} 2\delta^1_{\max}(T) + K \quad , \tag{31}$$

where $\delta_{\max}$ is the maximum node degree of the network and $\delta^V_{\max}$ is either 2 for any $T > 2$, 1 for $T = 2$ and 0 for no time window ($T = 1$). The term $\max_{i \in \mathcal{V}} |\mathcal{A}_i|$ corresponds to maximum number of anchors connected to a node. Finally, $K = \frac{1}{\sigma_N^2} + \frac{1}{\sigma_A^2} + \frac{1}{\sigma_V^2}$, where σ_N, σ_A and σ_V are the maximum values of matrices Σ_N, Σ_A and Σ_V, respectively.

In the end, we obtain a distributed method, as evidenced by the explicit formulation in Algorithm 1. We define $C_{(i\sim j,i)}$ as the element of C on row referring to edge $i \sim j$ and column referring to node i; N_i as the sub-matrix of N, whose rows correspond to node i; $\mathbf{s}_i$ and $\mathbf{x}_i$ as the concatenation of $s_i(\tau)$ and $x_i(\tau)$ over the time window; $\mathbf{w}_i$ and $\mathbf{y}_i$ as the concatenation of $w_{ik}(\tau)$ and $y_{ij}(\tau)$ over the respective edges and time window. Besides, the projections for each component are defined as $\mathcal{Y}_{ij}(t) = \{y_{ij}(t) : \|y(t)\| \leq d_{ij}(t)\}$, $\mathcal{W}_{ik}(t) = \{w_{ik}(t) : \|w_{ik}(t)\| \leq r_{ik}(t)\}$ and $\mathcal{S}_i(t) = \{s_i(t) : \|s_i(t)\| \leq v_i(t)\}$. The elements $F_1(\tau)$ trough $F_4(\tau)$ are given as

$$F_1(\tau) = \frac{L - \sum_{i\sim j \in \mathcal{E}_i} 1/\sigma^2_{ij}(\tau) - \sum_{k \in \mathcal{A}_i} 1/\varsigma^2_{ik}(\tau)}{L} \tag{32}$$

$$F_2(\tau) = \frac{1}{L\sigma^2_{ij}} \sum_{i\sim j \in \mathcal{E}_i} (x^\kappa_j(\tau) + C_{(i\sim j,i)}(\tau) y^\kappa_{ij}(\tau)) \tag{33}$$

$$F_3(\tau) = \frac{1}{L\varsigma^2_{ik}} \sum_{k \in \mathcal{A}_i} (w^\kappa_{ik}(\tau) + \alpha_k(\tau)) \tag{34}$$

$$F_4(\tau) = \frac{1}{L} \Sigma^2_V (N_i^T \mathbf{s}^\kappa_i - N_i^T N_i \mathbf{x}^\kappa_i) \tag{35}$$

Recalling that we stay under a convex formulation, it is noted that z^0 may be initialized with any value.

Algorithm 1 Hybrid convex localization

Input: L;
$z^0 = (x^0, y^0, w^0, s^0)$;
$\{d_{ij}(t), u_{ij}(t) : i \sim j \in \mathcal{E}\}$; $\{r_{ik}(t), q_{ik}(t) : i \in \mathcal{V}, k \in \mathcal{A}\}$; $\{v_i(t) \in \mathcal{V}\}$
Output: $\hat{x} = \{\hat{x}_i\}$

for all t **do**
  $\kappa = 1$; $z^1 = z^0$
  while some stopping criterion is not met, each node i **do**
    $z^\kappa_i = z^\kappa_i + \frac{\kappa-1}{\kappa}(z^\kappa_i - z^{\kappa-1}_i)$
    for all $t - T_0 \leq \tau \leq t$ **do**
      $x^{\kappa+1}_i(\tau) = x^\kappa_i(\tau) F_1(\tau) + F_2(\tau) + F_3(\tau) + F_4(\tau)$
    end for
    for all $i \sim j \in \mathcal{E}_i$ and $t - T_0 \leq \tau \leq t$ **do**
      $y^{\kappa+1}_{ij}(\tau) = P_{\mathcal{Y}_{ij}(\tau)}\Big(\frac{L - 1/\sigma^2_{ij}}{L} y^\kappa_{ij}(\tau) + \frac{1}{L\sigma^2_{ij}} C_{(i\sim j,i)}(\tau)(x^\kappa_i(\tau) - x^\kappa_j(\tau)) + \frac{\tilde{u}_{ij}(\tau)}{L}\Big)$
    end for
    for all $k \in \mathcal{A}_i$ and $t - T_0 \leq \tau \leq t$ **do**
      $w^{\kappa+1}_{ik}(\tau) = P_{\mathcal{W}_{ik}(\tau)}\Big(\frac{L - 1/\varsigma^2_{ik}}{L} w^\kappa_{ik}(\tau) + \frac{1}{L\varsigma^2_{ik}}(x^\kappa_i(\tau) - a_k(\tau)) + \frac{\tilde{q}_{ik}(\tau)}{L}\Big)$
    end for
    if $T_0 > 1$ **then**
      for all $t - T_0 + 1 \leq \tau \leq t$ **do**
        $s^{\kappa+1}_i(\tau) = P_{\mathcal{S}_i}\Big(\frac{L - 1/\sigma^2_{ij}}{L} + \frac{1}{L\sigma^2_i}(x^\kappa_i(\tau) - x^\kappa_i(\tau - 1)) + \frac{\tilde{v}_i(\tau)}{L}\Big)$
      end for
    end if
    $z^{\kappa+1}_i = (\mathbf{x}^{\kappa+1}_i, \mathbf{y}^{\kappa+1}_i, \mathbf{w}^{\kappa+1}_i, \mathbf{s}^{\kappa+1}_i)$
    $\kappa = \kappa + 1$
    broadcast x_i to all neighbours
  end while
  return $\hat{x} = \{\mathbf{x}^{\kappa+1}_i\}$
end for

5.3. Results

We compare our algorithm with EKF, a state-of-the-art method, with an equivalent data model. EKF requires initialization, for which we use the true position with a STD of $2m$. We also optimize its parameters for the considered trajectories, through a grid search method.

A common trajectory in AUV missions is the Lawnmower, as depicted in Figure 4. Considering noise levels obtainable with current instrumentation, we define $\sigma_d = 0.5m$ (for all distance measurements), $\sigma_v = 0.1m/s$ (for all speed measurements), $\kappa_a = 1000$ (for all bearing measurements) and $\kappa_v = 1000$ (for all heading measurements). Figure 5 represents the Mean Positioning Error for each algorithm along the trajectory, denoted as Mean Navigation Error. It is clear that, unlike EKF, our method is able to keep a constant error regardless of the vehicles' movement. While EKF slightly outperforms our method during linear parts, the inverse is observed for the remaining trajectory.

When we consider a more realistic case of outliers in measurements, the difference in performance increases. This is implemented by introducing a distance measurement with value $5d$ with a probability of 10%, where d is the true distance. Figure 6 contains the result of such experiment, where ranges between the outer node and one of the anchors are contaminated with outliers, evidenced by the peaks in position estimates. The line in red, representing results from our algorithm, presents smaller variations with outliers and does not require as much time as EKF (in blue) to recover. The difference between them also seems to accentuate during non

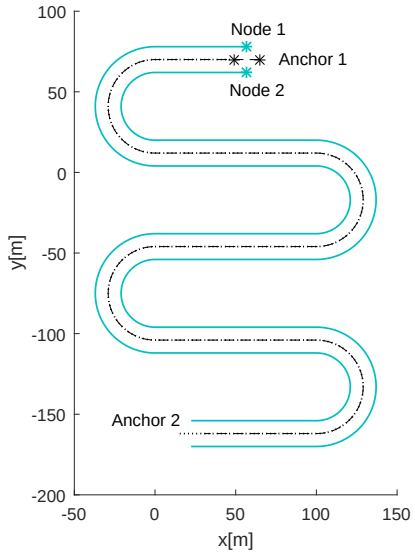

Figure 4: Lawnmower trajectory with two nodes, in solid blue line, travelling on both sides of two anchors, in dashed and dotted black lines. Note that both anchors describe similar trajectories but one is "behind" the other. The begging of each trajectory is marked with a star.

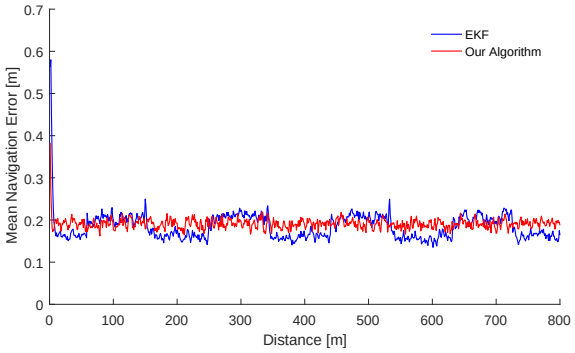

Figure 5: Comparison of Mean Navigation Error between EKF and our algorithm, during a lawnmower trajectory, averaged over 100 MC trials. The curved segments of the trajectory are evidenced by the higher error in EKF, where our method outperforms it. For the linear parts EKF shows a better accuracy.

linear trajectory parts.

We further extend this scenario for the situation where all range measurements related to the outer node are subject to outliers, not only with both anchors but also with the second node. The results, expressed in Figure 7, show an even greater difference between both performances, with our algorithm revealing superior robustness.

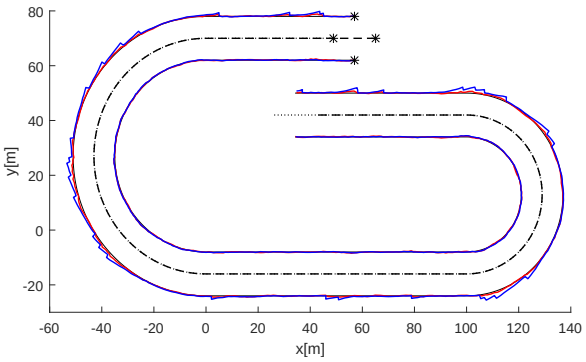

Figure 6: Distances between one of the anchors (dashed lines) and the outer node, are contaminated with outliers, evidenced by the peaks in both estimates. EKF, in blue, presents larger peaks, showing to be far less robust than our algorithm, in red.

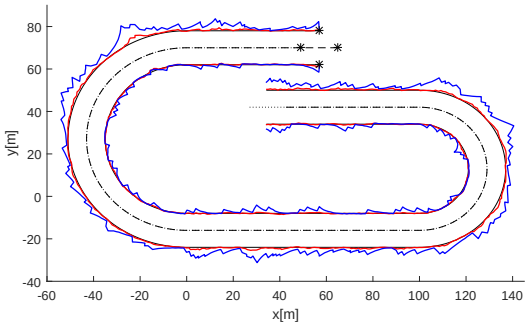

Figure 7: Distances between the outer node and all the other vehicles are contaminated with outliers. EKF, in blue, is affected to higher a extent by this problem and takes more time to recover from it than our algorithm, in red. The results are obtained during a lap trajectory performed by two anchors, in dashed lines, and two nodes, in black lines, with initial positions marked as a star.

6. Parameter Estimation

Until this point, the parameters σ and κ of error distributions have been assumed to be known a priori. In a real applications, these parameters could be estimated from previous experiments. However, this can also be included in the algorithm so that no previous knowledge is necessary.

Maximum likelihood estimators for the parameters of each distribution are easily found in the literature. Given a random variable X normally distributed with unknown mean and variance σ^2, the MLE for the latter parameter is given as [14]

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2. \tag{36}$$

For κ estimation, authors in [15], propose the use

of

$$\hat{\kappa} = \frac{\bar{\gamma}p - \bar{\gamma}}{1 - \bar{\gamma}^2}, \tag{37}$$

where x_i are independent and identically distributed sample unit vectors drawn from a Von Mises-Fisher distribution $vMF(\mu, \kappa)$, p is the dimension of x_i and $\bar{\gamma} = \frac{\| \sum_{i=1}^{n} x_i \|}{n}$.

6.1. Implementation

Evidently, there is no access to true values of our measurements to accurately estimate κ and σ. However, after each computation of position estimates by the algorithm, it is possible to compute estimated distances, angles and velocities which can be used as the "true value". So, exact values of κ and σ are not to be expected but if the position estimates are close to the true ones, then a similar value should be obtained. However, note that this two estimators influence each other and poor estimates in one will affect the other.

Denoting by $\hat{x}$ the value obtained from our position estimation, it is possible to compute the resulting distances and bearings as

$$\hat{d}_{ij}(t) = \|\hat{x}_i(t) - \hat{x}_j(t)\| \tag{38}$$

and

$$\hat{u}_{ij}(t) = \frac{\hat{x}_i(t) - \hat{x}_j(t)}{\|\hat{x}_i(t) - \hat{x}_j(t)\|}. \tag{39}$$

Given the obtained measurements d_{ij}, Equation (44) can now be used for variance estimation as

$$\hat{\sigma}_{ij}^2(t) = \frac{1}{t} \sum_{\tau=1}^{t} (d_{ij}(\tau) - \hat{d}_{ij}(\tau))^2 \tag{40}$$

and, given bearing measurements u_{ij}, Equation (45) is used for concentration parameter estimation as

$$\hat{\kappa}_{ij}(t) = \frac{\bar{\gamma}_{ij}(t)p - \bar{\gamma}_{ij}(t)}{1 - \bar{\gamma}_{ij}(t)^2} \quad , \tag{41}$$

where $\bar{\gamma}_{ij}(t) = \frac{\| \sum_{\tau=1}^{t} u_{ij}(\tau) - \hat{u}_{ij}(\tau) \|}{t}$. The same reasoning can be applied to node-anchor edges, with similar resulting equations.

Velocity measurements (β_i) require a more careful approach, given that a similar process using $\hat{\beta}_i(t)\Delta T = x_i(t) - x_i(t-1)$ does not produce good results. This problem is considered in [16] and the proposed solution is here implemented as $\hat{\beta}_i(t)\Delta T = \frac{5(x_i(t-3) - x_i(t-5))}{32} + \frac{4(x_i(t-2) - x_i(t-6))}{32} + \frac{(x_i(t-1) - x_i(t-7))}{32}$.

It should be noted that this approach introduces a time lag that should be taken into account during implementation. Recalling that $\hat{\beta}_i(t)$ is decomposed as $\hat{\beta}_i(t) = \hat{v}_i(t)\hat{V}_i(t)$, estimates for variance and concentration parameter are given as

$$\hat{\sigma}_i^2(t) = \frac{1}{t} \sum_{\tau=1}^{t} (V_i(\tau) - \hat{V}_i(\tau))^2 \tag{42}$$

and

$$\hat{\kappa}_i(t) = \frac{\bar{\gamma}_i(t)p - \bar{\gamma}_i(t)}{1 - \bar{\gamma}_i^2(t)} \quad , \tag{43}$$

with $\bar{\gamma}_i(t) = \frac{\| \sum_{\tau=1}^{t} v_i(\tau) - \hat{v}_i(\tau) \|}{t}$.

6.2. Simulation results

In Figure 10, we present the evolution of $\hat{\sigma}_d$ for all the distance measurements, where the true σ_d was set to $0.5m$. The estimates converge to σ_d, as desired, but it should be noted that a considerable number of measurements is necessary before this happens. Consequently, we choose a default initial value for all σ and κ (different from the real one), to be used during the first time steps of the trajectory. After that point, the estimated values are used instead. The introduction of parameter estimation does not seem to considerably decrease performance, when compared with an optimal case of known values, as Figure 11 expresses.

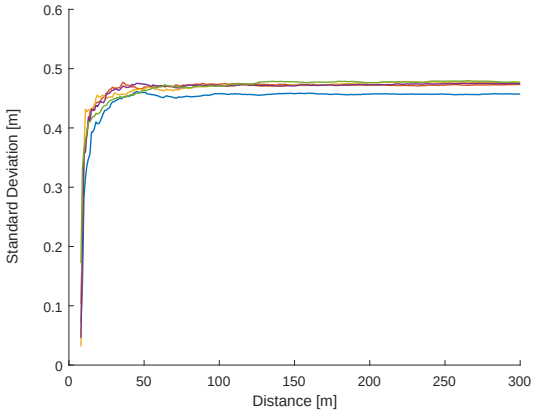

Figure 8: Evolution of estimates of $\hat{\sigma}_d$ for distance measurements. After some time, the estimates converge to values close to the true $\sigma_d = 0.5m$. The different lines correspond to σ_{ij} for different edges, all with the same value of STD.

7. Conclusions

In this work, we formulated the MLE for generic network localization problem, both in static and dynamic contexts, including range and bearing measurements between nodes. The introduction of angular information with an appropriate relaxation allowed the method to overcome some challenges usually encountered in similar relaxations for distances. For example, anchors are not required to be deployed at the boundary of the network, unlike many other approaches. When compared with an hybrid state-of-the-art SDP relaxation, our algorithm reveals superior accuracy and less susceptibility to network configuration.

While it is not necessary that all nodes measure bearings, it must be emphasized that each bearing

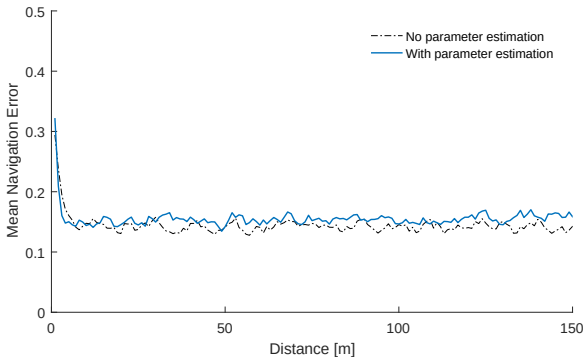

Figure 9: Comparison of performance between the method with parameter estimation and the optimal case, where the real parameter values are provided. Although the latter stays slightly below the former for most of the trajectory, the difference is almost negligible.

should have a correspondent distance measurement. Of course, range measurements can exist alone for some pairs of vehicles. This is due to our relaxation, causing these terms to be dependent on formulation for the ranges.

The method is extended to dynamic networks, in a horizon-based approach, and proven to be distributed. Comparison with EKF, the centralized state-of-the-art, showed that, in terms of accuracy, the performance is competitive for cases with variable motion patterns, where for mostly linear trajectories the EKF still slightly outperforms our distributed estimator. Nonetheless, EKF requires a good initialization for convergence, whereas our distributed convex method is proven to converge, without initialization requirements. Furthermore, the accuracy of our algorithm is constant throughout the trajectory, thus being more predictable than EKF. More importantly, our algorithm shows more robustness in the presence of outliers in measurements, which is a known issue in real-life scenarios.

Besides, with the parameter estimation process, no parameter tuning is required, so that the algorithm can be easily applied to new environments or technologies. However, it should be noted that this is a sensible step where abnormal values could start a chain of wrong position estimates, leading to equally wrong parameter estimates and so on. Although this was never verified in testing, even with discrepant initial values, it was also not proven otherwise, so that a formal prove would be desirable in a future approach. Our simulations show that the method maintains its performance, despite the estimated parameters and regardless of the initializations chosen.

References

[1] T. Yoo. DVL/RPM based velocity filter aiding in the underwater vehicle integrated inertial navigation system. *Journal of Sensor Technology*, 4, 2014.

[2] P. A. Miller, J. A. Farrell, Y. Zhao, and V. Djapic. Autonomous underwater vehicle navigation. *IEEE Journal of Oceanic Engineering*, 35(3):663–678, July 2010.

[3] Y. Ye P. Biswas. A distributed method for solving Semidefinite Programs arising from Ad Hoc wireless sensor network localization. *ACM Transactions on Sensor Networks*, 2003.

[4] P. Tseng. Second order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185, 2007.

[5] J. Nie. Sum of squares method for sensor network localization. *Computational Optimization and Applications*, 2006.

[6] David F. CrouseE, Richard W. Osborne, Krishna Pattipati, Peter Willet, and Yaakov Bar-Shalom. Efficient 2D sensor location estimation using targets of opportunity. *Jounal of Advances in Information Fusion*, 8, jun 2013.

[7] Ming-Yong LIU, Wen-Bai LI, and Xuan PEI. Convex optimization algorithms for cooperative localization in autonomous underwater vehicles. *ACTA AUTOMATICA SINICA*, 2010.

[8] P. Biswas, H. Aghajan, and Y. Ye. Semidefinite programming algorithms for sensor network localization using angle information. *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 2005.

[9] H. Naseri and V. Koivunen. Convex relaxation for maximum-likelihood network localization using distance and direction data. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, June 2018.

[10] C. Soares, J. Xavier, and J. Gomes. Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements. *IEEE Transactions on Signal Processing*, 63(17):4532–4543, Sept 2015.

[11] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. John Wiley and Sons, Inc., 2000.

[12] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 693–696, April 2009.

[13] L. Vandenberghe. Proximal gradient method. `http://www.seas.ucla.edu/~vandenbe/236C/lectures/proxgrad.pdf`, 2013. Accessed: 2018-08-10.

[14] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Inc., 2003.

[15] Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.

[16] C. Soares, J. Gomes, B. Q. Ferreira, and J. P. Costeira. LocDyn: Robust distributed localization for mobile underwater networks. *IEEE Journal of Oceanic Engineering*, 42(4):1063–1074, Oct 2017.