Improving Ranging-Based Location Estimation
with Rigidity-Constrained CRLB-Based Motion Planning

Justin Cano and Jerome Le Ny

Abstract— **Ranging systems can provide inexpensive, accurate, energy- and computationally-efficient navigation solutions for mobile robots. This work focuses on location and pose estimation in ranging networks composed of anchors with known positions as well as mobile robots modeled as rigid bodies, each carrying multiple tags to localize. Noisy distance measurements can be obtained between a subset of the nodes (anchors and tags), and the robots can move in order to improve the accuracy of the localization process, which depends on the geometry of the network. We propose a method to find trajectories for the robots leading to configurations that locally optimize this localization accuracy. These trajectories minimize a cost function based on the *constrained* Cramér-Rao Lower Bound (CRLB), where the constraints capture the information about the known distances between tags carried by the same robot. A primal-dual optimization scheme aims to enforce these distance constraints between tags in the motion planner as well. An important feature of the approach is that the gradient terms necessary to plan the motion can be computed essentially in closed form, thereby simplifying the implementation. We compare the proposed method to a naive two-stage algorithm that optimizes the positions and orientations of the robots independently. Simulation results illustrate the benefits of using the constrained optimization approach.**

I. INTRODUCTION

Mobile robots require accurate, real-time, computationally and energy efficient pose (position and orientation) estimation solutions in order to accomplish their tasks. Many sensing-modalities can be used to design these navigation systems, with systems based on radio-frequency (RF) communications (such as the GPS) or computer vision being currently among the most popular [1], [2]. One can also classify these systems based on the type of geometric information extracted from the measurements. This paper focuses on the use of ranging technologies, i.e., providing only distance measurements (no bearings), as can be obtained for example from RF devices such as Ultra-Wide Band (UWB) transceivers. UWB-based distance measurements can be used for example to design inexpensive yet highly accurate indoor navigation systems that require little processing power, in comparison for example to vision-based systems [3]–[8].

A ranging-based navigation system involves a network of nodes capable of measuring their relative distances, e.g., via signal time-of-flight measurements in the case of UWB nodes. Some of these nodes, called *anchors*, know their absolute positions, and the goal of the navigation system is to estimate the location of the remaining nodes, called *tags*.

The relative positions between the nodes strongly influence the accuracy with which this localization problem can be solved, a phenomenon called *Dilution Of Precision* (DOP) in the navigation literature [2, Chap. 7], which can also be given a statistical interpretation via the Cramér Rao Lower Bound (CRLB) [9, p. 184] on the variance of the tags' position estimates [10]. As a result, the node localization problem becomes coupled with the node placement problem, or, in the case of mobile robots trying to maintain an accurate position estimate as they move through their environment, with the motion planning problem. One can then use performance measures computed from the CRLB to solve localizability-constrained motion planning problems [11]. This is similar in spirit to other planning problems under uncertainty in robotics, see, e.g., [12]–[15].

We consider here scenarios where a number of robots moving in the plane carry *multiple* ranging tags and need to solve pose estimation and pose optimization problems on SE(2) while moving, instead of just a localization problem for the tags, as done in [11] for example. In other words, the motion planning problem for the nodes to maintain their localization accuracy should take into account the rigid constraints that link the tags carried by the same robot, both in the definition of the cost function and in the design of the motion strategies. One possible approach is to directly define a CRLB on $SE(2)$ [16], [17] in order to build a cost function capturing localization performance. However, using explicit parametrizations of $SE(2)$ to define the robot configurations leads to complex expressions that are difficult to optimize.

Instead, as explained in section II, we develop here an approach based on the *constrained* CRLB [18], adding distance constraints between the tags within each rigid body to define an appropriate cost function capturing the ability of the robots to localize. We consider here the two-dimensional case for simplicity, but the extension to $3D$ is straightforward. We then develop in Section IV a gradient based motion planning strategy for the robots that also takes these constraints into account, via a primal-dual optimization algorithm. The robot pose estimates are implicitly optimized by integrating these constraints in the motion planner. For comparison, we also design in section III a naive motion planner that simplifies the problem by using a two-step approach, first optimizing the location of one tag on a robot, before optimizing the robot's orientation. The simulations presented in V and in the accompanying video illustrate the benefits of using the more sophisticated approach.

II. BACKGROUND AND PROBLEM STATEMENT

As illustrated on Fig. 1, we fix a $2D$ global frame $\mathcal{G} = (0, \vec{x}, \vec{y})$ and consider a network of $N = n + m$ agents (nodes), each equipped with a distance measuring device such as an UWB transceiver, whose positions are denoted $\mathbf{p}_i = [x_i, y_j]^\top$, $1 \leq i \leq N$. We known the exact positions of the first n agents denoted A_i, $1 \leq i \leq n$, forming a set $\mathcal{A}$ and called *anchors*. The other m agents, called *tags*, need to be localized. They form a set $\mathcal{T}$ and are denoted T_j, $n < j \leq N$. Moreover, the tags are all carried by s rigid bodies (or robots) S_k, $1 \leq k \leq s$, whose configuration in SE(2) can be described by their pose, i.e., the position of their center of mass $\mathbf{p}_k^G = [x_k^G, y_k^G]^\top$ and their orientation θ_k. We assume that *each rigid body is carrying at least two tags* and that *all distances between tags on the same rigid body are known.*
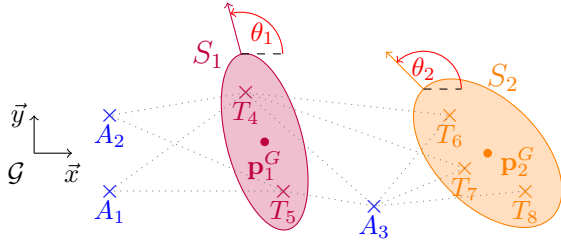

Fig. 1: Configuration with three anchors and two rigid bodies.

We suppose that certain pairs $\{i, j\}$ of agents (anchors and tags) are able to measure their relative distances $d_{ij} = ||\mathbf{p}_{ij}||_2$, where $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$. These distance measurements can then be used to estimate the tags' positions in the global frame $\mathcal{G}$. Our goal is to move the S_k robots from their starting pose in order to improve the accuracy of the tag localization process and hence *in fine* reduce the uncertainty in the estimates of the poses. The motion planning problem is stated in Section II-C, after we introduce the necessary background to evaluate this uncertainty quantitatively.

A. Equality Constrained Cramer-Rao Lower Bound

Let $\mathbf{x} \in \mathbb{R}^p$ be a parameter vector and $\mathbf{Y} \in \mathbb{R}^q$ a random observation vector, whose Probability Density Function (PDF) $f(\mathbf{y}; \mathbf{x})$ depends on $\mathbf{x}$. We suppose f to be sufficiently differentiable, as a function of $\mathbf{x}$. We define the $p \times p$ Fisher Information Matrix (FIM) of this PDF as follows:

$$\mathbf{F}(\mathbf{x}) = \mathbb{E} \left\{ \frac{\partial \ln f(\mathbf{Y}; \mathbf{x})}{\partial \mathbf{x}}^\top \frac{\partial \ln f(\mathbf{Y}; \mathbf{x})}{\partial \mathbf{x}} \right\}, \quad (1)$$

with $\mathbb{E} \{\cdot\}$ the expectation operator.

Theorem 1 (Equality-constrained CRLB [18]): Let $\mathbf{f}_c : \mathbb{R}^p \to \mathbb{R}^c$, for $c < p$, $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^p, \mathbf{f}_c(\mathbf{x}) = \mathbf{0}\}$, and consider a parameter vector $\mathbf{x} \in \mathcal{C}$ and a random vector $\mathbf{Y} \in \mathbb{R}^q$. Let $\mathbf{y} \mapsto \hat{\mathbf{x}}(\mathbf{y})$ be an unbiased estimator of $\mathbf{x}$, i.e., $\mathbb{E} \{\hat{\mathbf{x}}(\mathbf{Y})\} = \mathbf{x}$, computed from $\mathbf{Y}$ and satisfying the constraints $\hat{\mathbf{x}} \in \mathcal{C}$. Assume that $\hat{\mathbf{x}}$ has finite covariance. Denote the Jacobian of the constraint function $\frac{\partial \mathbf{f}_c}{\partial \mathbf{x}}$ and introduce a matrix $\mathbf{A}$ with p

rows and columns spanning the nullspace of $\frac{\partial \mathbf{f}_c}{\partial \mathbf{x}}$, We then have the following inequality :

$$\Sigma := \mathbb{E} \left\{ (\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\top \right\} \succeq \mathbf{A} \left(\mathbf{F}_c \right)^\dagger \mathbf{A}^\top =: \mathbf{B}_c \quad (2)$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse [19, p.21] and $\mathbf{F}_c := \mathbf{A}^\top \mathbf{F} \mathbf{A}$ is the *constrained* FIM. Here, the notation $\mathbf{M} \succeq \mathbf{M}'$ for $\mathbf{M}, \mathbf{M}'$ square symmetric matrices means that $\mathbf{M} - \mathbf{M}'$ is positive semidefinite.

B. Measurement Model and Information Matrix

Suppose that we can obtain noisy relative range measurements, e.g., from a *Two-Way Ranging* RF protocol [3, p.70], between pairs of agents $\{i, j\}$

$$\tilde{d}_{ij} = d_{ij} + \nu_{ij}, \text{ where } \nu_{ij} \sim \mathcal{N}(0, \sigma^2), \quad (3)$$

with independent noise samples. The agent positions constitute a parameter vector $\mathbf{x}$ that we want to estimate and the distance measurements a random observation vector $\mathbf{Y}$ that depends on $\mathbf{x}$. Thus, we can compute the FIM $\mathbf{F} \in \mathbb{R}^{2N \times 2N}$ from (1). We introduce the notation $\mathbb{1}_i(k)$ with $\mathbb{1}_i(k) = 1$ if (i) and (k) are communicating (measuring their distance) and $\mathbb{1}_i(k) = 0$ otherwise. Then, as shown in [10], $\mathbf{F}$ can be decomposed into 2×2 blocks so that one can write for each block (i, j), with $1 \leq i, j \leq N$,

$$\mathbf{F}_{ij} = -\frac{4}{d_{ij}^2 \sigma^2} \mathbf{p}_{ij} \mathbf{p}_{ij}^\top \mathbb{1}_i(j), \text{ if } i \neq j; \quad \mathbf{F}_{ii} = -\sum_{i, i \neq j}^{N} \mathbf{F}_{ij}. \quad (4)$$

This FIM is symmetric but not invertible. Indeed, it has a three-dimensional nullspace generated by rigidly translating and rotating all the agents in the plane, a fact that can be formally proved using rigidity theory [11].

Consider a graph $\mathfrak{G} = (\mathcal{E}, \mathcal{V})$ where the vertices $\mathcal{V}$ represent the agents and edges $\mathcal{E}$ correspond to the availability of inter-agent distance measurements. We introduce the *framework* $(\mathfrak{G}, \mathbf{p})$ where $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_N) \in \mathbb{R}^{2N}$ are the positions of all agents. Note that throughout this paper we use parentheses to denote vectors stacked in one large column vector. If $\mathbf{p}' = (\mathbf{p}_1', \ldots, \mathbf{p}_N') \in \mathbb{R}^{2N}$ is another position vector, we say that $\mathbf{p}'$ is *congruent* to $\mathbf{p}$ if $d_{ij}' = d_{ij}$ for each $1 \leq i, j \leq N$, $i \neq j$, where $d_{ij}' = ||\mathbf{p}_{ij}'||$. Two frameworks $(\mathfrak{G}, \mathbf{p})$ and $(\mathfrak{G}, \mathbf{p}')$ are called *equivalent* if for all $\{i, j\} \in \mathcal{E}$ we have $d_{ij}' = d_{ij}$. $(\mathfrak{G}, \mathbf{p})$ is called *rigid* if there exists $c > 0$ such that for all $(\mathfrak{G}, \mathbf{p}')$ equivalent to $(\mathfrak{G}, \mathbf{p})$, which moreover satisfy $||\mathbf{p}_i - \mathbf{p}_j'|| < c, \forall \{i, j\} \in \mathcal{V}$, we have that $\mathbf{p}$ is congruent to $\mathbf{p}'$ [20]. A rigid framework can be translated along $\vec{x}$ or $\vec{y}$ or rotated without changing the distances [21] and hence the position estimate. A possible solution to this issue is to fix a subset of the nodes (subset $\mathcal{A}$) and derive the constrained CRLB.

We can partition $\mathbf{F}$ defined in (1) as follows

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{\mathcal{A}} & \mathbf{F}_{\mathcal{A}\mathcal{T}} \\ \mathbf{F}_{\mathcal{A}\mathcal{T}}^\top & \mathbf{F}_{\mathcal{T}} \end{bmatrix} \text{ with } \begin{cases} \mathbf{F}_{\mathcal{A}} \in \mathbb{R}^{2n \times 2n}, \mathbf{F}_{\mathcal{A}\mathcal{T}} \in \mathbb{R}^{2n \times 2m} \\ \mathbf{F}_{\mathcal{T}} \in \mathbb{R}^{2m \times 2m}, \end{cases}$$

and apply Theorem 1 with the constraint function $\tilde{\mathbf{f}}_c(\mathbf{p}) = (\tilde{\mathbf{f}}_{c,1}(\mathbf{p}_1), \ldots, \tilde{\mathbf{f}}_{c,n}(\mathbf{p}_n))$, representing the knowledge of the

true anchor positions $\bar{\mathbf{p}}_i$, i.e., $\tilde{\mathbf{f}}_{c,i}(\mathbf{p}_i) = \mathbf{p}_i - \bar{\mathbf{p}}_i \in \mathbb{R}^2$, for $1 \leq i \leq n$. We then find $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{2m \times 2n} & \mathbf{I}_{2m} \end{bmatrix}^\top$, so that the $\mathbf{F}_c$ matrix in Theorem 1 is equal to $\mathbf{F}_\mathcal{T}$. This matrix is invertible if $(\mathfrak{G}, \mathbf{p})$ is rigid and $n \geq 2$, fixing the framework at a given place and orientation.

C. Motion Planning Problem

We aim to optimize the location of the tags in $\mathcal{T}$ by minimizing a criterion derived from the matrix $\mathbf{B}_c$ introduced in (2). We choose to minimize the lower bound on the mean-squared error (MSE) of the estimator $\hat{\mathbf{p}}_\mathcal{T}$ of the tag positions $\mathbf{p}_\mathcal{T} = (\mathbf{p}_{n+1}, \dots, \mathbf{p}_{n+m}) \in \mathbb{R}^{2m}$. Although minimizing the CRLB provides no formal guarantee that a even the best available position estimator will perform well, it is a standard practice to optimize localization accuracy in the literature on sensor placement [10] and navigation systems [22], [23], since the CRLB is a generalization of the concept of DOP.

Introduce the constrained set of tag locations

$$\mathcal{R} = \left\{ \mathbf{p}_\mathcal{T} \in \mathbb{R}^{2m} \,\middle|\, \forall k \in [1, s], \text{ if } (T_i, T_j) \in S_k, r_{ij} = 0 \right\}, \quad (5)$$

where $r_{ij} = \frac{1}{2}(\|\mathbf{p}_{ij}\|^2 - d_{ij}^2)$. Assuming we know all distances between tags on the same rigid body, the estimators $\hat{\mathbf{p}}_\mathcal{T}$ can use the property that $\mathbf{p}_\mathcal{T} \in \mathcal{R}$ and the CRLB should take this knowledge into account.

To define this CRLB, introduce the vector $\mathbf{r}(\mathbf{p}_\mathcal{T}) = (\mathbf{r}_1, \dots, \mathbf{r}_s) \in \mathbb{R}^K$, where $\mathbf{r}_k = [\dots r_{ij} \dots]^T$ corresponds to distance constraints for tags carried by S_k and K is the total number of such constraints. Hence, $\mathbf{p}_\mathcal{T} \in \mathcal{R}$ is equivalent to set $\mathbf{r}(\mathbf{p}_\mathcal{T}) = \mathbf{0}$. To use Theorem 1, the node positions $\mathbf{p}$ satisfy the constraints $\mathbf{f}_c(\mathbf{p}) = (\tilde{\mathbf{f}}_c(\mathbf{p}), \mathbf{r}(\mathbf{p}_\mathcal{T})) = \mathbf{0}$. We can compute $\frac{\partial \mathbf{f}_c}{\partial \mathbf{p}}$ and its kernel matrix $\mathbf{A}$, see Section IV. Then, we obtain the constrained FIM $\mathbf{F}_c$ and constrained CRLB $\mathbf{B}_c$. Any unbiased estimator of the positions $\hat{\mathbf{p}}_\mathcal{T}$ then satisfies the following scalar inequality on its MSE

$$\mathbb{E}\left\{ [\hat{\mathbf{p}}_\mathcal{T} - \mathbf{p}_\mathcal{T}]^\top [\hat{\mathbf{p}}_\mathcal{T} - \mathbf{p}_\mathcal{T}] \right\} = \mathrm{Tr}\{\Sigma\} \geq \mathrm{Tr}\{\mathbf{B}_c\},$$

where Σ is the covariance of the estimator.

This discussion leads us to define the *cost function* $J(\mathbf{p}_\mathcal{T}) = \mathrm{Tr}\{\mathbf{B}_c\}$, which serves as a potential field that we attempt to minimize by moving the robots. Alternatively, one could impose an upper bound on this function when using other optimization-based motion planners. We must also ensure that tag trajectories satisfy the constraints defined by $\mathcal{R}$ at all time. Hence, we aim to solve the problem

$$\mathbf{p}_\mathcal{T}^* = \underset{\mathbf{p}_\mathcal{T} \in \mathcal{R}}{\mathrm{argmin}} \ J(\mathbf{p}_\mathcal{T}), \text{ where } J(\mathbf{p}_\mathcal{T}) = \mathrm{Tr}\{\mathbf{B}_c\}, \quad (6)$$

to find a (locally) optimal placement for the tags, which would improve the accuracy of their location estimate and as a result the pose estimation for the robots, see Section V for some examples.

III. Center of Mass Motion Planner

A naive approach to enforcing the constraints (5) is to use a two-step approach: first, move the centers of mass $\mathbf{p}_k^G$ of the robots by minimizing $\tilde{J}(\mathbf{p}_\mathcal{T}) = \mathrm{Tr}\{F_\mathcal{T}^{-1}\}$, without

modifying their orientation; then, rotate the bodies to find an optimum orientation. To simplify the discussion, we assume here that a node (i) is present at the center of mass G_k of solid S_k.

A. Positioning Optimization

With ξ_i denoting one of the coordinates x_i or y_i of an agent (i) and $\xi_{ij} = \xi_i - \xi_j$, we have [19]

$$\frac{\partial \tilde{J}(\mathbf{p})}{\partial \xi_i} = \frac{\partial \mathrm{Tr}\{\mathbf{F}_\mathcal{T}^{-1}\}}{\partial \xi_i} = -\mathrm{Tr}\left\{ \mathbf{F}_\mathcal{T}^{-2} \frac{\partial \mathbf{F}_\mathcal{T}}{\partial \xi_i} \right\}. \quad (7)$$

The matrix $\frac{\partial \mathbf{F}}{\partial \xi_i}$ can be computed from the 2×2 blocks

$$\frac{\partial \mathbf{F}_{ij}}{\partial x_i} = \frac{8 \times 1_i(j)}{\sigma^2 d_{ij}^2} \begin{bmatrix} \frac{x_{ij}^3}{d_{ij}^2} - x_{ij} & y_{ij}\left(\frac{x_{ij}^2}{d_{ij}^2} - \frac{1}{2}\right) \\ \star & \frac{x_{ij}y_{ij}^2}{d_{ij}^2} \end{bmatrix},$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial y_i} = \frac{8 \times 1_i(j)}{\sigma^2 d_{ij}^2} \begin{bmatrix} \frac{y_{ij}x_{ij}^2}{d_{ij}^2} & x_{ij}\left(\frac{y_{ij}^2}{d_{ij}^2} - \frac{1}{2}\right) \\ \star & \frac{y_{ij}^3}{d_{ij}^2} - y_{ij} \end{bmatrix}, \quad (8)$$

for $\{i, j\} \in [1, N], i \neq j$, where the symbol $\star$ replaces symmetric terms. Indeed, all blocks $\frac{\partial \mathbf{F}_{kl}}{\partial \xi_i}$ can be computed from (8), see [11] for the details.

The gradient $\frac{\partial \tilde{J}(\mathbf{p})}{\partial \mathbf{p}_k^G}$ can be computed from the previous expressions for the node (i) placed at G_k. Defining $\mathbf{p}_k \in \mathbb{R}^{2|S_k|}$ the position vector for all tags carried by $\mathbf{S}_k$, we can follow the gradient flow by translating all the tags on S_k by the same vector

$$\mathbf{p}_k(l+1) = \mathbf{p}_k(l) - \alpha_k(l) \, \mathbf{1}_{|S_k|} \otimes \left[\frac{\partial \tilde{J}(\mathbf{p})}{\partial \mathbf{p}_k^G}(l) \right]^\top \quad (9)$$

with $\mathbf{1}_{|S_k|}$ an all-ones vector, $\otimes$ the Kronecker product and $\alpha_k(l) \in \mathbb{R}$ a decreasing stepsize chosen by Armijo's Rule for example [24, p. 35]. The gradient-based motion of a robot k is stopped when the gradient falls below a pre-defined threshold.

B. Orientation Optimization

Simply translating the robots satisfies the constraints (5) but does not necessarily result in optimal positions for all tags. In a second step, we can rotate the rigid body around its center of mass in order to further decrease $\tilde{J}(\mathbf{p})$. Since in two dimensions this is a one-dimensional optimization problem, one can simply gradually increment the orientation angle until a local minimum is found for $\tilde{J}$, or perform a full rotation of the robot by such increments to find a global minimum over the orientation angle.

IV. Rigid Body Motion Planner

In this section we present an algorithm to solve (6) directly and obtain a gradient flow that the robots can follow to optimize the localization of all their tags simultaneously.

A. Lagrangian Descent Formulation

Minimization of $J(\mathbf{p}_\mathcal{T})$ is subject to the equality constraint $\mathbf{r}(\mathbf{p}_\mathcal{T}) = \mathbf{0}$, which must remain satisfied as the robots move. To solve this constrained problem, we implemented the following first-order Lagrangian algorithm [24, p.528], written here with a reordering of the elements of $\mathbf{p}_\mathcal{T}$ to partition them between x and y coordinates $\mathbf{p}_\mathcal{T} = (\mathbf{x}_\mathcal{T}, \mathbf{y}_\mathcal{T})$:

$$\begin{cases} \mathbf{p}_\mathcal{T}(l+1) = \mathbf{p}_\mathcal{T}(l) - \beta(l)\left(\frac{\partial J}{\partial \mathbf{p}_\mathcal{T}(l)} + \begin{bmatrix} \boldsymbol{\lambda}^\top(l)\frac{\partial \mathbf{r}}{\partial \mathbf{x}_\mathcal{T}(l)} \\ \boldsymbol{\lambda}^\top(l)\frac{\partial \mathbf{r}}{\partial \mathbf{y}_\mathcal{T}(l)} \end{bmatrix} \right), \\ \boldsymbol{\lambda}(l+1) = \boldsymbol{\lambda}(l) + \delta \mathbf{r}(\mathbf{p}_\mathcal{T}(l)), \end{cases}$$
(10)

where $\beta(l)$ is decreasing stepsize following Armijo's rule [24] and γ, δ two constants. The scheme is stopped when $||\mathbf{p}_\mathcal{T}(l+1) - \mathbf{p}_\mathcal{T}(l)||_2 < \eta$, with η a given threshold.

B. Gradient Computation

1) Expression of the CRLB: The criterion $J(\mathbf{p}_\mathcal{T}) = \text{Tr}\{\mathbf{B}_c(\mathbf{p}_\mathcal{T})\}$ requires computing $\mathbf{B}_c$, which depends on the matrix $\mathbf{A}$ whose columns span the kernel of $\frac{\partial \mathbf{f}_c}{\partial \mathbf{p}}$, see Section II-C. The Jacobian of the constraint function is

$$\frac{\partial \mathbf{f}_c}{\partial \mathbf{p}} = \begin{bmatrix} \mathbf{I}_{2n} & \mathbf{0}_{2n \times 2m} \\ \mathbf{0}_{K \times 2n} & \text{diag}(\mathbf{R}_1, \dots \mathbf{R}_s) \end{bmatrix} \in \mathbb{R}^{(2n+K) \times 2N}$$

where $\mathbf{R}_k = \frac{\partial \mathbf{r}_k}{\partial \mathbf{p}_k}$ are called *rigidity matrices* of S_k [25]. Each of these matrices admits a null space spanned by three vectors [21], [25]: $\ker \mathbf{R}_k = \text{span}\{\mathbf{v}_1^k, \mathbf{v}_2^k, \mathbf{v}_3^k\}$ with $\mathbf{v}_1^k = \mathbf{1}_{|S_k|} \otimes [1,0]^\top$, $\mathbf{v}_2^k = \mathbf{1}_{|S_k|} \otimes [0,1]^\top$ and $\mathbf{v}_3^k = (\dots, \mathbf{v}_{3k,i}, \dots)$ where $\mathbf{v}_{3k,i} = [y_i, -x_i]^\top$ for the tag $T_i \in S_k$. Hence, $\mathbf{A} := [\mathbf{a}_1, \dots \mathbf{a}_s] \in \mathbb{R}^{2N \times 3s}$ with $\mathbf{a}_k = \begin{bmatrix} \mathbf{0}_{2n \times 3} \\ \mathbf{v}^k(\mathbf{p}_k) \end{bmatrix} \in \mathbb{R}^{2N \times 3}$ and the 3 columns of the matrix $\mathbf{v}^k(\mathbf{p}_k)$ obtained by placing the vectors $\mathbf{v}_1^k, \mathbf{v}_2^k, \mathbf{v}_3^k$ on the lines for the kth rigid body.

2) Gradient of the Cost Function: Let us introduce the notation $v[j]$ to represent the j-th component of the column vector $\mathbf{v}$. We compute the derivative of $\mathbf{A}$ with respect to each coordinates present in $\mathbf{p}_\mathcal{T}$. For each coordinate $\xi_i \in \{x_i, y_i\}$ for tag $T_i \in S_k$, we have :

$$\begin{cases} \frac{\partial \mathbf{a}_j}{\partial \xi_i} = [\mathbf{0}_{2n \times 2} \quad \mathbf{v}'_{k,\xi_i}], & \text{if } j = k; \\ \frac{\partial \mathbf{a}_j}{\partial \xi_i} = \mathbf{0}_{2n \times 3} & \text{elsewise.} \end{cases}$$

with $v'_{k,x_i}[2n + 2i + 1] = -1$ (resp. $v'_{k,y_i}[2n + 2i] = 1$) and $v'_{k,x_i}[j'] = 0$ for $j' \neq 2n + 2i + 1$ (resp. $j' \neq 2n + 2i$). We then form the $\frac{\partial \mathbf{A}}{\partial \xi_i}$ matrix from the $\frac{\partial \mathbf{a}_j}{\partial \xi_i}$ matrices.

Then, we compute the gradient of J with respect to ξ_i, with $T_i \in \mathcal{T}$:

$$\frac{\partial J}{\partial \xi_i} = \frac{\partial}{\partial \xi_i} \text{Tr}\left\{ \mathbf{A}\mathbf{F}_c^{-1}\mathbf{A}^\top \right\}$$
$$= 2\text{Tr}\left\{ \frac{\partial \mathbf{A}}{\partial \xi_i}\mathbf{F}_c^{-1}\mathbf{A}^\top \right\} - \text{Tr}\left\{ \mathbf{A}\mathbf{F}_c^{-1}\frac{\partial \mathbf{F}_c}{\partial \xi_i}\mathbf{F}_c^{-1}\mathbf{A}^\top \right\}$$

with $\frac{\partial \mathbf{F}_c}{\partial \xi_i} = \mathbf{A}^\top\frac{\partial \mathbf{F}}{\partial \xi_i}\mathbf{A} + \frac{\partial \mathbf{A}}{\partial \xi_i}^\top\mathbf{F}\mathbf{A} + \mathbf{A}^\top\mathbf{F}\frac{\partial \mathbf{A}}{\partial \xi_i}$,

since $\mathbf{F}_c = \mathbf{A}^\top\mathbf{F}\mathbf{A}$, where $\frac{\partial \mathbf{F}}{\partial \xi_i}$ is given in (8). We can build the whole gradient by concatenation

$$\frac{\partial J}{\partial \mathbf{p}_\mathcal{T}} = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{p}_{n+1}}, \dots \frac{\partial J}{\partial \mathbf{p}_N} \end{bmatrix} \text{ with } \frac{\partial J}{\partial \mathbf{p}_i} = \begin{bmatrix} \frac{\partial J}{\partial x_i}, \frac{\partial J}{\partial y_i} \end{bmatrix}^\top.$$

3) Gradient of the Rigidity Constraint: Let $\boldsymbol{\xi}_\mathcal{T} \in \{\mathbf{x}_\mathcal{T}, \mathbf{y}_\mathcal{T}\}$ containing all ξ_i coordinates of the agents of the set $\mathcal{T}$. The gradient of the rigidity function $\frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}_\mathcal{T}}$ is block diagonal because $\frac{\partial r_{ij}}{\partial \xi_{j'}} = 0$ if $\{T_i, T_j\}$ and $T_{j'}$ are carried by different rigid bodies. We can compute remaining values using following formulas:

$$\begin{cases} \frac{\partial r_{ij}}{\partial \xi_{j'}} = \xi_i - \xi_j, & \text{if } j' = i; \\ \frac{\partial r_{ij}}{\partial \xi_{j'}} = \xi_j - \xi_i, & \text{if } j' = j; \\ \frac{\partial r_{ij}}{\partial \xi_{j'}} = 0, & \text{otherwise.} \end{cases}$$

V. SIMULATIONS

A. Numerical Motion Planning Schemes

We present a comparative use case for rigid body motion planning. Let a network of $N = 4$ agents, with $n = 2$ fixed-position anchors A_1 and A_2. The remaining agents are $m = 2$ tags, T_3 and T_4, carried by rigid body S_1. Therefore, d_{34} is constant and known. An arbitrary initial value for the positions $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ is chosen and shown on Fig. 2). The goal is to deploy $\mathbf{p}_\mathcal{T} = [\mathbf{p}_3, \mathbf{p}_4]$ using the algorithms presented Sections III and IV.
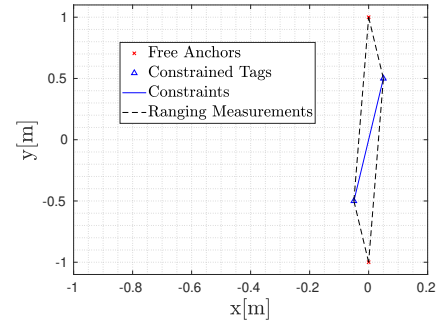

Fig. 2: Initial configuration.

To simplify the setup, we suppose that T_3 is located on G, the center of mass of S_1. For the *Center of Mass Motion Planner*, we have plotted a 30-step path on Fig. 3. For the first 15 steps of the trajectory, the algorithm follows the partial gradient scheme (9). We see that the global criterion $J(\mathbf{p})$ actually increases during this stage. This issue is due to the alignment of the four agents. Indeed, descending the gradient with respect to the center of mass is not equivalent to minimizing the global cost function. The naive approach can therefore in fact deteriorate the precision of the localization dramatically. For the chosen threshold κ, the rotation optimization stage starts at iteration 15. T_3 remains fixed as T_4 rotates until $J(\mathbf{p})$ reaches a local minimum.

The trajectory found by the *Rigid Constraint* motion planner, following the scheme (10), is presented on Fig.
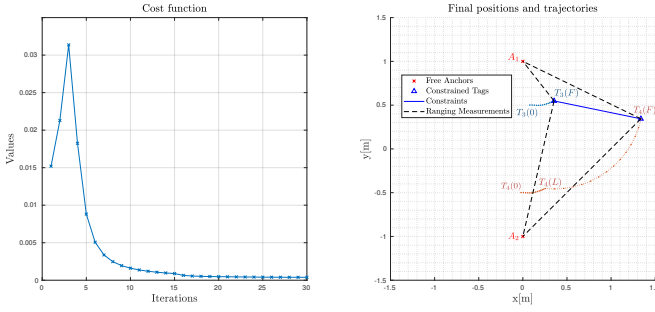
Fig. 3: Center of Mass motion planner.

4 for the first 20 steps. The cost function monotonically decreases along the path and the trajectory is intuitive, forming a circular arc. Indeed, the final tag positions are obtained by rotating the rigid body and essentially make the triangles $T_3 T_4 A_i, i \in \{1, 2\}$ isosceles. The rigidity constraint $f_c(l) = ||\mathbf{p}_3(l) - \mathbf{p}_4(l)||^2 - d_{34}(0)^2 = 0$ remains close to satisfied during the motion. Profiling analysis on `Matlab` shows that the runtime per iteration of the *Rigid Constraint* motion planner was 1.75 times longer than for the *Center of Mass* algorithm, which shows that the complexity of our proposed algorithm remains reasonable.
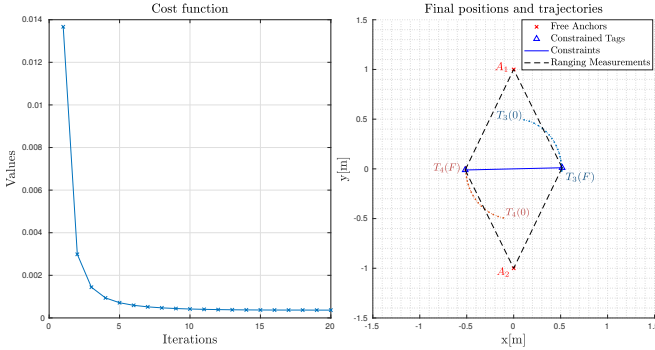

Fig. 4: Rigid constraints algorithm motion planning.

B. Performance with Least Squares Estimators

In order to demonstrate the localization improvement induced by motion planning on an actual estimator, we solve the following least squares problem at each step of the algorithm:

$$\hat{\mathbf{p}}_{\mathcal{T}} = \underset{\mathbf{p}_{\mathcal{T}} \in \mathbb{R}^{m \times 2}}{\operatorname{argmin}} \left\{ \sum_{i=n+1}^{m} \left(\sum_{j=1}^{n} [\tilde{d}_{ij} - ||\hat{\mathbf{p}}_i - \mathbf{p}_j||]^2 \right. \right.$$
$$\left. \left. + \sum_{\substack{k=n+1, k \neq i \\ (T_i, T_k) \in S_p}}^{m} [d_{ik} - ||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_k||]^2 \right) \right\}, \tag{11}$$
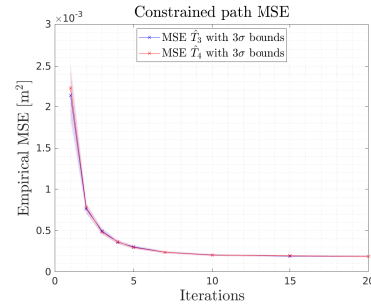
given the range measurements $\tilde{d}_{ij}$ and knowing the distances constraints d_{ik} within each rigid body S_p. We performed Monte Carlo simulations using the model (3), setting $\sigma = 0.1$ m, to evaluate empirically the MSE Q_E^i of the position estimate of node $i = 3$ at 4, over $M = 30000$ simulations,

at each step l of the trajectory

$$Q_E^i(l) = \frac{1}{M} \sum_{r=1}^{M} (\hat{\mathbf{p}}_i(l; r) - \mathbf{p}_i(l))^\top (\hat{\mathbf{p}}_i(l; r) - \mathbf{p}_i(l)).$$
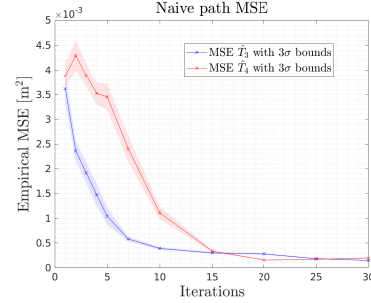
(a) Center of Mass motion planner.

(b) Rigid Constraints motion planner.

Fig. 5: Monte-Carlo simulations.

Monte-Carlo simulation results for the empirical MSE of the least squares estimate (11) at each step of the trajectory are shown on Fig. 5, plotted with the 3σ confidence bounds. On Fig 5a, we plotted the results for $Q_E^3(l)$ and $Q_E^4(l)$ using the *Center of Mass* motion planner. As noticed on Fig. 3, the algorithm optimizes the position of T_3, which is the center of mass of the rigid body. Indeed, the blue curve representing the estimate of T_3's position is seen to decrease monotonously. On the other hand, the red curve confirms a transient degradation of the positioning accuracy for T_4, as the CRLB-based criterion increases (see Fig. 3). Nevertheless, by the end of the simulation, the uncertainty on the position of T_4 has also decreased significantly. The same simulation for the *Rigid Constraint* motion planner shows on Fig. 5b a simultaneous reduction in position estimate uncertainty for both tags, confirming the results of Fig. 4.

C. Influence of Motion Planning on Rigid Body States Estimation

The objective of minimizing the tags' location uncertainty is actually to improve the pose estimation for the robots. With $\mathbf{x} = [x_G, y_G, \theta]^\top \in SE(2)$ the configuration of the robot in this example, the position $\mathbf{p}_i$ of tag T_i at each step l of the algorithm is given by

$$\mathbf{p}_i(l) = \mathbf{p}^G(l) + \mathbf{R}(\theta(l))\mathbf{a}_i := \mathbf{g}_i(\mathbf{x}),$$

where $\mathbf{R}(\theta(t))$ is a 2D rotation matrix and $\mathbf{a}_i$ a known vector, representing the tag's coordinates in the rigid body

frame. From the estimates $\hat{\mathbf{p}}_i(t)$ obtained using (11), we can formulate the following nonlinear least squares problem in order to estimate $\mathbf{x}$

$$\hat{\mathbf{x}}(l) = \underset{\substack{\mathbf{x} \in \mathbb{R}^2 \times \mathcal{S}_1 \\ i = n+1 \\ T_i \in S_p}}{\operatorname{argmin}} \left\{ \sum_{i=n+1}^{N} [\hat{\mathbf{p}}_i(l) - \mathbf{g}_i(\mathbf{x})]^\top [\hat{\mathbf{p}}_i(l) - \mathbf{g}_i(\mathbf{x})] \right\}. \tag{12}$$

We solve this problem with the Gauss-Newton algorithm [24]. Fig. 6 shows the results of 3000 Monte-Carlo simulations to evaluate the MSE for the body configuration estimates. The estimates $\hat{\mathbf{p}}_{\mathcal{T}}$ used in (12) are those from Fig. 5.
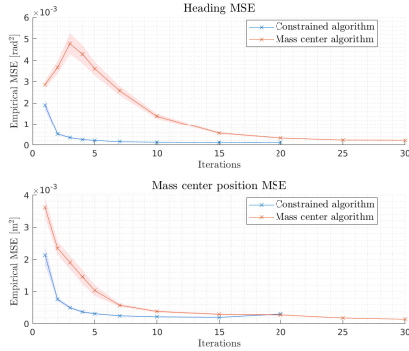

Fig. 6: Monte Carlo simulation for heading and center of mass position.

With the *Center of Mass* motion planner, the quality of the heading estimates initial decreases, because of the alignment between the anchors and tags. In contrast, the *Rigid Constraint* motion planner improves the accuracy uniformly on each state.

D. Application to an Unmanned Ground Vehicle

We applied the *Rigid Constraint* motion planner to the following scenario. The pose of a rover carrying three UWB tags (denoted T_4, T_5 and T_6) is estimated by relying on three anchors A_1, A_2 and A_3, whose absolute locations are known in the global frame $\mathcal{G}$, equal to $\mathbf{a}_1^{\mathcal{G}} = [-3,3]^\top$, $\mathbf{a}_2^{\mathcal{G}} = [3,3]^\top$ and $\mathbf{a}_3^{\mathcal{G}} = [3,-3]^\top$. The center of mass G of the robot center of mass is located at $\mathbf{p}^G$, the origin of the rover's frame $\mathcal{F}$ and barycenter of the tags. Tag positions in $\mathcal{F}$ are known, equal to $\mathbf{p}_4^{\mathcal{F}} = [0,-4/3]^\top$, $\mathbf{p}_5^{\mathcal{F}} = [-1/4,2/3]^\top$ and $\mathbf{p}_6^{\mathcal{F}} = [1/4,2/3]^\top$. The setup is shown on Fig 7.

Using (10), we computed a rigidity compliant trajectory $\mathbf{p}_{\mathcal{T}}^{\text{traj}} = \{\mathbf{p}_{\mathcal{T}}^{\mathcal{G}}(l), l \in [1,F]\}$ for the tag positions, solving the minimization problem (6). After this calculation, we solved an auxiliary least squares problems to estimate the trajectories of the center position $\mathbf{p}_G^{\text{traj}}$ and heading θ^{traj}. We define $\theta = \angle(\vec{x}, \mathbf{m})$ to be the angle between $\mathbf{m} = (\mathbf{p}_4^{\mathcal{G}} - \mathbf{p}_G^{\mathcal{G}})$ and the vector $\vec{x} = [1,0]$ of $\mathcal{G}$. We suppose to be able to reach exactly each planned position $\mathbf{p}_G(l)$ and heading $\theta(l)$ at iteration l. At iteration $l = F$, we introduce a malfunction for anchor A_3, which immediately ceases to emit. The algorithm automatically computes a new optimization trajectory $\mathbf{p}_{\mathcal{T}}^{\text{traj2}} = \{\mathbf{p}_{\mathcal{T}}^{\mathcal{G}}(l), l \in [F+1,T]\}$ to re-position the rover until convergence is reached at $l = T$.
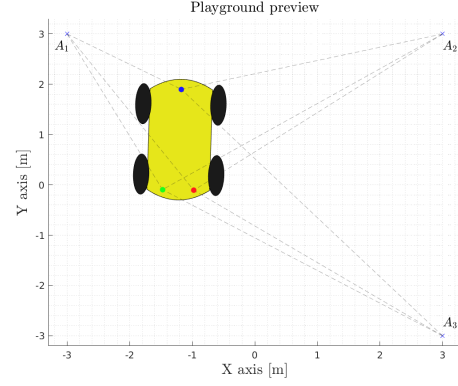

Fig. 7: Rover and anchors setup.

Detailed results and interactive graphics for this scenario are shown in the accompanying video. This includes the rover trajectory and plots of the evolution of the cost function and empirical MSE. For the latter plots, we used the Monte Carlo scheme presented in section V-B, with $M = 1000$ realizations. To do so, the least squares estimate (11) has been implemented, considering distance constraints between tags and using measurements with centered additive white Gaussian noise with $\sigma = 0.1$ m. The video demonstrates that we can obtain a smooth trajectory for the rover, which could be implemented on a real robot. Moreover, the degradation of localization accuracy induced by the anchor malfunction is clearly mitigated by the adjusted trajectory computed by the motion planner. This shows the practical usefulness of relying on implementable closed form formulas based on the CRLB.

VI. Conclusion

We presented a motion planning algorithm based on the Cramr-Rao lower bound (CRLB) to improve the localization accuracy in a ranging network with nodes carried by robots modeled as rigid bodies. The main application is in estimating the pose of the robots. The algorithm takes the distance constraints for the tags on the same robot into account by using an equality-constrained CRLB to formulate a cost function capturing the localization accuracy. These constraints are also enforced in a primal-dual optimization algorithm producing a robot trajectory converging to a local minimum of the cost function. The expressions of the gradients necessary to implement the algorithm can be computed in closed form, which simplifies the implementation. Future work will focus on real-world experiments, the inclusion of other constraints in the deployment (tasks, obstacles, robot dynamics) and developing distributed versions of this algorithm that scale to large networks.

REFERENCES

[1] P. Corke, *Robotics, Vision and Control*, ser. Springer Tracts in Advanced Robotics, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 73. [Online]. Available: http://link.springer.com/10.1007/978-3-642-20144-8

[2] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*, 2nd ed. Artech House, 2013.

[3] Z. Sahinoglu, S. Gezici, and I. Guvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge: Cambridge University Press, 2008. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511541056

[4] A. Prorok, "Models and Algorithms for Ultra-Wideband Localization in Single- and Multi-Robot Systems," PhD Thesis, Ecole Polytechnique Fererale de Lausanne (EPFL), 2013.

[5] D. Lymberopoulos and J. Liu, "The microsoft indoor localization competition: Experiences and lessons learned," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 125–140, 2017.

[6] M. Hamer and R. DAndrea, "Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization," *IEEE Access*, vol. 6, pp. 22 292–22 304, 2018.

[7] V. Mai, M. Kamel, M. Krebs, A. Schaffner, D. Meier, L. Paull, and a. R. Siegwart, "Local Positioning System Using UWB Range Measurements for an Unmanned Blimp," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2971–2978, Oct. 2018.

[8] J. Cano, S. Chidami, and J. Le Ny, "A Kalman Filter-Based Algorithm for Simultaneous Time Synchronization and Localization in UWB Networks," in *ICRA*, Montreal, QC, Canada, May 2019.

[9] A. J. Haug, *Bayesian estimation and tracking: a practical guide*. Hoboken, N.J: Wiley, 2012, oCLC: ocn767824937.

[10] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[11] J. Le Ny and S. Chauvire, "Localizability-Constrained Deployment of Mobile Robotic Networks with Noisy Range Measurements," in *2018 Annual American Control Conference (ACC)*, Jun. 2018, pp. 2788–2793.

[12] H. Takeda, C. Facchinetti, and J.-C. Latombe, "Planning the motions of a mobile robot in a sensory uncertainty field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1002–1017, 1994.

[13] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1999, pp. 35–40.

[14] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

[15] M. S. Ramanagopal, A. Phu-Van Nguyen, and J. Le Ny, "A motion planning strategy for the active vision-based mapping of ground-level structures," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 356–368, 2017.

[16] S. Bonnabel and A. Barrau, "An intrinsic Cramér-Rao bound on SO(3) for (dynamic) attitude filtering," *arXiv:1503.04701 [math, stat]*, Oct. 2015, arXiv: 1503.04701. [Online]. Available: http://arxiv.org/abs/1503.04701

[17] G. S. Chirikjian, "From Wirtinger to Fisher Information Inequalities on Spheres and Rotation Groups," in *2018 21st International Conference on Information Fusion (FUSION)*. Cambridge, United Kingdom: IEEE, Jul. 2018, pp. 730–736. [Online]. Available: https://ieeexplore.ieee.org/document/8455626/

[18] J. Gorman and A. Hero, "Lower bounds for parametric estimation with constraints," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1285–1301, Nov. 1990, conference Name: IEEE Transactions on Information Theory.

[19] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Technical University of Denmark, 2012.

[20] R. Connelly, "Generic Global Rigidity," *Discrete & Computational Geometry*, vol. 33, no. 4, pp. 549–563, Apr. 2005. [Online]. Available: http://link.springer.com/10.1007/s00454-004-1124-4

[21] T.-S. Tay and W. Whiteley, "Generating Isostatic Frameworks," *Struct. Topology*, vol. 11, Jan. 1985.

[22] S. Xue and Y. Yang, "Understanding GDOP minimization in GNSS positioning: Infinite solutions, finite solutions and no solution," *Advances in Space Research*, vol. 59, no. 3, pp. 775–785, Feb. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S027311771630597X

[23] J. J. Khalife and Z. Z. M. Kassas, "Optimal Sensor Placement for Dilution of Precision Minimization Via Quadratically Constrained Fractional Programming," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 2086–2096, Aug. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8528397/

[24] D. P. Bertsekas, *Nonlinear programming*, 3rd ed. Belmont, MA: Athena Scientific, 2016.

[25] Z. Sun, C. Yu, and B. D. O. Anderson, "Distributed optimization on proximity network rigidity via robotic movements," in *2015 34th Chinese Control Conference (CCC)*, Jul. 2015, pp. 6954–6960.