Ranging-Based Localizability-Constrained Deployment of Mobile Robotic Networks

Justin Cano, Member, IEEE, and Jerome Le Ny, Senior Member, IEEE

Abstract—In cooperative localization schemes for robotic networks relying on noisy range measurements between agents, the achievable positioning accuracy strongly depends on the network geometry. This motivates the problem of planning robot trajectories in such multi-robot systems in a way that maintains high localization accuracy. We present potential-based planning methods, where localizability potentials are introduced to characterize the quality of the network geometry for cooperative position estimation. These potentials are based on Cramér Rao Lower Bounds (CRLB) and provide a theoretical lower bound on the error covariance achievable by any unbiased position estimator. In the process, we establish connections between CRLBs and the theory of graph rigidity, which has been previously used to plan the motion of robotic networks. We develop decentralized deployment algorithms appropriate for large networks, and we use equality-constrained CRLBs to extend the concept of localizability to scenarios where additional information about the relative positions of the ranging sensors is known. We illustrate the resulting robot deployment methodology through simulated examples.

Index Terms—Multi-robot systems, Path planning, Cooperative localization

I. INTRODUCTION

MOBILE robots require accurate, computationally efficient and low power localization systems to navigate their environment and perform their assigned tasks. Positioning can rely on various technologies, e.g., wheel odometry, computer vision or long- and short-range radio frequency (RF) systems, each with distinct advantages and drawbacks, depending on the environment and requirements. For example, the most common methods of terrestrial localization rely on RF signals from Global Navigation Satellite Systems (GNSS) to achieve meter- to centimeter-level accuracy, but these systems do not operate indoors or when the line of sight to the satellites is obstructed, and are sensitive to interference.

Multiple robots can collaborate to improve the accuracy and coverage of their individual localization solution [3], [4]. In particular, they can leverage proximity [3], relative position [4], bearing [5], or distance measurements [6], [7] between them to estimate their individual positions in a common reference frame. Relative bearing measurements can be provided by monocular cameras for example, range measurements by short-range RF systems, and relative position measurements by LiDARs or stereo cameras. In this paper, we focus on collaborative localization in Multi-Robot Systems (MRS) using only range measurements. This is motivated by the fact that accurate distance measurements can be deduced from Time-of-Flight (ToF) measurements obtained from inexpensive short-range RF communication systems, e.g., Ultra-Wide Band (UWB) transceivers [8]–[10]. In particular, such systems associate distance measurements unambiguously with pairs of robots, simply by having the robots broadcast their IDs.

Once the robots have measured their relative distances, many algorithms exist to compute from these measurements an estimate of the robot positions, see, e.g., [11] for a recent survey. These algorithms can be centralized or decentralized, applicable to static or mobile networks, appropriate for real-time localization or require longer processing times, etc. Two major factors determine the ability of these algorithms to solve the position estimation problem and their accuracy. First, enough relative distance measurements should be available, which links the feasibility of the location estimation problem to the concept of rigidity [12]–[14] of the ranging graph corresponding to these measurements. Second, satisfying the graph-theoretic condition of rigidity is still insufficient to guarantee accurate localization of the individual agents, when measurement noise is inevitably present. For example, a group of robots that are almost aligned can form a rigid formation if enough range measurements are available, but can only achieve poor localization accuracy in practice. Indeed, the spatial geometry of the network strongly influences the accuracy of position estimates in the presence of measurement noise [15], a phenomenon known as Dilution of Precision (DOP) in the navigation literature [16, Chap. 7]. We call here localizability the ability to accurately estimate the positions of the individual robots of an MRS in a given geometric configuration, using relative measurements.

In contrast to static sensor networks or GNSS, an MRS can actively adjust its geometry, e.g., some of the robot positions and orientations, in order to improve its overall localizability. This results in a coupling between the motion planning and localization problem for the group.

Maintaining the rigidity of the ranging graph during the motion of an MRS is a stronger condition than maintaining its connectivity, but similar techniques can be used to address both problems. In particular, we can capture the degree of connectivity or rigidity of the graph using a function of the first non-zero eigenvalue of a type of Laplacian matrix, and guide the MRS along paths or configure its nodes in ways that increase this function. This is the approach adopted for exam-
ple in [17]–[19] for improving connectivity and in [20]–[23] for improving rigidity. This article builds on this principle to optimize localizability. Following an approach that we initially proposed in [1], [2], we leverage Cramér Rao Lower Bounds (CRLBs) [24, Chap. 7] to construct localizability potentials, which can then be used as artificial potentials [25] to drive the motion of an MRS toward geometric configurations promoting good localization.

The CRLB provides a lower bound on the covariance of any unbiased position estimate constructed from the relative range measurements available in the robot network.

Tighter covariance lower bounds exist, such as Barankin bounds [26], but an advantage of the CRLB is that it is relatively easy to compute and admits a closed-form expression for the problem considered here, assuming Gaussian noise [15]. Moreover, as we show in Section IV, the CRLB for Gaussian noise is in fact closely related to the so-called rigidity matrix of the ranging graph. This can be expected since the Gaussian CRLB is known to correspond to DOP expressions for least-squares estimators, which are implicitly derived in [20] for example and also linked to the rigidity matrix. The CRLB only provides a lower bound on estimation performance and there is generally no guarantee that a position estimator actually achieves it. Nonetheless, using this bound as a proxy to optimize sensor placement is a well accepted approach [27]. An important advantage of this approach is that the motion planning strategy becomes independent of the choice of position estimator implemented in the network.

Our main contributions and the structure of this paper are as follows. First, we formulate in Section II a novel motion planning problem allowing an MRS to optimize its localizability. This is done by minimizing appropriate cost functions based on the Fisher Information Matrix (FIM) appearing in the CRLB, as detailed in Section III. We derive in Section IV a closed-form expression for the FIM and establish an explicit connection with the weighted rigidity matrices introduced in [22], [23]. One of the benefits of establishing this connection is to see that various artificial potentials can be constructed from the FIM to capture localizability, as discussed in the literature on optimal experimental design [28] or optimal sensing with mobile robots, see, e.g., [27], [29], [30]. Some of these functions may be more conveniently optimized than the smallest nonzero eigenvalue, which is the standard potential for least-squares estimators, which are implicitly derived in [24, Chap. 7] to construct localizability potentials, proposed in [1], [2], we leverage Cramér Rao Lower Bounds (CRLBs) [31] to account for the presence of additional rigidity constraints. This can be viewed as an alternative and simpler approach to deriving intrinsic CRLBs on the manifold of rigid motions [32], [33]. Finally, Section VII illustrates the performance of the proposed algorithms in simulation for two simple robot deployment scenarios.

This article builds on the conference paper [1], which introduced the the concept of localizability potentials for the deployment of MRS in two dimensions. Here we extend the methodology to three dimensions, introduce new distributed optimization schemes, discuss useful properties on the FIM and make a clearer connection with rigidity theory. We also generalize the conference paper [2], which considered robots carrying multiple sensors, by developing the results in three dimensions and integrating the full relative position information in the CRLB rather than just relative distances, which is significantly more challenging. We demonstrate in simulation the improvement achievable with this extension.

Notation: We write vectors and matrices with a bold font. The all-one vector of size $p$ is denoted $1_p$. The notation $x = \text{col}(x_1, \ldots, x_n)$ means that the vectors or matrices $x_i$ are stacked on top of each other, and $\text{diag}(A_1, \ldots, A_k)$ denotes a block diagonal matrix with the matrices $A_i$ on the diagonal. The nullspace of a matrix $A$ is denoted $\ker A$. For $A$ and $B$ symmetric matrices of the same dimensions, $A \succeq B$ means that $A - B$ is positive semidefinite and $A \succ B$ that it is positive definite. If $A$ is a symmetric matrix, $\lambda_{\text{min}}(A)$ and $\lambda_{\text{max}}(A)$ denote its minimum and maximum eigenvalues. The time derivative of a vector-valued function $t \rightarrow x(t)$ is denoted $\dot{x}$.

The expectation of a random vector $x$ is denoted $\mathbb{E}[x]$ and its covariance matrix $\text{cov}[x] = \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]$. For a differentiable function $f : \mathbb{R}^p \rightarrow \mathbb{R}^q$, $\frac{\partial f(p)}{\partial p}$ represents the $q \times p$ Jacobian matrix of $f$, with components $\frac{\partial f_i(p)}{\partial p_j}$ for $1 \leq i \leq q$, $1 \leq j \leq p$. When $q = 1$, $\frac{\partial^2 f(p)}{\partial p \partial p^\top}$ denotes the Hessian, i.e., the square matrix with components $\frac{\partial^2 f(p)}{\partial p_i \partial p_j}$. Finally, for a set $S$, $1_S(i) = 1$ if $i \in S$ and 0 otherwise.

II. Problem Statement

Consider a set of $N$ nodes in the $n$-dimensional Euclidean space, where $n = 2$ or $n = 3$. We fix a global reference frame denoted $\mathbf{3} = (\mathbf{0}, \vec{x}, \hat{y}, \hat{z})$ if $n = 3$ or $\mathbf{3} = (\mathbf{0}, \vec{x}, \hat{y})$ if $n = 2$. For $1 \leq i \leq N$, we write the coordinates of node $i$ in that frame $p_i := [x_i, y_i, z_i]^\top$ if $n = 3$ or $p_i := [x_i, y_i]^\top$ if $n = 2$, and we let $\mathbf{p} := \text{col}(p_1, \ldots, p_N) \in \mathbb{R}^{nN}$ denote the global spatial configuration of the nodes, which can vary with time. As illustrated on Fig. 1, some of these nodes are carried by mobile robots, while others could remain at fixed locations. We suppose that the coordinates of a subset $\mathcal{K}$ of
the nodes are perfectly known in $\mathcal{G}$, for $1 < |\mathcal{K}| : = K < N$, and refer to these nodes as anchors. The anchors could be placed at fixed locations or they could be mobile, as long as we can precisely localize them via external means, e.g., using accurate GNSS receivers. The other nodes, also mobile or fixed and whose positions are unknown and need to be estimated, are called tags in the following. They form a set denoted $\mathcal{U}$, with $|\mathcal{U}| : = U = N - K$.

Next, we assume that $P$ pairs of nodes, called ranging pairs, can measure their relative distance (with each such pair containing at least one tag). For such a pair of nodes $(i, j)$, we denote $d_{ij}$ the true distance between the nodes and $\tilde{d}_{ij}$ a corresponding measurement, to which both nodes $i$ and $j$ have access. In the following, we consider measurement models assuming either additive Gaussian noise

$$\tilde{d}_{ij} = d_{ij} + \nu_{ij}, \quad \nu_{ij} \sim \mathcal{N}(0, \sigma^2), \quad (1)$$

or multiplicative log-normal noise

$$\tilde{d}_{ij} = d_{ij} e^{\mu_{ij}}, \quad \mu_{ij} \sim \mathcal{N}(0, \sigma^2), \quad (2)$$

where the noise realizations $\nu_{ij}$ or $\mu_{ij}$ are independent for all $i, j$. We collect all the measured distances $\tilde{d}_{ij}$ in the vector $\tilde{d} = [\ldots, \tilde{d}_{ij}, \ldots]^T \in \mathbb{R}^P$. We also define an undirected graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, called the ranging graph, whose vertices $\mathcal{V}$ are the $N$ nodes and with an edge in $\mathcal{E}$ for each ranging pair and for each pair of anchors. In particular, the subgraph of $\mathcal{G}$ formed by the anchors is a complete graph, which is consistent with the fact that the distances between anchors are implicitly known from their coordinates. Two nodes linked by an edge in $\mathcal{G}$ are called neighbors and we denote by $\mathcal{N}_i$ the set of neighbors of $i$ or neighborhood of $i$, for $1 \leq i \leq N$. Let $E = P + \frac{K(K-1)}{2}$ be the total number of edges in $\mathcal{G}$.

A concrete implementation of the previous set-up is as follows. The nodes could correspond to RF transceivers capable of measuring their distance with respect to other nodes within their communication radius. Radiolocation protocols such as Two-Way Ranging (TWR), Time of Arrival (ToA) or Time Difference of Arrival (TDoA) [8], [34] use the timestamps of messages exchanged by the transceivers to estimate the ToF of these messages and deduce distance measurements, which can be assumed to be of the form (1), at least under line-of-sight signal propagation conditions. Another ranging method consists in measuring the strength of a received signal (RSS) to deduce the distance to the transmitter using a path loss propagation model [34]. This method typically leads to a distance measurement model of the form (2), assuming again a simple radio propagation environment [15], [35].

We assume that the nodes implement a cooperative localization scheme, in order to jointly produce an estimate $\hat{p}$ of all their coordinates $p_i$ in $\mathcal{G}$, based on the noisy measurements $\tilde{d}$ and the knowledge of the anchor coordinates. As we explain in Section III, the value of $p$ itself strongly influences the achievable accuracy of its estimate. Hence, we introduce in that section some real-valued functions $J_{\text{loc}} : \mathbb{R}^{nN} \rightarrow \mathbb{R}$ that can serve as localizability potentials, i.e., such that a lower value for $J_{\text{loc}}(p)$ means that the performance of an estimator at configuration $p$ is expected to be better. A localizability potential can then serve as an artificial potential for motion planning [25], to guide or constrain the motion of an MRS to configurations that are favorable for accurate cooperative localization. For example, one can generate a sequence of configurations $p(0), p(1), \ldots$, for the MRS of increasingly better localizability, by following the gradient descent scheme

$$p_{i,k+1} = p_{i,k} - \gamma_k \left( \frac{\partial J_{\text{loc}}(p_k)}{\partial p_i} \right)^T, \quad (3)$$

for each mobile node $i$, with $\gamma_k > 0$ a sequence of appropriate stepsizes. The potential $J_{\text{loc}}$ can also be added to other potentials that enforce collision avoidance constraints, connectivity maintenance [19], desired final positions [36] or coverage control tasks [37], etc.

A key issue when relying on artificial potentials to provide goal configurations to an MRS is to ensure that each mobile node $i$ can compute the gradient $(\partial J_{\text{loc}}(p(k))/\partial p_i)^T$ with respect to its coordinates in (3) by exchanging information only with its immediate neighbors in the communication network, which we assume here to coincide with the ranging graph (although in general the anchors will not need to communicate with each other). This ensures scalability to large networks and improves the robustness of the network against the loss of nodes. The design of distributed gradient descent schemes for the localizability potentials is discussed in Section V.

In summary, the problem considered in this paper is to first define appropriate functions that can serve as localizability potentials and then design distributed gradient descent algorithms for these potentials in order to deploy an MRS with ranging sensors while ensuring that its cooperative localization scheme performs well. In addition, we show in Section VI how to adapt the definition of the localizability potentials and the gradient descent scheme to a more complex situation where multiple tags can be carried by the same robot, which introduces additional constraints on the positions $p$ that should be taken into account by localization and motion planning algorithms. This set-up can be used in practice to provide more accurate full pose estimates for the robots, including their orientations.

**Remark 1.** In practice, the tags have access to their position $p$ only through their estimates $\hat{p}$. As a result, the gradient descent scheme (3) cannot be directly implemented, and a standard approach is to compute and follow the gradient at the current estimate, i.e., use $\partial J_{\text{loc}}(\hat{p}(k))/\partial p_i$ in (3). Alternatively, (3) can also be used to compute a sequence of steps, i.e., plan a future trajectory for the MRS, in which case we assume at the planning stage that the agents will be able to track that trajectory perfectly. In this paper, as in much of the related literature, we do not consider the tracking errors due to the fact that only imperfect position estimates are obtained during the deployment of an MRS. At least, the fact that the planned configurations promote accurate localization tends to mitigate the impact of these errors.

**III. LOCALIZABILITY POTENTIALS**

This section is concerned with defining artificial potentials that can be used as localizability potentials. The proposed definitions require that we first recall some elements from estimation theory related to the CRLB.
A. Constrained Cramér-Rao Lower Bound

We assume that the position estimator implemented by the MRS is unbiased, i.e., satisfies $\mathbb{E}[\hat{p}] = p$. We then focus on finding configurations $p$ for which the error covariance matrix $\mathbb{E}[(p - \hat{p})(\hat{p} - p)\top]$ for $\hat{p}$, which is then also the covariance matrix $\text{cov}(\hat{p})$, is “small” in some sense. More precisely, since the error covariance depends on the specific estimator used and can be difficult to predict analytically, we use the CRLB, a lower bound on the covariance of any unbiased estimator, to quantify the quality of a configuration $p$. Although this implicitly assumes that an estimator can be constructed to achieve or approach this lower bound, this methodology is commonly used in optimal experiment design and sensor placement [27], [28]. In general, the CRLB corresponds to the inverse of the Fisher Information Matrix (FIM), which we define below.

**Definition 1** (FIM). Let $x \in \mathbb{R}^p$ be a deterministic parameter vector and $y \in \mathbb{R}^q$ a random observation vector, for some positive integers $p, q$. Define $f : \mathbb{R}^q \times \mathbb{R}^p \rightarrow \mathbb{R}^+$ the Probability Density Function (PDF) of $y$, which depends on the parameter $x$, so that we write $f(y; x)$. Under some regularity assumptions on $f$ (see [24, Chap. 14]), the $p \times p$ Fisher Information Matrix (FIM) of this PDF is defined as

$$
\mathbf{F}(x) = -\mathbb{E}_y \left[ \frac{\partial^2 \ln f(y; x)}{\partial \mathbf{x} \partial \mathbf{x}^\top} \right].
$$

(4)

The matrix $\mathbf{F}(x)$ is symmetric and positive semi-definite.

In the position estimation problem, the parameters of interest are the node coordinates in the vector $p \in \mathbb{R}^{nN}$, whereas the random observations are contained in the vector $d$. As computed in [15], the FIM of the PDF $f(d; p)$ is an $nN \times nN$ matrix that depends on $p$ and can be decomposed into $n \times n$ blocks $\mathbf{F}_{ij}$ such that

$$
\mathbf{F}_{ij}(p) = \mathbf{F}_{ij}(p_{ij}) = -\frac{1}{\sigma^2} \mathbf{p}_{ij} \mathbf{p}_{ij}^\top 1_{N_i}(j), \quad \text{if } i \neq j,
$$

$$
\mathbf{F}_{ii}(p) = -\sum_{j \neq i} \mathbf{F}_{ij},
$$

(5)

where $p_{ij} := p_i - p_j$, and $\kappa = 1$ for the additive noise model (1) or $\kappa = 2$ for the multiplicative noise model (2).

Note however that estimating the anchor positions is not needed, since the locations of these nodes are known. The fact that $p_i := p_i$ for all $i \in K$, with these coordinates $p_i$ known, should be taken into account by an estimator of the tag positions, and hence should also be taken into account when bounding the covariance of these estimators. We can rely on the theory of CRLBs with equality constraints on the estimated parameters in order to include these trivial constraints on the anchor positions and later in Section VI also additional rigid constraints on the tag positions.

**Theorem 1** (Equality constrained CRLB [31]). Let $x \in \mathbb{R}^p$ be a deterministic parameter vector and $y \in \mathbb{R}^q$ a random observation vector, for some positive integers $p, q$. Let $h : \mathbb{R}^p \rightarrow \mathbb{R}^c$, for $c \leq p$, be a differentiable function such that $h(x) = 0$. Let $\hat{x}$ be an unbiased estimate of $x$ also satisfying $h(\hat{x}) = 0$ and with finite covariance matrix. Define $\mathbf{F}_c := \mathbf{A}^\top \mathbf{F} \mathbf{A}$, the constrained Fisher Information Matrix, where $\mathbf{A}$ is any matrix whose columns span ker $\frac{\partial h}{\partial x}$, and $\mathbf{F}$ is the FIM defined in (4). Then, the following inequality holds

$$
\text{cov} [\hat{x}] \succeq \mathbf{A} (\mathbf{F}_c)^\dagger \mathbf{A}^\top =: \mathbf{B}_c
$$

(6)

where $\dagger$ denotes the Moore-Penrose pseudo-inverse [38, p. 21].

Consider now the problem of estimating the vector of tag coordinates $p_t \in \mathbb{R}^{nU}$ based on the distance measurements $d$ and knowledge of the anchor coordinates $p_K \in \mathbb{R}^{nK}$. Order the nodes so that $p = \text{col}(p_t, p_K)$, and partition the FIM defined in (5) accordingly as

$$
\mathbf{F} = \begin{bmatrix}
\mathbf{F}_{tt} & \mathbf{F}_{tk} \\
\mathbf{F}_{kt} & \mathbf{F}_{kk}
\end{bmatrix},
$$

(7)

with in particular $\mathbf{F}_{tt}$ a symmetric positive semi-definite matrix of size $nU \times nU$. We then have the following result.

**Proposition 1.** Let $\hat{p}_t$ be an unbiased estimate of the tag positions $p_t$, based on the measurements $d$ and the knowledge of the anchor positions $p_K$. Then

$$
\text{cov} (\hat{p}_t) \succeq \mathbf{F}_{tt}^\dagger (p).
$$

(8)

Proof. This result is a corollary of Proposition 5 stated below, with $\mathbf{c}_e = 0$ in (28) and so $\mathbf{A}_{tt} = \mathbf{I}_{nU}$. \hfill \square

B. Localizability Potentials and Optimal Design

Given (8), the following functions are possible candidates to define potential functions that penalize configurations of the ranging network leading to poor localizability

$$
J_A(p) = \text{Tr} \left\{ \mathbf{F}_{tt}^{-1}(p) \right\} \quad \text{(A-Optimal Design)},
$$

(9)

$$
J_D(p) = -\ln \det (\mathbf{F}_{tt}(p)) \quad \text{(D-Optimal Design)},
$$

(10)

$$
J_E(p) = -\lambda_{\min} (\mathbf{F}_{tt}(p)) \quad \text{(E-Optimal Design)},
$$

(11)

assuming in the first two cases that $\mathbf{F}_{tt}(p)$ is invertible. In the following, we refer to the functions $J_A$, $J_D$ and $J_E$ as the A-Opt, D-Opt and E-Opt potentials respectively, using standard terminology from optimal experiment design [28], [39].

In each case, configurations $p$ for which $J(p)$ takes large values correspond to geometries for which the error covariance matrix of an unbiased position estimator will necessarily be “large” in a sense defined by the choice of potential. Hence, for (9), we have from (8) that $J_A(p)$ is a lower bound on $\text{Tr} \{\text{cov}(\hat{p}_t)\}$, which represents the total mean-squared error (MSE) of the unbiased estimator $\hat{p}_t$. Similarly, (10) corresponds to a lower bound on $\ln \det (\text{cov}(\hat{p}_t))$, which would be equal (up to a constant) to the statistical entropy of $\hat{p}_t$, if this estimate were to follow a normal distribution. Finally, still assuming $\mathbf{F}_{tt} > 0$, minimizing $J_E$ in (11) aims to minimize the maximum eigenvalue of $\mathbf{F}_{tt}^{-1}$ (equal to $1/\lambda_{\min} (\mathbf{F}_{tt})$), which is a lower bound on the maximum eigenvalue or induced 2-norm of $\text{cov}(\hat{p}_t)$. Potentials like $J_E$ are often used to maintain the connectivity [17]–[19] or rigidity [22], [23] of an MRS, which are closely related problems.

Once a potential has been chosen, it can be used to move the nodes to configurations of low potential values, where the localization accuracy is expected to be high. This can be done...
for example by descending the gradient of the potential, as discussed in Sections V and VI.

Remark 2. Another a priori possible potential is

$$J_T(p) = -\text{Tr} \{ F_H(p) \}.$$  

Configurations $p$ that minimize this potential are called $T$-optimal designs [28]. However, in our case we can compute

$$J_T(p) = -\alpha \sum_{i,j \in E} d_{ij}^{2-2\alpha},$$

with $\alpha$ a positive constant. Hence, in the case of additive Gaussian noise (1), $\kappa = 1$ and $J_T$ is constant, so that it cannot be used to optimize $p$. In the case of multiplicative noise (2), we have $\kappa = 2$ so $J_T(p) = -\alpha \sum_{i,j \in E} d_{ij}^{-2}$ becomes a simple attractive potential. In this case, $J_T$ cannot be used alone as a potential, since its global minimum is trivially achieved when all agents occupy the same position. In view of these remarks, $J_T$ is not considered further in the following.

IV. PROPERTIES OF THE FISHER INFORMATION MATRIX

In this section, we study certain algebraic properties of the FIM that are useful for the design of algorithms in the next sections. For this, we establish connections between the FIM and rigidity theory.

A. Infinitesimal Rigidity

For the ranging graph $G = (E, V)$, the incidence matrix $H \in \mathbb{Z}^{E \times N}$ is defined by first assigning an arbitrary direction $i \rightarrow j$ to each edge $\{i, j\} \in E$, and then setting each element as follows:

$$H_{i,j} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j, \\ 0 & \text{otherwise.} \end{cases}$$

For concreteness, we use throughout the paper the lexicographic ordering to order the rows $i \rightarrow j$ and hence the rows of $H$. As a result, the rows of $H$ corresponding to pairs of tags (in $\mathcal{U} \times \mathcal{U}$) appear first, followed by pairs in $\mathcal{U} \times K$ and finally by pairs of anchors, in $K \times K$. Given a ranging graph $G$, a framework is a pair $(G, p)$, where the vector $p \in \mathbb{R}^{nN}$ contains the positions of all agents, as before.

The rigidity function $r : \mathbb{R}^{nN} \rightarrow \mathbb{R}^E$ of a framework $(G, p)$ is defined componentwise by

$$r(G, p)_{i \rightarrow j} = \frac{1}{2} \| p_{ij} \|^2, \quad \forall \{i, j\} \in E,$$

and its rigidity matrix $R(G, p) \in \mathbb{R}^{E \times nN}$ is the Jacobian $\partial r / \partial p$ of the rigidity function [12], [22], which can be written explicitly as

$$R(G, p) = \text{diag}(p^T_{ij}, ..., p^T_{ij}) [H \otimes I_n].$$

In other words, the row $i \rightarrow j$ of $R(G, p)$ is

$$[0 \ldots 0 p^T_{ij} 0 \ldots 0 -p^T_{ij} 0 \ldots 0]$$

with $p^T_{ij}$ occupying the $i^{th}$ block of $n$ coordinates and $-p^T_{ij}$ the $j^{th}$ block. Next, when the node positions vary with time, consider motions that do not change the distances between nodes in ranging pairs, in other words, motions that keep the rigidity function constant. These motions must then satisfy

$$\frac{dr(G, p)}{dt} = R(G, p) \frac{dp}{dt} = 0,$$

i.e., the corresponding velocity vectors $dp/dt$ must lie in the kernel of $R(G, p)$. This constraint is rewritten more explicitly in the following definition.

Definition 2. An infinitesimal motion of a framework. An infinitesimal motion of a framework $(G, p)$ is any vector $v = \text{col}(v_1, \ldots, v_N)$ in $\mathbb{R}^{nN}$, such that $v \in \ker R(G, p)$. Equivalently, for each edge $\{i, j\} \in E$, we have $p_i^T (v_i - v_j) = 0$.

Any framework admits a basic set of infinitesimal motions, namely, the Euclidean infinitesimal motions of the framework [12], [40], which can be defined for $n = 3$ as

$$\text{Euc}_1^p = \{ \text{col}(v + \omega \times p_1, \ldots, v + \omega \times p_n) | v, \omega \in \mathbb{R}^3 \},$$

and for $n = 2$, with the notation $p_i = [x_i, y_i]^T$,

$$\text{Euc}_2^p = \{ \text{col}(v + \omega \begin{bmatrix} y_1 \\ -x_1 \end{bmatrix}, \ldots, v + \omega \begin{bmatrix} y_n \\ -x_n \end{bmatrix}) | v \in \mathbb{R}^2, \omega \in \mathbb{R} \}.$$  

These infinitesimal motions correspond to the global rigid translations and rotations of the whole framework, and it is immediate to verify that the subspace $\text{Euc}_p^1$ is always contained in $\ker R(G, p)$. Infinitesimally rigid frameworks do not admit other infinitesimal motions, which would correspond to internal deformations.

Definition 3 (Infinitesimal rigidity). A framework $(G, p)$ in $\mathbb{R}^{nN}$ is called infinitesimally rigid if all its infinitesimal motions are Euclidean, i.e., if $\ker R(G, p) = \text{Euc}_p^1$.

The following result provides a basis of $\text{Euc}_p^2$ and is used in Section VI. When $n = 3$, with $e_x, e_y, e_z$ the standard unit vectors in $\mathbb{R}^3$, define $v_T = 1_N \otimes e_x$ as well as $v_R = \text{col}(e_x \times p_1, \ldots, e_x \times p_n)$, for $\xi \in \{ x, y, z \}$. Similarly, if $n = 2$ and $e_x, e_y$ are the standard unit vectors in $\mathbb{R}^2$, define $v_T = 1_N \otimes e_x, v_T = 1_N \otimes e_y$ and

$$v_R = \text{col}(\begin{bmatrix} -y_1 \\ x_1 \end{bmatrix}, \ldots, \begin{bmatrix} -y_n \\ x_n \end{bmatrix}).$$

Proposition 2. Suppose that $n \geq n$. If $n = 2$ and at least 2 nodes are at distinct locations, the dimension of $\text{Euc}_p^2$ is 3 and a basis of this subspace is given by $(v_T, v_T, v_R)$. If $n = 3$ and we have at least 3 nodes that are not aligned, the dimension of $\text{Euc}_p^3$ is 6 and a basis of this subspace is given by $(v_T, v_T, v_R, v_T, v_R, v_R)$.  

Proof. We provide a proof for $n = 3$, the case $n = 2$ is similar. The fact that the vectors in the proposition span $\text{Euc}_p^3$ is clear by definition, so it is sufficient to prove their independence. Consider a linear combination equal to zero

$$\alpha_1 v_T + \alpha_2 v_T + \alpha_3 v_T + \alpha_4 v_R + \alpha_5 v_R + \alpha_6 v_R = \text{col} (v + \omega \times p_1, \ldots, v + \omega \times p_n) = 0,$$
where \( \mathbf{v} = [\alpha_1, \alpha_2, \alpha_3]^T \) and \( \omega = [\alpha_4, \alpha_5, \alpha_6]^T \). Suppose that the nodes indexed by \( i, j \) and \( k \) are not aligned. We have from the equation above \( \mathbf{v} = -\omega \times \mathbf{p}_i \), and so
\[
\omega \times (\mathbf{p}_j - \mathbf{p}_i) = \omega \times (\mathbf{p}_k - \mathbf{p}_i) = 0.
\]
Since \((\mathbf{p}_j - \mathbf{p}_i)\) and \((\mathbf{p}_k - \mathbf{p}_i)\) are by assumption independent, this gives \( \omega = 0 \) and hence \( \mathbf{v} = 0 \). This proves the independence of the vectors in the proposition, which therefore form a basis of \( \text{Euclidean}_3 \).

### B. Relations between the Rigidity Matrix and the FIM

Throughout this section, we consider the set of nodes (tags and anchors) to be at positions \( \mathbf{p} \), with corresponding ranging graph \( \mathcal{G} \). This defines a framework \((\mathcal{G}, \mathbf{p})\), as discussed in the previous section. The FIM \( \mathbf{F} \) is given by (5), whereas the rigidity matrix \( \mathbf{R} := \mathbf{R}(\mathcal{G}, \mathbf{p}) \) is given by (13).

#### Proposition 3.
We have \( \mathbf{F} = \mathbf{R}^T \mathbf{Q} \mathbf{R} \), for the positive definite matrix \( \mathbf{Q} = \text{diag} \left( \frac{1}{1/(d_{ii}^2 \sigma_i^2)}, \ldots \right) \in \mathbb{R}^n \times \mathbb{R}^n \).

To explain this result, remark that \( \mathbf{F} \) in (5) has a structure similar to the Laplacian matrix \( \mathbf{L} \) of the graph \( \mathcal{G} \) [41, Chapter 12]. The expression of Proposition 3 then corresponds to the standard relationship \( \mathbf{L} = \mathbf{H}^T \mathbf{H} \) between the incidence matrix and the usual Laplacian of an undirected graph. Hence, the FIM \( \mathbf{F} \) can be considered as a type of weighted Laplacian matrix. In [22], matrices of the form \( \mathbf{R}^T \mathbf{Q} \mathbf{R} \), for any diagonal matrix \( \mathbf{Q} \), are called (weighted) “symmetric rigidity matrices”. Hence, with this terminology, Proposition 3 says that the FIM is a symmetric rigidity matrix, for a specific set of weights in \( \mathbf{Q} \) determined by the properties of the measurement noise model. In particular, these weights depend inversely on the (true) distances between ranging nodes.

**Proof.** Starting from (13), we have
\[
\mathbf{R}^T \mathbf{Q} \mathbf{R} = (\mathbf{H}^T \otimes \mathbf{I}_n) \text{diag} \left( \frac{1}{d_{ii}^2 \sigma_i^2}, \ldots \right) (\mathbf{H} \otimes \mathbf{I}_n).
\]
Hence, for \( i \neq j \), the block \( i, j \) of \( \mathbf{R}^T \mathbf{Q} \mathbf{R} \) is
\[
[\mathbf{R}^T \mathbf{Q} \mathbf{R}]_{ij} = \sum_{e \in E} H_{ei} H_{ej} Q_{ee} = -\frac{P_{ij} P_{ij}^T}{d_{ij}^2 \sigma^2} 1_{N_i}(j) = F_{ij},
\]
using the fact that \( H_{ei} H_{ej} = -1 \) if \( e \) is \( i \to j \) and 0 otherwise. Similarly, for all \( i \)
\[
[\mathbf{R}^T \mathbf{Q} \mathbf{R}]_{ii} = \sum_{e \in E} H_{ei} H_{ei} Q_{ee} = \sum_{j \in N_i} \frac{P_{ij} P_{ij}^T}{d_{ij}^2 \sigma^2} = F_{ii}.
\]

The following result then follows immediately from the fact that \( \mathbf{Q} \succeq 0 \) in Proposition 3.

#### Corollary 1.
We have \( \ker \mathbf{F} = \ker \mathbf{R} \).

We can now establish a link between infinitesimal rigidity and the invertibility of the partial FIM \( \mathbf{F}_U \) appearing in (7).

#### Theorem 2.
If the framework \((\mathcal{G}, \mathbf{p})\) is infinitesimally rigid and contains at least \( n \) anchors at distinct locations, and if moreover when \( n = 3 \) at least 3 of these anchors are not aligned, then \( \mathbf{F}_U \) is positive definite.

**Proof.** We give the proof in the more involved case \( n = 3 \). With the assumed ordering of nodes and edges, the rigidity matrix has the following block structure
\[
\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{R}_3 \end{bmatrix}, \quad \text{with } \mathbf{R}_1 \in \mathbb{R}^{p \times U}, \mathbf{R}_3 \in \mathbb{R}^{K(3-1) \times K}.
\]

In other words, the rows of the matrix \( \mathbf{R}_1 \) correspond to the edges internal to \( \mathcal{U} \) and between \( \mathcal{U} \) and \( \mathcal{K} \), whereas \( \mathbf{R}_3 \) is the rigidity matrix of the complete subgraph formed by the anchors and the links between them. Now, we have \( \ker \mathbf{F}_U = \ker \mathbf{R} \). Consider some vector \( \mathbf{x}_1 \in \mathbb{R}^U \) with \( \mathbf{x}_1 \in \ker \mathbf{R}_1 \). Then,
\[
\mathbf{R} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{R}_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{bmatrix} = 0,
\]

hence \( \text{col}(\mathbf{x}_1, \mathbf{0}) \) is in \( \ker \mathbf{R} \). Since \( \mathcal{G} \) is infinitesimally rigid, there must exist \( \mathbf{v}, \omega \in \mathbb{R}^3 \) such that
\[
\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{bmatrix} = \text{col}(\mathbf{v} + \omega \times \mathbf{p}_1, \ldots, \mathbf{v} + \omega \times \mathbf{p}_N).
\]

In particular, for the 3 anchors that are not aligned, indexed by \( i, j \), and \( k \), we must have
\[
\mathbf{v} + \omega \times \mathbf{p}_i = \mathbf{v} + \omega \times \mathbf{p}_j = \mathbf{v} + \omega \times \mathbf{p}_k = 0.
\]

From this, we conclude as in the proof of Proposition 2 that \( \mathbf{v} = \omega = 0 \), which in turns implies \( \mathbf{x}_1 = 0 \). Hence \( \ker \mathbf{F}_U = \{0\} \), i.e., \( \mathbf{F}_U \succeq 0 \).

#### Remark 3.
If we have only one tag, then one can show that \( \mathbf{F}_U \) is invertible if and only if we have at least \( n + 4 \) anchors and the nodes’ locations span an affine space of full dimension \( n \) (i.e., we have \( n + 3 \) non aligned nodes if \( n = 2 \), and \( 4 \) non coplanar nodes if \( n = 3 \)). Note that if we have just \( n \) anchors, we cannot localize uniquely the tag in general, even with perfect measurements, because the intersection of \( n \) spheres in \( \mathbb{R}^n \) gives two possible locations. Hence, even when \( \mathbf{F}_U \) is invertible, the localization problem might not be uniquely solvable. Unicity of the localization solution can be characterized by the stronger notion of global rigidity [14], which however is more complex to check if \( n = 2 \) and for which no exact test is currently known if \( n = 3 \).

Theorem 2 can be used to produce an initial node placement and choose ranging links to guarantee that \( \mathbf{F}_U \) is already invertible at the start of the deployment. For this, we should ensure that \((\mathcal{G}, \mathbf{p})\) is infinitesimally rigid. One convenient way to satisfy this condition (in fact, the stronger condition of global rigidity) is to construct a triangulation graph [14], [42]: starting from a set of at least \( n + 1 \) anchors, we add tags one by one, with each new tag connected to at least \( n + 1 \) previous nodes that are in general position (3 non-aligned nodes if \( n = 2 \), 4 non-coplanar nodes if \( n = 3 \)). Although this construction requires more anchors and links than the strict minimum necessary for the invertibility of \( \mathbf{F}_U \), the resulting network supports efficient distributed localization algorithms that are robust to measurement noise [42].
V. DISTRIBUTED GRADIENT COMPUTATIONS FOR THE LOCALIZABILITY POTENTIALS

In order to implement the gradient descent scheme (3), in Section V-A we provide analytical forms for the gradients of the localizability potentials (9), (10) and (11). Then, in Sections V-B and V-C, we describe decentralized deployment algorithms by showing how each agent can compute its components of the gradient of the chosen localizability potential, using its own local information as well as data obtained from its neighbors in the ranging graph.

A. Partial Derivatives of the FIM

Irrespective to the potential considered, we need to evaluate the derivative of the FIM \( F_U \) in (7) with respect to any coordinate \( \xi_i \in \{x_i, y_i, z_i\} \) of a mobile agent \( i \) (anchor or tag) located at \( p_i = [x_i, y_i, z_i]^T \). We provide formulas for the case \( n = 3 \), the case \( n = 2 \) being similar. Define the notation \( \xi_{ij} = \xi_i - \xi_j \) and \( \gamma_{ij} = \frac{\kappa}{\sigma^2 \delta^{(i,j-1)}} I_N(j) \). For \( F_{ij}, i \neq j \), the 3 \times 3 blocks introduced in (5), we find

\[
\begin{align*}
\frac{\partial F_{ij}}{\partial x_i} &= \gamma_{ij} \begin{bmatrix}
 x_i^2 x_{ij} & 2 x_i x_{ij}^2 - \frac{d_{ij}^2 y_{ij}}{\kappa} & 0 \\
 2 x_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{2 \kappa} & x_{ij} y_{ij}^2 - \frac{d_{ij}^2 z_{ij}}{\kappa} & 0 \\
 0 & 0 & \delta_{ij}
\end{bmatrix}, \\
\frac{\partial F_{ij}}{\partial y_i} &= \gamma_{ij} \begin{bmatrix}
 x_i y_{ij}^2 - \frac{d_{ij}^2 y_{ij}}{2 \kappa} & 2 y_i y_{ij}^2 - \frac{d_{ij}^2 y_{ij}}{\kappa} & 0 \\
 2 x_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{2 \kappa} & x_{ij} y_{ij}^2 - \frac{d_{ij}^2 z_{ij}}{\kappa} & 0 \\
 0 & 0 & \delta_{ij}
\end{bmatrix}, \\
\frac{\partial F_{ij}}{\partial z_i} &= \gamma_{ij} \begin{bmatrix}
 x_i y_{ij}^2 & x_i y_{ij} & y_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{\kappa} \\
 x_i y_{ij} & x_i y_{ij} + \frac{d_{ij}^2 y_{ij}}{\kappa} & y_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{2 \kappa} \\
 y_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{\kappa} & y_i^2 y_{ij} - \frac{d_{ij}^2 y_{ij}}{2 \kappa} & \delta_{ij}
\end{bmatrix},
\end{align*}
\]

where the symbol \( \ast \) replaces symmetric terms. These expressions are sufficient to compute the whole matrix \( \partial F_{ij}/\partial \xi_i \), because \( F_{ij} = F_{ji}, F_{kk} = -\sum_{i \in N_k} F_{ki} \), and \( \partial F_{kl}/\partial \xi_i = 0 \) if \( k \neq l \) and \( i \notin \{k, l\} \).

Using standard differentiation rules [38], the partial derivatives of the A-Opt potential (9) are

\[
\begin{align*}
\frac{\partial J_A(p)}{\partial \xi_i} &= \partial \text{Tr} \left\{ F_U^{-1} \right\} = -\text{Tr} \left\{ F_U^{-2} \partial F_U/\partial \xi_i \right\}.
\end{align*}
\]

Similarly, we can compute the derivatives of the D-Opt potential (10) as

\[
\begin{align*}
\frac{\partial J_D(p)}{\partial \xi_i} &= -\partial \ln \det F_U/\partial \xi_i = -\text{Tr} \left\{ F_U^{-1} \partial F_U/\partial \xi_i \right\}.
\end{align*}
\]

Finally, if \( \lambda_{\min}(F_U) \) is a non-repeated eigenvalue with associated unit norm eigenvector \( v \), we can compute the derivative of the \( E \)-Opt potential (11) as [43, p. 565]

\[
\frac{\partial J_E(p)}{\partial \xi_i} = -\lambda_{\min}(F_U)/\partial \xi_i = -v^T \partial F_U/\partial \xi_i v.
\]

Hence, we can in principle compute the gradient of the chosen localizability potential, using the expressions for the FIM and its derivatives. However, in practice we would also like to be able to implement these computations in a distributed manner, in order to obtain deployment strategies that can scale to large multi-robot systems.

B. Decentralized Gradient Computations for the D- and A-Opt Potentials

We propose now a new method to estimate in a distributed way the gradient of the \( D \)- and \( A \)-Opt potentials at a given configuration \( p \), which have similar expressions, see (16) and (17). As mentioned in Remark 1, we assume that the nodes have already executed a localization algorithm such as the one in [42] to estimate \( p \), and we ignore the effect of the location estimation error in the gradient computation. Hence, we omit \( p \) from the notation in the following, writing \( F_U \) instead of \( F_U(p) \). The method essentially relies on inverting \( F_U \) in a decentralized manner, which we discuss first.

1) Auxiliary Problem: Suppose that each tag \( i \in U \) knows initially a matrix \( E_i \in \mathbb{R}^{n \times n} \), for some integer \( n \), and the tags need to compute \( F_U^{-1} E \) in a distributed manner over the network \( G \), where \( E = \text{col}(E_1, \ldots, E_U) \in \mathbb{R}^{nU \times m} \). This is equivalent to solving in a decentralized manner the linear system \( F_U x = E \), with the variable \( x \in \mathbb{R}^{nU \times m} \). A special case of this problem is to compute \( F_U^{-1} \), when \( E = I_{nU} \).

Consider the following system of differential equations

\[
\dot{x}(t) = -F_U x(t) + E, \quad x(0) = x_0.
\]

If \( F_U \succ 0 \), as guaranteed for instance by Theorem 2, then \( -F_U \) has strictly negative eigenvalues, i.e., is stable, so the solution \( x(t) \) to the system (19) converges to the solution \( F_U^{-1} E \) of the linear system as \( t \to \infty \), no matter the choice of initial condition \( x_0 \). A discrete-time version of the flow (19) can be implemented for \( k \geq 0 \) as

\[
x_{k+1} = x_k - \eta_k (F_U x_k - E),
\]

for some stepsizes \( \eta_k \), which reads more explicitly for each tag \( 1 \leq i \leq U \)

\[
x_{i,k+1} = \eta_k \sum_{j \in N_i \cap U} F_{ij} (x_{i,k} - x_{j,k}) + \left( I_n + \eta_k \sum_{j \in N_i \cap U} F_{ij} \right) x_{i,k} + E_i.
\]

Again, the iterates \( x_k \) converge to the desired solution \( F_U^{-1} E \) if we choose for example \( \eta_k = \eta \) constant and sufficiently small (namely, as long as \( \eta < 2/\lambda_{\max}(F_U) \)). The iterations (20) can be implemented in a decentralized manner by the tags, i.e., at each step \( k \) tag \( i \) only need to exchange its matrix \( x_i \) with its neighboring tags. This also requires that tag \( i \) knows \( F_{ij} \) for \( j \in N_i \), which is the case if prior to the iterations, the nodes (tags and anchors) broadcast their position (estimates) to their neighbors. When the iterations have converged, the \( n \times m \) matrix \( x_i \) at tag \( i \) represents the \( i \)-th block of rows of \( F_U^{-1} E \), i.e., \( F_U^{-1} E = \text{col}(x_1, \ldots, x_U) \).

Remark 4. The iterations (20) correspond to Richardson iterations to solve the linear system \( F_U x = E \) in a decentralized
way [44]. Other distributed iterative methods could be used, such as the Jacobi over-relaxation iterations

$$x_{i,k+1} = (1 - \eta)x_{i,k} + \eta F_{ii}^{-1} \left( E_i - \sum_{j \in N_i \cap U} F_{ij} x_{j,k} \right),$$

with potentially better convergence properties, but a detailed discussion of such alternatives, which can be found in [44, Chapter 2], is outside of the scope of this paper.

2) Application to compute $\partial J_D/\partial \xi_i$: To implement the gradient descent scheme (3) for D-optimization, each mobile node $i$ (tag or anchor) needs to compute $\partial J_D/\partial \xi_i$ for $\xi_i \in \{x_i, y_i, z_i\}$, which is given by (17). Denote $M = F_{ii}^{-1} \in \mathbb{R}^{nU \times nU}$ and its $n \times n$ blocks $M_{ij}$, for $1 \leq i, j \leq U$. First, the tags run the iterations (20), with the matrix $E = I_{nU}$. That is, tag $j$ uses the matrix $E_j = e_j^i \otimes I_n$, where $e_j^i$ is the $j$th unit vector in $\mathbb{R}^n$. After convergence, tag $j$ stores an approximation of the matrix $M_j = [M_{jj}, \ldots, M_{jU}] \in \mathbb{R}^{n \times nU}$. Next, note from (15) that the only $n \times n$ non-zero blocks $\partial F_{ij}/\partial \xi_i$, with $0 \leq k, l \leq U$, are those for which: a) $k = l$ and $k \in N_i$; b) $k = l = i$; c) $k = i$ and $l \in N_i$; or d) $l = i$ and $k \in N_i$. Moreover, if $i$ is a mobile anchor (so $i \geq U + 1$), only case a) can occur. From this remark, we can derive the following expressions. If $i \in U$,

$$\frac{\partial J_D(p)}{\partial \xi_i} = \sum_{j \in N_i \cap U} \text{Tr} \left\{ (M_{jj} + M_{ii} - 2M_{ij}) \frac{\partial F_{ij}}{\partial \xi_i} \right\}, \quad (21)$$

and if $i \in K$,

$$\frac{\partial J_D(p)}{\partial \xi_i} = \sum_{j \in N_i \cap K} \text{Tr} \left\{ M_{jj} \frac{\partial F_{ij}}{\partial \xi_i} \right\}. \quad (22)$$

Since we assume that each node knows an estimate of its coordinates and of its neighbors’ coordinates, node $i$ can obtain from its neighbor tags $j$ the terms $\text{Tr} \{ M_{jj} \frac{\partial F_{ij}}{\partial \xi_i} \}$, and also compute the terms $\text{Tr} \{ (M_{ii} - 2M_{ij}) \frac{\partial F_{ij}}{\partial \xi_i} \}$ if $i \in U$. Hence, overall this provides a method allowing each mobile node $i$ to compute $\partial J_D/\partial \xi_i$ by communicating only with its neighbors. Algorithm 1 summarizes the distributed gradient computation procedure for D-optimization.

**Algorithm 1: D-Opt distributed gradient computation**

**Data:** Each node $i$ knows an estimate of its $p_i$ from a localization algorithm, or exactly if $i \in K$.

**Result:** Each mobile node $i$ knows $\partial J_D(p_i)/\partial p_i$.

Each node $i \in U \cup K$ broadcasts $p_i$ to its neighbors; The tags run the iterations (20) until acceptable convergence, with $E_j = e_j^i \otimes I_n$ for tag $j$, and each tag $j$ stores the resulting matrix $M_j$; Each mobile tag $i$ computes

$$\sum_{j \in N_i \cap U} \text{Tr} \left\{ (M_{jj} - 2M_{ij}) \frac{\partial F_{ij}}{\partial \xi_i} \right\};$$

Each tag $j$ computes and sends $\text{Tr} \{ M_{jj} \frac{\partial F_{ij}}{\partial \xi_i} \}$ to each of its mobile neighbors $i \in N_j$ (i tag or anchor); Each mobile node $i$ computes its gradient using (21) or (22);

The same steps can be used to compute the gradient (16) at each mobile node for A-optimization. The only difference is that the matrices $M_i$ above should represent rows of $F_{ii}^{-1}$ instead of $F_{ii}^{-1}$. For this, the tags first compute the rows $\tilde{M}_i$ of $F_{ii}^{-1}$ using the iterations (20). Then, we restart these iterations but now replacing the matrices $E_i = e_i^i \otimes I_n$ by $\tilde{M}_i$. This computes an approximation of $F_{ii}^{-1}F_{ii}^{-1} = F_{ii}^{-2}$, as desired.

C. Decentralized Computation of E-Opt Gradient

The decentralized computation of the gradient of the E-Opt potential can be done using the methodology developed in [19] for the standard Laplacian, also used in [22] for the symmetric rigidity matrix. Hence, our presentation is brief and focuses on adapting this methodology to $F_{ii}(p)$.

Using the sparsity of $F_{ii}$, if $i \in U$, we can rewrite (18) as

$$\frac{\partial J_E(p)}{\partial \xi_i} = \sum_{j \in N_i \cap U} (v_i - v_j)^T \frac{\partial F_{ij}}{\partial \xi_i} (v_i - v_j)^T + v_i^T \sum_{j \in N_i \cap U} \frac{\partial F_{ij}}{\partial \xi_i} v_j, \quad (23)$$

and if $i \in K$,

$$\frac{\partial J_E(p)}{\partial \xi_i} = \sum_{j \in N_i \cap K} v_j^T \frac{\partial F_{ij}}{\partial \xi_i} v_j. \quad (24)$$

where $v = \text{col}(v_1, \ldots, v_U) \in \mathbb{R}^{nU}$. Computing these expressions requires a decentralized algorithm to estimate the components of $v$, a unit norm eigenvector associated with $\lambda_1 := \lambda_{\min}(F_{ii})$.

1) Power-iteration eigenvector estimator: To compute $v$ in a decentralized manner, consider the solution $t \mapsto w(t) \in \mathbb{R}^{nU}$ to the following differential equation, adapted from [19],

$$w = -[\beta F_{ii} + \mu(nU)^{-1}\|w\|^2 - 1]I_n w, \quad (25)$$

with an initial condition $w_0 := w(0)$ and $\beta, \mu > 0$.

**Proposition 4.** If $\mu > \lambda_1 \beta$ and $w_\infty^T v \neq 0$, then the solution $w(t)$ to (25) converges to an eigenvector $w_\infty$ of $F_{ii}$, associated with $\lambda_1$ and proportional to $v$.

**Proof.** Follows from the argument in the appendix of [19]. □

In practice, we can choose $w_0$ randomly to fulfill the condition $w_\infty^T v \neq 0$ with probability one. To set the gains $\beta, \mu$, note that $\text{Tr} \{ F_{ii} \} > \lambda_1$ since $F_{ii} > 0$. Then, for the additive measurement noise model (1), we have $\text{Tr} \{ F_{ii} \} \leq 2P$. So, if we choose $\beta = \sigma^2/(2P)$ and $\mu > 1$, the condition of Proposition 4 is satisfied. For the log-normal model (2), we have $\text{Tr} \{ F_{ii} \} \leq \frac{2}{\sigma^2} \sum_{i,j} d_{ij}^2$. Hence, if we set again $\beta = \sigma^2/(2P)$ and now $\mu > 1/d_{\min}^2$, such that $d_{ij} \geq d_{\min}$ for all $i, j$, then the condition of Proposition 4 is satisfied. The minimum distance $d_{\min}$ between robots could be enforced as part of a collision avoidance scheme.

An estimation algorithm for $v$ is obtained by discretizing (25), leading to the following iterations for each agent $i \in U$

$$w_{i,k+1} = w_{i,k} - \eta \left( \mu(1 - s_k)w_{i,k} + \beta \sum_{l \in (N_i \cup \{i\}) \cap U} F_{il} w_{l,k} \right), \quad (26)$$
where \( \eta_k > 0 \) is a sufficiently small step-size and \( s_k := \|w_k\|^2/nU \). All the terms in (26) can be obtained locally by node \( i \) using one-hop communication with its neighbors, except for the global average \( s_k \), which can be computed by a consensus algorithm as described next. The last step is to normalize \( w_\infty \), obtained after convergence in (26). This can again be done by each individual agent, since \( v := w_\infty/\sqrt{nUS_\infty} \) is a unit-norm vector.

2) Estimation of \( s_k \) via a consensus algorithm: Since \( s_k = \|w_k\|^2/(nU) = \frac{1}{U} \sum_{i=1}^U (\|w_{i,k}\|^2/n) \), this term can be computed by the tags using a decentralized averaging consensus algorithm.

We assume for simplicity that the graph of the tags \( G_U \) is connected. To solve the averaging problem, each tag \( i \) initializes a variable \( \hat{s}_{i,0} := \|w_{i,k}\|^2/n \). Then, they execute in a distributed manner the iterations

\[
\hat{s}_{i+1} = L \hat{s}_i, \forall l \geq 0,
\]

where \( \hat{s}_i = \text{col}(\hat{s}_{i,1}, \ldots, \hat{s}_{i,l}) \), and \( L \) is a doubly stochastic matrix of weights \( L_{ij} \) associated with the edges of \( G_U \) (i.e., \( \sum_{k=1}^U L_{ik} = \sum_{k=1}^U L_{ji} = 1 \), for \( 1 \leq i \leq U \), and \( L_{ij} = 0 \) if \( j \notin N_i \)), for instance the Metropolis-Hastings weights

\[
\begin{align*}
L_{ij} &= 1_{N_i \cap U}(j)(1 + \max(|N_i|, |N_j|))^{-1}, \forall i \neq j, \\
L_{ii} &= 1 - \sum_{k=1}^U L_{ik}.
\end{align*}
\]

We then have \( \hat{s}_i \rightarrow s_k U \) [45, p. 58], so that each tag knows after convergence the scalar value \( s_k \) needed for (26).

Algorithm 2 summarizes the decentralized computation of the estimate \( \hat{v}_i \) of the \( i \)-th component of \( v \) by a given tag \( i \in U \). After decentralized estimation of \( v \) by the tags, each mobile agent \( i \) can compute its components of the gradient of \( J_E \) from (23) or (24) by communicating with its neighbors.

**Algorithm 2: Estimation of \( v_i \) by tag \( i \in U \).**

**Data:** \( w_{i,0} \) random, \( L, \mu, \beta, n_{iter}, \hat{n}_{iter} \)

for \( 0 \leq k \leq n_{iter} \) do

\[
\hat{s}_{i,0} = \|w_{i,k}\|^2/n;
\]

for \( 0 \leq l \leq \hat{n}_{iter} \) do

\[
\hat{s}_{i+1} = L_i \hat{s}_i + \sum_{j \in N_i \cap U} L_{ij} \hat{s}_{j,l};
\]

end

compute \( w_{i,k+1} \), setting \( s_k := \hat{s}_{i,\hat{n}_{iter}} \) in (26).

end

transmit \( \hat{v}_i := \frac{w_{i,n_{iter}}}{\sqrt{nUS_{i,n_{iter}}}} \) to the neighborhood;

**Remark 5.** If \( G_U \) is not fully connected, there exists an \( U \times U \) permutation matrix \( P \) such that \( F_U := (P \otimes I_n)^{-1} F_U (P \otimes I_n) = \text{diag}(F_{S_1}, \ldots, F_{S_l}, \ldots) \) is block diagonal, where each \( S_i \) represents a subset of connected tags. Hence, the minimal eigenvalue \( \lambda \) of \( F_U \) is among the minimal eigenvalues \( \lambda_{S_i} \) of the blocks \( F_{S_i} \). Therefore, each subset \( S_i \) can use Algorithm 2 to compute its eigenvector \( v_{S_i} \) associated to \( \lambda_{S_i} := v_{S_i}^T F_{S_i} v_{S_i} \). On the other hand, the graph \( G \) with all nodes is assumed rigid and hence fully connected. This allows comparing the \( \lambda_{S_i} \) through the network \( K \) formed by the anchors in order to find \( \lambda := \min_{S_i} \lambda_{S_i} \) corresponding to the subset \( S^* \). Since \( F_U \) is block diagonal, its eigenvector

**VI. LOCALIZABILITY OPTIMIZATION FOR RIGID BODIES**

A. Constrained Localizability Optimization

In this section, we consider scenarios where mobile robots can carry several tags, see Fig. 2. Hence, the relative motion and position of some tags are constrained by the fact that they are attached to the same rigid body. More generally, let \( f : \mathbb{R}^n U \rightarrow \mathbb{R}^C \) be a known function defining \( C \) constraints \( f_i(p_U) = 0 \) that the tag positions must satisfy, and define the feasible set

\[
C := \{ p = \text{col}(p_U, p_K) \in \mathbb{R}^{nN} | f_i(p_U) = 0 \}.
\]

To use the CRLB as localizability potential, the bound should now reflect the fact that localization algorithms can leverage the information provided by the constraints to improve their performance. We use the following result generalizing Proposition 1.

**Proposition 5.** Assume that the tag positions are subject to the constraints (28).

Let \( A_{Ul}(p_U) \) be a matrix whose columns span \( \ker \partial h/\partial p_U \) (which depends on \( p_U \) in general). Let \( \hat{p}_U \) be an unbiased estimate of the tag positions \( p_U \), based on the measurements \( d \), the knowledge of the anchor positions \( p_K \), and the knowledge of the constraints (28). Then

\[
\text{cov}^{\hat{p}_U} \geq B_U(p), \quad (29)
\]

where

\[
B_U(p) := A_U[A_U^T F_U A_U]^T A_U^T. \quad (30)
\]

**Proof.** We have both the trivial constraint \( f_U(p_U) = p_K - p_K^* = 0 \) with \( p_K^* \) the known positions of the anchors, and the equality constraint \( f_i(p_U) = 0 \). Define \( h(p) = \text{col}(f_i(p_U), f_i(p_K)) \). We then have:

\[
\frac{\partial h}{\partial p} = \begin{bmatrix}
\frac{\partial f}{\partial p_U} & 0 \\
0 & I_{nK}
\end{bmatrix}.
\]

We apply the result of Theorem 1, with the matrix \( A \) in (6)

\[
A = \begin{bmatrix} A_U & 0 \\
0 & 0
\end{bmatrix}, \quad F_c = A_U^T F_U A_U, \quad B_c = \begin{bmatrix}
A_U^T F_U A_U^T & 0 \\
0 & 0
\end{bmatrix}.
\]

In (6), the \( nU \times nU \) top-left corner of the matrix inequality gives (29) for the covariance of \( p_U \). The other parts of the bound (6) are trivial (0 \( \geq 0 \)) and correspond to the fact that a reasonable estimate \( \hat{p} = \text{col}(p_U, p_K) \) should set \( p_K = p_K^* \), so that \( \hat{p}_U \) will have zero covariance.  

Note that to simplify the notation, we have omitted in (30) to state the dependencies \( A_{Ul}(p_U) \) and \( F_U(p) \). From the matrix-valued bound (30), we can define constrained localizability potentials as in Section III-B. Here, for conciseness, we only consider the A-Opt potential

\[
J_c(p) := \text{Tr} \{ B_U(p) \}. \quad (31)
\]

Moreover, the desired tag positions should also respect the constraints specified by (28).
In other words, we aim to adjust the positions of the mobile nodes (anchors or tags) in order to minimize, at least locally, the potential \( J_\epsilon \) in (31) subject to the constraints \( \mathcal{C} \).

For this, we replace the gradient-descent method (3) by the following first-order primal-dual method [46, p. 528]:

\[
\begin{aligned}
\begin{cases}
\mathbf{p}_{k+1} = \mathbf{p}_k - \eta_k \left( \frac{\partial J_\epsilon}{\partial \mathbf{p}} + \Lambda_k^T \frac{\partial E}{\partial \mathbf{p}} \right) \mathbf{B}_\epsilon^T, \\
\Lambda_{k+1} = \Lambda_k + \delta \mathbf{f}_k(\mathbf{p}_{U I}, k),
\end{cases}
\end{aligned}
\]

where \( \eta_k \in \mathbb{R} \) is a sequence of decreasing stepsizes (following for instance Armijo’s rule [46]), \( \delta \) a fixed parameter and \( \Lambda_k \) are dual variable iterates. The scheme (32) provides a sequence of waypoints converging toward a local constrained optimum \( \mathbf{p}^* \). Feasibility of the constraints (28) is not maintained during the iterations (32), but the algorithm contributes to keeping \( \mathbf{p}_{k+1} \) close to \( \mathcal{C} \).

In addition, for each iterate \( \mathbf{p}_k \) that we actually want to use as waypoint for motion planning (some iterates could be skipped), since (28) represents rigidity constraints, we can enforce feasibility by computing for each robot the pose minimizing the distance between the desired and achievable tag locations, in a least-squares sense (this corresponds to a standard pose estimation problem [47, Section 8.1]).

In principle, we know the relative positions of the tags in \( \mathcal{U}_r \) in the robot frame of reference, so this information should be included in the CRLB. First, however, we only include the information about relative distances between tags in each group, as this leads to simpler algorithms.

In this case, in the framework of Section VI-A, \( f \) has one component for each pair of tags \( \{i, j\} \) in the same group \( \mathcal{U}_r \), of the form

\[
f^{ij}(\mathbf{p}_{U I}) = ||\mathbf{p}_{ij}||^2 - d_{ij}^2,
\]

where \( d_{ij} \) is perfectly known. If we order these components by listing all pairs of tags in the same set \( \mathcal{U}_1, \mathcal{U}_2, \ldots \), we obtain for the Jacobian

\[
\frac{\partial E}{\partial \mathbf{p}_{U I}} = \text{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_R),
\]

where \( \mathbf{R}_r \) is the rigidity matrix defined in Section IV-A, for the framework formed by a complete graph among the tags in group \( \mathcal{U}_r \).

Because the framework within each group is infinitesimally rigid, the kernel of each matrix \( \mathbf{R}_r \) is spanned by three explicitly known vectors if \( n = 2 \), or six if \( n = 3 \), as described in Proposition 2. By completing these vectors with zeros, we can form the matrix \( \mathbf{A}_{U I} = [\mathbf{A}_1 \ldots \mathbf{A}_R] \) with \( nU \) rows and \( 3R \) if \( n = 2 \) or \( 6R \) if \( n = 3 \) columns spanning the kernel of (33).

For example, based on the discussion above Proposition 2, if \( n = 2 \) we can take \( \mathbf{A}_r = [\mathbf{v}_{T r}^1 \mathbf{v}_{T r}^2 \mathbf{v}_{R r}^1] \), with \( \mathbf{v}_{T r}^2|_{2i-1} = 1 \), \( \mathbf{v}_{T r}^2|_{2i} = 1 \), \( \mathbf{v}_{R r}^1|_{2i-1} = -y_i \) and \( \mathbf{v}_{R r}^1|_{2i} = x_i \) for all \( i \in \mathcal{U}_r \) and zeros everywhere else.

From these explicit expressions of \( \mathbf{A}_{U I} \), we can also immediately compute the derivatives \( \partial \mathbf{A}_{U I}/\partial \xi_i \), for \( \xi_i \in \{ x_i, y_i, z_i \} \).

Since determining \( \mathbf{A}_{U I}(\mathbf{p}_{U I}) \) allows us to compute \( J_\epsilon(\mathbf{p}_{U I}) \) using (30), the only missing element to execute the iterations (32) is the gradient of \( J_\epsilon \). For this, assuming that \( \mathbf{F}_c := \mathbf{A}_{U I}^T \mathbf{F}_{U I} \mathbf{A}_{U I} \) is invertible, we obtain

\[
\frac{\partial J_\epsilon}{\partial \xi_i} = \frac{\partial}{\partial \xi_i} \text{Tr}\left\{ \mathbf{A}_{U I} \mathbf{F}^{-1} \mathbf{A}_{U I}^T \right\}
= 2 \text{Tr}\left\{ \mathbf{F}^{-1} \mathbf{A}_{U I} \mathbf{F}^{-1} \mathbf{A}_{U I}^T \right\} - \text{Tr}\left\{ \mathbf{A}_{U I} \mathbf{F}^{-1} \mathbf{F} \mathbf{F}^{-1} \mathbf{A}_{U I}^T \right\}

= 2 \text{Tr}\left\{ \mathbf{F}^{-1} \mathbf{A}_{U I} \mathbf{F} \mathbf{F}^{-1} \mathbf{A}_{U I}^T \right\} - \text{Tr}\left\{ \mathbf{B}_{U I} \mathbf{F} \mathbf{F}^{-1} \mathbf{A}_{U I}^T \right\}.
\]

C. CRLB with Constrained Relative Positions

To improve its accuracy, a position estimator can in fact use the full knowledge of the relative positions (RP) \( \mathbf{p}_{ij}^r \) in the frame of robot \( r \) for each pair of tags \( (i, j) \) carried by the same robot \( r \). Correspondingly, a CRLB should be derived for this case. To simplify the presentation, we assume in this section that each robot carries at least two tags.

To obtain the CRLB, let us first introduce \( R \) new parameters \( \mathbf{\theta} := \text{col}(\theta_1, \ldots, \theta_R) \), one for each robot, where \( \theta_i \in \mathbb{R}^q \), with \( q = 1 \) if \( n = 2 \) and \( q = 3 \) if \( n = 3 \). Then, for the extended
set of parameters \( p_U = (p_U, \theta) \) and the measurements (1) or (2), we denote the extended FIM

\[
\hat{F}_U = -\mathbb{E} \left\{ \frac{\partial^2 \ln f(d; p_U)}{\partial p_U \partial \hat{p}_U} \right\} = \begin{bmatrix} \mathbf{F}_U & 0_{nU,qR} \\ 0_{qR,nU} & 0_{qR,qR} \end{bmatrix}.
\]

(35)

In the following, we add constraints between the tag positions and the parameters \( \theta \), in such a way that the latter then represent the robot orientations in exponential coordinates, and the FIM \( \hat{F}_U \) is then appropriately changed using Theorem 1.

It is convenient to number and order the tags as follows. Consider robot \( r \in \{1, \ldots, R\} \) and associated tags \( U_r \), using the notation of Section VI-B. Pick one tag in \( U_r \), denoted in the following \( 1^r \). The other tags of \( U_r \) are denoted \( 2^r, \ldots, U_r \), with \( U_r = \|U_r\| \). We group these latter tags by robot and list them in the order

\[
p_o := \text{col}(p_{2^r}, \ldots, p_{U^r_1}), \ldots, p_{U^r_2}, \ldots, p_{U^r_R}) \in \mathbb{R}^{n(U-R)},
\]

(36)

from robot 1 to robot \( R \). The positions of the \( R \) tags \( 1^r \) are also grouped in the vector

\[
p_r := \text{col}(p_{1^r}, \ldots, p_{1^r}) \in \mathbb{R}^{nR}.
\]

Then, we have \( \hat{p}_U = \text{col}(p_o, p_r, \theta) \).

Next, for each tag \( j^r \in U_r \) other than \( 1^r \), we add the constraint \( f^{(r,j^r)}(p_{1^r}, p_{j^r}, \theta_r) = 0 \in \mathbb{R}^n \), where

\[
f^{(r,j^r)}(p_{1^r}, p_{j^r}, \theta_r) = p_{j^r} - p_{1^r} - \exp([\theta_r]_x) p_{1^r,j^r},
\]

(37)

with the notation (depending if \( n = 2 \) or \( n = 3 \))

\[
[\theta]_x = \begin{bmatrix} 0 & -\theta & 0 \\ \theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{if } \theta \in \mathbb{R},
\]

\[
[\theta]_x = \begin{bmatrix} 0 & -\theta_y & \theta_x \\ \theta_y & 0 & -\theta_z \\ \theta_z & \theta_x & 0 \end{bmatrix}, \quad \text{if } \theta = [\theta_x, \theta_y, \theta_z]^T \in \mathbb{R}^3.
\]

There are \( U_r - 1 \) constraints of the form (37) for robot \( r \), each of dimension \( n \). With these constraints, the matrix \( \exp([\theta_r]_x) \) represents the rotation matrix from the world frame \( \mathfrak{F} \) to the frame of robot \( r \), using the exponential coordinate representation [48]. These constraints then simply represent the choice of coordinates for the known vector \( p_{1^r,j^r} \) in the robot frame to the (unknown) coordinates \( p_{1^r,j^r} \) in frame \( \mathfrak{F} \). Define in the following the notation

\[
\Phi^{(r,j^r)}_{\theta_r} := \exp([\theta_r]_x) p_{1^r,j^r}, \quad \text{for } j^r \in U_r, 1 \leq r \leq R.
\]

Remark 6. Recall that when \( n = 2 \), we have simply

\[
\exp([\theta]_x) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},
\]

and when \( n = 3 \), \( \exp([\theta]_x) \) can be computed efficiently using Rodrigues’ formula [48, Proposition 3.1].

Considering (37) for all \( R \) robots, we obtain \( U - R \) constraints on the parameters \( p_{U_r} \), each of dimension \( n \). We list these constraints in the same order as for \( p_o \) in (36) and denote them \( f_{RP}(p_o, p_r, \theta) = 0 \). For the constrained CRLB, we are interested in the kernel of the Jacobian matrix of \( f_{RP} \). Remark that with the chosen ordering of tags and constraints, we have

\[
\frac{\partial f_{RP}}{\partial p_o} = \mathbf{I}_{n(U-R)}.
\]

If we define

\[
\mathbf{N} := \begin{bmatrix} \frac{\partial f_{RP}}{\partial p_o} & \frac{\partial f_{RP}}{\partial \theta} \end{bmatrix},
\]

(38)

then immediately

\[
\mathbf{A}_{RP} := \text{span} \left\{ \ker \frac{\partial f_{RP}}{\partial \theta} \right\} = \text{span} \left\{ \ker \left[ \mathbf{I}_{n(U-R)} \quad \mathbf{N} \right] \right\}
\]

(39)

Indeed, \( \frac{\partial f_{RP}}{\partial \theta} \) is of rank \( n(U-R) \), so \( \mathbf{A}_{RP} \) should have \( nU + qR - n(U-R) = (n+q)R \) independent columns, and clearly

\[
\frac{\partial f_{RP}}{\partial \theta} \mathbf{A}_{RP} = -\mathbf{N} + \mathbf{N} = 0.
\]

Hence, it is sufficient to compute \( \mathbf{N} \) to obtain \( \mathbf{A}_{RP} \).

Proposition 6. The matrix \( \mathbf{N} \) in (38) is defined by

\[
\mathbf{N} = \text{col} \left\{ \left( \mathbf{N}^{(r,j^r)} \right)_{1 \leq r \leq R, 2 \leq j^r \leq U^r_r} \right\} \in \mathbb{R}^{n(U-R) \times (n+q)R},
\]

where the blocks \( \mathbf{N}^{(r,j^r)} \) are stacked in the same order as for \( p_o \) in (36) and are of the form

\[
\mathbf{N}^{(r,j^r)} = - \begin{bmatrix} 0 & \mathbf{I}_n & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{N}^{(r,j^r)} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & \mathbf{N}^{(r,j^r)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & \mathbf{I}_n \end{bmatrix},
\]

(40)

with \( \mathbf{G}_i \in \mathbb{R}^{n \times n} \) and \( \mathbf{H}_i \in \mathbb{R}^{n \times q} \). The matrix \( \mathbf{N}^{(r,j^r)} \) is obtained by taking the partial derivatives of \( f^{(r,j^r)} \) in (37) with respect to the coordinates of \( p_{1^r} \), which gives the block \( \mathbf{G}_r = -\mathbf{I}_n \), and with respect to the coordinates of \( \theta_r \), which gives the block \( \mathbf{H}_r = -\mathbf{N}_r^{(r,j^r)} \in \mathbb{R}^{n \times q} \). All other blocks are zero. The computation of \( \mathbf{H}_r \) comes from the fact that \( \theta_r = \theta_r \mathbf{W} \) when \( n = 2, q = 1 \), and \( \theta_r = \theta_r \mathbf{W}_x + \theta_r \mathbf{W}_y + \theta_r \mathbf{W}_z \) when \( n = 3, q = 3 \).

With the matrices \( \hat{F}_U \) and \( \mathbf{A}_{RP} \) defined in (35) and (39), we can follow the discussion of Section VI-A and define \( \mathbf{B}_{RP} := \mathbf{A}_{RP} \hat{F}_U \mathbf{A}_{RP}^T \mathbf{A}_{RP}^T \) to obtain a CRLB taking the RP constraints into account. We can choose the cost function

\[
J_c(p) = \text{Tr} \left\{ \mathbf{C} \mathbf{B}_{RP} \mathbf{C}^T \right\},
\]

(40)

similarly to (31), where \( \mathbf{C} = [\mathbf{I}_{nU} \quad 0_{nU,qR}] \) is introduced to select only the uncertainty in the position estimates.

To compute the gradient with respect to \( p \) for (32), similarly to (34), we have, for \( \xi \in \{x, y, z\} \):

\[
\frac{\partial J_c}{\partial \xi} = 2 \text{Tr} \left\{ \mathbf{C} \frac{\partial \mathbf{A}_{RP}}{\partial \xi} \mathbf{D}^T \right\} - \text{Tr} \left\{ \mathbf{D} \frac{\partial \mathbf{F}}{\partial \xi} \mathbf{D}^T \right\},
\]
with \( D := CA_{RP}F_c^{-1} \), assuming \( F_c = A_{RP}^\top \hat{F}_d A_{RP} \) is invertible. To compute the derivative \( \partial A_{RP} / \partial \xi_i \), it is sufficient to know how to compute the terms \( \partial \Phi_{\theta, \xi}^{(r,j')} / \partial \xi_i \). Since

\[
\Phi_{\theta, \xi}^{(r,j')} = p_{j'} - p_{1r}
\]

from the constraint (37), then

\[
\frac{\partial \Phi_{\theta, \xi}^{(r,j')}}{\partial j'} = e_{\xi}, \quad \frac{\partial \Phi_{\theta, \xi}^{(r,j')}}{\partial \xi_i} = -e_{\xi},
\]

for \( \xi \in \{x, y, z\} \) and \( \partial \Phi_{\theta, \xi}^{(r,j')} / \partial \xi_j = 0 \) if \( j \notin \{1^r, j'\} \).

VII. SIMULATIONS

In this section, we present two deployment scenarios. The first is a structure inspection problem by a multi-robot network maintaining localizability while the task is performed. The second concerns the deployment of an Unmanned Ground Vehicle (UGV) carrying several tags, where we include the distance and relative position constraints in the CRLB-based potential.

A. Cooperative Structure Inspection

Consider \( N = 5 \) mobile robots, where three of them are anchors, i.e., \( \mathcal{K} = \{3, 4, 5\} \), assumed to be independently perfectly localized (e.g., via RTK GNSS), and the other two are tags, i.e., \( \mathcal{U} = \{1, 2\} \). Each robot carries an UWB transceiver to communicate and take ranging measurements with any other robot, via TWR [49], [50]. The ranging measurements follow the model (1). The tags are required to go underneath an \( L \times H := 6 \text{ m} \times 10 \text{ m} \) rectangular structure, represented on Fig. 3, in order to inspect it. Once under the structure, the tags lose access to the independent localization system (e.g., because GNSS signals are blocked). The anchors’ task is then to provide accurate localization for the tags, using the algorithms described in this paper.

1) Motion Planner Design: First, we assume that the tags \( \mathcal{U} = \{1, 2\} \) perform the inspection by following specified straight paths under the structure, described by the sequences of waypoints \( p_{1l}^d = \{L/3, al\}^\top \) and \( p_{2l}^d = \{2L/3, al\}^\top \) with \( a = 0.1 \text{ m}, 1 \leq l \leq 100 \).

The tags follow this path without taking their localization performance into account. On the other hand, the purpose of the mobile anchors is to provide accurate localization for the tags, measured here by \( J_{\text{loc}}(p) = J_D(p) \) for its computational efficiency. Moreover, the anchors must not wander away from the tags nor go under the structure, to maintain good localization. Hence, we introduce a potential \( J_{\text{con}}(p_K) = \sum_{i \in \mathcal{K}} \sum_{e \in B_i} g(d(i, e), d_s) \) penalizing the anchors if they approach the boundaries of specified bounding boxes \( B_i \), see Figure 3. We use standard repulsive potentials \( g(d(i, e), d_s) = 0.5(1/d(i, e) - 1/d_s)^21_{d(i,e)<d_s}, \) where \( d(i, e) \) is the distance between agent \( i \) and edge \( e \) of box \( B_i \), and \( d_s = 1.5 \text{ m} \) defines the influence region of the edge [48].

Therefore, we define the overall potential for the anchors as \( J_{\text{anchors}}(p) = K_I J_D(p) + K_c J_{\text{con}}(p_K) \) where \( K_I, K_c > 0 \) are constant parameters. The anchors \( i \in \mathcal{K} \) implement the descent gradient scheme

\[
P_{i,k+1} = P_{i,k} - K_I \left( \frac{\partial J_D}{\partial P_{i,k}} - K_c \frac{\partial J_{\text{con}}}{\partial P_{i,k}} \right). \tag{41}
\]

For \( \xi_i \in \{x_i, y_i\} \), we compute \( \partial J_D / \partial \xi_i \) analytically by (17).

The expressions of the derivatives \( \partial J_{\text{con}} / \partial \xi_i \) of the repulsive potential are standard [48].

The gradient descent scheme is used to obtain desired waypoints for the anchors, which we can track using controllers on the robots. For concreteness, assume that all robots are identical with monocycle kinematics [51, Chap. 4]

\[
\dot{x}_M = v \cos(\theta), \quad \dot{y}_M = v \sin(\theta), \quad \dot{\theta} = \omega \tag{42}
\]

where \( \omega \) and \( v \) are the rotational and translational velocities (see Fig 4). \((M, \vec{x}, \vec{y})\) is the robot frame and \( \theta \) is the robot heading with respect to \( \vec{y} \). The transceiver’s coordinates in the robot frame are \( p^r = [\alpha, \beta]^\top \), and we assume that \( \alpha \neq 0 \).

Fig. 3. Trajectories for the cooperation inspection scenario.

with \( \mathbf{u} \in \mathbb{R}^2 \) the coordinates of the anchor velocity in \( \vec{y} \), the following Proportional-Integral (PI) controller

\[
\dot{\mathbf{u}} = K_p (p_d(t) - p(t)) + K_i \int_0^t (p_d(\tau) - p(\tau)) d\tau, \tag{43}
\]

for \( K_p, K_i > 0 \), allows the anchors to track the desired trajectory \( p_d \). This provides a velocity command \( \mathbf{u} := [v, \omega]^\top \) for the robot, since \( \mathbf{u} = T(\theta) \) [48, p. 529] with

\[
T(\theta) = \frac{1}{\alpha} \begin{bmatrix} \alpha \cos \theta - \beta \sin \theta & \alpha \sin \theta + \beta \cos \theta \\ -\sin \theta & \cos \theta \end{bmatrix}
\]
2) Simulation and Performance Analysis: We set the initial positions for the tags to \( \mathbf{p}_{1,0} = [1, -1/2]^T \), \( \mathbf{p}_{2,0} = [5, -1/2]^T \) and for the anchors to \( \mathbf{p}_{3,0} = [-2, 0]^T \), \( \mathbf{p}_{4,0} = [-1.5, 0]^T \), \( \mathbf{p}_{5,0} = [8, 0]^T \). We choose the parameter values in (41) as \( \hat{K}_f = 2 \) and \( K_r = 1/100 \). To take into account the physical constraints on the robot velocities, we bound the stepsizes \( \|\mathbf{p}_{i,k+1} - \mathbf{p}_{i,k}\| \leq \Delta_{vel} \) by implementing the iterations

\[
\mathbf{p}_{i,k+1} = \mathbf{p}_{i,k} - \frac{\partial J}{\partial \mathbf{p}_{i,k}} \times \min \left\{ 1, \frac{\Delta_{vel}}{\|\partial J/\partial \mathbf{p}_{i,k}\|} \right\}.
\]

In our simulation, we set \( \Delta_{vel} = 0.2 \) m. We set \( \alpha = 0.5 \) m and \( \beta = 0.5 \) m and the PI controller gains are \( K_p = 3 \), \( K_i = 0.5 \). The controller (43) follows the trajectory computed by (41) with a maximum tracking error of about 10 cm.

As shown by the trajectories on Fig 3, the tags follow their assigned path and the anchors maintain the localizability. Initially, all robots are nearly aligned, a geometry with poor localizability. As shown on Fig. 5, the anchor deployment quickly and significantly decreases the localizability potential.

In Fig. 6, at each iteration \( k \) and for each tag \( i \), we plot the empirical total MSE, i.e., \( \hat{MSE}_{i,k} = \frac{1}{M} \sum_{\rho=1}^M \sigma_{i,k}^2 \), evaluated via \( M \) Monte Carlo simulations computing \( \sigma_{i,k}^2 = ||\hat{\mathbf{p}}_{i,k} - \mathbf{p}_{i,k}||^2 \), with \( \hat{\mathbf{p}}_{i,k} \) the estimate of \( \mathbf{p}_{i,k} \) at simulation \( \rho \), obtained by solving the least squares problem

\[
\hat{\mathbf{p}}_{i,k} = \arg\min_{\mathbf{p}_{i,k} \in \mathbb{R}^D} Q(\mathbf{p}_{i,k}) := \sum_{\rho=1}^M \sum_{i \epsilon \mathcal{K}} [\tilde{d}_{i,j,k} - \sigma_{i,k}^2]^2, \quad (44)
\]

where \( \tilde{d}_{i,j,k} := ||\hat{\mathbf{p}}_{i,k} - \mathbf{p}_{j,k}|| \) (with \( \mathbf{p}_{j,k} \) the anchor position if \( j \in \mathcal{K} \)) and \( \sigma_{i,k}^2 \) the measurements at simulation \( \rho \), obtained with the Gaussian additive model (1) for \( \sigma = 10 \) cm. We simulate \( M = 500 \) realizations and give 3\( \sigma \) confidence bounds \( b_+, b_- \) on Fig. 6. These bounds are defined by \( b_{\pm,k} = \hat{MSE}_{i,k} + 3\sigma_{i,k}/\sqrt{M} \), where \( \sigma_{i,k}^2 = \frac{1}{M-1} \sum_{\rho=1}^M [\tilde{d}_{i,k}^\rho - \hat{MSE}_{i,k}]^2 \) is the empirical variance of the samples. We remark that the motion planning method improves the precision of the estimates significantly during the first 25 iterations of motion planning, dividing the MSE of tag 1 by a factor of ten. We plot \( \ln \det \Sigma_k \) on Fig. 5, with \( \Sigma_k \) the empirical covariance of \( \hat{\mathbf{p}}_{U,k} \) obtained using (44), which shows that localizability as defined by the D-Opt criterion is properly maintained.

B. Deployment of An UGV Carrying Several Anchors

Here we illustrate the results of Section VI and the performance difference between leveraging information only on relative distances or on the full relative positions. Consider the robot shown on Fig. 7, following the kinematic model (42) and carrying two tags \( \mathcal{U} = \{1, 2\} \) placed at positions \( \mathbf{p}_1^* = [1, 0]^T \) and \( \mathbf{p}_2^* = [-1, 0]^T \) in the robot frame, centered at \( \mathbf{p}_M = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2) \). Three fixed anchors \( \mathcal{K} = \{3, 4, 5\} \) are placed at the coordinates \( \mathbf{p}_3 = [-5, 5]^T \), \( \mathbf{p}_4 = [5, -5]^T \) and \( \mathbf{p}_5 = [5, 5]^T \) in the absolute frame. All nodes communicate and obtain range measurements with each other, following the Gaussian additive model (1) with \( \sigma = 0.1 \) m. The heading of the robot is \( \theta \) and \( \exp[\theta]_x \) is the rotation matrix between \( \mathcal{F} \) and the robot frame.

In scenario (D), we include the constraint \( d_{12} = 2 \) m as in Section VI-B, and define the cost function as (31). In scenario (RP), we include the constraint \( \mathbf{p}_{12}^* = [2, 0]^T \) as in Section VI-C and define the cost function as (40), so that it can be compared to the previous one. We compute the potentials and their derivatives with the results of Section VI and implement the scheme (32) to compute a sequence of desired poses, which are then reached using the pose controller presented in [52]. At \( k = 0 \), the initial configuration of the robot in both cases is given by \( \mathbf{p}_M(0) = [-15, -4]^T \) and \( \theta(0) = -\pi/8 \). The cost and robot trajectories are shown on Fig. 8. The steady state configuration prescribed by (32) is feasible for the robot, thanks to the dual penalization of the rigidity constraint.

The following constrained least-squares estimators \( \hat{\mathbf{p}}_{U}^D \) and \( \hat{\mathbf{p}}_{U}^{RP} \) of \( \mathbf{p}_U \) are implemented in scenarios (D) and (RP)

\[
\begin{align*}
\hat{\mathbf{p}}_{U}^D & = \arg\min_{\mathbf{p}_U \in \mathbb{R}^D} Q(\mathbf{p}_U), \\
\text{s.t.} \; d_{12} - d_{12}^* = 0
\end{align*}
\]

and

\[
\begin{align*}
\hat{\mathbf{p}}_{U}^{RP} & = \arg\min_{\mathbf{p}_U \in \mathbb{R}^D} Q(\mathbf{p}_U), \\
\text{s.t.} \; \mathbf{p}_{21} - \exp[\hat{\theta}]_x \cdot \mathbf{p}_{12} = 0
\end{align*}
\]

where \( \hat{\theta} := \arctan(\hat{y}_{21}, \hat{x}_{21}) \) and \( Q(\mathbf{p}_U) \) is defined in (44). We evaluate the localization performance by computing the empirical MSE \( \hat{MSE}_{k} := \frac{1}{2}[\hat{MSE}_{1,k} + \hat{MSE}_{2,k}] \) for the two tag positions, using the same process as in VII-A2, with

\[
\text{Fig. 5. Localizability potential } J_D(\mathbf{p}(k)) \text{ and empirical plot of } \ln \det \text{ cov}(\hat{\mathbf{p}}(k)).
\]

\[
\text{Fig. 6. Empirical mean-squared error.}
\]

\[
\text{Fig. 7. Robot equipped with two tags.}
\]
M = 500 simulations. The results shown in table I indicate that the motion significantly improves the estimate accuracy. Moreover, the relative position constraints provides a clear improvement compared to only using the relative distance information.

|   | $MSE_D$ | Confidence | $MSE_R$ | Confidence | ET  |
|---|---------|------------|---------|------------|-----|
| (D) | 4.28 m$^2$ | ±0.03 m$^2$ | 0.98 m$^2$ | ±0.02 m$^2$ | 1.70 s |
| (RP) | 2.97 m$^2$ | ±0.04 m$^2$ | 0.63 m$^2$ | ±0.002 m$^2$ | 1.89 s |

Table I also provides the Execution Times (ET) of the deployment algorithms for all the steps shown on Fig. 8. The simulation is coded in Matlab R2018b and runs on a computer equipped with an Intel i7 processor. The ET for the (RP) scenario is about 10% larger than for (D), due to the increased complexity to evaluate A and its derivative. In summary, compared to (D), deployment using (RP) leads to a significant improvement of the precision and a moderated increase of the computation time.

VIII. CONCLUSION AND PERSPECTIVES

This paper presents deployment methods applicable to Multi Robots Systems (MRS) with relative distance measurements, which enforce good localizability. Constrained Cramér-Rao Lower Bounds (CRLB) are used to predict the localization error of a given configuration, assuming Gaussian ranging measurement models. A connection between Fisher information matrices and rigidity matrices is highlighted, which yields useful properties, e.g., for initial MRS placement.

The CRLB is used to design artificial potentials, so that gradient descent schemes can be developed to plan robot motions that enhance the overall localizability of the network. Moreover, we show how to distribute the execution of the resulting algorithms among the robots, so that they only need to communicate with their neighbors in the ranging graph. Finally, we extend the methodology to MRS with robots carrying multiple tags, again leveraging the theory of equality-constrained CRLBs. Future work could focus on using the proposed measures of localizability to constrain other types of motion planning algorithms.

ACKNOWLEDGEMENTS

The authors wish to thank Drs. Éric Chaumette, Gaël Pagès and Ali Naouri from ISAE-Supaéro (France) for helpful discussions.

REFERENCES

[1] J. Le Ny and S. Chauvière, “Localizability-constrained deployment of mobile robotic networks with noisy range measurements,” in American Control Conference (ACC), Milwaukee, WI, Jun. 2018, pp. 2788–2793.
[2] J. Cano and J. Le Ny, “Improving ranging-based location estimation with rigidity-constrained CRLB-based motion planning,” in International Conference on Robotics and Automation (ICRA), Xi’an (China), 2021.
[3] J. Sheu, W. Hu, and J. Lin, “Distributed localization scheme for mobile sensor networks,” IEEE Transactions on Mobile Computing, vol. 9, no. 4, pp. 516–526, Apr. 2010.
[4] A. Prorok, A. Bahr, and A. Martinoli, “Low-cost collaborative localization for large-scale multi-robot systems,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 2012.
[5] J. Xu, M. Ma, and C. L. Law, “AOA cooperative position localization,” in IEEE Global Telecommunications Conference (GLOBECOM), New Orleans, LA, Nov. 2008.
[6] M. Wei, R. Aragues, C. Sagues, and G. C. Calafiore, “Noisy range network localization based on distributed multidimensional scaling,” IEEE Sensors, vol. 15, no. 3, pp. 1872–1883, Mar. 2015.
[7] L. Carlino, D. Jin, M. Muma, and A. M. Zoubir, “Robust distributed cooperative RSS-based localization for directed graphs in mixed LoS/NLoS environments,” arXiv:1810.08098, Dec. 2018. [Online]. Available: http://arxiv.org/abs/1810.08098
[8] Z. Sahinoglu, S. Gezici, and I. Guvenc, Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols, Cambridge: Cambridge University Press, 2008.
[9] M. W. Mueller, M. Hamer, and R. D’Andrea, “Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation,” in IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA: IEEE, May 2015, pp. 1730–1736.
[10] J. Cano, S. Chidami, and J. Le Ny, “A Kalman filter-based algorithm for simultaneous time synchronization and localization in UWB networks,” in IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May 2019.
[11] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, “Collaborative sensor network localization: Algorithms and practical issues,” Proceedings of the IEEE, vol. 106, no. 6, pp. 1089–1114, 2018.
[12] T.-S. Tay and W. Whiteley, “Generating isostatic frameworks,” Struct. Topology, vol. 11, Jan. 1985.
[13] R. Connolly, “Generic Global Rigidity,” Discrete & Computational Geometry, vol. 33, no. 4, pp. 549–563, Apr. 2005.
[14] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur, “A theory of network localization,” IEEE Transactions on Mobile Computing, vol. 5, no. 12, pp. 1663–1678, 2006.
