ARLCL: Anchor-free Ranging-Likelihood-based Cooperative Localization

Dimitris Xenakis, Antonio Di Maio, Torsten Braun
{dimitrios.xenakis, antonio.dimaio, torsten.braun}@unibe.ch
Communication and Distributed Systems, Institute of Computer Science, University of Bern, Switzerland

***Abstract*—Positioning estimations of wireless sensors can be enhanced via sensor collaboration. To enable this, various methods have been proposed; yet, most do not leverage the entire collective knowledge, which also involves the estimation's uncertainty. In this article, we introduce Anchor-free Ranging-Likelihood-based Cooperative Localization (ARLCL); a novel anchor-free and technology-agnostic localization algorithm that utilizes inter-exchanged ranging signals from sensors to enable their simultaneous positioning. Ranging technologies with easy-to-model propagation properties, such as UWB or LiDAR are among the first beneficiaries that ARLCL is targeting. To examine its applicability, however, even to signals that are noisier and often unsuitable for ranging, we assess ARLCL with real-world BLE RSS measurements. At the same time, we consider deployments that typically induce flip-ambiguity, being a major problem in cooperative localization. We provide an extensive comparison against the most widely-adopted optimization method (Mass-Spring) but also against the recent likelihood-based approach (Maximum Likelihood - Particle Swarm Optimization). The results showed that ARLCL outperformed the baselines in almost all scenarios. Our gain in positioning accuracy is also found to be positively correlated to both the swarm's size and the signal's quality, reaching an improvement of 40%.**

***Index Terms*—Cooperative Localization, Network Localization, Mesh ranging, Relative Positioning, Wireless Sensor Networks**

I. Introduction

The ability to know accurately where an entity is situated in space is gaining increasing importance. Localization technologies are being progressively adopted and standardized, providing positioning via different techniques. In this work, we focus on Cooperative Localization (CL) technologies. These are of great interest as they leverage noisy, yet redundant spatial information to provide estimations of increased accuracy. Such systems have broad applications, from Vehicular Ad-hoc Networks (VANETs) and swarms of Unmanned Aerial Vehicles (UAVs) to submarines and extraterrestrial drones.

In CL systems, for each node to be referenced to a geographic coordinate system, connectivity is required towards anchor nodes whose positions are already known. Otherwise, the swarm's relative positions become geographically unreferenced and invariant under isometric transformations. The existence of anchors is a requirement in most CL solutions. However, in many cases, the exact geographic coordinates of nodes are not of major importance. Instead, the relative positioning and distances remain the prime interest.

At the same time, CL systems remain highly susceptible to flip-ambiguities. Viewing the system as a graph, rigidity theory shows that flip-ambiguity occurs when a sub-graph flips across an axis without edge constraints violations [1]. In other words, when sub-groups of correctly positioned nodes (in the sub-group) are connected with other such sub-groups, incorrectly. Literature acknowledges flip-errors as a major challenge that becomes prominent in scenarios involving anchor-less systems and swarms with irregular deployments [2], [3].

Our research aims to increase the performance of services that localize nodes by measuring the distance (ranging) between them and, if available, towards other reference sources. This requires addressing several key challenges in the field, with arising research questions on how can a CL system become independent from anchors while remaining robust against flip-ambiguities, and how can such a system be a generalized framework that can support any ranging technology.

Having addressed these challenges, we present a novel anchor-free CL framework that supports ranging with any signal of modellable uncertainty, from Global Navigation Satellite Systems, WiFi, and Bluetooth, to acoustic waves. Likewise, any ranging technique can also be used, including Time Difference of Arrival (TDoA), Time of Arrival (ToA), or even Received Signal Strength (RSS). In the case of RSS, its low acquisition cost and simplicity make its utilization quite tempting. Yet, its unpredictability due to a plethora of noise sources such as Non-Line-of-Sight (NLOS) propagation, shadowing, etc., often renders it unsuitable for robust solutions.

Our work's contribution is the synthesis of the following:

1) A novel ranging-likelihood CL method called ARLCL that is not only independent from anchors and any prior information about the nodes' positions but is also technology-agnostic and, thus, suitable for all ranging technologies.

2) An extensive evaluation against today's most adopted and latest approaches, targeting deployments with irregular shapes and using real-world collected Bluetooth measurements.

3) To facilitate the reproduction of our results and future comparisons, we openly release ARLCL source code[1].

The remainder of this paper is organized as follows: Section II provides a review of the most relevant work on ranging-based CL. Section III presents our system model. ARLCL, our proposed method for Cooperative Localization, is presented in Section IV. Section V describes the evaluation framework for assessing the performance of ARLCL against the most established CL approach and describes the outcomes. Finally, conclusions and future works are outlined in Section VI.

[1]https://doi.org/10.5281/zenodo.7552461

II. Related Work

In a typical CL system, the unknown positions of a swarm of nodes are estimated based on inter-exchanged signals and communication with nodes at known positions (*anchors*). The localization is performed by a solver function in a centralized or distributed manner. The estimation technology may vary; from RSS-based ranging given a path-loss model [4], to complex ranging approaches such as Doppler frequency [5]. However, due to multipath conditions and environmental dynamics, measurements are affected by noise, rendering the associated problem highly non-convex and often intractable.

Non-Bayesian approaches based on Least Squares (LS) have been widely used, with recent notable contributions introducing systems of dynamic ranging models to support changing environments [4]. These approaches typically make use of Gauss-Newton, Particle Swarm Optimization [6], or Steepest Descent methods to iteratively converge to optimal solutions. However, when the problem becomes highly nonlinear, reaching the global optimum becomes challenging. For that reason, a relaxation on the objective function using Semi-Definite Programming (SDP) [7] and Second-Order Cone Programming (SOCP) [4] has been suggested. Yet, transforming a non-convex optimization function into a convex one via relaxation introduces the risk of converging to a global optimum far from the initial solution set.

Another non-Bayesian approach where positions are treated as unknown deterministic parameters is the Maximum Likelihood Estimation (MLE) [8]. Contrary to LS, MLE exploits the statistical properties of the noise sources, making the estimation more robust to measurement errors. Although it is asymptotically efficient when enough measurements are available, its complexity is increased, with its performance remaining dependent on the solver's initialization.

Several Bayesian methods have also been proposed that consider the positions as a realization of a random variable. Here, a posterior distribution of positions is determined for the estimations. One notable example is Sum-Product Algorithm over a Wireless Network (SPAWN) [9], a framework for the systematic design of CL algorithms based on factor graphs and the Sum-Product Algorithm (SPA) [16]. Even though SPAWN offers high performance and is well generalizable and applicable not only to Bayesian and classical methods, but also to both distributed and centralized scenarios, it suffers from high computational complexity and relies on anchors.

The Generalized Approximate Message Passing (GAMP) procedure [17] is an alternative to SPAWN. This has been used in the 3D Geographical Information Enhanced Universal Cooperative Localizer (3D GIE-UCL) [5] as a more simplified, but still robust CL approach for VANETs in 3D space and under varied ranging methods. Moreover, this work introduces GPU acceleration support and a lightweight importance sampling (particles) for estimating the marginal posterior Probability Density Function (PDF) of the nodes' positions. Nevertheless, some critical points reflect the need again for anchors and even building map information.

Soft Range Information (SRI) localization [10], [18] is based on machine-learning methods. According to SRI, the solver does not rely solely on the estimation for the most likely distance, but instead it is able to account for the likelihood of all possible distances (ranging-likelihood). However, this method is demonstrated using highly accurate Round Trip Time (RTT) measurements, while additionally, no assessment is provided for anchor-free scenarios.

Among the aforementioned works, [7] has been the only Anchor-Free CL system. This problem is tackled by other approaches using non-probabilistic Mass-Spring (MS) models such as the Anchor-Free Localization (AFL) [11], or the Distributed Anchor-Free Localization (DAFL) [12]. Here, the distances are depicted as springs of varying equilibrium, connecting different node masses [19]. In that sense, the system's overall equilibrium state represents a placement solution of minimum tensions. The most recent Anchor-Free CL approach assessed with RSS measurements is Particle-Assisted Stochastic Search (PASS) [15]. Unlike MS, PASS is a centralized Bayesian method based on Maximum Likelihood - Particle Swarm Optimization (ML-PSO) [6], [20]. Its particles offer a search mechanism for the non-convex objective posterior location estimator that can find the global optimum with a higher probability [20].

Although MS [11] and ML-PSO [6], [20] constitute the basis for the most adopted and latest RSS-based approaches respectively, they either neglect the measurement's uncertainty or assume a Gaussian positional error. This introduces a strong assumption since the position error depends on the distance between nodes following a log-normal law [19]. Moreover, they do not utilize mechanisms to mitigate flip-ambiguities which, especially in irregular deployments, impose a great challenge for CL systems [3], [21], [22]. This means that when the nodes' deployment does not span towards all directions (e.g. passengers inside a train), it becomes difficult for the solver to produce estimations without flips across the non-extended sides, as there is no spatial information from neighbors available. ARLCL, on the contrary, respects these points and is designed to support both centralized and distributed setups. For the evaluation of our method, we offer an implementation of these three approaches, and we openly release the source code as a common base for comparisons.

Table I presents a summary of the discussed related works.

III. System Model

This section provides a comprehensive understanding of the system model and lays the foundation for the subsequent presentation of the proposed methodology and evaluation. In summary, we consider a network of wireless devices, each capable of transmitting and receiving data packets. We outline the true and estimated positions of the devices, the measurement sets, and the probabilistic signal propagation model for the radio signals. Finally, we formulate the localization problem as the minimization of an error function, aiming to estimate the devices' positions based on the set of measurements.

TABLE I
SUMMARY OF RELATED WORKS

Method	Opt. Approach	Ranging	AF	Distr.	Major Contribution
S. Chang et al. [4]	LS & SOCP	RSS	-	-	Supporting unknown path-loss model
S. Tomic et al. [8]	MLE & SDP/SOCP	RSS	-	-	Supporting unknown path-loss model
3D GIE-UCL [5]	GAMP & Particle-based	Diverse	-	✓	GPU accelerated for real-time operation
SPAWN [9]	Factor Graphs	RTOA	-	✓	Well generalizable, supporting mobility and sensor fusion
SRI [10]	Ranging-Likelihood	RTT	-	✓	Robust to the non-linearity of ranging noise
AFL [11]	MS	-	✓	✓	Became the basis for many future works
DAFL [12]	MS	RSS	✓	✓	Complete decentralization compared to AFL
S. Nawaz & S. Jha [13]	MS	-	✓	-	Showed the value of constraints for distant nodes
J. Eckert et al. [14]	MS	Acoustic	✓	✓	Good scalability reported
G. Calafiore et al. [7]	LS & SDP	-	✓	✓	Use of steepest descent with auto-computed step sizes
PASS [15]	Particle-based	RSS	✓	-	Good convergence of the objective function
ARLCL	Ranging-Likelihood	RSS	✓	✓	Robust to the non-linearity of ranging noise

Without loss of generality due to the dimensionality reduction in Euclidean space, we consider a scenario in which N wireless devices, indexed by an ordered set $\mathcal{A} = \{1, 2, \ldots, N \geq 3\} \subseteq \mathbb{N}$, are located on a bidimensional plane. Let $\mathcal{P} = \{\boldsymbol{p}_i \mid i \in \mathcal{A}\}$ denote the set of the devices' true positions, where $\forall i \in \mathcal{A} : \boldsymbol{p}_i \in \mathbb{R}^2$. We assume that $\forall (i,j) \in \mathcal{A}^2, i \neq j : \boldsymbol{p}_i \perp\!\!\!\perp \boldsymbol{p}_j$. Each device i can receive and transmit wireless data packets from at least two other devices. We note that no full connectivity is required with the rest of the swarm. For each device i, let $\mathcal{B}_i \subset \mathcal{A}, i \notin \mathcal{B}_i$ denote the set of its neighbors, defined as the set of those devices that have received at least one data packet from device i. For every k-th data packet sent by device i and received by j, device j computes a *measurement* $z_{i,j,k} \in \mathbb{R}^M$, where the value of M and the physical meaning of each component of the M-dimensional vector depends on the set of the M different techniques used to compute ranging measurements (e.g., ToA, Angle of Arrival (AoA), RSS). In this paper, for sake of clarity, we assume ranging based on RSS signals only ($M = 1$, $z_{i,j,k} \in \mathbb{R}$). We assume that every device $j \in \mathcal{B}_i$ has received $K_{i,j} \in \mathbb{N}$ data packets from device $i \in \mathcal{A}$. For each couple of devices $(i,j) \in \mathcal{A} \times \mathcal{B}_i$, we define the *measurement set* $\boldsymbol{z}_{i,j} = \{\boldsymbol{z}_{i,j,k} : k \in \{1, \ldots, K_{i,j}\}$ as the set of all measurements computed by receiver device j for all $K_{i,j}$ data packets sent by device i. We define the set of all measurement sets in the system as $\mathcal{Z} = \{\boldsymbol{z}_{i,j} \mid i \in \mathcal{A}, j \in \mathcal{B}_i\}$.

ARLCL is not bound to some ranging technology and has already been verified using Ultra-WideBand (UWB) transceivers deployed loosely across our building. Yet, for the current paper's assessment, we consider Bluetooth Low Energy (BLE) RSS measurements to assess its performance even for a technology that is typically unsuitable for ranging. The signal power of such RF signals decays proportionally to $d^{-\alpha}$, where α is a path loss exponent (PLE) that depends on the environment and d is the distance from the signal source. For Free-Space Path Loss (FSPL) models, $\alpha = 2$, while studies suggest alternative values of α for other propagation environments [23]. Assuming $P'(0)[\mathrm{W}]$ as the signal power at the source, the signal power $P'(d)[\mathrm{W}]$ at distance $d[\mathrm{m}]$ from the source is $P'(d) = P'(0) \cdot \left(\frac{4\pi d}{\lambda}\right)^{-\alpha}$, where λ is the signal's wavelength. We formulate the signal power $\bar{P}(d)$ in dBm at distance d from the source as

$$\bar{P}(d) = 10 \log \left(\frac{P'(d)}{1\,\mathrm{mW}} \right) = P_0 - 10\alpha \log \left(\frac{d}{1\,\mathrm{m}} \right) \quad (1)$$

where $P_0[\mathrm{dBm}] = 10 \log \left(\frac{P'(0)}{1\,\mathrm{mW}} \cdot \frac{\pi 4\,\mathrm{m}}{\lambda} \right)$ is the power at $1\,\mathrm{m}$.

We assume that every RSS measurement received by node i from any node $j \in \mathcal{B}_i$ is normally distributed with average $\bar{P}(d_{i,j})$ and constant variance σ_{dB}^2 (2).

$$\boldsymbol{z}_{i,j,k} \sim \mathcal{N}\left(\bar{P}(d_{i,j}), \sigma_{dB}^2\right), \quad \forall k \in K_{i,j} \quad (2)$$

$\forall k \in K_{i,j}$ we can write $\boldsymbol{z}_{i,j,k}$ as the sum of the expected power $\bar{P}(d_{i,j})$, expressed in dBm, and a zero-mean Gaussian noise $\epsilon_{i,j}$, expressed in dB, with variance σ_{dB}^2 (3). It is worth mentioning that $\bar{P}(d_{i,j})$ depends non-linearly on the distance $d_{i,j}$ between nodes i and j, and that the random variables $\{\epsilon_{i,j} \sim \mathcal{N}(0, \sigma_{dB}^2) \mid \forall i \in \mathcal{A}, j \in \mathcal{B}_i\}$ are mutually independent and identically distributed (i.i.d.).

$$\boldsymbol{z}_{i,j,k} = \bar{P}(d_{i,j}) + \epsilon_{i,j}, \quad \forall j \in \mathcal{B}_i, \forall i \in \mathcal{A}, \forall k \in K_{i,j} \quad (3)$$

We assume that signal propagation follows the probabilistic model $\mathcal{L}_z(\boldsymbol{d})$, which is generic and applicable to any ranging technology. This model can relate a measurement $\boldsymbol{z}$ to an a posteriori PDF $f(d \mid \boldsymbol{z})$ conditioned on the measurement $\boldsymbol{z}$. Although this function could be used to extract a single-distance estimation (e.g. the expected value or mode), ARLCL uses the entire distribution during the localization process. Using the whole distribution instead of point estimates allows us to take advantage of the whole information contained in the PDF about the relative likelihood of every possible distance. Different ranging-modeling approaches may lead to different density functions. These always remain proportional to the product of the ranging likelihood and the prior distribution of the measurements. We assume an uninformative prior for the distribution of any node's position ($\pi_0(\boldsymbol{p}_i) = 1, \forall i \in \mathcal{A}$), so the model family that ARLCL requires may be defined as:

$$\mathcal{L}_{\boldsymbol{z}}(d) \propto f(d \mid \boldsymbol{z}). \quad (4)$$

The set $\mathcal{P}$ is unknown and we aim to compute its estimate $\widehat{\mathcal{P}}$ based on a set of measurements $\mathcal{Z}$. When the device $j \in \mathcal{B}_i$

performs *ranging* towards node i, it means that it produces an estimate $\widehat{d}_{i,j}$ of the true distance $d_{i,j} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \in \mathbb{R}$ between them. Since each measurement $\boldsymbol{z}$ is noisy, the estimated distance $\widehat{d}_{i,j}$ is also noisy. Given the pairwise noisy measurement set $\mathcal{Z}$ of N wireless sensor nodes located at positions $\mathcal{P}$, the final CL problem ARLCL aims to solve is to compute an estimate $\widehat{\mathcal{P}}$ that minimizes the error function

$$\varepsilon(\mathcal{P}, \widehat{\mathcal{P}}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{p}_i - \widehat{\boldsymbol{p}}_i\|^2, \quad \boldsymbol{p}_i \in \mathcal{P} \text{ and } \widehat{\boldsymbol{p}}_i \in \widehat{\mathcal{P}}. \quad (5)$$

IV. Methodology

This section presents the methodology of the cooperative localization framework, ARLCL. First, we explain, with the help of factor graphs, the ranging model's role in the final objective function, where measurements are related probabilistically to underlying distances. Then, we detail the offline and online key phases of the ARLCL process (during which the estimated positions are iteratively refined), addressing factors such as the adjustment of the ranging model, optimization order, and positioning convergence.

A. From Measurements to Objective Function

ARLCL requires a ranging model, often acquired during an offline phase, which is associated with some Line-of-Sight (LOS) or NLOS environment. Such a ranging model is used to relate a measurement to an underlying distance and is a typical requirement in CL systems. From a graphical model perspective [24] let us introduce the function $\psi_{i,j}$ as the *clique potential* between node i and j, described by our ranging model $\mathcal{L}_{\boldsymbol{z}}(d)$. We also define the *local potential* ψ_i as the representation of the prior probability distribution for $\boldsymbol{p}_i$. We can model $\boldsymbol{p}_i$ as a random variable in a Markov Random Field [9], [25], and consider all inter-node measurements $\boldsymbol{z}_{i,j}$. For the uniform cases, where we have no information about the i-th node's position, ψ_i can be replaced with an improper prior $\psi_i = 1$. The pdf $f(\{\boldsymbol{p}_i\}_{i\in\mathcal{A}} \mid \mathcal{Z})$ of the joint posterior probability distribution of the positions of all nodes in the system is proportional to the nodes' position likelihood (6) multiplied by the node's position prior probability. The Hammersley-Clifford theorem allows us to write:

$$f(\{\boldsymbol{p}_i\}_{i\in\mathcal{A}} \mid \mathcal{Z}) \propto \prod_{i\in\mathcal{A}} \psi_i(\boldsymbol{p}_i) \prod_{j\in\mathcal{B}_i} \psi_{i,j}(\boldsymbol{p}_i, \boldsymbol{p}_j). \quad (6)$$

We could use Bayesian estimators such as Minimum Mean Squared Error (MMSE) or Maximum A Posteriori (MAP) on the marginal posterior distribution of $\boldsymbol{p}_i$ to obtain a point estimate $\widehat{\boldsymbol{p}}_i$. This implies the integration of (6) as

$$f(\boldsymbol{p}_i \mid \mathcal{Z}) = \int_{\mathbb{R}^{N-1}} f\left(\{\boldsymbol{p}_i\}_{i\in\mathcal{A}} \mid \mathcal{Z}\right) \mathrm{d}\boldsymbol{p}_{i:N\setminus i} \quad (7)$$

where $\boldsymbol{p}_{i:N\setminus i}$ denotes the positions of all nodes except i's. The number of nodes has an exponential impact on the complexity of the marginalization. Depicting the network system as a factor graph enables us not only to simplify the marginalization task, but also to distribute the computation across all nodes [24]. A factor graph is a bipartite graph that can be used to represent the factorization of a joint distribution like (6) into local functions. The graph's vertices are classified into two types: the variable nodes $\boldsymbol{p}_i$ (circles representing the random variables of the distribution) and the factor nodes ψ_i and $\psi_{i,j}$ (squares representing a specific factor in the probability distribution). The edges connecting the variable nodes to the factor nodes represent the functional dependence between the variables and the factors. Figure 1 provides a representation of a CL network as a factor graph. Here, the factor ψ_i represents the prior information about i's position, namely $\psi_i(\boldsymbol{p}_i)$ in (6), and the factors $\psi_{i,j}$ represent the ranging likelihood between two variable nodes. Each edge shows on which variable, a factor depends. The grey area enclosing all available factors dependent on $\boldsymbol{p}_2$ designates the information required for producing an estimation about $\boldsymbol{p}_2$.

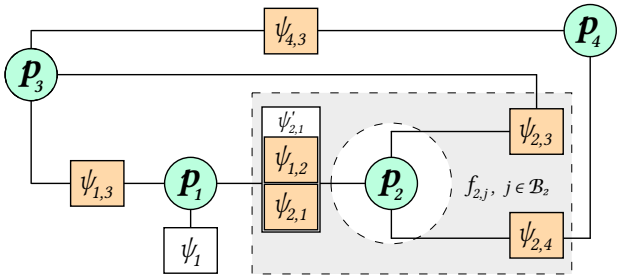

Fig. 1. Depiction of a CL network as a factor graph.

Obtaining now the marginal distribution ("belief") of each variable $\boldsymbol{p}_i$ in (7) becomes easier through message passing. The Sum-Product Algorithm (often referred to as Belief Propagation) is a suitable candidate for such a technique [24]. It is an iterative algorithm that allows us to calculate the marginal posterior distribution $f(\boldsymbol{p}_i \mid \mathcal{Z})$ over a graph through the exchange of statistical messages between adjacent nodes.

According to the SPA scheme [16], the exchanged messages can be of two types; namely, messages from a factor-node $\psi_{i,j}$ or ψ_i to an adjacent variable-node $\boldsymbol{p}_i$ (denoted as $m_{\psi_{i,j}\to\boldsymbol{p}_i}(\boldsymbol{p}_i)$ or $m_{\psi_i\to\boldsymbol{p}_i}(\boldsymbol{p}_i)$ respectively), and messages from a variable node to a factor node (denoted as $m_{\boldsymbol{p}_i\to\psi_{i,j}}(\boldsymbol{p}_i)$). As depicted in Figure 1, the bidirectional ranging measurements can allow the corresponding factor-node $\psi_{i,j}$ to exchange messages with exactly two variable nodes. These types of messages are given as

$$m^{(t)}_{\psi_{i,j}\to\boldsymbol{p}_i}(\boldsymbol{p}_i) \propto \int \psi_{i,j}(\boldsymbol{p}_i, \boldsymbol{p}_j) m^{(t)}_{\boldsymbol{p}_j\to\psi_{i,j}}(\boldsymbol{p}_j)\, \mathrm{d}\boldsymbol{p}_j, \quad (8)$$

$$m^{(t)}_{\psi_i\to\boldsymbol{p}_i}(\boldsymbol{p}_i) \propto \psi_i(\boldsymbol{p}_i) \quad (9)$$

where t denotes the iteration step of SPA. Essentially, $m_{\boldsymbol{p}_j\to\psi_{i,j}}(\boldsymbol{p}_j)$ embodies the collective knowledge of $\boldsymbol{p}_j$ that has been fused with the participation of all its neighbors,

except the neighbor towards which it is propagated. These are the second type of messages (i.e., from a variable node to a factor node), which are given as

$$m^{(t)}_{\boldsymbol{p}_i \to \psi_{i,j}}(\boldsymbol{p}_i) \propto m^{(t-1)}_{\psi_i \to \boldsymbol{p}_i}(\boldsymbol{p}_i) \prod_{k \in \mathcal{B}_i \setminus j} m^{(t-1)}_{\psi_{i,k} \to \boldsymbol{p}_i}(\boldsymbol{p}_i). \quad (10)$$

After collecting at the variable-node $\boldsymbol{p}_i$ all messages from the adjacent factor-nodes, the belief of the node used to approximate its posterior distribution is given by

$$b^{(t)}_i(\boldsymbol{p}_i) \propto m^{(t)}_{\psi_i \to \boldsymbol{p}_i}(\boldsymbol{p}_i) \prod_{j \in \mathcal{B}_i} m^{(t)}_{\psi_{i,j} \to \boldsymbol{p}_i}(\boldsymbol{p}_i) \quad (11)$$

Figure 1 also shows that the factor graph of a CL system is typically not cycle-free. This means there are no specific "leaf" and "root" nodes to designate the beginning and end of the message-passing process. Therefore, the computed marginals are also not guaranteed to be exact. In such a "loopy" graph, the belief propagation becomes cyclic and dependent on the message passing order and iteration depth, until some convergence criterion is reached. Optimal strategies for these will be provided later in this section.

The collective knowledge that node i has acquired at some step t, about its neighbor's positions and the ranging measurements they have towards it, as expressed in (11), suffices to produce an estimation of its own position. Following this iterative scheme, ARLCL leverages this knowledge but does not require any initial information about the nodes' positions. Hence, at step $t = 0$, each node can be assigned any position, which will be progressively refined throughout all the iteration steps. Ultimately, according to (4) and since $\boldsymbol{p}_i, \boldsymbol{p}_j$ are mutually independent with no prior information available, each factor-node contributing to the computation of $\widehat{\boldsymbol{p}}_i$ reflects the ranging likelihood between the two nodes, where

$$\mathcal{L}(\boldsymbol{p}_i, \boldsymbol{p}_j \mid \boldsymbol{z}_{i,j}) \propto f(\boldsymbol{p}_i \mid \boldsymbol{z}_{i,j}, \boldsymbol{p}_j). \quad (12)$$

The product of the contributing beliefs in (12), where the position of each neighbor at step $t-1$ is assumed to be known, leads to the objective function that ARLCL is considering for computing the current most probable position of node i

$$\widehat{\boldsymbol{p}}^{(t)}_i = \arg\max_{\boldsymbol{p} \in \mathbb{R}^2} \prod_{j \in \mathcal{B}_i} f(\boldsymbol{p} \mid \boldsymbol{z}_{i,j}, \widehat{\boldsymbol{p}}^{(t-1)}_i). \quad (13)$$

An instance of this optimization is illustrated in Figure 2 (which also corresponds to the optimization shown in Figure 7). Here, four neighbor-nodes have acquired from node 9, four individual measurements. Based on a common ranging model, these measurements have produced four different beliefs about node-9's position. Equation (13) is used to fuse these beliefs into an objective function (contours basemap) that can be maximized to estimate the position of node 9. Along with the objective function and node-9's optimized position, Figure 2 also illustrates part of the individual belief of neighbor-node 3 (as expressed between their current positions). Although the actual belief resembles the shape of

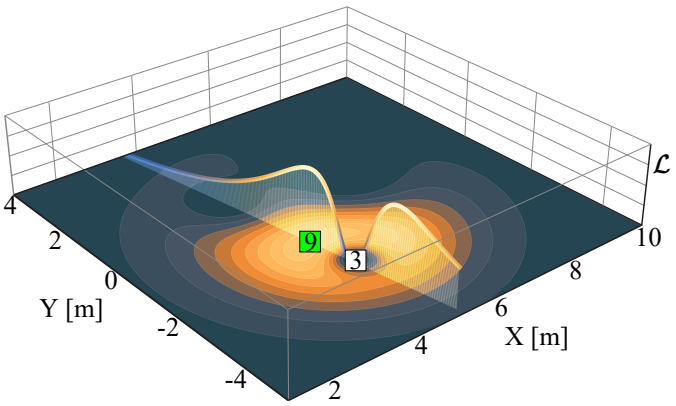

Fig. 2. A slice of neighbor-node's 3 belief towards node 9.

a "volcano" (given an optimization in two-dimensional space), a slice of this "volcano" towards node 9, is actually depicted.

The optimization of (13) may be performed in an MLE or a MAP framework. In this paper, we demonstrate MAP, constructing the posterior distribution by integrating (2), as

$$f(d \mid \boldsymbol{z}) = \frac{f(\boldsymbol{z} \mid d) f(d)}{\int_0^{+\infty} f(\boldsymbol{z} \mid \delta) f(\delta)\, \mathrm{d}\delta} \quad (14)$$

Since the produced likelihood

$$\mathcal{L}(\boldsymbol{z}_{i,j} \mid d_{i,j}) = \frac{\exp\left\{-\frac{1}{2}\left(\frac{\boldsymbol{z}_{i,j} - (P_0 - 10\alpha \log(d_{i,j}))}{\sigma_{dB}}\right)^2\right\}}{\sigma_{dB}\sqrt{2\pi}} \quad (15)$$

has the form $\kappa \cdot \exp\{g(\boldsymbol{z}_{i,j}, \boldsymbol{p}_i, \boldsymbol{p}_j)\}$, we can reduce its computation cost by reforming it as

$$\widehat{\boldsymbol{p}}^{(t)}_i = \arg\max_{\boldsymbol{p} \in \mathbb{R}^2} \sum_{j \in \mathcal{B}_i} g(\boldsymbol{p} \mid \boldsymbol{z}_{i,j}, \widehat{\boldsymbol{p}}^{(t-1)}_i) \quad (16)$$

where the parameters of g depend on the parameters of f.

B. ARLCL Process

ARLCL is assembled into two phases; the offline phase, where policies and parameterizations are set, and the online phase, where the nodes' positions are iteratively refined.

A) Offline phase

During the online phase, several mechanisms and policies are considered that need to be set in the offline phase. These are the selection of the ranging model's parameters, the nodes' optimization order, and when to terminate the optimization.

1) Adjustment of the ranging model: The acquisition of a representative ranging likelihood model can be made with the collection of training data during a survey. Such a process is a typical requirement and foundation for most CL systems [10], [26]–[28], including both MS and ML-PSO methods. Equation (2) is then fitted (typically in a least-squares approach), and the parameters of g are determined. These parameters could alternatively be obtained from the literature, yet, different environments and hardware introduce different ranging responses.

2) Optimization order: The nodes' enumeration in $\mathcal{A}$ defines the system's optimization order. This constitutes a vital step for ARLCL as it grants it a flip-ambiguity mitigation capability and faster convergence. In practice, the true swarm's shape may expand in various directions and neighborhood densities. Therefore, it is reasonable to execute the localization in a spatial order that prevents the process from bouncing randomly across distant neighborhoods, for which there might be little effective information available. This results in improved spatial continuity of the optimization and, thus, reduced occurrence of clusters that are flipped relatively to other clusters.

To disfavor a spatially incoherent optimization, ARLCL prioritizes the nodes for which stronger beliefs are available from their neighbors. Such effective neighbors often form a common neighborhood. In general, the effectiveness of some $\boldsymbol{z}_{i,j}$ (measured by neighbor-node j) reflects the likelihood of the mode distance from which it could have been taken. In other words, it reflects the likelihood at the distance where (14) is globally maximized when conditioned on the measurement.

To quantify the volume of information about node i, ARLCL considers the measurements from its b most effective neighbors by accumulating their likelihoods. Since (14) produces densities that are unimodal and are monotonically increasing with the RSS, we could instead consider the RSS directly to avoid unneeded computations. Based on the above, the available information per node (offered by its neighbors) can be quantified and used to order the nodes accordingly.

3) Positioning convergence: On each iteration step t and according to the optimization order, the position of a single node is refined. When all nodes have been considered (i.e., before $t = N+1$), the system assesses whether the optimization should stop. This point signifies the end of a single cycle c (the period of which is N iteration steps) and the beginning of a new one. Our criterion is based on the trend of the positioning changes in the last w cycles. Let a_c be the average distance change between two successive cycles $c-1$ and c, with

$$a_c = \frac{2}{\sum_{i=1}^{N} |\mathcal{B}_i|} \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{B}_i} \left| \hat{d}_{i,j}^{(c-1)} - \hat{d}_{i,j}^{(c)} \right|, \quad \forall j > i \tag{17}$$

The positions used to compute $\hat{d}_{i,j}^{(0)}$ (and thus, a_1) are those assigned to the nodes at their initialization. Our statistic considers, across different cycles, the various changes in neighbor-distances $\hat{d}_{i,j}$ and not the changes of $\boldsymbol{p}_i$ themselves. Otherwise, it would become more prone to spatial fluctuations of entire neighborhoods often occurring during a CL optimization.

Let $\mathbb{O}$ denote the set of positive odd integers greater than 2. Let then $A^{(c)} = (a_{c-w+1}, a_{c-w+2}, \ldots, a_c)$ be a rolling contiguous sub-sequence of the last w successive statistics, with $w \leq c$ and $w \in \mathbb{O}$. The earliest possible element in $A^{(c)}$ is a_1. Moreover, the relation $w \leq c$ renders the computation of the first available window $A^{(c)}$ infeasible until c has become as large as w. Based on the current window's state at some cycle c, the convergence estimator can be acquired as

$$T^{(c)} = \frac{12}{w\,(w^2-1)} \sum_{i=1}^{(w-1)/2} i \left(A^{(c)}_{(w+1)/2+i} - A^{(c)}_{(w+1)/2-i} \right) \tag{18}$$

To assess whether convergence has been reached at cycle c, ARLCL checks whether the rolling trend $T^{(c)}$ has reached a predefined threshold T_0. This can be typically set close to 0.

The analytic expression of (18) is derived to enable a direct and fast computation of the scale factor of a linear regression, when applied to a set of odd observations which are indexed equidistantly. Here, the summation's coefficient becomes a precalculated constant. Hence, aside from its speed, the biggest strength of our metric is that it can identify the cycle, after which no further improvement in accuracy can be achieved without knowing what this accuracy is.

B) Online phase

ARLCL can operate either in a centralized arrangement or in a distributed one, where an elected node undertakes the coordination of the optimization steps. In this mode, every node i is responsible for estimating (as an individual optimization step) its own position after having received from all its neighbor-nodes: 1) their ranging measurements $\boldsymbol{z}_{i,j}$ and 2) the current estimations about their current positions $\boldsymbol{p}_j^{(t-1)}$.

1) System initialization: At the initial stage, assuming that the set of measurements $\mathcal{Z}$ has become available, $\mathcal{A}$ and $\mathcal{B}_i$ are constructed according to the optimization order policy and each node i is assigned a random initial position $\widehat{\boldsymbol{p}}_i$. Prior information about positions in $\widehat{\mathcal{P}}$ from previous optimization instances can be used to enable estimations for moving nodes. Nevertheless, ARLCL does not require any for the first estimation. Moreover, although the utilization of anchor-nodes would enable the positioning estimations to be referenced, our method does not require any anchor. This results in a set of estimated positions that are placed in an arbitrary reference system, where they remain relatively correct. The first cycle starts once the system initialization has been completed.

2) Execution of the optimization-cycle: This stage involves the consecutive repetition of the next three stages (3-5) until all nodes in the queue have been optimized (i.e. when $t = N$).

3) Construction of the next step objective function: The optimization of the next node in the queue (namely, node i) begins. The latest known position $\boldsymbol{p}_j^{(t-1)}$ of each neighbor $j \in \mathcal{B}_i$ is selected, together with $\boldsymbol{z}_{i,j}$. These pairs are used to construct the individual function components $g(\cdot)$ of the objective function (16). We should emphasize that ARLCL does not assume or require full connectivity between the nodes (either for remote or static scenarios). By the end of this stage, all pairs have been parsed, and (16) has been constructed.

4) Optimization of the objective function: In this stage, the produced objective function is optimized, and its global maximum represents the best estimate $\widehat{\boldsymbol{p}}_i^{(t)}$. ARLCL implements by default the Nelder–Mead method, a classical method for unconstrained optimization without derivatives. Yet, it does not dictate the use of any specific optimization framework.

5) Update of the currently optimized node: After the estimation $\widehat{\boldsymbol{p}}_i^{(t)}$ is retrieved, it becomes for the next steps (where node i will be used as neighbor-node j) the prior knowledge about node j's position (i.e., $\widehat{\boldsymbol{p}}_j^{(t-1)}$).

6) Convergence evaluation: With the completion of the optimization-cycle c and if $c \geq w$, ARLCL can check for convergence to assess whether the process can stop. If not, then all previous stages besides the initialization are recursively repeated. These six stages are outlined in Algorithm 1.

Algorithm 1: Online Phase of ARLCL

Input: N, $\mathcal{Z}$, b, trained params of $g(\cdot)$, any $\widehat{\mathcal{P}}_{\text{prior}}$
Define:
$t \leftarrow 1$, $c \leftarrow 1$ `// Optimization's iterators`
$T_0 \leftarrow 0$, $w \leftarrow 7$ `// Convergence params`
CL Initialization:
Build $\mathcal{A}$, $\mathcal{B}_i$, $\forall i \in \mathcal{A}$, and $\widehat{\mathcal{P}}$ (with latest $\widehat{\boldsymbol{p}}_i^{(0)}$, $\hat{d}_{i,j}^{(0)}$)
$b \leftarrow \frac{2}{\sum_{i=1}^{N}|\mathcal{B}_i|}$, $q \leftarrow \frac{12}{w(w^2-1)}$ `// Constants`
Repeat Optimization Cycle:
  for $i \in \mathcal{A}$ **do**
    $\widehat{\boldsymbol{p}}_i^{(t)} \leftarrow \arg\max_{\boldsymbol{p}\in\mathbb{R}^2} \sum_{j\in\mathcal{B}_i} g(\boldsymbol{p} \mid \boldsymbol{z}_{i,j}, \widehat{\boldsymbol{p}}_i^{(t-1)})$
    Update $\widehat{\mathcal{P}}$ with $\widehat{\boldsymbol{p}}_i^{(t)}$
  if $c \geq w$ **then**
    $T^{(c)} \leftarrow q\sum_{i=1}^{(w-1)/2} i\left(A^{(c)}_{(w+1)/2+i} - A^{(c)}_{(w+1)/2-i}\right)$
    if $T^{(c)} \geq T_0$ **then**
      return *final* $\widehat{\mathcal{P}}$ *with stats:* $[T^{(c)}, c, A]$)
  else
    $a_c \leftarrow b\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}_i}\left|\hat{d}_{i,j}^{(c-1)} - \hat{d}_{i,j}^{(c)}\right|, \quad \forall j > i$
    Update $A^{(c)}$ up to a_c
    $c \leftarrow c + 1$

V. Performance Evaluation

We present ARLCL's evaluation experiment against MS [11] and ML-PSO [6], [20] (source code is also released).

The baseline parameters [Opt:10^5, $Step$:10^{-3}] and [Opt:10^3, c_1:0.7, c_2:0.3, w:0.9, $particles$:700] for MS and ML-PSO respectively, where chosen to provide high optimization score instead of optimization speed. The evaluation was performed under extended indoor shapes that have been shown to be affecting flip-ambiguity the most [21]. Although RSS ranging with BLE is highly susceptible to environmental conditions (NLOS, multipath, etc.), we do not compare different ranging approaches (where these are a major factor); rather, the optimization approaches that are based on a common ranging model. Therefore, giving importance to the shape conditions instead of NLOS becomes more sensible.

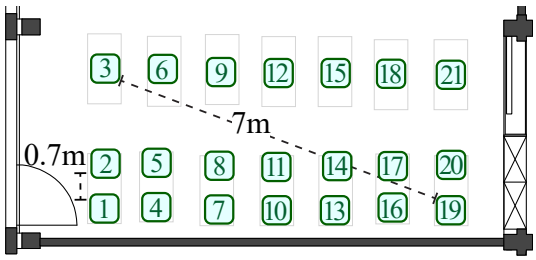

Fig. 3. Nodes' positions during the data collection.

To develop a shared indoor ranging model, 21 BLE enabled Raspberry Pi nodes were deployed ten times across a lecture room (Figure 3). Every node had, at each time, a random individual orientation. Their true positions and distances were tracked using a Leica BLK360 Imaging Laser Scanner. For a duration of 120 s, each node i was transmitting (at 50 Hz) and receiving (from every other node j) advertising packets, whose RSS was recorded. A total of 18 million measurements were collected. To identify the parameters α (path loss exponent), P_0 (reference RSS at 1 meter), and σ_{dB}, the measurements were fitted to our ranging likelihood model $\mathcal{L}(\boldsymbol{z}_{i,j} \mid d_{i,j})$.

To enable the assessment of different swarm sizes and shapes within the already irregular deployment, the initial set of 21 nodes was used to generate 19 groups corresponding to N number of nodes, with $N \in \{3, \ldots, 21\}$. Each group contained C_N combinations of N chosen nodes. Constrained by the maximum number of combinations $\binom{21}{N}$, for $N = 20$ and $N = 21$, we set $C_{20} = 21$ and $C_{21} = 1$ respectively. For all other groups, we set $C_i = 200, \forall i \in \{3, \ldots, 19\}$. In these cases, to determine which combinations to keep out of the possible $\binom{21}{N}$ ones, a sorted pool was used, holding the average node distances within each swarm. Finally, a grid selection of C_N swarms was performed from the sorted pool. For example, the 1st node combination in the group of $N = 3$ is {1,2,4}, while the 200th combination is the swarm {1, 3, 21}. In total, 3422 swarm instances were selected.

In total, 100 different initialization scenarios were examined per single instance to reduce the standard error in the methods' comparison. For every scenario, each component of $\widehat{\boldsymbol{p}}_i^{(0)}$ was assigned a random value from -10 m to 10 m.

For each generated scenario, the true distance $d_{i,j}$ between two nodes was used to acquire, based on our likelihood model $\mathcal{L}(\boldsymbol{z}_{i,j} \mid d_{i,j})$, the distribution of $\boldsymbol{z}_{i,j}$. We assessed the impact of variable noise levels in the measurements by assuming that every node j received $K_{i,j} = K \in \{1, \ldots, 20\}, \forall (i,j) \in \mathcal{A} \times \mathcal{B}_i$ RSS measurements from each neighbor node i.

Considering $z_{i,j}$ to be the average of all K samples, we use this information from now on to infer the nodes' distance. The generated measurements were shared between all methods.

Our experiment involves (per CL method) the position estimation of every node in 7.6 million system combinations of a specific swarm size, set of nodes, initialization state, and noise level. To quantify the difference between $\mathcal{P}$ and $\widehat{\mathcal{P}}$ and, thus, the estimation's performance $\varepsilon(\mathcal{P}, \widehat{\mathcal{P}})$ in each system solution, we utilize the cost function defined in (5). However, since the performed localization is anchor-free, the difference between the ground truth $\mathcal{P}$ and the estimation $\widehat{\mathcal{P}}$ also involves a random and unknown translation, rotation, and

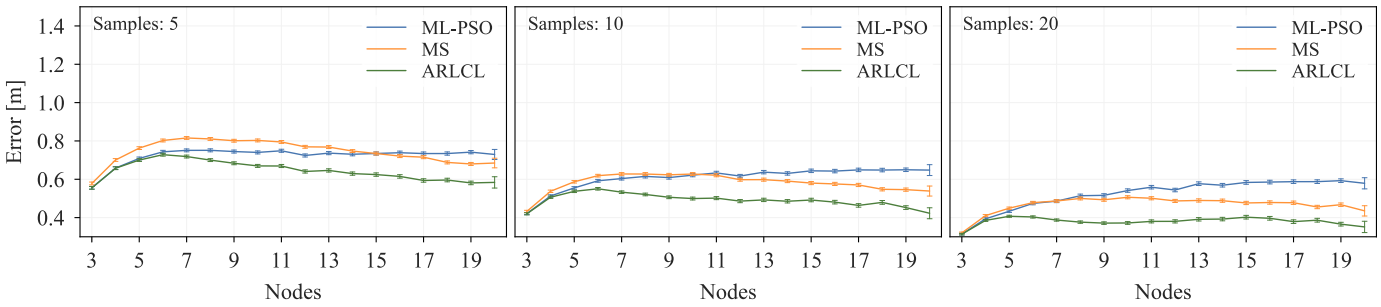

Fig. 4. Positioning error per number of nodes for 3 sample-size groups (5/10/20-RSS samples).

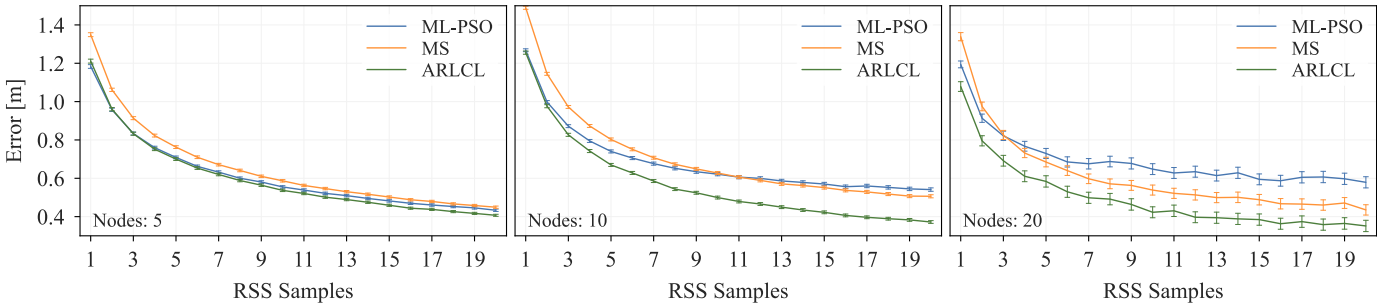

Fig. 5. Positioning error per number of samples for 3 node-size groups (5/10/20-nodes).

possibly reflection between the two sets. Therefore, before evaluating the cost function, we first eliminate this difference by performing a rigid and correspondence-based registration between the sets $\mathcal{P}$ and $\widehat{\mathcal{P}}$.

The HPC UBELIX cluster at Bern University was utilized to support this extensive evaluation computationally, allocating 4000 CPU cores across 100 compute nodes for a week.

A. Evaluation Results

The parameters offering the best fit of (15) to our collected training data, are: $\alpha \approx 1.274$, $P_0 \approx -53$ and $\sigma_{dB} \approx 9.74$.

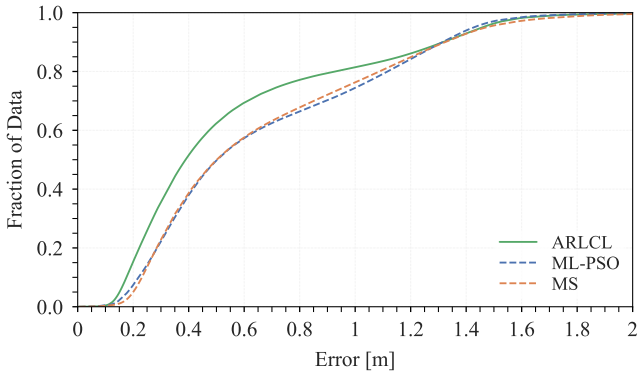

Fig. 6. Cumulative Distribution Functions (CDFs) of the positioning errors.

Figure 6 presents the cumulative error probability in meters for all evaluated scenarios. We see that up to the 85$^{\text{th}}$ percentile, ARLCL produced estimations of higher accuracy.

More specifically, Figures 5 and 4 show the positioning errors of the evaluated methods by the number of nodes and samples. The confidence intervals of 2 standard errors are also presented, which become larger for the cases of 20 nodes due to fewer evaluations. We see that in almost every case, ARLCL outperforms both baselines. Figure 4 suggests that our gain becomes less for very small swarms, which is explained due to the low probability of flips in such cases. This is also why the left group in Figure 5 ($N = 5$) shows a smaller gain than the other two groups. In contrast, we see a positive correlation in our gain as the swarm increases. According to the plots, the error reduction for the combination of maximal settings (i.e., $N = 20$, $K = 20$) that is introduced by ARLCL is 40% compared with ML-PSO and 24% compared with MS. Hence, ARLCL can make progressively better use of the available collective information as the swarm's size increases and utilize additional measurements more effectively.

Comparing ML-PSO to MS, we notice that the probabilistic ML-PSO outperforms MS for small swarms or limited RSS samples. However, as the measurements become less noisy (with the increase of samples), utilizing the likelihood distribution of the measurements in their method becomes less helpful. At the same time, when the number of nodes increases, its complexity makes the discovery of the global optimum more challenging, leading to even worse performance than MS.

Another observation is that the amount of noise affects the positioning error monotonically (Figure 5). This means that factors impacting the measurements, such as NLOS, reflections, dynamic environment, etc., should cause similar trends. Regarding the error at a different number of nodes, a threshold seems to exist (close to small sizes) before and after

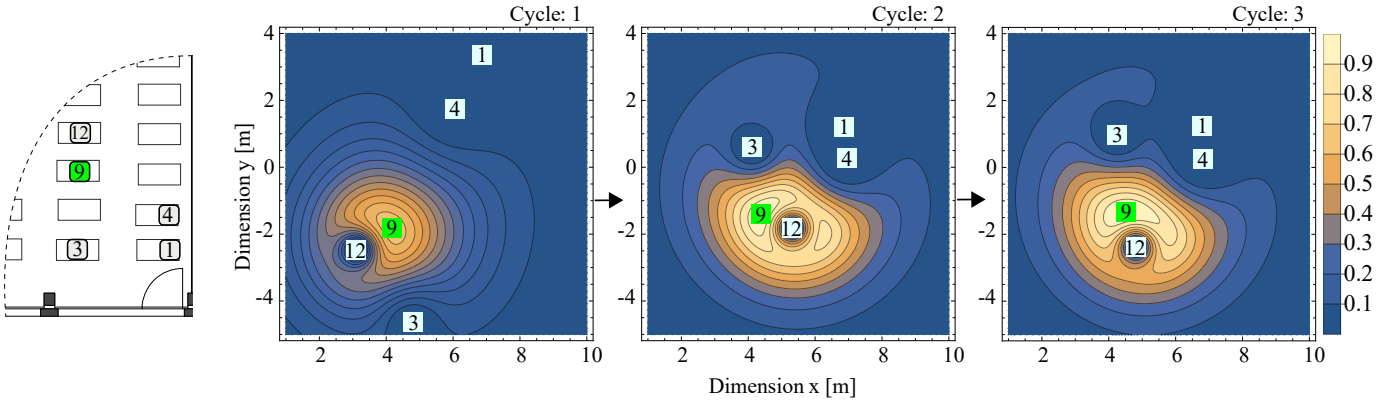

Fig. 7. Evolution of the objective function (contours basemap) for estimating the position of node 9 based on its neighbors' beliefs.

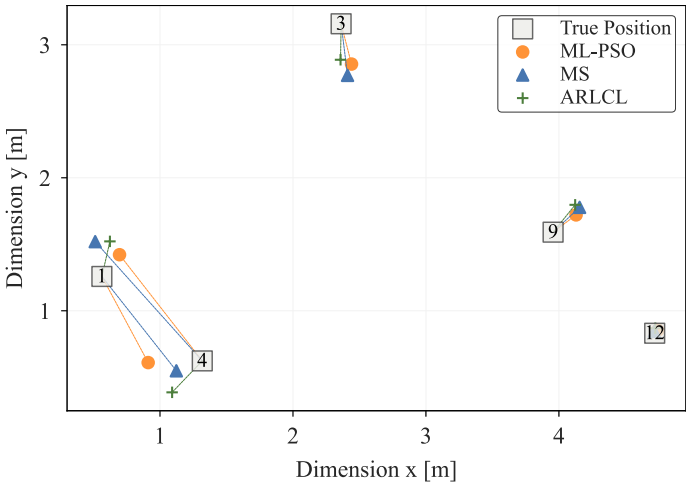

Fig. 8. Estimation residuals (Swarm {1,3,4,9,12}, 36$^{\text{th}}$ iteration, $K = 19$).

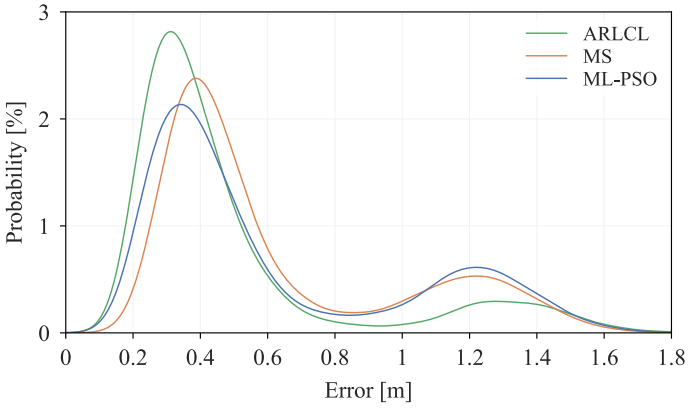

Fig. 9. Distribution of positioning errors for the cases of $N = 10$, $K = 10$.

which the accuracy improves monotonically with a long right tail. This happens due to flip-errors [21], where a subgroup of nodes with relatively correct positions has been connected with another such subgroup, incorrectly. It results from inadequate spatial information in the measurements (bonds) for characterizing the correct spatial relationships, leading inevitably to misaligned node clusters. Their occurrence starts at an early point (e.g., $N = 5$) and then increases with the number of nodes due to the coincidental increase of the likelihood for a weak bond to arise. Figure 8 shows an example of such flips. We see that ML-PSO and MS flipped nodes 1 and 4. Yet, ARLCL can produce a better estimation within three cycles as depicted in Figure 7. Here, although the estimations are non-georeferenced and reflected (as no anchors exist), they remain relatively correct. To stress the impact of flip-ambiguity in the positioning errors, we show in Figure 9 the distribution of errors for the average case of $N = 10$, $K = 10$. We see that flips are very common, leading to a bimodal distribution of errors. However, ARLCL has greater robustness against them.

Although no hard convergence detection was used to interrupt the optimization process for the methods, in the case of ARLCL (where the refinements are progressive across the different cycles), we examined the resulting trajectories. Figure 10 presents, per swarm size and across 20 cycles, the average convergence towards the accuracy of the final estimation. We observe that the bigger the swarm is, the slower the system converges to the final accuracy. For swarms of low size ($N \leq 5$), the estimations reach their maximum accuracy at the 2$^{\text{nd}}$ cycle. On the contrary, for bigger swarms, the required cycles for achieving the system's maximum possible accuracy increase. Lastly, ARLCL's convergence time for a swarm of size ($N = 21$) and with a random initialization was empirically observed to be approximately between 2 and 4 seconds.

VI. Discussion and Conclusion

In this paper, we presented and demonstrated Anchor-free Ranging-Likelihood-based Cooperative Localization, a novel localization framework that considers the cross-ranging measurements between wireless nodes to estimate their positions. This is performed under a refining scheme of successive optimization cycles offering flip-ambiguity mitigation. Our method leverages the a priori knowledge of the ranging uncertainty to assign proper importance to every distance.

An evaluation was also presented against Mass-Spring and Maximum Likelihood - Particle Swarm Optimization CL methods. We showed, for a ranging technology known to be

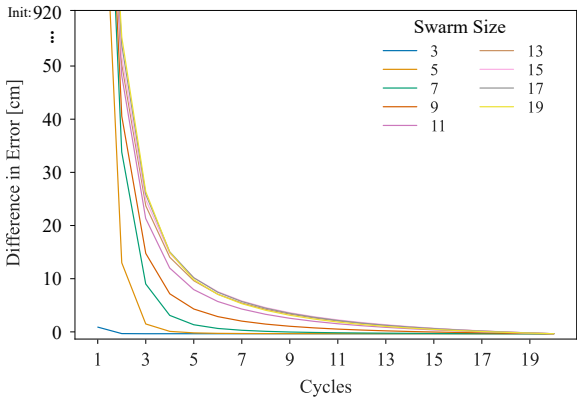

Fig. 10. Error's convergence towards its final state.

noisy, that ARLCL introduced an improvement in positioning accuracy under almost every setting and up to 40%. Our method was able to extract, by the swarm's increasing intercommunication, gradually more spatial information compared to the baselines. The experiment was designed with the effect of flip-ambiguity in mind (and less of the environmental conditions), because the ranging model, measurements, and technology remained exactly the same for all evaluated methods. Therefore, assessing a factor impacting the measurement itself becomes of secondary importance. Our demonstrated gain is still expected for either open-space scenarios (e.g., localization in air, sea) or more complex and dynamic indoor spaces. It is worth mentioning that preliminary results of a work in progress, where we deployed 40 UWB transceivers across $450\,\mathrm{m}^2$ indoors and under real NLOS conditions, have shown further performance gain.

As a technology-agnostic method, ARLCL is applicable to any CL system that uses ranging for its estimations. It does not depend on any prior positioning information or the existence of anchors (although it can use them for geo-referencing). Tracked entities in such systems can span from crowds of pedestrians and vehicles to UAVs and swarms of robots.

Further improvement in ARLCL's performance is theoretically possible, opening up ideas for future work. For example, researchers could develop a method for statistically identifying and rejecting flips in the produced solutions or explore ways to continuously update and refine the ranging-likelihood model used by the system. The latter could possibly be enabled based on federated learning over the exchanged messages.

References

[1] H. Ping, Y. Wang, D. Li, and T. Sun, "Flipping Free Conditions and Their Application in Sparse Network Localization," *IEEE Transactions on Mobile Computing*, vol. 21, pp. 986–1003, 2022.

[2] A. A. Kannan, B. Fidan, and G. Mao, "Robust Distributed Sensor Network Localization Based on Analysis of Flip Ambiguities," in *IEEE Global Telecommunications Conference*, (New Orleans, USA), 2008.

[3] A. A. Kannan, B. Fidan, and G. Mao, "Analysis of flip ambiguities for robust sensor network localization," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2057–2070, 5 2010.

[4] S. Chang, Y. Li, H. Wang, W. Hu, and Y. Wu, "RSS-Based cooperative localization in wireless sensor networks via second-order cone relaxation," *IEEE Access*, vol. 6, pp. 54097–54105, 2018.

[5] S. Wang and X. Jiang, "Three-Dimensional Cooperative Positioning in Vehicular Ad-hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 937–950, 2021.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE.

[7] G. C. Calafiore, L. Carlone, and M. Wei, "A distributed technique for localization of agent formations from relative range measurements," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 42, no. 5, pp. 1065–1076, 2012.

[8] S. Tomic, M. Beko, and R. Dinis, "RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2037–2050, 2015.

[9] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, 2009.

[10] S. Mazuelas, A. Conti, J. C. Allen, and M. Z. Win, "Soft Range Information for Network Localization," *IEEE Transactions on Signal Processing*, vol. 66, no. 12, pp. 3155–3168, 2018.

[11] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, (Los Angeles, California, USA), pp. 1–13, ACM, 2003.

[12] C. Xunxue, S. Zhiguan, and L. Jianjun, "Distributed localization for anchor-free sensor networks," *Journal of Systems Engineering and Electronics*, vol. 19, no. 3, pp. 405–418, 2008.

[13] S. Nawaz and S. Jha, "A graph drawing approach to sensor network localization," in *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, MASS*, (Pisa, Italy), p. 12, IEEE, 2007.

[14] J. Eckert, F. Villanueva, R. German, and F. Dressler, "Distributed mass-spring-relaxation for anchor-free self-localization in sensor and actor networks," in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, (Lahaina, USA), IEEE, 2011.

[15] B. Zhou and Q. Chen, "On the Particle-Assisted Stochastic Search Mechanism in Wireless Cooperative Localization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4765–4777, 2016.

[16] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[17] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *IEEE International Symposium on Information Theory - Proceedings*, (St. Petersburg, Russia), 2011.

[18] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft Information for Localization-of-Things," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2240–2264, 2019.

[19] 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, 2005.

[20] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative Sensor Network Localization: Algorithms and Practical Issues," 6 2018.

[21] O. H. Kwon, H. J. Song, and S. Park, "Anchor-free localization through flip-error-resistant map stitching in wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, 2010.

[22] Q. Guo, K. Zhang, T. Du, and S. Qu, "A localization algorithm reducing flip ambiguities for WSNs with measurement errors," in *2016 3rd International Conference on Informative and Cybernetics for Computational Social Systems, ICCSS 2016*, pp. 143–150, Institute of Electrical and Electronics Engineers Inc., 10 2016.

[23] A. A. M. Saleh and R. A. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, 1987.

[24] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[25] A. T. Ihler, *Inference in Sensor Networks: Graphical Models and Particle Methods*. PhD thesis, Massachusetts Institute of Technology, 3 2005.

[26] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint Indoor Localization and Radio Map Construction with Limited Deployment Load," *IEEE Transactions on Mobile Computing*, vol. 14, 5 2015.

[27] S. He, W. Lin, and S. H. Gary Chan, "Indoor Localization and Automatic Fingerprint Update with Altered AP Signals," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 1897–1910, 7 2017.

[28] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.