IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 February 2023, accepted 13 March 2023, date of publication 24 March 2023, date of current version 29 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261567

APPLIED RESEARCH

Self-Localization of Ultra-Wideband Anchors: From Theory to Practice

PABLO CORBALÁN[1], (Member, IEEE), GIAN PIETRO PICCO[1], (Senior Member, IEEE), MARTIN COORS[2], AND VIVEK JAIN[2], (Senior Member, IEEE)

[1]Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy
[2]Bosch Research, Sunnyvale, CA 94085, USA

Corresponding author: Gian Pietro Picco (gianpietro.picco@unitn.it)

This work was supported in part by the Italian Government via the NG-UWB Project (MIUR PRIN 2017).

ABSTRACT Ultra-wideband (UWB) radios are increasingly exploited for localization in complex deployments with tens or even hundreds of anchor nodes, whose position must be measured accurately: a long and error-prone *manual* chore. *Self-localization* techniques can estimate anchor positions *automatically*, from relative distances acquired via UWB, but are often evaluated i) only with a handful of anchors all in range, a far cry from the large, multi-hop setups above, and ii) in simulation, therefore neglecting *system* aspects and undermining immediate use in real contexts. We tackle the problem from a different, *practical* viewpoint. First, we exploit *three real-world, large-scale, multi-hop UWB testbeds*, a unique asset in the literature. Second, we build upon state-of-the-art multidimensional scaling (MDS) that, unlike recent UWB-based techniques, is not limited to a specific type of localization or infrastructure, is computationally lightweight, and does not require training. Third, we integrate MDS with in-field distance estimation, yielding a complete, immediately usable self-localization system. Our evaluation, both in simulation and in the testbeds above, analyzes extensively the parameter space (e.g., the impact of ranging errors or anchor connectivity) and shows that anchor positions are determined quickly and accurately, minimizing manual labor without significant detriment to the accuracy of the localization system relying on them.

INDEX TERMS Anchor positioning, localization, multidimensional scaling, ranging, ultra-wideband.

I. INTRODUCTION

The accuracy and reliability of localization systems based on ultra-wideband (UWB) radios is gradually fostering their adoption in a plethora of sectors [1], [2], [3], [4]. As these systems become part of the fabric of our daily lives, practical issues concerned with their deployment in big numbers and over large areas arise. Covering an area of interest (e.g., a factory, a mall, an office floor) involves tens or even hundreds of nodes, whose fast *and* reliable deployment is key to contain the operational costs associated with this technology and boost its acceptance [3], [4], [5].

A. MANUAL ANCHOR LOCALIZATION

Knowing the accurate positions of the deployed *anchors*, i.e., the reference nodes w.r.t. which target localization occurs, is crucial to its performance. This measurement, typically performed *manually* via specialized equipment (e.g., a laser meter), is tedious and effort-demanding, especially with many anchors across large areas. Worse, it is error-prone; even with expert personnel, human errors may still occur that lead to degraded localization performance, thus requiring debugging sessions further increasing effort.

To offer a concrete example, we had first-hand experience with Leica 3D Disto [6] which is a cross between a surveyor's robotic total station and a handheld precision Laser Distance Meter (LDM). We used Leica 3D Disto to obtain ground truth positions in our industrial plant testing facility, described later; similar alternatives exist on the market. Although these specialized tools are a significant leap forward compared to LDM, they are still expensive, labor intensive, and require training. Further, they require the station and the deployed anchors to be in complete line of sight (LOS). In a large installation, this cannot be guaranteed from a single point

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed Kheir.

towards all anchors. Therefore, to localize all anchors, the station must be moved to several points in the target area; this requires recalibration of ground truth along with expertise to convert all positions to a common coordinate system.

B. ANCHOR SELF-LOCALIZATION

An alternative approach is to *automatically* compute the anchor positions via the distance estimation (*ranging*) obtained with the on-board UWB radio, enabling the system to perform anchor *self-localization*. This approach is very appealing, as it minimizes the manual effort involved in the measurement of anchor positions. For instance, the Leica 3D Disto above could be used to localize a few anchors from a single point, and delegate the remaining ones to the self-localization system. Further, the latter would enable localization even in scenarios where human intervention is impractical or even impossible [7]. Actually, it was precisely these observations, grounded in industrial use cases and experiences, that motivated us to investigate the in-field, practical feasibility of self-localization to support, or completely replace, existing manual methods.

C. DOES IT WORK OVER LARGE-SCALE, MULTI-HOP AREAS?

This approach is of practical interest only if the anchor positions automatically estimated are accurate enough, and errors are not detrimental to the overall localization performance.

Several techniques exist that tackle this goal (§II). Unfortunately, they are evaluated, often via simulation, only in a small-scale, single-hop scenario with a handful of anchors in range of each other. For instance, a recent work [8, Table 1] reports 8 self-localization solutions for UWB. In all cases, the anchors are always in range of each other. Two solutions are evaluated only in simulation; the remaining rely on experiments using at most 8 anchors, and covering at most a 16×16 m^2 area (with 5 anchors). In the same paper [8], the authors offer a slightly larger evaluation with 15 anchors over an area of 41×26 m^2.

These setups are a valid means to evaluate the techniques at hand, but are a far cry from the large-scale, multi-hop UWB scenarios increasingly targeted by industry where, in contrast, a manual approach is clearly impractical or even prohibitive and self-localization essentially becomes a must. Therefore, the question above remains open.

Further, by focusing on evaluating their novel techniques, these studies often neglect the self-localization system enabling their use, undermining immediate applicability (and repeatability) in real contexts.

D. SEEKING AN ANSWER FROM A PRACTICAL VIEWPOINT

In this paper, we approach the problem from an opposite perspective that, however, is of immediate practical relevance.

First, we exploit *three real-world, large-scale, multi-hop testbeds* deployed in an industrial setting (PLANT)

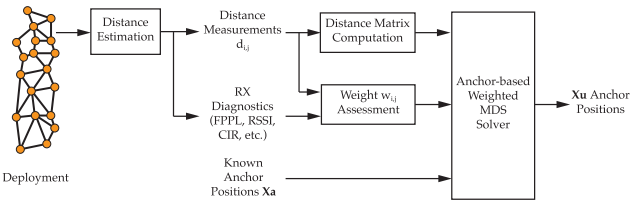


FIGURE 1. Anchor self-localization system architecture. After deploying the wireless nodes, a distance acquisition system (e.g., UWB TWR) measures the distance $d_{i,j}$ between neighboring nodes, optionally storing received (RX) signal diagnostics. The distance estimates $\hat{d}_{i,j}$ together with the known node positions $\mathbf{X_a}$ are used in a weighted MDS solver to determine the unknown anchor positions $\mathbf{X_u}$.

and two indoor areas of the University of Trento (DEPARTMENT, RECEPTION). They contain up to 36 anchors over $\sim$3000 m^2 and have very different geometries and non-line-of-sight (NLoS) conditions. The size and realism of *any* of these testbeds, representative of many use cases, is already significantly higher than what commonly reported in the UWB literature. The availability of *three* of them enables us to explore real-world setups with different environmental conditions and geometries, significantly affecting distance estimation and, ultimately, localization.

Second, unlike the aforementioned studies, *we do not propose yet another novel technique* to solve the problem of self-localization we formalize in §III, and rely instead on multidimensional scaling (MDS). This well-known and general technique has been successfully applied to different localization technologies, including UWB. The reason of our choice is that, in comparison with many UWB-specific approaches (§II), MDS does not pose requirements on the localization technique or infrastructure used, is computationally lightweight, and does not require training.

Specifically, we rely on a *weighted* version of MDS [9]. In contrast to the classical one, which requires full-mesh connectivity among anchors, this weighted variant does not and is therefore applicable to the large-scale, multi-hop scenarios we target where many anchors are not in range with each other. Further, this variant exploits knowledge of a few anchor positions to estimate the unknown ones, removing the need for the rigid body transformations required by other common approaches.

Third, we integrate the MDS algorithm with the mechanics of acquiring in-field ranging estimates, yielding *a full-fledged, immediately usable self-localization UWB system* whose high-level operation is shown in Fig. 1. Self-localization begins by measuring the distance $d_{i,j}$ between each anchor and its neighbors, via two-way ranging (TWR) [10], [11], [12]. This first step populates a distance matrix, input to the chosen state-of-the-art MDS algorithm (§IV) that performs the actual computation of unknown positions.

E. A REAL-WORLD SYSTEM EVALUATION

After describing in more detail the system implementation (§V), we evaluate its performance in our testbeds (§VI).

The combination of these two elements, and the resulting findings, are the main contribution of this paper. To the best of our knowledge, this is the first study that ascertains the performance of a full-fledged self-localization UWB system in-field and across different testbeds in large, multi-hop areas with many anchors.

Self-localization is affected by several parameters, determined by deployment and configuration choices as well as environmental conditions. Therefore, before the actual testbed experiments, we exploit the geometric layout of our testbeds in simulation to understand the sensitivity of the algorithm to these parameters and the conditions under which it works at its best, deriving the insights and guidelines driving the in-field experimental campaign.

Results indicate that, across all three real-world testbeds, our system can estimate the position of all anchors with a mean error well below 50 cm despite the presence of ranging errors of higher magnitude. Further, we analyze the performance of a localization system tracking a mobile target, and demonstrate that the error in estimating the anchor positions yields only a minor increase in the target positioning error w.r.t. a configuration in which the position of anchors is manually determined.

Our findings show how self-localization provides an effective and practically-relevant means to significantly reduce the human effort in the deployment of UWB systems. Specifically, the quantitative evidence we report from real-world, large-scale, multi-hop areas, enables a concrete understanding, largely missing in the literature, of the benefits and limitations of self-localization in these contexts.

Of course, these are biased by our specific choice of techniques, systems details, and environments. Further, there is also margin for improvements and future work on the topic. We encourage both *repeatability* of our results and follow-up research by publicly releasing[1] our system and datasets. We outline these opportunities (§VII) before concluding remarks (§VIII).

II. RELATED WORK

The problem of determining the positions of wireless nodes received attention since the early days of wireless sensor networks [13]. Most approaches first measure the distance between neighbors, then estimate all node positions with an algorithm based either on MDS or multi-lateration. In general, distance estimates are obtained using UWB two-way ranging (TWR) exchanges [11], [12], [14] or, in the case of narrowband radios (e.g., WiFi, Bluetooth, IEEE 802.15.4), from significantly less accurate received signal strength (RSS) measurements [15], [16].

A. MULTIDIMENSIONAL SCALING

Early approaches exploited mere connectivity information, estimating the distance between nodes as the number of communication hops between them [17]. The resulting distance matrix is input to an MDS solver yielding only a rough position estimate, due to the inaccuracy of distance estimation. The distributed MDS approach in [18] computes the relative positions of neighbors as local maps that are then stitched together into a global one. The latter can be aligned to a known coordinate system by applying a translation, a rotation, and possibly a reflection to the estimated positions. The work in [15] proposes an alternate distributed approach encompassing known anchor positions in the computation, to reduce the computational load and improve accuracy, and exploiting non-binary weighting functions, e.g., to down-weight measurements based on their confidence.

More recently, Franco et al. [9] reformulated the MDS minimization problem to directly take into account known anchor positions in a centralized approach. As a result, in each iteration only unknown positions are computed. This simplifies the problem and removes the need for roto-translations, as estimated positions can potentially be aligned with known ones in the target coordinate system directly during the minimization process. This reformulation, however, has been tested only in simulation and a small setup with a handful of UWB nodes.

B. MULTI-LATERATION

Other recent approaches targeting UWB, e.g., the auto-positioning feature offered by Decawave [19] and the works in [7], [20], and [21] are based on multi-lateration. After measuring the distance between nodes, these approaches iteratively build a coordinate system by assigning a 1^{st} node to the origin (0, 0, 0) and placing a 2^{nd} one on the positive x-axis at ($d_{1,2}$, 0, 0). The coordinates of the 3^{rd} and 4^{th} nodes, derived in closed form, set the positive direction of the y- and z-axis. The remaining node positions are determined based on these 4 nodes and available distance information, e.g., using a non-linear least squares solver. In comparison, MDS provides a more elegant and efficient formulation of the problem, *simultaneously* computing the position of all nodes. Moreover, including known anchor positions in MDS removes the need for a rigid body transformation to align the built coordinate system to the desired one, generally yielding more accurate estimates.

C. OTHER APPROACHES

Another set of UWB-specific approaches tackles the problem by introducing additional system and/or operational constraints. The solution in [22], evaluated experimentally with only 2 unknown anchors, relies on a variant of time-difference of arrival (TDoA) to self-localize anchors, posing additional synchronization and infrastructure constraints that may not be necessary if alternative techniques are used for the actual localization. Other works rely on a mobile tag to acquire ranging measurements from the deployment area, increasing their number and diversity. In [23], the fusion of inertial measurement unit (IMU) and UWB data from a free-roaming tag enables simultaneous localization and mapping (SLAM),

[1] https://github.com/d3s-trento/selfloc

yielding anchor localization as a necessary by-product; however, the system is demonstrated only in simulation. Instead, in [24] the mobile tag is constrained to move on a known path and its data matched against accurate ground truth; the latter is provided by RTK GPS and generally difficult to achieve indoor. Finally, the work in [8] extends the multi-lateration approach in [20] by focusing specifically on the NLoS problem and analyzing the channel impulse response (CIR) associated to the UWB signal via machine learning. As such, this approach is computationally heavy, requires extensive training, and is difficult to transfer to different environments.

D. IMPACT ON THIS PAPER

Unlike these studies, we are not proposing our own self-localization technique. Instead, our goal is to ascertain whether existing ones, hitherto evaluated only in simulation or small, single-hop scenarios, are applicable to the large-scale, multi-hop ones increasingly targeted by UWB applications and exemplified by our three real-world testbeds (§I).

Nevertheless, comparing in-field and on multiple testbed all the techniques above, would be a daunting and somewhat unreasonable task, given that their implementation is often not available and/or limited to simulation and therefore glossing over details of practical relevance. Indeed, our other stated goal is to provide a full-fledged, immediately usable self-localization UWB system, along with qualitative guidelines informing its use and quantitative evidence of the performance that can be attained in-field. Together, these considerations forced us to select a single technique.

Our choice landed on the MDS-based approach by Franco et al. [9], due to the following reasons. As mentioned, MDS is a well-known and general technique, successfully applied in several contexts and with different technologies, and more efficient and flexible than approaches based on multi-lateration. Further, unlike the last set of approaches above, it does not pose additional system constraints, is computationally lightweight, and does not require training. Finally, the specific weighted variant in [9] lifts the constraint of classical MDS that all anchors must be in range, therefore directly addressing large-scale, multi-hop scenarios, despite having been evaluated only in simulation and small-scale, single-hop ones.

We describe next the self-localization problem and the techniques we employ to solve it.

III. PROBLEM STATEMENT

Consider a network deployment of N nodes in an M-dimensional space, $M \in \{2, 3\}$, and position coordinates $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}]^T \in \mathbb{R}^{N \times M}$. We arrange $\mathbf{X}$ so that $\mathbf{X} = [\mathbf{X_u}, \mathbf{X_a}]^T$, where $\mathbf{X_u}$ denote the n unknown coordinates in $\mathbf{X}$ and $\mathbf{X_a}$ the m anchor positions known a priori. Our goal is to determine the unknown $\mathbf{X_u}$ positions that satisfy the measurable distance $d_{i,j}$ between nodes i and j, $\forall i, j \in \{1 \ldots N\}$, and respect the known $\mathbf{X_a}$ coordinates.

The unknown anchor positions $\mathbf{X_u}$ can be determined with overall knowledge of the Euclidean distance between all anchors, i.e., in 2D:

$$d_{i,j} = \left\| \mathbf{x_i} - \mathbf{x_j} \right\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

The $d_{i,j}$ distances are typically estimated manually with specialized equipment, e.g., laser meters. In contrast, our automatic, in-band approach estimates them *directly* via the UWB radio (§V) by using two-way ranging [10], [25], [26].

The corresponding $\hat{d}_{i,j}$ estimates can be arranged as

$$\hat{D} = \begin{pmatrix} 0 & \hat{d}_{1,2} & \cdots & \hat{d}_{1,N} \\ \hat{d}_{2,1} & 0 & \cdots & \hat{d}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{N,1} & \hat{d}_{N,2} & \cdots & 0 \end{pmatrix} \tag{2}$$

where $\hat{D}$ is a square and symmetric *distance matrix* of size $N \times N$ whose main diagonal elements $\hat{d}_{i,i} = d_{i,i} = 0$.

Although ideally $\hat{d}_{i,j} = \hat{d}_{j,i}$, this does not hold in practice. Each measurement $\hat{d}_{i,j} = d_{i,j} + \eta$ is affected by a noise η that depends on a variety of system and environmental factors, e.g., NLoS conditions. However, the matrix can be easily made symmetric by setting $\hat{d}_{i,j} = \hat{d}_{j,i}$ to the median of the distribution resulting from multiple measurements. Further improvements are enabled by techniques that, e.g., reject outliers, weight measurements, and perform NLoS identification and mitigation based on RX diagnostics [27], [28] (Fig. 1).

On the other hand, *not all distances in (2) can be measured*. In large deployments, anchors are separated by multiple wireless hops, beyond the communication range enabling pairwise distance estimation. In practice, this limits the $\hat{d}_{i,j}$ measurements available, a crucial aspect to be accounted for in determining the unknown positions $\mathbf{X_u}$.

Finally, the distance between the m known anchors can be directly *computed* as $d_{i,j} = \left\| \mathbf{x_i} - \mathbf{x_j} \right\|$. This reduces the overhead of acquiring distance estimates and the computational load of the self-localization algorithm, described next.

IV. ANCHOR SELF-LOCALIZATION

We illustrate the salient aspects of the state-of-the-art MDS algorithm [9] we use to determine the unknown anchor positions $\mathbf{X_u}$ via the known m positions $\mathbf{X_a}$ and the distance matrix $\hat{D}$. The algorithm is based on a weighted variant of MDS. We first introduce classical MDS and how it can be used towards self-localization. We then discuss its main drawback, i.e., the need for a complete distance matrix $\hat{D}$, not available in our context (§III), and how the weighted variant overcomes this problem. We add the possibility to account for known anchor positions, enabling the algorithm to directly estimate only the n unknowns. Finally, we discuss the impact of the initial estimate input to the algorithm in relation to local minima.

A. MULTIDIMENSIONAL SCALING

MDS is a technique that, given a set of objects, aims at finding a low-dimensional representation in which their distance

closely matches the measured pairwise "dissimilarity" in the original high-dimensional representation [15], [29]. In our context, MDS can determine the Cartesian coordinates of anchors given the pairwise distances between them. If distance measurements are error-free and *all* pairwise distances are available, classical MDS provides a closed-form solution to determine the anchor coordinates. In practice, this is not the case and techniques based on non-linear least squares (NLLS) via majorizing functions like SMACOF [30], [31] are typically used to solve the problem. An MDS algorithm essentially tries to find the representation $\mathbf{X}$ that minimizes the mismatch between the measured dissimilarity $\hat{d}_{i,j}$ and the distances in $\mathbf{X}$ as follows

$$\min_X \mathcal{S}(\mathbf{X}) = \min_X \sum_{i<j\leq N} \left(\hat{d}_{i,j} - d_{i,j}(\mathbf{X})\right)^2 \quad (3)$$

B. WEIGHTED MDS

The equation above requires the all-to-all pairwise distance estimates $\hat{d}_{i,j}$ between nodes, unfeasible when they are not within communication range. In other words, classical MDS cannot cope with missing data (i.e., links) and requires full connectivity among anchors. To overcome this, a weight $w_{i,j}$ can be associated to each $d_{i,j}$ measurement, leading to [15] and [18]

$$\min_X \mathcal{S}(\mathbf{X}) = \min_X \sum_{i<j\leq N} w_{i,j} \left(\hat{d}_{i,j} - d_{i,j}(\mathbf{X})\right)^2 \quad (4)$$

Missing links have zero weight and do not affect the cost function, while available and measured links have $w_{i,j} = 1$. This is the weighting function used in our experiments (§VI). Alternatively, if information on the noise on distance estimates is available, less accurate measurements can be down-weighted [15]. Similarly, if multiple measurements are taken per distance $d_{i,j}$, the weight could be set based, e.g., on their variance.

C. ACCOUNTING FOR KNOWN ANCHORS

The MDS algorithm determines the low-dimensional coordinates of $\mathbf{X}$ in a new relative coordinate system. The new coordinates must then be transformed to the desired coordinate system via a rigid body transformation including a rotation, translation, and possibly a reflection. To this end, at least 3 points (4 in 3D) must be measured in both coordinate systems. If we have m known positions ($\mathbf{X_a}$), we can directly include their coordinates in the MDS minimization problem as in [9], removing the need for these transformations and facilitating the accurate estimation of the unknown node coordinates.

D. LOCAL MINIMA

As many other NLLS algorithms, MDS suffers from local minima. Iterative MDS solvers require as input an initial estimate $\mathbf{X^0} = [\mathbf{X_u^0}, \mathbf{X_a}]^T$. For the algorithm to converge to the desired minimum, $\mathbf{X_u^0}$ must be *well-positioned*, especially if there are significant errors in the distance measurements $\hat{d}_{i,j}$ and/or their number is small, due to a low density of anchors. To deal with this issue, iterative solvers consider *multiple* random initial estimates $\mathbf{X^0}$, computing each corresponding solution and reporting as the final output the one with the lowest residual cost from (4). This does not guarantee convergence to the optimal solution but increases the likelihood to find it. Increasing the number of initial estimates also increases the computational cost, which is acceptable if the algorithm is run sparingly, e.g., only after deploying the nodes.

V. SYSTEM IMPLEMENTATION

We offer details of our full-fledged, immediately usable self-localization system (Fig. 1) combining in-field acquisition of ranging estimates with the techniques described above. The system, used in our experimental campaign (§VI), targets the Decawave DW1000 UWB transceiver [32] and is developed atop Contiki OS [33].

A. ACQUIRING AND REPORTING DISTANCE ESTIMATES

The in-band acquisition of distance estimates via UWB is performed via two-way ranging, which comes in several variants. SS-TWR is the simplest, with the lowest overhead but also lowest accuracy [11]. In PLANT, we used asymmetric DS-TWR [12], a classic scheme achieving higher accuracy at the cost of higher message overhead, and therefore latency and energy consumption. Instead, in our latest deployments in DEPARTMENT and RECEPTION we use SS-TWR with a clock drift compensation based on carrier frequency offset (CFO) estimation [26], a recent technique yielding accuracy akin to DS-TWR and the same overhead as SS-TWR. Concurrent ranging [14], also recently proposed, could be another viable option. However, its significant improvements in latency, update rate, and energy consumption, are largely irrelevant in common scenarios where anchors are mains-powered and their distance acquired only sporadically. Further, these benefits come at the cost of a slightly lower ranging accuracy, crucial in self-localization.

We exploit the out-of-band network, present in all of our testbeds, to schedule tests and collect the $\hat{d}_{i,j}$ distance estimates. First, we determine the neighborhood of each anchor via a connectivity assessment and discard links with a packet reception rate $PRR < 25\%$. Based on the resulting connectivity graph among anchors, we then instruct each one, in sequence, to range against a given set of neighbors. This simple solution schedules ranging exchanges, and in general UWB communication, to guarantee the absence of collisions.

The reliance on an out-of-band network is not an issue for most UWB applications, where it is already present. For instance, in time-difference of arrival (TDoA) schemes [34], [35] a wired or wireless network enables the collection of the time information acquired by anchors, necessary to compute the target position at a central server. In situations where this out-of-band infrastructure is impractical or impossible to deploy, the scheduling of ranging exchanges must be performed in-band, over the UWB network. In this case,

a multi-hop routing protocol, e.g., the RPL standard [36], can be used to support i)the downward traffic necessary to instruct each anchor to perform ranging, and ii) the upward traffic to collect the $\hat{d}_{i,j}$ values. We exploited this approach in the PLaNS project [7]; however, its detailed description and evaluation is outside the scope of this paper.

B. COMPUTING THE ANCHOR POSITIONS

The distance estimates collected at the server are used to populate the distance matrix $\hat{D}$ (§III). We mitigate the effect of noise by setting $\hat{d}_{i,j} = \hat{d}_{j,i}$ to the median of the distribution yielded by the multiple measurements available. Further, we compute the distance between *known* anchors as $\|\mathbf{x_i} - \mathbf{x_j}\|$ and set the corresponding elements in $\hat{D}$.

This distance matrix is then input to the algorithm in §IV, developed atop the Python implementation of MDS in `scikit-learn` [37]. We extend the original source code with the possibility to add i) weights to each distance estimate, coping with missing data due to limited network connectivity, and ii) known node positions in each SMACOF step. The algorithm runs on a personal computer, which receives the in-field measured distance estimates as input and computes offline the estimated anchor positions.

VI. FROM THEORY TO PRACTICE

We now turn our attention to the performance of self-localization in our real-world settings.

UWB Testbeds: We exploit the following large-scale, multi-hop deployments representative of many UWB use cases:

- PLANT: A 28-anchor deployment in a large industrial plant, covering a rectangular area of ∼3000 m^2. It is characterized by the presence of metallic objects and NLoS conditions, typical of industrial settings.
- DEPARTMENT: A 36-anchor deployment spanning an entire floor at the University of Trento. The anchors cover an area of 80 m × 40 m, but are deployed mostly along corridors, which are very narrow (2.7 m) and long (Fig. 10). This yields a very challenging geometry for localization, yet representative of many indoor applications.
- RECEPTION: A 19-anchor deployment covering a total of 720 m^2 in the reception floor of the University of Trento. The area is L-shaped (Fig. 11a), with two nearly-separated areas connected only by a few NLoS links.

Anchors are always placed on the ceiling of the target area.

Metrics: Our assessment revolves around the following error metrics, for which we report the mean μ and standard deviation σ, along with percentiles of absolute errors. The *ranging error* between anchors enables us to characterize the *input* to the self-localization algorithm. It is computed for each link as $\hat{d}_{i,j} - d_{i,j}$, where $\hat{d}_{i,j}$ is the distance estimate and $d_{i,j} = \|\mathbf{x_i} - \mathbf{x_j}\|$ is computed from ground-truth anchor coordinates. This metric informs us of the discrepancies one can expect in the real world, and therefore of the extent to which an inaccurate input affects the quality of the self-localization *output*. The latter is directly captured by the *anchor positioning error*, computed as the difference $\|\hat{\mathbf{x}}_\mathbf{i} - \mathbf{x_i}\|$ between the estimated position and the ground-truth one. Moreover, in §VI-C we report the *target positioning error*, quantifying the impact anchor positioning errors induced by self-localization bear on the accuracy of the overall localization system.

Outline: We first analyze via simulation, exploiting the geometric layout of our testbeds, the impact of key environmental, system, and configuration parameters on the resulting anchor positioning accuracy (§VI-A). This step is necessary to understand what are the tradeoffs at stake, and to inform the configuration and setup of the self-localization system when actually used in our three testbeds (§VI-B). The experimental campaigns i) offer a validation of our simulation results, and ii) concretely demonstrate the accuracy of self-localization one can expect in real-world, large-scale scenarios; both aspects are rarely found in the related literature, let apart in combination (§II). Along the same lines, we close the circle by ascertaining the impact of the inaccuracies in anchor position induced by self-localization on the overall accuracy of the localization system (§VI-C). This provides researchers and practitioners with a *concrete* understanding, again largely missing in the literature, of the benefits and limitations of self-localization, albeit inevitably biased by the techniques, systems, and environments we use.

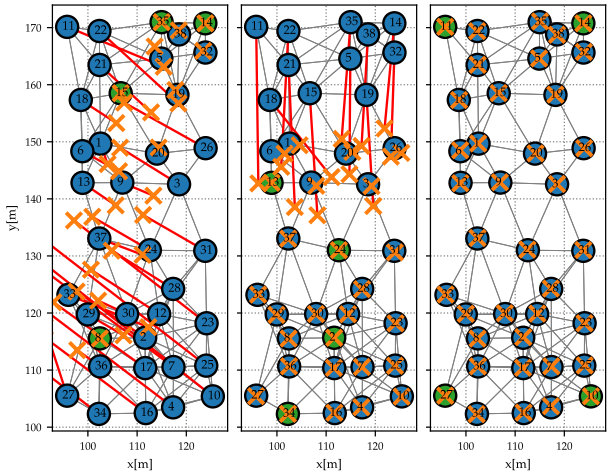
A. UNDERSTANDING THE PARAMETER SPACE

The performance of the self-localization algorithm depends on several parameters, of different nature. The choice of which and how many anchors should have a known, manually-determined position, and the density of all anchors throughout the target area (§VI-A2–§VI-A3) are determined in-field at deployment time. The amount of noise on ranging errors (§VI-A4) is instead primarily determined by the environment. Finally, accuracy directly descends from the initial estimate passed as input to the self-localization algorithm (§VI-A5). In this section, we exploit simulation to analyze the impact of each parameter separately and quickly, deriving the insights and guidelines we apply in our real-world evaluation in §VI-B.

1) EXPERIMENTAL SETUP

We report simulations based on the geometric characteristics of PLANT, to keep presentation concise. This testbed is a system operational in an industrial context, whose anchor placement is significantly more irregular than the other two. Our real-world results (§VI-B) are based on the 28 anchors available during the limited period of time when we were allowed to perform measurements without disrupting the facility operation. Here, we use instead the 38-anchor placement originally planned, as it enables us to explore in simulation the parameter space in a larger setup.

The simulations exploit the very same implementation of the self-localization algorithm we described in §V.



(a) Random #1. (b) Random #2. (c) Boundary.

FIGURE 2. Positioning performance with 4 known anchor positions (green circles) selected randomly (2a and 2b) and on the boundary of the area (2c). Grey lines denote available wireless links, blue circles unknown anchors, orange crosses estimated anchor positions, and red lines the distance of the latter from ground-truth anchor positions.

The difference is in the input provided to it, and specifically the distance matrix $\hat{D}$, which is synthetically generated. To determine the neighborhood of each anchor, we assume a UWB communication range of 15 m, which achieves a connectivity similar to what observed in the real-world ranging traces in PLANT (Fig. 8). We then populate $\hat{D}$ with the computed distance between anchors, and account for noise by adding a ranging error from a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 15$ cm. This simple model controls noise with a single parameter, simplifying both the interpretation of simulation results and their computation. However, it does not account for the distance between anchors, their antenna orientations, or the presence of NLoS; these aspects are nonetheless investigated in our real-world testbed experiments (§VI-B–§VI-C). Finally, as the performance of MDS is affected by local minima (§IV), we execute the algorithm with 128 different random initialization coordinates $\mathbf{X_u^0}$.

We use all these defaults unless otherwise specified, when we change them to analyze specific configuration or environmental conditions.

2) IMPACT OF THE POSITION OF KNOWN ANCHORS

The performance of the anchor self-localization algorithm depends on which anchors have a known position (§IV). We analyze the impact of their number and placement, useful to the personnel in charge of deployment. There are two main attributes to consider for anchor selection: i) anchor position and ii) connectivity. We start our evaluation by looking at the impact of the former on the estimation of $\mathbf{X_u}$.

Fig. 2 compares the estimated and true anchor coordinates in PLANT, considering 4 known positions. In Fig. 2a–2b the selection is completely random; this is somewhat unrealistic but allows us to analyze the algorithm behavior and elicit how poorly-placed known anchors prevent correct estimation of the unknown positions.

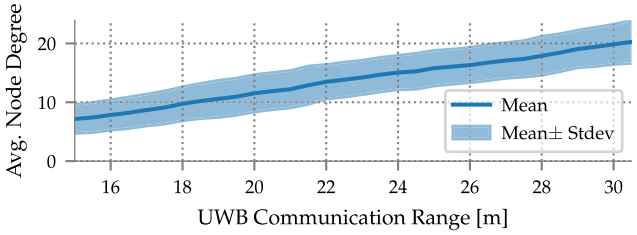


FIGURE 3. Average node degree (i.e., number of links per node) and standard deviation as a function of the communication range in PLANT.

In Fig. 2a, the bottom half of the area contains only one known position (anchor 8); the algorithm is unable to distinguish whether its neighbors should be located on its right or left, leading to very large errors. A similar behavior occurs in the top half despite the presence of three known anchors (14, 15, 35), due to their placement and, more generally, sparse connectivity. For instance, some unknown anchors (11, 18, 21, 22) are to the left of *all* three known ones above; therefore, MDS can comply with the input distance matrix $\hat{D}$ by placing these unknowns correctly to the left but also incorrectly below the known anchors, yielding the large errors. A denser connectivity would yield more elements in $\hat{D}$, further constraining the MDS algorithm, and ultimately improving estimates, as analyzed later (Fig. 4).

In Fig. 2b, all randomly-selected known anchors are in the bottom half, yielding accurate estimates for the anchors in this area. In contrast, the estimates in the top half exhibit large errors, despite the fact that they *satisfy* the distance measurements $\hat{d}_{i,j}$ available. Indeed, recall that missing distance estimates are assigned a zero weight in the cost function. As a result, the algorithm may position these anchors in areas whose connectivity, while satisfying the input constraints, would be quite different from the real one.

In contrast, in Fig. 2c the 4 known anchor positions are selected on the boundary of the deployment. In this case, the algorithm accurately estimates the position of all unknown anchors with a mean error $\mu = 10$ cm, standard deviation $\sigma = 4$ cm, and a maximum error of only 20 cm. Setting the known positions at the boundary of the deployment encloses the unknown anchors to a confined area, making it easier for the algorithm to correctly determine the unknown positions and limiting the likelihood to suffer from large errors as observed in Fig. 2a and 2b. These anchors are also typically the easiest to visually identify on-site, therefore helping the personnel in charge of measuring the actual anchor positions, e.g., using a map and a laser meter.

3) IMPACT OF ANCHOR DENSITY AND NUMBER OF KNOWN POSITIONS

The self-localization algorithm (§IV) is sensitive to the overall *density of anchors*, both unknown or known. Moreover, concerning the latter, it is sensitive not only to the position (§VI-A2), but also to the *number of known anchors*.

To investigate the former issue, we change the communication range from the default of 15 m to 30.5 m, in steps

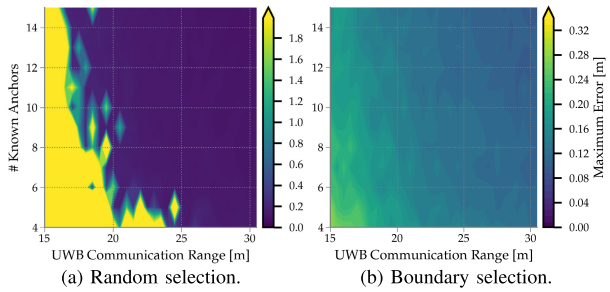


FIGURE 4. Maximum positioning error as a function of the communication range and the number of known anchors across 15 iterations per configuration with a random (4a) and boundary (4b) selection of known anchors. Note the different scale of the right colorbar.

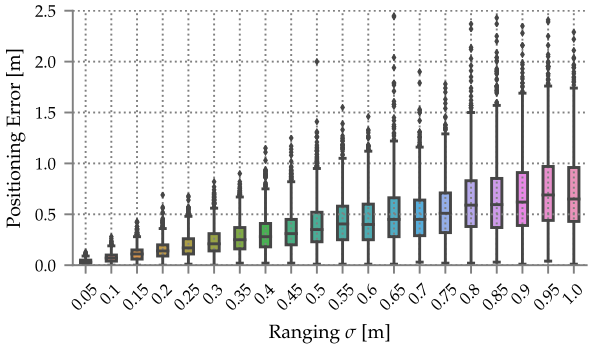


FIGURE 5. Anchor positioning error boxplots as a function of the ranging standard deviation σ. Small dots represent samples outside the 1.5× inter-quartile range of the distribution.

of 0.5 m. This changes the density of anchors and therefore the number of their neighbors (Fig. 3). Increasing density increases the information available to self-localization, improving the estimation of the unknown anchor coordinates.

To investigate the latter issue, we change the number of known anchors from 4 to 15, in steps of 1, for each communication range tested. Increasing the number of known positions decreases the unknowns to determine, simplifying the localization problem and increasing accuracy.

We consider both i) a random anchor selection to further illustrate the impact of a poor anchor choice, and ii) a selection with all known anchors on the boundary. We run each configuration 15 times, changing the ranging errors introduced in the distance matrix $\hat{D}$.

Fig. 4 shows two heatmaps with the *maximum* positioning error obtained per configuration considering the random (4a) and boundary (4b) anchor selection cases. Using a random selection, self-localization requires at least 6 known anchor positions (16% of all anchors in Fig. 2) and a high density of >11 neighbors (Fig. 3) to reliably and accurately determine the unknown anchor positions. Interestingly, with an average density of ≈8 neighbors, the algorithm cannot determine all unknown positions accurately even with 15 known positions (40%), reasserting the importance of sufficient connectivity—at least with a random selection. Indeed, selecting the known anchors on the boundary of the deployment area improves performance dramatically, yielding a maximum error ≤45 cm in all configurations tested (Fig. 4b). Further, the number of known anchors bears only a marginal impact, suggesting that the dominating factor is their selection on the boundary and their connectivity.

4) IMPACT OF RANGING ERRORS

To examine how ranging errors affect the anchor positioning accuracy, we change the default ranging error distribution $\mathcal{N}(0, \sigma^2)$ in §VI-A1 by increasing σ from 5 cm to 1 m in steps of 5 cm. For each value of σ, we run 30 iterations producing different Gaussian distance errors. We select the same 4 known anchor positions as in Fig. 2c.

Fig. 5 shows that, as expected, the anchor positioning error increases with the ranging error. When NLoS or otherwise noisy links are predominant, the anchor positioning accuracy decreases significantly and rapidly. However, in the common case where most links are LoS and relatively few are NLoS, we can generally expect a ranging precision with $\sigma \leq 25$ cm, resulting in a positioning error ≤50 cm. This is usually acceptable and bears limited impact on the overall localization accuracy, as we demonstrate in §VI-C, while our automated self-localization toolchain dramatically decreases the deployment setup time and effort.

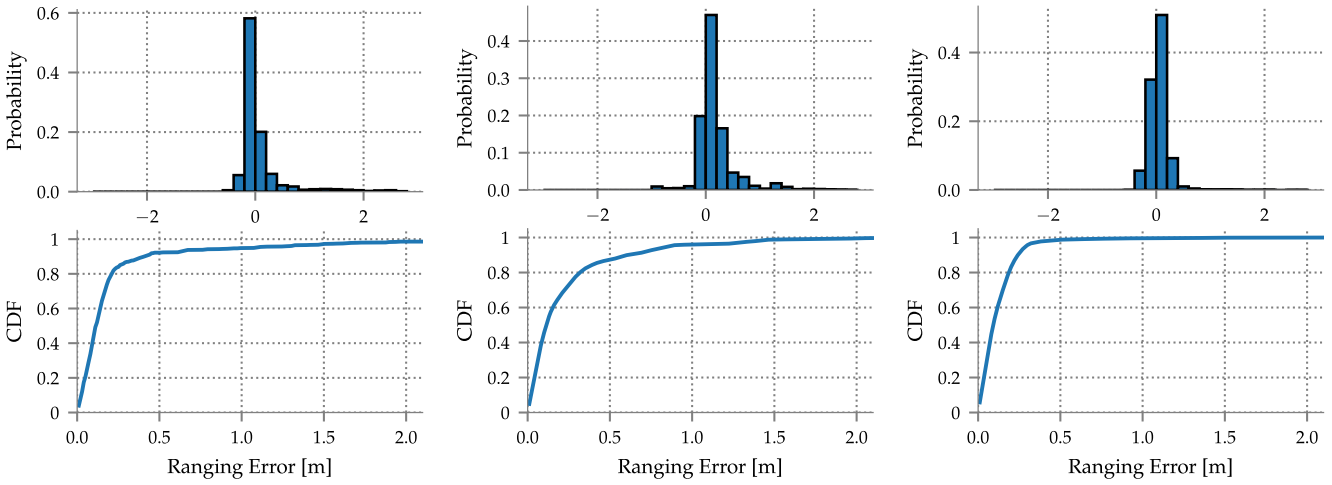
5) IMPACT OF INITIAL ESTIMATES

As mentioned in §IV, MDS suffers from local minima; the initial position $\mathbf{X^0}$ is crucial in ensuring convergence. Therefore, we cannot rely on a *single* random $\mathbf{X^0}$, rather we must compute the solution with several and select as the final output the one that provides the lowest cost. This, however, increases the computational overhead of the approach.

We analyze these issues by changing the number of initial positions $\#\mathbf{X^0}$ from 1 to 256 with a power-of-two step. We execute 15 iterations per configuration, and report the mean, standard deviation, and maximum values of execution time and positioning error, itself aggregated across all estimated unknowns. We use default settings (§VI-A1) and the same 4 known anchor positions of Fig. 2c.

Table 1 shows the results as a function of the number of initial estimates. As expected, the execution time increases with $\#\mathbf{X^0}$, while the positioning error decreases. Nevertheless, these results allow us to identify the minimum number of iterations required to obtain a reasonable error, which is key to verify the practical applicability of the approach, and a precious indication at deployment time. Under the conditions evaluated, to obtain a mean error <50 cm we need $\#\mathbf{X^0} \geq 16$ initial estimates. Nevertheless, the maximum positioning error can be much higher; to limit it to <1 m, we need at least 64 iterations. Interestingly, the corresponding execution time is only a handful of seconds; even with 256 iterations, the worst-case execution time is 20.22 s on the rather old laptop model (MacBook Pro 2013) we used.

These values confirm the practical applicability of the technique, as embodied in our system. The computational overhead is entirely acceptable in absolute terms, especially considering that the self-localization computation is likely to be run only during the anchor deployment.



(a) PLANT. (b) DEPARTMENT. (c) RECEPTION.

FIGURE 6. Normalized histogram and CDF of the ranging error across all measurements in the three studied deployments.

TABLE 1. Execution time and positioning error as a function of the number of initial $\mathbf{X^0}$ values considered in the algorithm using a ranging error distribution with $\sigma = 15$ cm, 15 iterations with different random errors, and the 4 known anchor positions in Fig. 2c.

	Positioning Error [m]			Execution Time [s]			
$\#\mathbf{X^0}$	mean	stdev	max	mean	stdev	min	max
1	3.46	4.48	24.57	1.34	0.22	1.16	1.95
2	3.24	4.51	21.76	1.39	0.27	1.23	2.19
4	1.66	3.08	20.68	1.68	0.26	1.45	2.29
8	1.10	2.58	20.32	1.90	0.33	1.63	2.82
16	0.29	1.34	16.57	2.25	0.18	1.97	2.57
32	0.14	0.66	14.78	3.44	0.60	2.75	4.86
64	0.10	0.06	0.35	5.30	1.10	4.22	8.41
128	0.10	0.06	0.35	8.59	1.20	7.16	10.76
256	0.10	0.06	0.36	15.26	1.96	12.88	20.22

B. REAL-WORLD ANCHOR SELF-LOCALIZATION

We now quantify the performance of self-localization in our three real-world, large-scale, multi-hop testbeds. We begin by characterizing the quality of ranging estimates, inevitably affected by errors induced by the environment (§VI-B2). As these estimates are collected by our system and provided as input to the MDS algorithm (§IV), they crucially affect the accuracy of the anchor positions returned as output, analyzed next (§VI-B3).

1) EXPERIMENTAL SETUP

In all testbeds, each anchor is composed of a UWB node paired with an embedded PC; the latter enables communication access to a central server via Ethernet. In PLANT, an anchor consists of a custom Linux computer interfaced with a UWB platform designed by the company owning the deployment, based on the Decawave DW1000 UWB transceiver [32]. Anchors use channel 2 with center frequency $f_c = 3993.6$ MHz and 499.2 MHz bandwidth.

In DEPARTMENT and RECEPTION, anchors consists of an EVB1000 board [38] attached to a Raspberry Pi v3 (RPi) and a J-Link programmer. The UWB radios use channel 7 with $f_c = 6489.6$ MHz and 900 MHz bandwidth.

All experiments use the highest 6.8 Mbps data rate.

2) ANALYSIS OF RANGING ERRORS

To appreciate the quality of the output of self-localization, an analysis of the quality of the input is necessary. We report the ranging errors present in our real-world testbeds, and investigate whether their presence can be inferred from the PHY-level indicators available on the UWB radio.

a: ACCURACY AND PRECISION

Fig. 6 shows the normalized histogram and cumulative distribution function (CDF) of the ranging error distributions across more than 1.2k, 163k, and 111k estimates covering all available links ($PRR \geq 25\%$, §V) in PLANT, DEPARTMENT, and RECEPTION, respectively.

The best ranging performance is obtained in RECEPTION with $\mu = 4$ cm and $\sigma = 20$ cm and percentiles of the absolute error of $95^{th} = 29$ cm and $99^{th} = 61$ cm. This testbed spans two open areas where most anchors enjoy LoS conditions, except for the few connecting the two areas. At the other extreme, in DEPARTMENT anchors are placed in long and narrow corridors, with corners and ceiling half-walls causing several NLoS links, responsible for a long-tailed error distribution where $99^{th} = 1.94$ m and the error is ≥ 87 cm in 5% of the estimates. As a consequence, the overall accuracy ($\mu = 20$ cm) and precision ($\sigma = 57$ cm) are also significantly worse. PLANT shows a similar long tail of errors ≥ 1 m in 5% of the estimates, this time caused by the complexity of the industrial environment, despite obtaining slightly better performance with $\mu = 8$ cm and $\sigma = 44$ cm.

b: CAN WE IDENTIFY ERRORS?

These long-tailed error distributions are a challenge for the self-localization algorithm, ultimately preventing accurate positioning of *all* anchors (§VI-B3). Therefore, based on our extensive datasets, we investigate whether a correlation

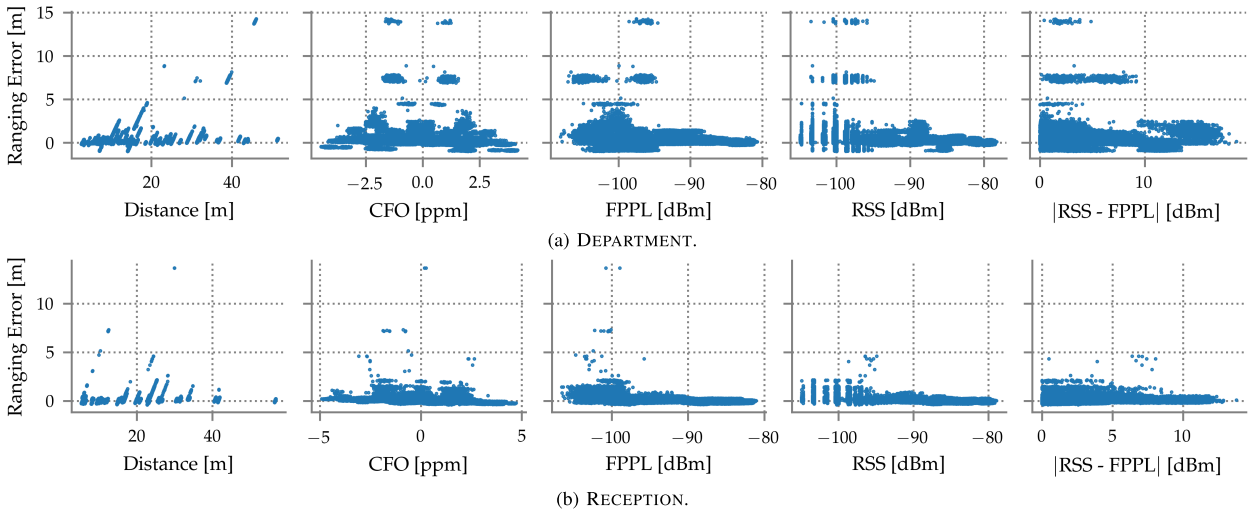


FIGURE 7. Ranging error vs. distance, CFO, FPPL, RSS, and |RSS − FPPL| across more than 163k and 111k ranging estimates in Department and Reception, respectively. Errors do not exhibit a clear correlation with these measured radio features.

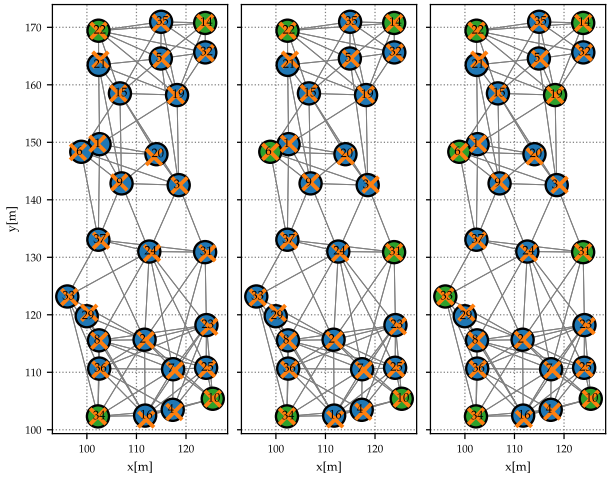


FIGURE 8. Positioning performance with 4, 6, and 8 known anchor positions (green circles) in the boundary of Plant. Grey lines denote available wireless links, blue circles unknown anchors, and orange crosses estimated anchor positions.

exists between these errors and PHY-level indicators available on the radio. A positive answer would enable us to reliably discard faulty estimates, increasing the self-localization accuracy.

We consider the relation between ranging errors and i) the ground-truth distance, ii) the carrier frequency offset (CFO) between the ranging initiator and responder, iii) the first path power level (FPPL), iv) the received signal strength (RSS), and v) their difference |RSS − FPPL|. Concerning the latter indicator, according to Decawave [39] links with |RSS − FPPL| > 10 dBm are generally in NLoS, while links with |RSS − FPPL| < 6 dBm are in LOS.

The scatter plots in Fig. 7 report results for Department and Reception, our largest datasets. We observe no direct relation between the ranging errors and the ground-truth distance or the CFO. Instead, large errors often have lower RSS and FPPL values; however, many other links with comparably low values yield very accurate ranging estimates. Finally, the NLoS indicator suggested by Decawave is similarly unreliable, given that the largest errors in our dataset have |RSS − FPPL| < 10 dBm.

We conclude that these simple radio features cannot be used to identify errors, at least in our testbeds, and therefore are not included as part of the ranging computation. More sophisticated techniques are the subject of active research, as briefly discussed later (§VII), and could in principle be used to improve the ranging accuracy; however, they are outside the scope of this work.

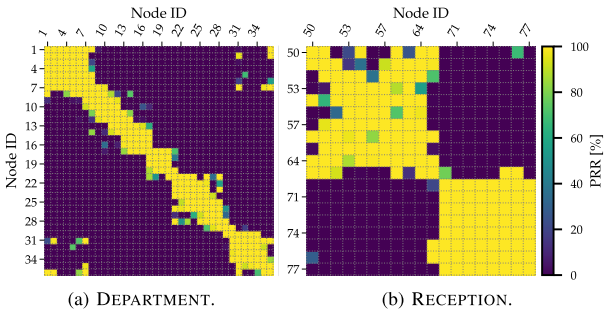
Instead, we investigate next whether the self-localization algorithm is robust enough to yield accurate anchor positions even in the presence of these long-tailed error distributions.

3) PERFORMANCE OF ANCHOR POSITIONING

We now investigate the performance of the self-localization algorithm (§IV) in determining the positions of the unknown anchors based on the ranging traces previously analyzed.

We first consider Plant, as this was the focus of our simulation-based analysis (§VI-A). Fig. 8 shows the estimated vs. ground-truth anchor coordinates with 4, 6, and 8 known anchor positions on the boundary, respectively 14%, 21%, and 28% of the total. Overall, the performance is quite good despite the long-tailed errors in Fig. 6a; the maximum error is ≈1 m in all cases, even if 5% of the ranging estimates are affected by errors ≥ 1 m, confirming the robustness of the system. With only 4 known anchors, the positioning error has $\mu = 44$ cm, $\sigma = 23$ cm, and $90^{th} = 72$ cm. Results improve with 6 anchors ($\mu = 33$ cm, $\sigma = 24$ cm, $90^{th} = 68$ cm) and 8 anchors ($\mu = 32$ cm, $\sigma = 25$ cm, $90^{th} = 63$ cm) albeit only marginally, as suggested by the heatmap in Fig. 4b.

We now switch our attention to the results in Department and Reception, where we have a larger ranging dataset and more freedom in performing additional tests, due to logistics. Specifically, and before delving into the positioning performance, we investigate in more depth the connectivity in these



(a) DEPARTMENT. (b) RECEPTION.

FIGURE 9. Network connectivity heatmap in DEPARTMENT and RECEPTION. In the former, nodes are deployed in long, narrow corridors where each node generally has 3 to 12 good neighbors. In RECEPTION, we have two clearly distinguished open areas, poorly connected between them.

testbeds because, as seen in §VI-A3, it is key for the correct operation of the algorithm.

Fig. 9 depicts connectivity using a heatmap whose colors represent *PRR* values. In DEPARTMENT, anchors are deployed across 4 long and narrow corridors, severely constraining connectivity. Anchors generally have at most 3 to 6 good neighbors, except for anchors 1–7 that are deployed in a small open area and enjoy higher connectivity. Instead, in RECEPTION there are two large open areas; anchors in the same area are highly connected. On the other hand, anchors across the two areas are poorly connected via few NLoS links, as the key anchors (50 and 65) have an obstacle in front that severely impacts the *PRR* with nodes 70–77.

Fig. 10 and 11a compare the estimated vs. true anchor positions in DEPARTMENT and RECEPTION, respectively. In both cases, we select only 4 known anchor positions (11% and 21%, respectively) on the boundary of the target area.

In DEPARTMENT, the placement of anchors along corridors severely constrains the spatial resolution along one of the coordinate axis. Moreover, there are several NLoS links (e.g., anchor 8 with 1–6 or anchor 21 with 22–28), hampering correct estimation of the anchor positions. Nonetheless, we observe that the main dimension in the corridor is estimated accurately, e.g., the x-axis coordinates of anchors 8–16 and 22–29. The other dimension, affected by poor spatial resolution, could be improved in a post-processing step or with manual validation. On the other hand, the position of the unknowns in the small open area comprising anchors 1–7 is estimated with an error $\leq$71 cm, despite we provided the algorithm with only one known anchor in the area.

In RECEPTION, performance is significantly better, arguably due to higher connectivity between anchors and more reasonable geometry. Across the unknown anchors, the positioning error has $\mu = 24$ cm, $\sigma = 18$ cm with a minimum error of 3 cm and a maximum of only 58 cm, despite the large size of the target area. The largest errors are for anchors 51, 62, 64 (slightly above the true positions) and 65 (slightly to the right), probably due to the large ranging error of 1.22 m between anchors 65 and 70 and the 30 cm overestimation between 65 and the known anchor 50, due to obstacles and walls yielding NLoS.

C. SELF-LOCALIZATION ERRORS: DO THEY MATTER?

To ascertain whether the self-localization techniques and system evaluated here are ultimately of practical interest for real-world applications, we assess the impact of the above inaccuracies in estimating anchor positions on the accuracy of the localization system exploiting them. To this end, we compare the accuracy of tracking a mobile target in RECEPTION using the true, manually-measured anchor coordinates vs. using those estimated by our self-localization approach (§VI-B3). We focus on RECEPTION because its L-shaped geometry offers an interesting mix of deployment conditions. The bottom area (Fig. 11, $y < 16$ m) is long, narrow, and with an irregular anchor placement, overall yielding a more challenging setup w.r.t. the top area, wider and with anchors regularly and more densely placed. These two areas, hereafter NARROW and REGULAR, are representative of corresponding real-world situations; we therefore analyze their separate impact on performance in addition to the aggregate one over RECEPTION.

Nevertheless, acquiring ground-truth trajectories of mobile targets over the wide area covered by RECEPTION is very challenging. Therefore, we resort to the following methodology. We first generate the trajectories in simulation, which enables us to easily define ground truth to compare against and ascertain the *absolute* impact of self-localization on localization accuracy. Moreover, we determine the *distance* between the target positions determined with true vs. estimated anchor coordinates, as a measure of the *relative* impact induced by the latter. This is actually a very important metric because, in real applications, ground truth is not known a priori and therefore not available for comparison. In other words, we answer the question: *What would be the difference in target positioning by using true vs. estimated anchor positions?* If the difference is small enough, choosing one over the other should not really matter in practice. Interestingly, this latter metric does not require ground truth, and we can therefore measure it with relative ease based on real trajectories in our testbed.

1) SIMULATED TRAJECTORY

We define the ground-truth trajectory as a round-trip along the L-shaped area, spanning both NARROW and REGULAR. We simulate 3200 positions; however, due to the geometry of RECEPTION and the simulated trajectory, only 661 ($\sim$20%) are in REGULAR. For each ground-truth position, we select the 4 closest anchors and simulate 4 ranging estimates with zero-mean Gaussian error $\mathcal{N}(0, \sigma^2)$ and the default $\sigma = 15$ cm (§VI-A1). Then, we use an NLLS solver to compute the target positions $\hat{\mathbf{P}}_\mathbf{T}$, based on the true anchor coordinates $\mathbf{X}$, and $\hat{\mathbf{P}}_\mathbf{E}$, based on the estimated ones $\hat{\mathbf{X}}$.

Fig. 11a visually compares both trajectories against the ground-truth one. The latter is actually not visible, as it is accurately captured by both $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$; both these trajectories are actually very close to each other. The largest differences are found near anchors 62–64, whose estimated positions are slightly above the true ones, and therefore alters the trajectory $\hat{\mathbf{P}}_\mathbf{E}$ accordingly.

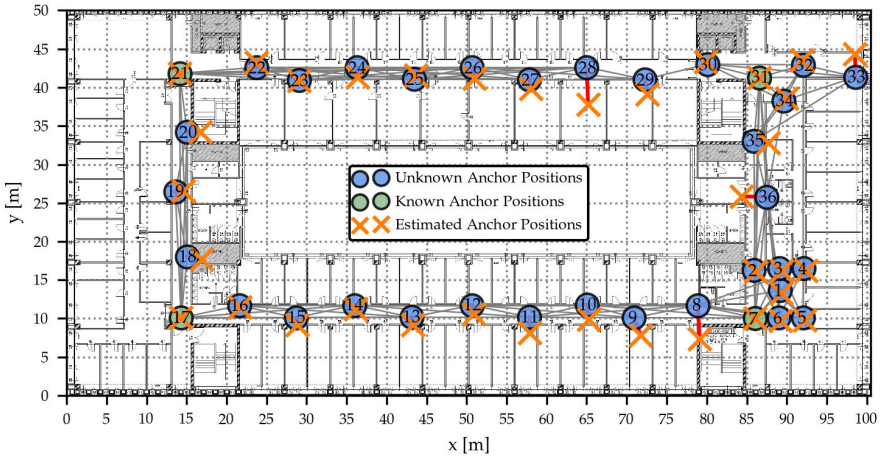
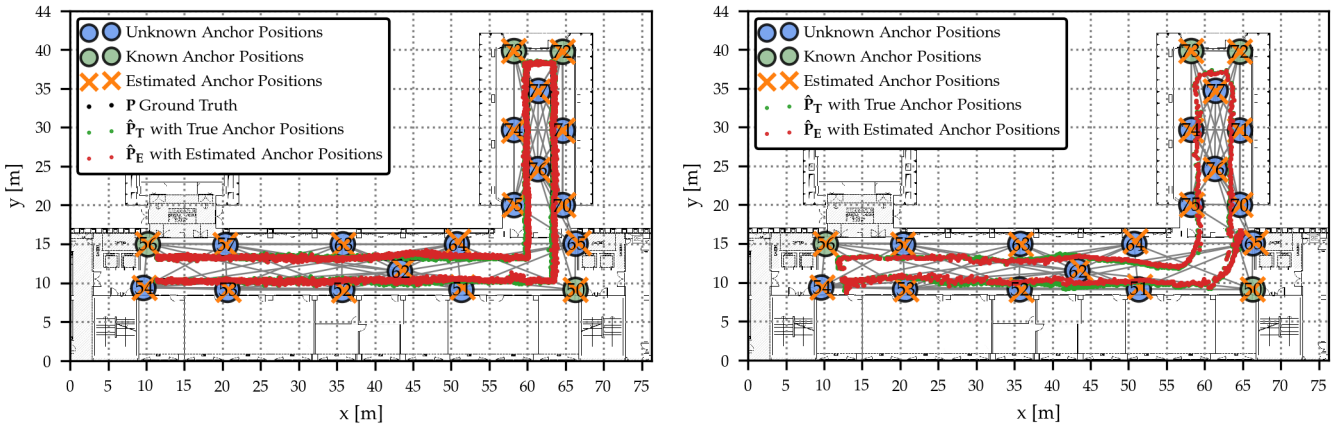


FIGURE 10. Anchor self-localization in DEPARTMENT. Grey lines denote available wireless links for ranging.



(a) Simulated trajectory. (b) Real trajectory.

FIGURE 11. Ascertaining the impact of self-localization errors on the overall localization accuracy of target position tracking in RECEPTION.

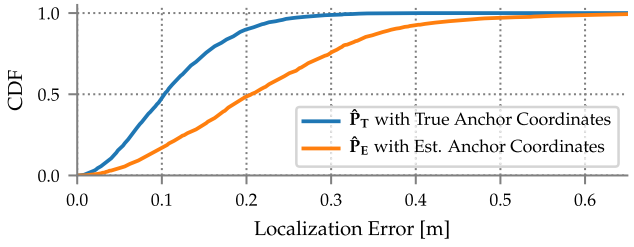


FIGURE 12. CDF of the target positioning error w.r.t. ground truth in Fig. 11.

The CDF of the positioning error (Fig. 12) offers an alternate, quantitative view. Using true anchor positions achieves a mean error $\mu = 17$ cm with $\sigma = 10$ cm, yielding $50^{th} = 16$ cm and $95^{th} = 35$ cm. As expected, the error increases using estimated anchor positions. However, it remains $\leq$51 cm for 95% of the 3200 location samples, with $50^{th} = 25$ cm, $\mu = 26$ cm, and $\sigma = 14$ cm.

Instead, Fig. 13 analyzes the distance $||\hat{\mathbf{P}}_\mathbf{E} - \hat{\mathbf{P}}_\mathbf{T}||$ between the target position in $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$, highlighting the impact of geometry and anchor placement. In REGULAR, anchors 70–77 provide very good coverage, yielding a small average distance (8 cm) between location samples and very small variations overall ($99^{th} \leq 15$ cm). In NARROW, the irregular placement of anchors 50–65 in the longer, narrower area yields higher average distance (23 cm) and variations ($99^{th} \leq$ 50 cm). The CDF aggregated over both areas is dominated by NARROW, due to the larger number of samples. Next, we investigate to what extent these results hold in practice.

2) REAL TRAJECTORY

We run experiments where a person carrying a DW1000-equipped platform walks in RECEPTION along a path similar to the simulated one (Fig. 11a). The target position is acquired by TALLA, a system enabling TDoA localization across large, multi-hop anchor deployments. The difference Δt_i in the RX timestamps of packets, sent by the target, at a given anchor i w.r.t. a reference one represents the equation of a hyperbola. The actual position is determined by solving a NLLS problem minimizing the squared difference between the measured and theoretical Δt_i estimates. Details about the TDoA solver and the overall system design are in [34]. We obtain $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$ by running the TALLA solver with true and estimated anchor positions on the same experiment traces.

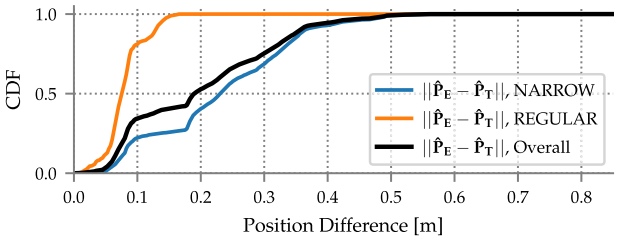


FIGURE 13. Simulation: CDF of the distance between the estimated positions $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$ in Fig. 11 using $\mathbf{X}$ and $\hat{\mathbf{X}}$, respectively, in RECEPTION.

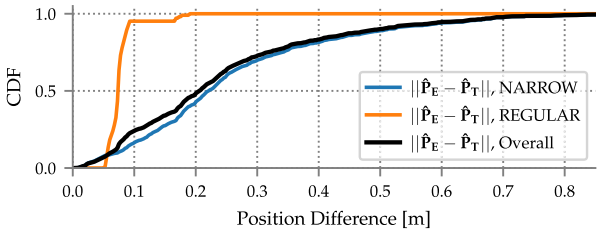


FIGURE 14. TALLA: CDF of the distance between the estimated positions $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$ of a mobile node using $\mathbf{X}$ and $\hat{\mathbf{X}}$, respectively, in RECEPTION.

Fig. 11b visually compares the resulting trajectories. As mentioned, we do not have accurate ground truth for the mobile target and therefore analyze only the distance between the target positions in $\hat{\mathbf{P}}_\mathbf{T}$ and $\hat{\mathbf{P}}_\mathbf{E}$, whose CDF is shown in Fig. 14. Similar to our simulation results, $\hat{\mathbf{P}}_\mathbf{E}$ follows $\hat{\mathbf{P}}_\mathbf{T}$ closely and with a slight difference in the two areas. In REGULAR, the average of the distance $||\hat{\mathbf{P}}_\mathbf{E} - \hat{\mathbf{P}}_\mathbf{T}||$ (8 cm) and its variability ($99^{th} \leq 18$ cm) are nearly the same as in simulation. In NARROW, the average distance (26 cm) and its variability ($99^{th} \leq 84$ cm) are slightly higher, arguably due to the impact of real-world conditions exacerbated by geometry and the high sensitivity to measurement errors of TDoA localization. As in the simulated case, the CDF aggregated over the whole RECEPTION is dominated by the more challenging NARROW area; REGULAR accounts for only 104 (∼10%) of the 1079 total samples.

Overall, our results with real trajectories (Fig. 14) are in good agreement with those in simulation (Fig. 13). This points to the fact that, were target mobility ground truth available in the former, one could expect very good accuracy, similar to Fig. 12.

VII. DISCUSSION

We summarize our findings, reflect on their immediate applicability, and concisely highlight lessons learned and limitations that may orient future work on the topic.

A. MAIN FINDINGS AND APPLICABILITY

Our *quantitative* analysis in multiple real-world, large-scale, multi-hop deployments shows that the self-localization approach we considered and embodied in our system introduces only a relatively small anchor positioning error (§VI-B). Moreover, our analysis of the impact of this error on localization (§VI-C) shows that the difference between target positions acquired with true and estimated anchor positions is similarly small. In both cases, the error magnitude depends on the geometry of the target area, the connectivity among anchors, and the environmental conditions affecting the accuracy of the ranging estimates among anchors used as input. For instance, Fig. 14 shows that, even in the same RECEPTION testbed, different areas (top vs. bottom) introduce a different bias on the resulting trajectory (8 vs. 26 cm on average).

At first sight, these values may seem high, relative to the decimeter-level ranging accuracy typically associated with UWB. However, two observations hold. First, we showed that, in practice, the latter accuracy can be worse depending on the environment (§VI-A4). Second, even with the extra bias induced by self-localization, the overall, absolute error remains in line with the requirements of many applications exploiting localization [1], [2], [3], [4], for which sub-meter accuracy is enough, in contrast with the coarse, meter-level one offered by other RF-based localization technologies [4].

For these applications, the findings in this paper reassert the role of self-localization as a practical tool to facilitate the deployment of UWB-based localization systems by dramatically reducing the time and effort to setup the anchor infrastructure, specifically focusing on the large-scale, multi-hop settings that require an automated approach.

On the other hand, the errors above, however small, may be unacceptable in applications that demand the highest accuracy possible enabled by UWB, e.g., drone navigation [40] or detection of fine-grained human mobility patterns [41], for which manual anchor localization may remain the only viable option.

B. SELF-LOCALIZATION IN 3D

In this study, we considered only 2D localization, common to many applications including those typically run in our testbeds, whose anchors are consequently all placed on the ceiling at the same or very similar height. In contrast, 3D localization requires diversity in the positioning of anchors along the z-axis.

If a proper anchor configuration is deployed, the algorithm in §IV can directly output the 3D coordinates of the unknown positions based on the distance matrix $\hat{D}$. Still, its ability to accurately estimate the height of the unknown anchors depends on the z-axis resolution, generally poor due to the low spread of anchors along that axis compared to the (x, y) plane. Nonetheless, we observe that the true height of each anchor can be easily, reliably, and independently measured, e.g., with a LDM. These true heights can also be incorporated in the MDS algorithm as in [9].

C. ROLE OF KNOWN ANCHORS

The selection of anchors whose positions $\mathbf{X_a}$ are known is key for accurate self-localization (§VI-A); a poor selection may result in large positioning errors (Fig. 2). Known anchors should be at the boundary of the target area, enclosing unknown positions. In our testbeds, a minimum of 4 known anchors at the boundary are required for acceptable accuracy; a higher number, which increases manual effort, yields

only marginal improvements. However, deployments with different geometries and/or across even larger areas may benefit from more known anchor positions. Moreover, known anchors should also be well-connected (i.e., with several neighbors), providing enough distance $\hat{d}_{i,j}$ estimates as input to the algorithm.

Finally, their selection should privilege anchors that enjoy LoS connectivity. For instance, our initial experiments in RECEPTION relied on anchor 65 as a known one, chosen purely from geometrical considerations on the map (Fig. 11), and achieved very poor results. Visual inspection evidenced that this anchor is placed on an area of the ceiling hampering LoS with all neighbors; selecting it as a known anchor forces the algorithm to treat inaccurate NLoS distance information as reliable, with detriment to accuracy.

D. NLoS DETECTION AND ERROR MITIGATION

More generally, we have shown in §VI-A4 and §VI-B3 that the performance of self-localization strongly depends on the ranging error. In the TWR variants we used, this is caused mainly by an incorrect time-of-arrival (ToA) estimation, which adds a significant offset to ranging estimates in NLoS conditions. We have shown that these cannot be detected based on simple indicators of the UWB transceiver. However, the latter also provides sophisticated RX information in the form of the measured channel impulse response (CIR). Existing works [27], [28] exploit CIR information combined with machine learning and deep learning techniques to detect whether a link is in NLoS and even estimate and mitigate the resulting errors. Enhancing our system with NLoS detection and mitigation is outside the scope of this paper, although part of our future work. On the other hand, in-field inspection of the target area in many cases can help identify NLoS links between anchors and down-weight or discard the corresponding measurements.

VIII. CONCLUSION

The widespread adoption of UWB-based localization systems hinges on the ease of deploying them quickly and reliably. This clashes with the current practice of carefully positioning localization anchors via manual measurements, a labor-intensive and error-prone process whose drawbacks are exacerbated when many anchors must be deployed across large and complex areas.

This paper concretely and quantitatively shows that an alternate, *automated* approach in which the system self-localizes anchors is *practically* applicable to these large-scale, multi-hop scenarios. To this end, we i) integrate state-of-the-art techniques hitherto evaluated only via simulation or small-scale setups into a full-fledged system that covers the entire gamut from in-field, in-band acquisition of distance information via UWB to the output of estimated anchor positions, and ii) evaluate the resulting system in three large, multi-hop real-world testbeds with different characteristics, eliciting quantitatively the impact of the information input to self-localization (e.g., ranging errors and anchor connectivity) on the accuracy of the output anchor position estimates.

Our results show that manual estimation of only a handful of anchor positions is sufficient to estimate large numbers of unknown ones quickly and accurately, minimizing manual labor without significant detriment to the accuracy of the localization system relying on them. We argue that the latter accuracy is in line with the requirements of many UWB-based localization systems, for which the significant reduction in human labor is a crucial factor currently hampering their adoption in large-scale settings.

Finally, to enable researchers and practitioners to immediately build upon, improve, or replicate our results, we publicly release our system implementation as open source, along with the datasets of the experimental campaigns we described.

REFERENCES

[1] UWB Alliance. *Applications*. Accessed: Feb. 27, 2023. [Online]. Available: https://uwballiance.org/applications/

[2] FiRa Consortium. *UWB Use Cases*. Accessed: Feb. 27, 2023. [Online]. Available: https://www.firaconsortium.org/discover/use-cases

[3] I. V. Bourne, "The rise of indoor positioning: A 2016 global research report on the indoor positioning market," IndoorAtlas, Oulu, Finland, Tech. Rep., 2016. [Online]. Available: https://www.indooratlas.com/wp-content/uploads/2016/09/A-2016-Global-Research-Report-On-The-Indoor-Positioning-Market.pdf

[4] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2017.

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

[6] *Leica 3D Disto*. Accessed: Feb. 27, 2023. [Online]. Available: https://shop.leica-geosystems.com/learn/laser-distance-measuring/3d-disto

[7] E. Varriale, P. Corbalán, T. Istomin, and G. P. Picco, "PLaNS: An autonomous local navigation system," in *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, Oct. 2018, pp. 1722–1727.

[8] M. Ridolfi, J. Fontaine, B. V. Herbruggen, W. Joseph, J. Hoebeke, and E. D. Poorter, "UWB anchor nodes self-calibration in NLOS conditions: A machine learning and adaptive PHY error correction approach," *Wireless Netw.*, vol. 27, no. 4, pp. 3007–3023, May 2021.

[9] C. Di Franco, M. Marinoni, E. Bini, and G. C. Buttazzo, "Dynamic multidimensional scaling with anchors and height constraints for indoor localization of mobile nodes," *Robot. Auto. Syst.*, vol. 108, pp. 28–37, Oct. 2018.

[10] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Standard IEEE 802.15.4-2011, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/6012487

[11] Y. Jiang and V. C. M. Leung, "An asymmetric double sided two-way ranging for crystal offset," in *Proc. Int. Symp. Signals, Syst. Electron.*, Jul. 2007, pp. 525–528.

[12] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," in *Proc. 13th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2016, pp. 1–4.

[13] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2001, pp. 1655–1663.

[14] P. Corbalán and G. P. Picco, "Ultra-wideband concurrent ranging," *ACM Trans. Sensor Netw.*, vol. 16, no. 4, pp. 1–41, 2020.

[15] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[16] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, p. 25, Dec. 2013.

[17] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2003, pp. 201–212.
[18] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2652–2661.
[19] *MDEK1001 Kit User Manual Module Development & Evaluation Kit for the DWM1001*, Decawave, Dublin, Ireland, 2017.
[20] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, pp. 22292–22304, 2018.
[21] C. M. Almansa, W. Shule, J. P. Queralta, and T. Westerlund, "Autocalibration of a mobile UWB localization system for ad-hoc multi-robot deployments in GNSS-denied environments," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, 2020, pp. 1–11.
[22] A. Vashistha, A. Gupta, and C. L. Law, "Self calibration of the anchor nodes for UWB-IR TDOA based indoor positioning system," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 688–693.
[23] Q. Shi, S. Zhao, X. Cui, M. Lu, and M. Jia, "Anchor self-localization algorithm based on UWB ranging and inertial measurements," *Tsinghua Sci. Technol.*, vol. 24, no. 6, pp. 728–737, Dec. 2019.
[24] A. D. Preter, G. Goysens, J. Anthonis, J. Swevers, and G. Pipeleers, "Range bias modeling and autocalibration of an UWB positioning system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.
[25] M. McLaughlin and B. Verso, "Asymmetric double-sided two-way ranging in an ultrawideband communication system," U.S. Patent 15 500 633, Mar. 1, 2018. [Online]. Available: https://patents.google.com/patent/US10488509B2/en
[26] I. Dotlic, A. Connell, and M. McLaughlin, "Ranging methods utilizing carrier frequency offset estimation," in *Proc. 15th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2018, pp. 1–6.
[27] I. Güvenc, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, Dec. 2007, Art. no. 271984.
[28] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17429–17441, 2018.
[29] Scikit-learn. *Multi-Dimensional Scaling (MDS)*. Accessed: Nov. 21, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/manifold.html#multidimensional-scaling
[30] J. De Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, Eds. North Holland Publishing Company, 1977, pp. 133–146.
[31] J. D. Leeuw and P. Mair, "Multidimensional scaling using majorization: SMACOF in R," *J. Stat. Softw.*, vol. 31, no. 3, pp. 1–30, 2009.
[32] *DW1000 Data Sheet*, Decawave, Dublin, Ireland, 2016.
[33] P. Corbalán, T. Istomin, and G. P. Picco, "Poster: Enabling Contiki on ultra-wideband radios," in *Proc. 15th Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, 2018, pp. 171–172.
[34] D. Vecchia, P. Corbalan, T. Istomin, and G. P. Picco, "TALLA: Large-scale TDoA localization with ultra-wideband radios," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.
[35] J. Tiemann, F. Eckermann, and C. Wietfeld, "ATLAS—An open-source TDOA-based ultra-wideband localization system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–6.
[36] T. Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, Mar. 2012.
[37] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.
[38] *Decawave ScenSor EVB1000 Evaluation Board*, Decawave, Dublin, Ireland, 2013.
[39] *DW1000 User Manual—Version 2.18*, Decawave, Dublin, Ireland, 2017.
[40] J. Tiemann and C. Wietfeld, "Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–7.
[41] F. Hachem, D. Vecchia, M. L. Damiani, and G. P. Picco, "Fine-grained stop-move detection in UWB-based trajectories," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2022, pp. 111–118.

PABLO CORBALÁN (Member, IEEE) received the M.Eng. degree in telecommunication engineering from the Miguel Hernández University of Elche, Spain, in 2014, and the Ph.D. degree in information and communication technology from the University of Trento, Italy, in 2020. He is currently a Software System Engineer with NXP Semiconductors, where he works on the design of UWB solutions for mobile systems and actively contributes to UWB standardization efforts. His research interests include low-power wireless communications and localization. His research has received several awards, including the Best Paper Award from EWSN, in 2018, and IPIN, in 2019; and the Best Poster Award from EWSN, in 2016.

GIAN PIETRO PICCO (Senior Member, IEEE) is currently a Professor with the Department of Information Engineering and Computer Science (DISI), University of Trento, Italy. His research interests include software engineering, middleware, and networking, and currently oriented toward wireless sensor networks, the Internet of Things and cyber-physical systems, mobile computing, and large-scale distributed systems. The research performed in his group combines theoretical study and in-field validation in real-world applications, and has led to several awards, including the Most Influential Paper from ICSE'07 (for a paper published a decade earlier) and the Best Paper Awards from IPSN, in 2009, 2011, and 2015; PerCom, in 2012; EWSN, in 2018; and IPIN, in 2019. He is an Associate Editor of *ACM Transactions on Sensor Networks* (TOSN), IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (TSE), and the *Pervasive and Mobile Computing Journal*. He is also the founding Co-Editor-in-Chief of the *ACM Transactions on Internet of Things* (TIOT).

MARTIN COORS received the Diploma degree in electrical engineering from RWTH Aachen, Germany, in 1997, and the Ph.D. degree in electrical engineering, in 2004. From 1997 to 2002, he was a Research Assistant with RTWH Aachen. In 2002, he joined Bosch Engineering GmbH, Abstatt, Germany. He is currently with the Bosch Research and Technology Center, Sunnyvale, USA, in the area of wireless systems research and UWB indoor localization.

VIVEK JAIN (Senior Member, IEEE) received the bachelor's degree in electronics and communication engineering from the Indian Institute of Technology Roorkee, in 2002, and the Ph.D. degree in computer science and engineering from the University of Cincinnati, in 2007. In 2006, he joined Bosch and brings in extensive domain knowledge and research experience in automotive, industrial, building, and residential networking applications. He is currently an Lead Expert with Bosch Research, where he leads research on ML/AI models for wireless, RF localization and sensing, intelligent connectivity, cooperative wireless networks, and lightweight implementation. He has created and led several innovations with Bosch, including the perfectly keyless product for which he received the Robert Bosch Innovation Award, in 2019. He was served as a reviewer/TPC for more than 40 diverse international journals and conferences. He has also served as a Judge for CES 2022 Innovation Awards and "Create the Future" design contest by SAE Media Group, from 2020 to 2022.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement