Multihop Self-Calibration Algorithm for Ultra-Wideband (UWB) Anchor Node Positioning

Ben Van Herbruggen, Stijn Luchie, Jaron Fontaine, and Eli De Poorter

***Abstract*—Ultra-wideband (UWB) is an emerging technology for indoor localization systems with high accuracy and excellent resilience against multipath fading and interference from other technologies. However, UWB localization systems require the installation of infrastructure devices (anchor nodes) with known positions to serve as reference points. These coordinates are of utmost importance for the performance of the indoor localization system as the position of the mobile tag(s) will be calculated based on this information. Currently most large-scale systems require manual measurement of the anchor coordinates, which is a time-consuming and error-prone process. Therefore, we propose an algorithmic approach whereby based on measurements of the position of a small random chosen subset of anchors, the position of all other anchors is calculated automatically by collecting distances between all anchors with two-way-ranging UWB. In this article we present a three stage algorithm which contains: 1) an initialization phase; 2) a global optimization phase; and 3) an optional extra calibration phase with a mobile node. In contrast to related work, our approach also works in multihop environments with severe non-line-of-sight effects. In a real world multihop Industry 4.0 environment with metal racks as obstacles and 18 UWB nodes, the algorithm is able to localize the anchors with an mean absolute error of only 21.6 cm.**

***Index Terms*—Infrastructure, localization, ultrawideband (UWB), UWB self-calibration.**

I. Introduction

INDOOR localization systems are installed and used in various scenarios, e.g., for warehouse management tasks, drone navigation and sport tracking. A technology that has emerged as one of the most flexible standards and that can be used in most indoor environments is ultra-wideband (UWB). UWB is a radio frequency (RF) technology that estimates distances between devices with cm-level accuracy [1]. By combining these estimated distances, the system can derive the exact position of a generic asset, also called a tag [2]. In most infrastructure-based cases [3], the predicted position is relative to the infrastructure nodes, which are called anchors. If the system uses fixed anchor nodes to derive the position of the mobile tags, the accuracy of this process is highly dependent on the accuracy with which the positions of the anchors are known. In literature, the procedure to calculate the location of the anchors is called calibration [4]. High accuracy in calibration can be achieved by means of dedicated hardware and meticulous measurements [5], for example using laser measurement tools by professional surveyors. However, as the interest in UWB indoor localization systems (ILS) grows, diverse application requirements tend to increase the complexity of the calibration procedure. This is typically due to the amount of hardware that is installed as well as the specific locations, e.g., on high ceilings in a warehouse or on steep rocks down in a cave. Installing anchor nodes and manually measuring their coordinates is an expensive and time-consuming task. It is one of the main drawbacks of any UWB infrastructure-based ILS. Therefore, solutions to simplify and improve the estimation of the positions of the fixed anchor nodes are one of the top priorities in the transition of UWB into everyday life [6].

To reduce the installation costs, self-calibration mechanisms are designed. By adopting these techniques, the anchors can derive their positions in an automated manner, without any human intervention. Typically, the anchors first determine the distance between each other (e.g., using UWB measurements [1]) and use this information to calculate their most likely positions. However, most scientific solutions assume all anchors can reach each other, i.e., they are in direct line-of-sight (LOS) and distance estimation is at decimeter level accurate. However, this is not the case in large deployments, such as offices which consist of multiple open areas separated by walls. In these conditions, not all anchors are within range of each other, and those that are in range often suffer from degraded accuracy due to non-line-of-sight (NLOS) conditions between the anchor nodes. As such, current state-of-the-art (SOTA) solutions from scientific literature cannot be used in realistic large-scale deployments.

Anchor topologies are often divided in two main categories: 1) single; and 2) multihop environments. In single-hop environments all possible anchor combinations can range with each other. On the other hand, in multihop environments some pairs of anchors are unable to reach each other. These anchor pairs need intermediate relay nodes to communicate with each other. This article proposes a self-calibration algorithm suitable for multihop networks with multiple nodes. We designed a three stage solution to solve self-calibration in a large multihop network with anchors operating in NLOS conditions.

Manuscript received 9 February 2023; revised 28 April 2023; accepted 5 May 2023. Date of publication 16 May 2023; date of current version 16 June 2023. *(Corresponding author: Ben Van Herbruggen.)*

The authors are with the IDLab, Department of Information Technology, Ghent University-imec, 9000 Ghent, Belgium (e-mail: ben.vanherbruggen@ugent.be; stijn.luchie@ugent.be; jaron.fontaine@ugent.be; eli.depoorter@ugent.be).

Digital Object Identifier 10.1109/JISPIN.2023.3276826

The main contributions of the article are as follows.

1) Design of a self-calibration algorithm that is suitable for deployments with both 1) multihop UWB conditions whereby not all anchor nodes are in range with each other and 2) NLOS conditions between the anchor nodes due to obstacles or walls with a random subset of anchor nodes defining the coordinate system.
2) Simulations on different anchor constellations to analyze the influence of multihop topologies and the scalability of the algorithms.
3) Experimental evaluation of the algorithm in a large-scale UWB testbed with 18 nodes in an industrial environment of 25 × 11 m^2 with metallic blocking objects.
4) Analysis of the performance of the algorithm in relation to the number of initially known anchor node positions and their layout on the overall accuracy.
5) Publishing the dataset as open-source data promoting future research.[1]

The rest of this article is organized as follows. Section II describes SOTA approaches used for anchor node self-calibration. Next, Section III describes the self-calibration algorithm. The evaluation environment and hardware are described in Section IV. Afterward, the performance of our algorithm is analyzed in Section V, followed by future directions to improve the performance in Section VI. Finally, Section VII concludes this article.

II. Related Work

A wide range of different approaches to realize self-calibration-alike problems exists in scientific literature [6]. To date, most of these studies focus on Wi-Fi or BLE technologies, often using simulation studies rather than large-scale realistic deployments. Solutions to (self-)calibrate the anchors in UWB ILS are not yet abundant. The few existing UWB self-calibration solutions focus mainly on single-hop localization rather than multihop localization. In this related work section, we will first discuss UWB self-calibration single-hop solutions, followed by giving an overview of multihop self-calibration solutions.

A. UWB Single Hop Self-Calibration

The UWB solutions that are presented in this paragraph all share two important aspects, i.e., they are designed for LOS conditions and evaluated in a single-hop setup. Typically, the number of anchors is very limited, between 4 and 8 in a single-hop scenario. Whenever experimentally evaluated, the evaluation scenario is relatively small, i.e., anchors are maximum 20-m apart. What differs in each solution is the approach to derive the location of the anchor nodes. Batstone et al. [7] used random sampling paradigm (RANSAC) and minimal solvers. Their solution is computationally light as it sequentially merges intermediate solutions applied to smaller datasets. The article includes experiments conducted with their Loco-positioning UWB system. Anchors were installed in a 3 × 2 × 2-m^3 volume space and results show an average accuracy of 15 cm. Moreover, 6 anchors were used in these experiments. Vashistha et al. [8] evaluated a self-calibration solution based on differential time-difference-of-arrival (DTDOA) using 2 anchors only and showed a root mean square error (RMSE) between 20 and 30 cm. As this solution uses a time-difference-of-arrival (TDOA)-based approach, the synchronization between nodes becomes crucial [9]. This aspect is also relevant in Hamer et al. [10]. The authors exploit the network synchronization to combine the self-calibration process with the actual mobile nodes' localization, i.e., by sniffing the self-calibration messages exchanged among the anchors. The actual self-calibration happens using distributed gradient descent. The reported RMSE is 97 mm, in this case. Another solution that is built on gradient descent approach, is described in Ridolfi et al. [11]. This article differs from the others for mainly two reasons, i.e., the experiments are conducted in realistic NLOS conditions and UWB-specific features are exploited with machine learning and adaptive physical layer (PHY) settings to improve the self-calibration performance. This solution allows the anchors to derive their own location with 2.4-cm error only, when machine learning is used on the channel impulse response (CIR) signals. Work on improving the UWB signal model is done in [12]. The authors use a bias correction model, which considers factors, such as power levels and antenna orientation, to improve the actual UWB ranging estimations. Their solution has an accuracy less than 10 cm on average for 4 anchors in a 16 × 6 m^2 area. A combination of multiple technologies is presented [13]. In this article, UWB is used in combination with data from low-cost inertial measurements unit (IMU) sensors. Self-calibration is performed by applying a simultaneous localization and mapping (SLAM) technique and using error-state Kalman filter (ESKF) to fuse UWB and IMU inputs. The only requirement for the correct functioning of the mechanism is to let a mobile node freely move in the space of interest to collect enough data for the calibration itself. In this case, the validation uses computer simulations, which show RMSEs of around 5 cm for each space component. Yu et al. [14] introduced an algorithm called SELF-CAL. It combines Markov state transition equation (MSTE) to calculate the state vectors of all coordinates of the anchors and an iterative trilateral localization technique. The error is on average around 10 cm to (self-)position 4 anchors in a 12 × 8 m^2 room. Finally, in [15], Almanasa et al. proposed a collaborative algorithm for online auto calibration. The 4 anchors used for validation are in LOS and use least squares estimator (LSE) to combine the derived distances. Interestingly, the calibration process is simulated and reported errors are based on the size of the covered area. The average accuracy goes from 4 cm for 1 m^2 room, up to 135 cm if anchors are deployed in a 144 m^2 area. In Table I, we summarize the most relevant papers on UWB single-hop self-calibration, and we show how our work differs from prior scientific publications. Most notably our

[1] Dataset available at https://github.com/BenVanHerbruggen/UWB-dataset-for-industrial-self-calibration.

TABLE I
OVERVIEW OF SELF-CALIBRATION SOLUTIONS FOR UWB ANCHOR NODES

Solution	Conditions	Algorithm	#Nodes	Area	Topology	Average Accuracy	Experimental data	Ground truth
Batstone et al. [7]	LOS	RANSAC and minimal solvers	6	3×2 m^2	single-hop	15 cm	✓	MOCAP
Hamer et al. [10]	LOS	Distributed GD	8	6×7 m^2	single-hop	9.7 cm	✓	hand
Vashistha at al. [8]	LOS	DTDOA	2	15×15 m^2	single-hop	30 cm	✓	NA
Shi et al. [13]	LOS	SLAM and ESKF	5	20×20 m^2	single-hop	15 cm	X	simulations
De Preter et al. [12]	LOS	MLE	4	17×6 m^2	single-hop	10 cm	✓	RTK-GPS
Yu et al. [14]	LOS	SELF-CAL	4	10×8 m^2	single-hop	10 cm	✓	lasermeter
Almanasa et al. [15]	LOS	LSE	4	144 m^2	single-hop	135 cm	✓	NA
Ridolfi et al. [11]	LOS/NLOS	GD with UWB optimizations	4	10×4 m^2	single-hop	2.4 cm	✓	lasermeter
Our solution	**LOS/NLOS**	**BFGS and KF**	**18**	**25×11 m^2**	**multi-hop**	**23.3 cm**	✓	**lasermeter**

TABLE II
OVERVIEW OF MULTIHOP LOCALIZATION SOLUTIONS

Solution	Topology	#Nodes	Simulated	Algorithm
Ji et al. [18]	Centralized	400		MDS
Biswas et al. [19]	Centralized	200	✓	SDP
Capkun et al. [20]	Distributed	400	✓	Local to global maps
Ihler et al. [21]	Distributed	10	✓	PF
Kannan et al. [22]	Centralized	200	✓	SA

algorithm can operate in NLOS conditions and in large multihop environments.

B. Multihop Localization

The solutions presented in the previous paragraph are validated and designed for small-scale networks. As UWB is gaining popularity and aims to be used in many different scenarios, a more appropriate study on large-scale UWB self-calibration is needed. However, in scientific literature there are no solutions where many UWB anchors need to be calibrated in large environments. Nevertheless, multihop localization is a well-researched topic [16], [17]. Thus, in this section we comment on some of the most common techniques. An overview of these solutions is given in Table II.

In previous single-hop solutions, the common assumption was that all the nodes in the networks are in reach of each other or at least all the anchors can communicate with a subset of anchors with known positions. Scaling up this approach means to overcome this limitation. Hence, multihop algorithms deal with sets of anchors that are out-of-reach from the known anchors, if any.

In [18], the authors design a distributed positioning method based on multidimensional scaling (MDS) technique. More specifically, MDS is used to calculate the relative positions of the nodes. To deal with missing distances, which happens frequently in realistic indoor conditions, the authors propose an iterative version of MDS. Their solution is simulated in MATLAB using 400 nodes. Results show that the iterative nature of their method allows accurate position estimates while eliminating the accumulation of the measurement error.

Biswas et al. [19] tackled the localization problem using a semidefinite programming (SDP) relaxation-based method. The approach is to convert the nonconvex quadratic distance constraints into linear constraints. Their simulations with over 200 nodes show that very few anchors are required to estimate the position of the mobile tags. Moreover, this seems to work well even when anchors are not placed in the optimal locations within the network.

Authors in [20] designed a distributed, infrastructure-free positioning algorithm that only uses distances between nodes to build a relative coordinate system. It is referred to as self-positioning algorithm (SPA). A location reference group is chosen and maintained over time. The system exploits the awareness of the location of the nodes to make sure that messages are sent out in the right direction. As previous papers, simulations with 400 nodes show that it is possible to build a coordinate system without a centralized knowledge of the network topology. However, further work to improve accuracy and enable this solution in real-world applications must follow.

In [21], the estimation of the position of the nodes is done with a generalization of a particle filter (PF), i.e., nonparametric belief propagation (NBP). An advantage of this approach is the fact that it is relatively easy to implement and can adopt many statistical models. Moreover, it can represent multimodal uncertainty. The authors applied a graphical model framework to solve the problem. They concluded that the method is robust and can keep the computational load low, e.g., by appropriate message scheduling. However, the authors highlighted that this powerful tool still needs to prove its potential when it comes to large ad-hoc networks.

Kannan et al. [22] defined a simulated annealing (SA)-based algorithm to deal with erroneous and noisy data that is used in positioning systems. The cost function represents the quantitative measure of the quality of the coordinate estimates. The goal is to minimize this function. Similarly, to previous solutions, 200 nodes were used in their simulations. They show that their SA solution performs better than others, such as SDP. A common problem that arises with SA is flip ambiguities and the authors highlight how this concern must be tackled in future research. To summarize, the above solutions have been designed mainly for simulations, and do not consider UWB specific communication features nor do they evaluate the solution on a real-life large-scale deployment. Both limitations have been tackled in this article.

III. Self-Calibration Algorithm

The proposed algorithm will be executed in three different stages. In the first stage, an initial position for each anchor is determined. Next, these found positions are improved in the second stage. Finally, a mobile node is guided between the anchors to further tune the anchor position and remove inaccuracies of the algorithm due to bad NLOS links.

A. Stage 1: Initialization of an Initial Solution

The first stage utilizes a heuristic approach to solve the anchor localization problem stated before. As input for stage 1 of the algorithm, the following input is required.

1) *A set of foreknown anchor coordinates S_m:* At least three anchor coordinates need to be available for the algorithm. By assuming that at least three coordinates are known, all other coordinates can be approximated in the same coordinate system as the surrounding building. In this way of working, no mapping between the algorithms coordinate system and the environments coordinate system has to be made. To evaluate the proposed algorithm ten different sets of initial nodes are selected. An out-of-the-box system should work with any combination of initial anchor sets.
2) *The distance between all anchors:* To obtain the distance between the anchor nodes, all anchors are initially programmed to range 100 times to the other anchors. The mean of the measured distances is used for the algorithm together with the standard deviation on the found distances.

Next, the anchors with unknown positions are added sequentially to the set of known anchor coordinates S_m. We prioritize the anchors with the highest number of connections to anchors in S_m. Once an anchor is selected to be added to the set, its position is calculated based on the current set of known anchors S_m. An algorithmic representation of this method is given in Algorithm 1.

When adding new anchors, stage 1 optimizes a cost function $f(x_i, y_i)$. The cost function chosen for stage 1 is inspired by geometry and is given by (1). The cost of choosing a particular position (x_i, y_i) for anchor i is the difference between the measured distance $\hat{d}_{im}$ and the calculated distance $d_{i,m}$ between a positioned anchor and anchor i, summed over all fixed anchors that range with this anchor i. The optimal point for the new anchor is determined by a linear least squares (LLS) approach [23]

$$\begin{aligned} f(x_i, y_i) &= \sum_{m\in S_m}\left(\sqrt{(x_m-x_i)^2+(y_m-y_i)^2} - \hat{d}_{i,m}\right) \\ &= \sum_{m\in S_m}\left(d_{i,m}-\hat{d}_{i,m}\right) \end{aligned} \tag{1}$$

with:

1) (x_m, y_m): coordinate of a positioned anchor m;
2) (x_i, y_i): coordinate of anchor i;
3) $\hat{d}_{i,m}$: measured distance between anchors m and i;
4) $d_{i,m}$: calculated distance between anchors m and i.

Algorithm 1: Stage 1: Geometric Initialization Approach.

```
Input: $S_m$ , $\hat{d}$
Output: $S_m$
while size($S_m$) $<$ #nodes do
    for node $n$ not in $S_m$ do
        score$_n$ $=$ count ($\hat{d}_{S_m,n}$)
    Select node $n$ with highest score$_n$
    for $i < 10$ do
        assign $n$ random position
        $x_i, y_i = LLS(n, S_m)$
        cost$_i$ $= f(x_i, y_i)$
    pick solution $i$ with lowest cost
    add $n = (x_i, y_i)$ to $S_m$
```

B. Stage 2: Optimization of the Initial Solution

Stage 1 initialized all positions sequentially (one by one). This means that at timestamp t, only the anchors that have already been positioned by the algorithm can be used to position a node with unknown position. After stage 1, all anchor nodes now have been given a preliminary position. Stage 2 of the algorithm aims to correct any multihop errors that accumulated through stage 1. Since all nodes now have a preliminary estimated position, stage 2 can use the full topology of the network to try to improve the accuracy of the self-calibration.

The constraints for solving the topology are inspired by a physical interpretation, namely, that the measured distances between different anchor pairs can be represented by springs. These springs exert a force along the direction they are pulled on, which is proportional to the extension or compression of the spring. The relation between the displacement and the executed force is presented in (2). This extension or compression force k will be larger in case the initial anchor node position was incorrect (e.g., in case the anchor node position does not adequately match the actual distance measurements between anchor nodes)

$$F = k * \Delta d. \tag{2}$$

In a physical context, this k depends on the spring's material and construction. In our algorithm, the displacement variable k represents the level of suspected error of an UWB link. We assume that the most trustworthy links will have small variations in the measured distances and these links will hence be considered as more reliable measurements in the self-calibration algorithm. For every link, the reciprocal of the standard deviation of the UWB signal will be employed as the value of k. The total energy of the system will be minimized and is given as follows:

$$E = \frac{1}{2} * \sum_{n=1}^{N} \frac{1}{\sigma_n} * \Delta d_n^2. \tag{3}$$

Our solver implements the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS-B) gradient descent algorithm [24] to find an optimal solution for this network by minimizing the defined cost function.

BFGS tries to eliminate the adverse effects of the Newton method by approximating the inverse of the Hessian $\mathbf{H}_k$. BFGS can be placed between gradient descent (GD) and the Newton method and as such acts as a quasi-Newton method. The time complexity of BFGS is $\mathcal{O}(n^2)$. Optimizations of this algorithm exist to get the time complexity lower. Like BFGS, L-BFGS uses an estimate of $\mathbf{H}_k$ to steer its search through variable space. However, L-BFGS only stores some vectors which represent the implicit approximation. L-BFGS maintains a history of the past i updates of the state $\vec{x}$ and gradient $\nabla f(\vec{x})$, where generally, the history size i can be small (often $i < 10$). These updates are used to perform operations where $\mathbf{H}_k$ is required implicitly.

C. Stage 3: Improving the Locations by Applying an Kalman Filter

It is possible that in some regions anchor nodes are deployed too sparsely and unable to reach sufficient other anchor nodes to be calibrated with high accuracy. To improve the calibration of these isolated anchor nodes, stage 3 of the algorithm introduces a mobile tag that can be moved to regions with high anchor node positioning uncertainty. The moving tag is manually walked throughout the area of uncertainty and ranges with all locally available anchors. The position of the tag is estimated, and position outliers can thus be linked with incorrectly positioned nearby anchor nodes. Stage 3 will update the coordinates of these anchor nodes using a Kalman filter. Kalman filters exist in many variants depending on the nature of the input data. The filter models all probabilistic distributions as Gaussian distributions defined by a mean and covariance matrix. The filter updates these parameters over time when new data are processed.

The Kalman procedure is outlined in Algorithm 2. The filter tracks the estimated state of the tag's position and speed on both the x and y-axis, as well as the positions of the anchors and the uncertainty of the estimate. In total, there are $4 + 2n$ parameters to predict, where n represents the number of anchors. The filter operates in two steps: 1) prediction; and 2) update. During each iteration, the prediction step is performed first. The new state is predicted based on the previous state and the uncertainty of the prediction, which is estimated based on various process noises in the system. In the update step, measurements from UWB sensors are incorporated. The difference between the predicted and measured values is used to calculate the Kalman gain, which represents the certainty of the measurement. This is then used to calculate the updated state. Every new UWB measurement triggers both steps, resulting in an adjusted state for the Kalman filter.

First, for the prediction step in the Kalman filter we define the state transition matrix $\mathbf{A}$

$$A = \begin{bmatrix} 1 & 0 & dT & 0 & 0 & \dots \\ 0 & 1 & 0 & dT & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}. \tag{4}$$

Matrix R denotes the uncertainty in the movement of the anchors and tag. The anchors are static and do not move, therefore this part of the matrix will be zero.

$$R = \begin{bmatrix} R1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ with } \quad R1 = G \cdot \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \cdot G^T \tag{5}$$

$$\text{and} \quad G = \begin{bmatrix} \frac{dT^2}{2} & 0 \\ 0 & \frac{dT^2}{2} \\ dT & 0 \\ 0 & dT \end{bmatrix}. \tag{6}$$

Q depicts the uncertainty from the measurements. This uncertainty influences all parameters. This implies a diagonal matrix of size $(4 + 2n) \times (4 + 2n)$ with the elements on the diagonal equal to σ_{UWB}

$$Q = \sigma_{\text{UWB}} \cdot \begin{bmatrix} \mathbf{1} \end{bmatrix}_{[(4+2n)\times(4+2n)]}. \tag{7}$$

The second part of the Kalman filter is the update step. For the update step a new state transition matrix $\mathbf{C}$ is defined, which updates the position of the anchor as well if the algorithm processes a range with this anchor. First, the distance between anchor i and the tag is calculated based on the current state of the Kalman filter (KF). Next, the contributions along both axes are calculated, as described by (8) based on the coordinates of the tag (x^t, y^t) and anchor i (x_i^a, y_i^a)

$$x_{\text{contr}} = \frac{x^t - x_i^a}{\mathrm{d}_{t,a_i}}, \qquad y_{\text{contr}} = \frac{y^t - y_i^a}{\mathrm{d}_{t,a_i}} \tag{8}$$

$$C = \begin{bmatrix} x_{\text{contr}} & y_{\text{contr}} & 0 & 0 & \dots & -x_{\text{contr}} & -y_{\text{contr}} & \dots \end{bmatrix}. \tag{9}$$

Finally, the measurement y_t that will be used at time t in the update step of the KF can be calculated by distracting the distance between the anchor and the current state of the filter d_{t,a_i} from the distance measurement $(\hat{d})$ between tag and anchor

$$y_t = \hat{d}_{t,a_i} - d_{t,a_i}. \tag{10}$$

The Kalman filter is evaluated in the lab environment and therefore its parameters are determined based on previous research in the same environment. The measurement variance of the UWB ranges is 25 cm. It is a rather conservative estimation of the noise on the UWB ranges as the technology can reach cm-level accuracy in clear LOS conditions. The process variance is 0 for all variables corresponding to the anchor nodes as they are at fixed positions. The process variance of the parameters of the tag is chosen high (10 000) for both coordinates as we do not know the initial position of the tag and we assume that the velocity of the tag is almost zero (0.001) when starting the Kalman filter.

The final KF of stage 3 is given in Algorithm 2. The total time complexity of the algorithm is $\mathcal{O}(k * (4 + 2n)^{2.376})$ with k the number of measurements taken during the run and n the number of anchors in the network [25].

Algorithm 2: Stage 3: Kalman Filter.

```
Input: $\vec{\mu}_{t-1}$, $\sum_{t-1}$, $y_t$
Output: $\vec{\mu}_t$, $\sum_t$
Prediction step
  $\vec{\mu}_t = \mathbf{A}_t \vec{\mu}_{t-1}$
  $\sum_t = \mathbf{A}_t \sum_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$
Update step
  $\mathbf{K}_t = \sum_t \mathbf{C}_t^T (\mathbf{C}_t \overline{\sum}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$
  $\vec{\mu}_t = \vec{\mu}_t + \mathbf{K}_t (y_t - \mathbf{C}_t \vec{\mu}_t)$
  $\sum_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \overline{\sum}_t$
with:
  • $\vec{\mu}_t$: state of anchors and tag at time $t$
  • $\sum_t$: covariance matrix of the state at time $t$
  • $y_t$: measurement at time $t$
  • $\mathbf{A}_t$: State Transition Matrix
  • $\mathbf{R}_t$: uncertainty on movement of the tag
  • $\mathbf{C}_t$: State Transition Matrix
  • $\mathbf{Q}_t$: uncertainty on measurements
```

IV. Measurement Setup and Data Collection

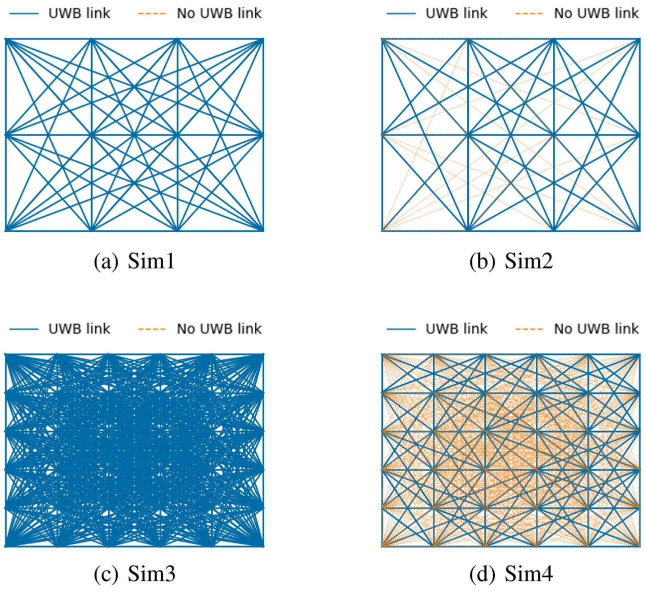
This section describes the two evaluation environments that are used: 1) simulation; and 2) an industrial testbed environment.

A. Simulation Environment

Four simulated scenarios are used to evaluate the algorithm as follows: 1) LOS versus NLOS; 2) single versus multihop; 3) dense versus sparse networks; 4) low versus high dilution of precision (DOP). As explained in Section III, the distance between two nodes is used as input in the self-calibration algorithm. One measurement, representing one UWB link, contains the addresses of both anchors, the mean and the standard deviation of the simulated distances. Errors occur on these ranges and can be divided into two parts. First, errors occur due to propagation faults when traveling from sender to receiver, which are called environment dependent errors in the remainder of this work. Second, hardware dependent errors can be introduced: a bias can be added due to the internal configuration in the chips which process the signal; whether or not caused by the receive power level [26]. Therefore, when generating the measurements between anchor nodes, the next steps are taken as follows.

1) For each measurement, an offset is added, which can be both negative and positive, from a Gaussian distribution with variance σ_a, to simulate propagation faults (environment dependent error).
2) Set a bias on the actual measurement. This bias is drawn from a Gaussian distribution with variance σ_b which can be altered to simulate different hardware statistics (hardware dependent error).
3) 100 samples are drawn and the mean and standard deviation are calculated.

The range for UWB packets to propagate is limited and reaches 5–100 m, depending on the PHY settings that were used. As the scale of the network is limited, the choice was made to cap the range of the simulated data at a maximum distance of d which is the same for all anchors. Restricting the range of a simulated UWB sensor ensures that multihop environments can be simulated. Using this method, a wide range of environments can be created. For example, a grid-like UWB system is depicted in Fig. 1.



(a) Sim1

(b) Sim2

(c) Sim3

(d) Sim4

Fig. 1. Different anchor constellations are simulated to evaluate the functionality of the first two stages of the proposed algorithm. (a) Sim 1 and (b) Sim 2 contain 12 nodes, while (c) Sim 3 and (d) Sim 4 have 36 nodes. The simulations contain both single (a+c) and multihop (b+d) scenarios.

Simulations can create a very diverse set of environments. Therefore, careful consideration should be taken on which environments to test the different stages. First, the algorithm should be able to work in the simplest environments. Hence a simulation was created where all nodes can range with one another, are ordered in a symmetrical optimal constellation, and the number of nodes is limited to 12. Second, the algorithm should perform reasonably well when not all nodes are in range with each other. Simulation 2 has the same anchor constellation as simulation 1 but range is limited to create a multihop scenario. Third, an environment is needed where there are a larger number of nodes available, and all of these nodes can range with each other. This environment can be considered as a larger version of simulation 1 with much denser anchor node deployment. Finally, a large-scale scenario where not all nodes can range with each other is defined. After the two stages, the algorithm should be able to position all anchor nodes reasonably well for all four simulations. The set of initial nodes S_m will be exactly the same for all scenarios, selected in one of the corners of the anchor topology to investigate the effects of multihop behavior in the setup. The different simulation environments and their settings are depicted in Table III.

NLOS environments are difficult to simulate using this method, except when a significant variance is used in the second step of the simulation. Consequently, the algorithms must be tested in true-to-life environments as well.

TABLE III
SETTINGS OF THE FOUR SIMULATION ENVIRONMENTS

Scenario	#nodes	range [m]	#samples	#hops	node density [m^2/node]	links per anchor
Sim1	12	40	100	1	50.0	11
Sim2	12	25	100	2	50.0	8.5
Sim3	30	70	100	1	69.4	35
Sim4	30	22	100	5	13.9	18

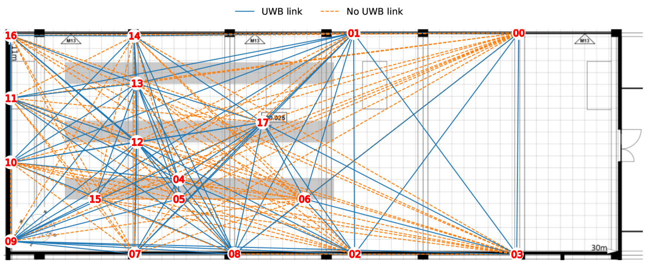
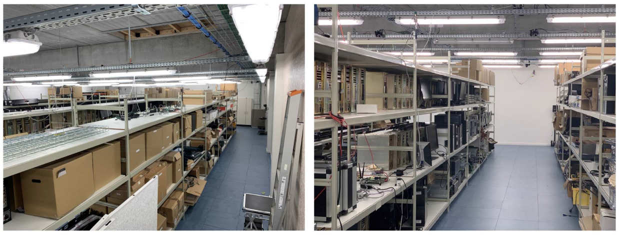


Fig. 2. UWB data are collected in a real world industry environment containing 18 anchors which are mainly installed between the racks. Many of the UWB links are failing in this environment, making it a multihop environment.



(a) (b)

Fig. 3. Industrial environment is created with multiple UWB anchors between the metal racks.

B. Industrial Warehouse Environment

The real-world environment that is used to perform self-calibration is the IIoT lab located at the iGent building [27]. This environment is specifically used for UWB experiments and partially uses a motion capture (MOCAP) system to capture ground truth. The IIoT lab has an area of approximately 25 × 11 m^2. It houses two sections, one open space, and one space were metal racks are placed. In total, 18 anchors will be used in the IIoT lab. The placement and connectivity of the anchors and warehouse racks are shown in Fig. 2. The anchors are densely installed in the lab and remotely accessible to emulate different possible scenarios. The anchors are placed between the racks so that in every aisle at least 2 anchors are present on top of the racks and one anchor at the end of the aisle. The wider left aisle has more anchors to evaluate more detailed warehouse localization scenarios. The environment of the anchors is shown in Fig. 3. The UWB anchors are Wi-Pos nodes [28] and contain the Decawave DW1000 UWB transceiver, a Zolertia remote with sub-GHz radio for creating a wireless backbone and a flexible antenna interface. A second radio technology is desirable in large deployments to use as backbone network for synchronization and data reporting. The UWB devices are programmed to transmit at channel 5 (center frequency of 6489.6-GHz and 499.2-MHz bandwidth), a preamble length of 1024 symbols, a pulse repetition frequency (PRF) of 64 MHz, and the default transmit gain as proposed by the manufacturers user manual [26]. Ten different sets of predefined anchor sets will be considered in this challenging industrial environment. Every anchor will be in one or two initial sets to generalize the results.

TABLE IV
MEAN ABSOLUTE ERROR FOR THE FOUR SIMULATION SCENARIOS AND THE ACCURACY FOR DIFFERENT SIZES OF PREDEFINED ANCHOR SETS

	stage 1 [cm]			stage 2 [cm]		
Anchors:	3	4	5	3	4	5
Sim1	20.1	18.0	15.6	10.1	9.8	9.0
Sim2	30.4	25.8	25.5	19.3	18.2	18.0
Sim3	27.1	21.7	19.9	13.0	10.6	10.2
Sim4	44.6	42.0	37.7	35.1	34.1	33.6
mean	30.5	26.8	24.7	19.4	18.2	17.7

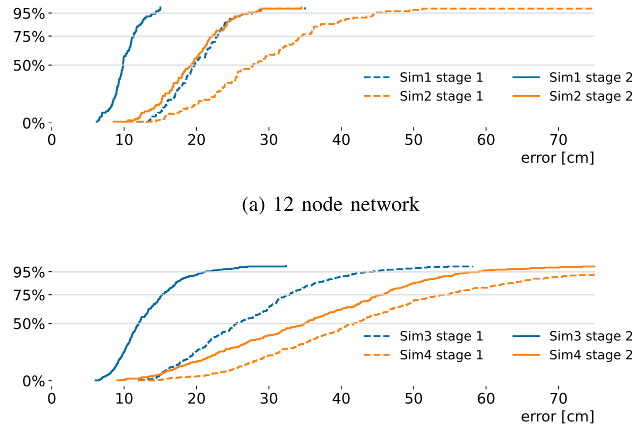
A minor improvement is noticeable when more anchors are known at the initialization of the algorithm.

V. DISCUSSION

A. Simulation

To have a clear assessment of the performance of the algorithm, it is run in simulated anchor constellations. Three randomly chosen anchor nodes are initially added to S_m, the set of known anchors and the coordinates of the other anchors are calibrated in the same reference coordinate system. An initial set of 3 anchor nodes corresponds to 25% of the nodes in the smaller constellations (Sim 1 and Sim 2) and 8.3% for the larger constellations (Sim 3 and Sim 4). All other anchors are evaluated toward their simulated position. The simulations were run with a hardware-dependent bias of 10 cm and the environment-dependent error has a variance of 30 cm. Both values are experimentally validated in a lab environment and comply with the standard industry performance of ILS. The results for both stages are summarized in Table IV.

1) Stage 1: When all nodes are added sequentially to the network, we see that the mean absolute error (MAE) on the anchor locations is the smallest in the simplest scenario Sim1. In a single hop network with 12 nodes, the anchors can be localized in stage 1 up to 20.1-cm accurate. When multihop is introduced in the same network, the MAE of the anchors positions increases to 30.4 cm. Errors are thus propagating through the network. For scenarios Sim 3 and Sim 4, with larger amounts of anchors, the MAE is 27.1 cm for the single hop variant. The error is higher than in small networks as the already localized anchors are also used for adding the new node to the network and therefore earlier deviations are propagating through the network. In multihop variants, the furthest points are localized purely on previously localized anchors and are therefore less accurate than points localized on the fixed anchors. Scenario Sim4 has an MAE of 44.6 cm, which is the highest of all scenarios. The total cumulative error distribution for stage 1 is shown in Fig. 4.



(a) 12 node network

(b) 36 node network

Fig. 4. Cumulative distribution functions of the different scenarios show a high increase in accuracy of the second stage of the algorithm with 3 anchors in the initial anchor set. (a) In smaller networks with only 12 anchors, the two-stage algorithm in multihop environment is as good as the single hop stage 1 for the majority of the nodes. (b) In larger networks, much more variation on the error is present.

2) Stage 2: The second stage is designed to improve the global network as one system. It takes as input the anchor locations that were found after stage 1 and tries to minimize the total cost energy of the network. For all four scenarios the MAE of the anchor position is lower after executing stage 2. For scenario Sim 1, the MAE is dropped by 10 cm down to a value of 10.1 cm. For Sim 2, the improvement is even higher: stage 2 succeeds in lowering the error with 11.1 cm down to 19.3 cm. This multihop variant is still more difficult to localize than the single hop variant. The constellation with more anchor nodes has a very high improvement for the single hop scenario. After stage 2 the MAE is only 13.0 cm, almost 14.1 cm better than the anchor locations after stage 1. The multihop variant has the smallest decrease in MAE of the anchor positions from 44.6 cm after stage 1 down to 35.1 cm after stage 2. The multihop character clearly influences the localization accuracy of the self-calibration algorithm. The total cumulative error distribution for the stage 2 is shown in Fig. 4. The first stage of a single-hop scenario is similar (for the small network) or even better (for the larger network) than the second stage of the multihop network.

B. Industrial Environment

1) Stage 1: Running stage 1 on the 10 configurations with predefined anchors at the IIoT lab yields the following results: The median absolute error is 28.7 cm while the MAE is 40.8 cm. There are a few configuration that result in high outlier values. 75% of the anchors are approximated within 59 cm of their actual position and 95% of the anchors within 83.0 cm. After stage 1, the standard deviation of the absolute anchor localization is 25.5 cm. Fig. 5 shows the cumulative distribution functions (CDFs) for this stage compared to the stages executed afterward. The range of mean error observed during the first stage 1 at the IIoT lab, which falls between 41.7 to 38.2 cm for the total 30 configurations, indicates that there is considerable potential for enhancing the accuracy. For every new anchor, stage 1 requires three ranges to already positioned anchors. Stage 1 is thus highly dependent on the order of adding anchors to the self-calibrated solution. Stage 2 is specifically designed to overcome this effect.

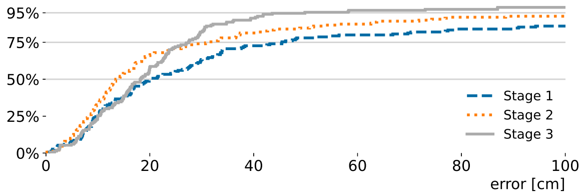


Fig. 5. Results in the IIoT lab with three initial nodes. Stage 2 increases the accuracy of the anchor self-calibration problem in the IIoT lab compared to stage 1. Stage 3 is added robustness to the procedure by lowering the error of the higher percentiles.

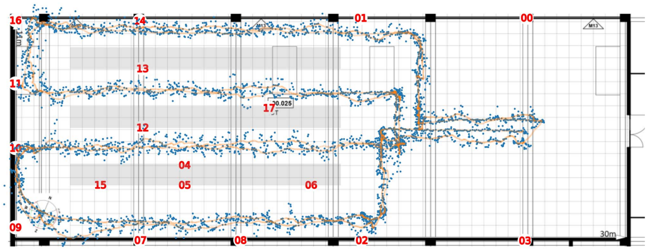


Fig. 6. Localized points of the mobile node when collecting data for stage 3. In the open space area (without racks) the points are better cluttered to straight lines showing better ranging performance there. All corridors in the warehouse were walked and data are collected near all anchor nodes.

2) Stage 2: The median absolute error is now 22.0 cm, 23% lower than after stage 1. The mean absolute error is 28.2 cm. Compared to stage 1, better results are obtained for every configuration of predefined anchors. This is expected as all available data are used by stage 2; generally, more (good) data leads to better results. The standard deviation of the anchor errors is 15.0 cm, showing a higher robustness of the results after the execution of both stages. The 75th and 95th percentile are 24.5 and 57.5 cm, respectively. All metrics show that executing stage 2 is beneficial to the self-calibration algorithm in both accuracy and robustness of the final solution. Both stages have the same input data and computation complexity is low. Another perspective of analysis can be obtained by looking at the CDFs of stage 2, then comparing them to the CDFs of stage 1, as shown in Fig. 5. For all anchor nodes, stage 2 improves the accuracy of the self-calibration.

3) Stage 3: Stage 3 uses a mobile tag to collect additional data. A trajectory between the racks has been executed, but no ground truth positions of the mobile tag trajectory are used. The positions that were estimated by the KF used in stage 3 are shown in Fig. 6, together with a moving average showing the path. The median difference between the calibrated anchors and their actual position is now lowered to 19.1 cm. The mean is slightly higher than the median at 21.6 cm. When looking at the CDFs for the different anchor configurations after stage 3, it can be observed that the 95 percentile is 35.4 cm and also a 7.6 cm standard deviation between the anchor errors, which both shows robust self-calibration. The majority of the anchors was calibrated accurately but in stages 1 and 2 some outlier values occur and one anchor can be in the order of 1 m off. This can be explained by one bad range and limited ranges with this anchor. For example anchor node 15 has only three successful links in the dataset and is often wrongly calibrated after the first two stages. Stage 3 is able to resolve this problem. This situation demonstrates the purpose and the usefulness of having a mobile tag be used in the self-calibration process.

C. Influence of UWB Settings

The self-calibration algorithm takes the measured ranges between the anchor nodes as input. In literature, many authors mention that the quality and range from an UWB link is dependent on the used PHY settings. The self-calibration algorithm was executed on a second dataset that was collected with a more conservative PHY layer that is used as control scenario to validate the algorithm. A lower center frequency at channel 3 lowers the path loss and attenuation on the link, expanding the range. A longer preamble increases the changes for correctly receiving a packet with a lower receive power. With these settings, we see the number of links in the IIoT lab increasing from 80 up to 141 out of the total 153 anchor combinations present in the lab. The multihop environment has been transformed in a quasi single-hop environment, with only a few missing links (only 8% of the links failed). The UWB packets are received with higher signal strengths and this results in a decrease of the mean absolute ranging error between the anchors of 69%. With these PHY settings and three known initial nodes, the three stages can reach mean accuracies of 23.3, 18.1, and 20.6 cm, respectively. Using the same environment with more available links closes in on the single-hop scenario and improves anchor self-calibration significantly which confirms the conclusions of the simulations and the challenging PHY settings used in the evaluation.

D. Influence of the Number of Measured Anchors

The previous results assumed three anchor nodes to be measured by the system installer. Some infrastructure installations might permit easy and accurate determination of more of these nodes. During this evaluation no assumptions were made on the selection of these initial node and the installer can chose which are best suited to measure by hand. In this paragraph we will investigate the performance of the self-calibration algorithm if more anchors are initially known. The subsets of fixed anchors are expanded with each time one or two additional anchors and are based on the same anchor combinations.

1) Simulations: In the simulations, there is a small improvement in the MAE when adding more nodes to the predefined set of anchor nodes S_m after stage 2. In Table IV, an overview is presented and it can be concluded that adding more anchors will definitely increase the robustness of the system and using a predefined set S_m with 5 anchor nodes still results in the best performance for the anchor localization.

2) IIoT lab: In the IIoT lab the same behavior is observed (Table V). When 5 anchors (27.8% of the nodes in the lab) are measured initially, the anchor nodes can be localized up to 21.9-cm accurate, an accuracy decrease of 0.3 cm than when only 3 anchors were known. However, the median and 75th percentile are still improving by a few centimeters. The MAE is at 26.1 and 22.4 cm after stage 1 and stage 2, respectively, when starting with five known nodes. If using four anchors (22.2% of the nodes in the lab) in the predefined set, the mean of stage 1 is high (41.7 cm), caused by some wrongly located nodes, but when applying stage 2 and stage 3, the accuracy with four predefined anchor nodes is already better than the 3 anchor nodes cases.

TABLE V
ABSOLUTE ERROR FOR DIFFERENT SIZES OF PREDEFINED ANCHOR SETS IN THE IIOT LAB. AN IMPROVEMENT IS NOTICEABLE WHEN MORE ANCHORS ARE KNOWN AT THE INITIALIZATION OF THE ALGORITHM

	stage 1 [cm]			**stage 2 [cm]**			**stage 3 [cm]**		
size S_m:	3	4	5	3	4	5	3	4	5
mean	40.8	41.7	38.2	28.2	25.1	26.1	21.6	20.6	21.9
std	25.5	26.0	34.7	15.0	10.1	12.0	7.6	5.2	15.8
median	28.7	29.7	25.5	22.0	22.6	23.1	19.1	19.5	17.3
75 th	59.0	48.7	28.5	24.5	28.5	27.2	22.6	24.0	20.6
95 th	83.0	88.6	103.5	57.5	42.8	46.6	35.4	28.6	47.5

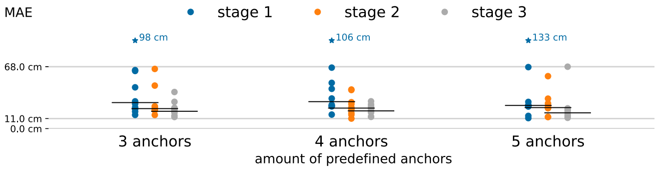


Fig. 7. Three times ten different known subsets are chosen. Using more anchors and/or more stages results in a more robust self-calibration of the anchor nodes and decreases the influence of the initial set.

E. Influence of the Selected Measured Anchors

The proposed algorithm assumes that 3 anchor coordinates are known. However, depending on which anchor node positions are initially measured, the performance of the algorithm might be influenced. Ideally, the outcomes are as independent of the initial choice of anchor nodes as possible. A good representation of this impact is the variation of the MAE in case different initial anchor nodes are selected. In Fig. 7, the MAE value for every configuration is given depending on each stage and on the number of predefined anchor nodes. When three nodes are chosen, the MAE is around 21.6 cm after the three stages but on the graph we also find predefined anchor sets with higher errors up to 133 cm, more than ten times the error of the best chosen anchor set. The variation on the final MAE in the anchor position decreases when more stages are used and when more anchors are predefined in the lab settings. Depending on the requirement of the indoor localization system, the installer might choose to measure more anchors. If less accuracy is required, stage 3 can be left out for faster self-calibration.

F. Influence of Cost Function

Choosing the spring constant k to be dependent on the standard deviation of the range measurements proved to be a good choice. If the spring constant is kept constant over all links, the error of the IIoT lab self-calibration is almost 34.3-cm higher after stage 2 and 1.5-cm higher after stage 3. During stage 3

TABLE VI
AFTER APPLYING GAUSSIAN NOISE TO THE ANCHOR COORDINATES THE SYSTEM GETS LESS ACCURATE

σ_{anchor} **[cm]**	0	10	20	40	80	160	320
MAE [cm]	0	5.4	7.6	26.7	56.3	102.8	132.8

more measurements are available and not weighted, making the influence smaller. With a constant k, the errors are almost doubled compared to a variable cost function per link. Nonetheless, an open question remains whether the standard deviation is the best parameter for this purpose. In future work, it could be investigated whether the estimate of the error of the ranges could be used instead of the standard deviations. In literature, machine learning is used for LOS/NLOS detection [29] and error mitigation [30] of the links. This estimated ranging error information could be used as as an alternative indicator of the trustworthiness of the measurement during phase 1 and 2 of the self-calibration.

G. Influence of Anchor Accuracy on Systems Accuracy

The required accuracy of the anchors depends on the intended system performance. To evaluate the influence of the anchor coordinates accuracy on the actual positioning, an extra simulation is done. First, 50 points are randomly selected in the IIoT lab and the UWB links to the anchor nodes are simulated. Next, Gaussian noise is added to the anchor coordinates. Finally, the selected points are estimated based on the noisy anchor coordinates and the simulated UWB links. In Table VI, the standard deviation of the noise on the anchor coordinates and corresponding MAE of the selected points is given. When the noise added to the anchor coordinates is increased, the positioning of the selected points is less accurate. A Gaussian distribution with σ equal to 20 cm, the order that our self-calibration algorithm was able to reach, shifts the MAE of the system with only 7.6 cm. Therefore the proposed algorithm can be adopted in many UWB applications.

VI. Future Work

This work has presented an innovative method for anchor calibration with limiting the manual effort for the installer of the UWB ILS and during the discussion potential future research determined. To start with stage 1, we expect better results if not only the number of links to already positioned nodes (S_m) is considered but also the DOP value of the anchor combination. If the neighbors are geometrically better positioned, the new anchors are more likely to be accurately localized. Second, in stage 2 further investigation can be performed on the optimal value of the spring constant k. Up to now, the standard deviation has been a good candidate for estimating the link quality but more advanced metrics based on other parameters, such as receive and first path power could probably boost the algorithm by better selecting the links and based on the estimated quality of the links, the algorithm could also assess the final anchor accuracy. Also, in stage 3, the path taken by the tag impacts the accuracy improvements. The authors chose to walk through the entire space wherever physically possible. However, path optimizations can include: 1) focusing on regions of high uncertainty (e.g., close to anchor nodes with large remaining spring force) and/or; 2) avoiding NLOS environments as much as possible to avoid using less accurate measured distances and/or; 3) optimization of the PHY UWB properties for optimal input for the Kalman filter. Future work can investigate the best guidelines for the path of the tag for each type of environment and on the selection of the initial chosen nodes.

VII. Conclusion

Anchor self-calibration is a game changer in the deployment of UWB ILS. The tasks of the system installer are limited to mount all anchors and measure a minimal subset of 3 anchors to define the coordinate system. In simulations, the presented algorithm reached an MAE of 30.5 cm after the initialization phase (stage 1) which was decreased to 19.4 cm with the optimization stage 2. In addition to the simulations, the algorithm is also validated in an experimental testbed, where 18 anchors are installed with metal racks occluding the direct path of many links. With three predefined anchors, we found an accuracy of 40.8 cm after stage 1, a 28.2 cm after stage 2, and 21.6 cm after stage 3. In the proposed algorithm, standard deviation was introduced as quality metric of the link. Without this quality metric to weight the individual links during self-calibration, the accuracy is almost 34-cm less accurate after the two first stages. If the known set of anchors is expanded to five anchor nodes, the three-stage algorithm is capable to increase the robustness of the anchor self-calibration. This result is still higher than the manual laser meter accuracy levels, but our algorithm lowers the manual effort of measuring the nodes and permits easy recalibration when anchors are moved. Executing the algorithm periodically could expose unexpected changes in anchor locations (when batteries are switched or during cleaning of the building).

Acknowledgment

The authors would like to thank Dr. Matteo Ridolfi for his technical support and UWB self-calibration expertise during the design of the algorithm.

References

[1] K. Witrisal, S. Hinteregger, J. Kulmer, E. Leitinger, and P. Meissner, "High-accuracy positioning for indoor applications: RFID, UWB, 5G, and beyond," in *Proc. IEEE Int. Conf. RFID*, 2016, pp. 1–7.

[2] M. Ridolfi et al., "Experimental evaluation of UWB indoor positioning for sport postures," *Sensors*, vol. 18, no. 1, 2018, Art. no. 168.

[3] N. Macoir et al., "UWB localization with battery-powered wireless backbone for drone-based inventory management," *Sensors*, vol. 19, no. 3, 2019, Art. no. 467.

[4] G. Shi and Y. Ming, "Survey of indoor positioning systems based on ultra-wideband (UWB) technology," in *Wireless Communications, Networking and Applications*. New Delhi, Delhi, India: Springer, 2016, pp. 1269–1278.

[5] P. R. Wolf, "Surveying and mapping: History, current status, and future projections," *J. Surveying Eng.*, vol. 128, no. 3, pp. 79–107, 2002.

[6] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–27, 2021.

[7] K. Batstone, M. Oskarsson, and K. Åström, "Towards real-time time-of-arrival self-calibration using ultra-wideband anchors," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, 2017, pp. 1–8.

[8] 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 on Internet of Things*, 2018, pp. 688–693.

[9] V. Djaja-Josko and J. Kolakowski, "A new method for wireless synchronization and TDOA error reduction in UWB positioning system," in *Proc. 21st Int. Conf. Microw., Radar Wireless Commun.*, 2016, pp. 1–4.

[10] M. Hamer and R D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, pp. 22292–22304, 2018.

[11] M. Ridolfi, J. Fontaine, B. Van Herbruggen, W. Joseph, J. Hoebeke, and E. De Poorter, "UWB anchor nodes self-calibration in NLOS conditions: A machine learning and adaptive PHY error correction approach," *Wireless Netw.*, vol. 27, pp. 1–17, 2021.

[12] 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 Navigation*, 2019, pp. 1–8.

[13] Q. Shi, S. Zhao, X. Oui, 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, 2019.

[14] W. Yu, X. Zhao, and G. Sun, "Self-calibration of anchor positions for indoor localization," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2017, pp. 581–586.

[15] 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," 2020. [Online]. Available: https://arxiv.org/abs/2004.06762

[16] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007.

[17] S. Severi, G. Abreu, and D. Dardari, "A quantitative comparison of multihop localization algorithms," in *Proc. 7th Workshop Positioning, Navigation Commun.*, 2010, pp. 200–205.

[18] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp. 2652–2661.

[19] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, 2004, pp. 46–54.

[20] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Comput.*, vol. 5, no. 2, pp. 157–167, 2002.

[21] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[22] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based localization in wireless sensor network," in *Proc. IEEE Conf. Local Comput. Netw. 30th Anniversary*, 2005, pp. 513–514.

[23] Y. Wang, "Linear least squares localization in sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, pp. 1–7, 2015.

[24] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, pp. 550–560, 1997.

[25] C. Montella, "The Kalman filter and related algorithms: A literature review," Lehigh Univ., Bethlehem, PA, USA, Tech. Rep., 2011. [Online]. Available: https://www.researchgate.net/publication/236897001_The_Kalman_Filter_and_Related_Algorithms_A_Literature_Review

[26] Decawave Ltd., "Dw1000 user manual," Version 2.18, 2017. Accessed: May 1, 2023. [Online]. Available: https://www.qorvo.com/products/d/da007967

[27] IDLab, Ghent Univ., Ghent, Belgium, "IIoT testbed," 2017, Accessed: Oct. 25, 2022. [Online]. Available: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm/

[28] B. Van Herbruggen et al., "Wi-PoS: A low-cost, open source ultra-wideband (UWB) hardware platform with long range sub-GHZ backbone," *Sensors*, vol. 19, no. 7, 2019, Art. no. 1548.

[29] F. Che et al., "Feature-based generalized Gaussian distribution method for NLOS detection in ultra-wideband (UWB) indoor positioning system," *IEEE Sensors J.*, vol. 22, no. 19, pp. 18726–18739, Oct. 2022.

[30] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter, "Edge inference for UWB ranging error correction using autoencoders," *IEEE Access*, vol. 8, pp. 139143–139155, 2020.