Wireless Networks (2021) 27:3007–3023
https://doi.org/10.1007/s11276-021-02631-0

ORIGINAL PAPER

UWB anchor nodes self-calibration in NLOS conditions: a machine learning and adaptive PHY error correction approach

Matteo Ridolfi[1] · Jaron Fontaine[1] · Ben Van Herbruggen[1] · Wout Joseph[2] · Jeroen Hoebeke[1] · Eli De Poorter[1]

Accepted: 16 April 2021 / Published online: 7 May 2021

Abstract
Ultra-wideband (UWB) positioning performance is highly related to the accuracy of the coordinates of the fixed anchor nodes, which form the system infrastructure. The process of determining the position of the anchors is called calibration. In an anchor-based system, it is crucial for the fixed nodes to know their locations with the highest possible accuracy. However, in certain situations, it is almost impossible to perform the calibration manually, e.g., during emergency interventions. Moreover, calibration is always delicate and time-consuming. We designed an effortless and accurate self-calibration algorithm that does not require any manual intervention to precisely pinpoint the position of the anchors. This paper presents an innovative algorithm that combines machine learning and exploits the time resolution capabilities of UWB with adaptive physical settings to enable the automatic calibration of the fixed anchor nodes, even in realistic NLOS (non-line-of-sight) conditions. The self-calibration algorithm combines iterative gradient descent to pinpoint the positions of the anchors and uses error detection and correction from a convolutional neural network. Moreover, the algorithm can use a different set of settings for each anchor pair. This is done to ensure the most robust and accurate communication between nodes. Extensive measurements were carried out to allow anchors to estimate distances among each others. Distances were then combined and processed by the self-calibration algorithm. Experimental evaluation in two complex and large environments with many obstacles and reflections shows that accuracy reached by the algorithm is about 2.4 cm on average and 95th percentile is 5.7 cm, in best case. The results refer to the relative positions among the anchors. Results prove that in order to precisely calibrate the anchors nodes in an UWB positioning system, high correctness can be obtained by combining the accuracy of UWB together with deep learning and adaptive PHY modulation schemes.

Keywords Ultra-wideband · Self-calibration · Deep learning · Adaptive physical settings

1 Introduction

Over the last couple of decades, various solutions to track and position objects indoor have emerged. As curious as it sounds, to pinpoint mobile assets in indoor environments using most wireless technologies, the infrastructure nodes need to be localized first. Whether it is WiFi fingerprinting [1] or ultra-wideband (UWB) calibration [2], there is typically an initial phase that involves measuring some parameters, which will later be used to position other nodes. However, this calibration procedure may be expensive and time consuming or even impossible at times. To overcome these drawbacks, self-calibration algorithms have been designed. Self-calibration or auto-calibration consists in determining the exact locations of the anchor nodes that are then used by the mobile tags to pinpoint their positions. Calibrating the infrastructure nodes is the standard procedure of each positioning system that uses a fixed architecture to compute the location of mobile assets [3]. Normally, this operation involves dedicated techniques such as surveying or mapping [4], making this step expensive and tedious. Hence, the necessity to automate the calibration procedure. This is useful in various

✉ Matteo Ridolfi
matteo.ridolfi@ugent.be

[1] IDLab, Department of Information Technology, Ghent University - IMEC, 9000 Ghent, Belgium

[2] IMEC-WAVES, Ghent University, 9000 Ghent, Belgium

scenarios, not only in large scale environments but also when nodes cannot be installed permanently, e.g. historical buildings or in emergency situations.

In this paper, we focus on a specific standard, which is largely used for indoor positioning systems (IPSs): ultra-wideband. This technology has the advantage of providing accurate timing information, which is in turn used to precisely compute the Time of Flight (ToF) of UWB signals. Once the ToF is known, little needs to be done to estimate the distance between two nodes, i.e., multiplying it by the propagation speed. With a bandwidth of minimum 500 MHz, UWB is capable of providing distance estimations with cm-level accuracy, even in challenging circumstances [5]. Although UWB systems reach outstanding positioning performance in terms of accuracy [6], installation time and costs remain probably the largest limitation [7]. The plug-and-play feature that ideally every real-time system should have, is not so straightforward in this context. The infrastructure nodes in UWB systems, i.e. anchors, have to be calibrated, meaning that their location must be known with the highest possible accuracy. Calibration procedures often require special equipment (laser meters, surveyor etc.) and an accurate planning, which also means long installation time, thus large costs. However, one of the claims of the latest UWB devices available on the market [8], is the low target price per device, which can only be kept small if costs to deploy the systems are also relatively small. Therefore, the need to simplify the calibration procedure by means of the so called self-calibration techniques, becomes crucial. As a matter of fact, various techniques already exist to solve this problem, as we will cover them in the next section. Nevertheless, most of these methods are not mature enough, either because not specifically designed for UWB or due to the fact that UWB-specific characteristics are not exploited. The main contribution of this paper is a new algorithm that combines iterative gradient descent (GD) techniques to position the anchors with machine learning algorithms to identify and minimize the errors of the ranging estimations by using the full diagnostic data available for the UWB radio messages [9]. This is achieved by applying a convolutional neural network to the algorithm. Moreover, our solution implements an adaptive PHY-layer modulation to make sure that each communication happens with the best possible settings for that specific scenario. Changing modulation schemes is beneficial to make the communication among anchors as robust as possible. In addition, we compare different possibilities to improve the self-calibration by evaluating several metrics such as accuracy and convergence rate. Finally, we discuss the main implications that would derive from extending the self-calibration to a multi-hop network with dozens of nodes in large environments.

The rest of the paper is organized as follows. Section 2 provides an exhaustive overview of existing self-calibration and machine learning techniques applied to UWB as these are the most researched topics that we discuss. Section 3 proposes the innovative algorithm to improve the self-calibration procedures for any UWB IPSs. Afterwards, the algorithm is evaluated in Sects. 4 and 5, with future research directions in Sect. 6, followed by conclusions in Sect. 7.

2 Related work

We split this section in two parts to give the reader a more clear view over the state of the art of both anchor nodes auto-calibration and machine learning approaches applied to UWB IPSs, which are both well researched topics.

2.1 Self-calibration for UWB anchor nodes

In this paragraph, all the solutions are based on the same assumption, i.e. the coordinates of the anchors nodes are unknown to the system. Moreover, these solutions are all UWB specific, meaning they are designed for UWB positioning system, in contrast with other papers, with solutions that are not technology specific [10–12].

Batstone et al. [13] experimentally evaluates the use of the Random sampling paradigm (RANSAC) and minimal solvers applied to the self-calibration enigma. The advantage of their method is that a solution can be found for a small dataset and then merged sequentially to a previous solutions, leading to computationally lighter algorithms to increase performance. The authors conducted tests using Bitcraze Crazyflie quadcopter and their Loco-positioning system, which uses UWB. A total of six anchors were installed in a small room (3m $\times$ 2m $\times$ 2m). Their relative distances were measured with a laser distance meter in the 2D experiments and with a Motion Capture (MOCAP) system for the full 3D experiment. Results show that the positions of the anchors were estimated with an accuracy of around 15 cm.

In Hamer et al. [14], authors start from a set of three nodes to define a reference system. Anchors are communicating with each other in order to obtain a first estimate on their relative distances. After this initial step, their coordinates are refined using distributed gradient descent. The reported Root Mean Square Error (RMSE) is lower than ten centimeters, i.e. 97 mm. In addition, by accurately synchronizing the network, the mobile nodes can exploit the messages being exchanged by the anchors to compute their position using Time Difference of Arrival (TDOA). In this example, the room used is also relatively small with sizes of 6x7x3.5 m^3.

Synchronization is again crucial in Vashistha et al. [15], wherein anchors range with each other according to a predefined transmission scheme. In this case, their location is estimated using Differential TDOA (DTDOA). Once calibrated, anchors are then used to pinpoint the mobile nodes positions. Experiments using 2 unlocalized anchors, report RMSE from 20 cm up to 30 cm in realistic office conditions (15x15 m^2).

UWB used in combination with low-cost Inertial Measurements Unit (IMU) data, is the solution proposed by Shi et al. [16] for an easy-to-use self-calibration mechanism. This is done by applying Simultaneous localization and mapping (SLAM) technique and using Error-State Kalman Filter (ESKF) to fuse UWB and IMU inputs. The system can provide the position and six degrees of freedom tag pose. The only requirement for the correct functioning of the mechanism is to let the tag freely move in the space of interests to collect enough data for the calibration itself. Simulations reveal RMSE of around 5 cm for each space component.

The calibration of anchors is achieved in De Preter et al. [17] by moving around the mobile nodes too. After an initialization phase, the coordinates of the infrastructure nodes are refined exploiting a predefined and known path, that the tag has to follow. Achieved accuracy is less than 10 cm on average for 4 anchors in a 16x6 m^2 area, thanks to a bias correction model, which take into account other factors such as power levels and antenna orientation. However, the full model is not applicable in indoor environments unless a ground truth system is available, i.e. the authors use RTK GPS to tune the anchors coordinates.

Yu et al. [18] describes an algorithm called SELF-CAL, which is composed of two separate steps. Firstly, authors apply Markov State Transition Equation (MSTE) to calculate the state vectors of all coordinates of the anchors and secondly the position of the anchors are estimated by implementing an iterative trilateral localization technique. The error is in line with previous solutions, i.e. average error of 10 cm to self-localize 4 anchors in a 12m $\times$ 8m room.

In [19], Almanasa et al. propose a collaborative algorithm for online auto calibration to shorten the calibration procedure and increase the accuracy at the same time. A least squares estimator (LSE) is used to combine the ranges from 4 anchor nodes in LOS conditions. The calibration process is simulated and authors reported errors 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 1 we summarize and present an overview of the reported papers. In addition, we also list our solution, which differs from the others mainly for the operating conditions, i.e. NLOS.

We presented several different approaches, which attempt to answer the same question: what are the coordinates of the anchor nodes? All of the previous techniques are designed and tested with a specific localization technology, i.e. UWB. As we mentioned in the introduction, UWB is a special standard [20]. Its high temporal resolution is the key of its success in accurately determining the distance between two nodes. This is indeed what the literature we have reported in this section shows. However, none of the listed papers actually exploits any other characteristics or features, which are typical of UWB [21], rather than simply estimating distances. One solution [17] actually looks at improving ranging estimation by studying the dependency of the accuracy on received power and antenna orientation. However, exclusively looking at power levels gives still limited advantages, whereas looking at the entire signal, e.g. by studying the Channel Impulse Response (CIR), can improve significantly the performance of the calibration procedure. In this paper, we exploit many features (different PHY modulations and CIRs) to create a robust and general framework that can be used in any working situation. Moreover, in Table 1, none of the described solutions works in NLOS conditions. On the contrary, in this paper we also investigate the impact of NLOS on the overall accuracy with a large scale real life office environment.

2.2 Machine learning for UWB IPS

In this section, the presented related work focuses on machine learning applied to the UWB domain to improve accuracy. More specifically, the literature utilizes machine learning to detect NLOS / LOS conditions or to correct ranging errors.

The authors of [22–24] propose NLOS / LOS detection using neural networks, support-vector machines and decision trees. The proposed machine learning algorithms perform well, with detection accuracies between 87% and 100%. This detection can increase the precision of indoor positioning systems when sufficient LOS exists between tag and certain anchors and by omitting ranges between tag and NLOS anchors. However, in complex environments often most signals are NLOS, leaving no clear LOS signals for further positioning calculations. This marks the desire to correct ranging errors in complex indoor environments with NLOS signal characteristics.

In [25], the authors correct ranging errors that are induced by different antenna orientations. The proposed solution applies machine learning, specifically a deep neural network, on a dataset that was captured in a indoor and outdoor LOS environment. The authors achieved < 20 mm accuracy after correction by the deep neural network on a dataset with a maximum error of 100 mm.

Table 1 Summary of self-calibration solutions for UWB anchor nodes

Solution	Type	Conditions	Algorithm	Anchors	Hop	Size	Average accuracy
Batstone et al. [13]	Experimental	LOS	RANSAC and minimal solvers	6	Single	3 m × 2 m × 2 m	15 cm
Hamer et al. [14]	Experimental	LOS	Distributed gradient descent	8	Single	6m × 7m × 3.5m	9.7 cm
Vashistha et al. [15]	Experimental	LOS	DTDOA	2	Single	15m × 15m	30 cm
Shi et al. [16]	Simulation	LOS	SLAM and ESKF	5	Single	20m × 20m × 15m	15 cm
De Preter et al. [17]	Experimental	LOS	MLE	4	Single	16m × 6m	10 cm
Yu et al. [18]	Experimental	LOS	SELF-CAL	4	Single	12m × 8m	10 cm
Almanasa et al. [19]	Simulation	LOS	LSE	4	Single	1 m^2 and 144 m^2	4 cm and 135 cm
Our solution	**Experimental**	**LOS/ NLOS**	**GD with UWB optimizations**	**4**	**Single**	**10m × 4m**	**2.4 cm**

In [26] and [27], machine learning for NLOS ranging correction in indoor environments is applied. Support vector machines and neural networks are used respectively to predict the errors caused by signal attenuation or multi-path reflections. The authors in these papers achieve an average error after correction of 100 mm and 179 mm, respectively. Solutions in [26] and [27] show great potential in improving indoor positioning systems. However, the errors in the collected datasets are small to start from ($<$ 900 mm). The impact and limitations of machine learning-based error correction in environments with large errors is still an open research question.

Different to existing work, with our solution we aim to explore the capabilities of convolutional neural networks to (i) improve ranging accuracy by predicting the error based on reduced CIR information (ii) target environments with high errors (up to a few meters) caused by attenuation, reflection, and multi-path effects (iii) integrate and evaluate machine learning as part of a self-calibration algorithm for UWB anchor nodes.

3 Approach

In this section, we present and discuss our UWB self-calibration algorithm. We tackle the problem by designing a gradient descent algorithm. Moreover, we apply multiple techniques in order to improve the accuracy of the self-calibration procedure. Firstly, we introduce the initialization phase and the gradient descent algorithm that are used to calibrate the anchors. This is followed by the three techniques that we applied together with the GD to further improve the accuracy, namely averaging over multiple ranges, machine learning and adaptive UWB PHY settings.

3.1 Gradient descent to compute the anchors coordinates

To initialize the anchors, we follow a similar approach to the one described in [14]. The goal of the initialization is to define a reference system in which the anchors will be placed. Therefore, we assume that no anchors have a known position and at least 3 are available, exception made for z-coordinate, which we will assume to be known. Another requirement is that all the ranges between each pair of these three anchors are available. Initialization starts by defining the origin of the reference system. At first, anchor one A_1 is assumed to be the origin (1), the second anchor A_2 defines the x-axis (2) and the third one's coordinates A_3 are calculated using the distances between the three anchors (3), wherein d_{ij} is the distance between A_i and A_j, $(i,j = 1,2,3)$.

$$(x_1, y_1, z_1) = (0, 0, 0) \tag{1}$$

$$(x_2, y_2, z_2) = (d_{21}, 0, 0) \tag{2}$$

$$(x_3, y_3, z_3) = \left(\frac{x_2^2 + \mathrm{d}_{31}^2 - \mathrm{d}_{32}^2}{2x_2}, \sqrt{\mathrm{d}_{31}^2 - x_3^2}, 0\right) \tag{3}$$

Once the reference system is defined, any new anchor is placed in it by computing its position with a least squares algorithm with 2 or more ranges, depending on how many are actually available.

After initialization, the positions of the anchors are refined and improved by applying the following algorithm based on gradient descent optimization. As in [14], the loss function *loss* is defined as the difference between the reported UWB distance d_{ij} between A_i and A_j and their calculated 2D distance.

$$loss = (d_{ij} - ||A_i - A_j||)^2 = \left(d_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)^2 \quad (4)$$

The objective is to minimize the summation of the loss function over all anchors pairs, as shown in Algorithm 1. The gradient descent consists in calculating the step size as derivative of the loss function multiplied by the learning rate. This tells the algorithm how large the step towards the minimum of the loss function should be. Eventually, in step 4 of Algorithm 1, the coordinates of the anchor are updated. This process is iterated until the algorithm converges, i.e., setting a convergence parameter or when the number of iterations is larger than one thousand, in our case.

Algorithm 1: Optimizing anchors coordinates using gradient descent

1 Define the loss function:

$$\sum_{i,j=1}^{N} loss = \sum_{i,j=1}^{N} (d_{ij} - ||P_i - P_j||)^2 = \sum_{i,j=1}^{N} \left(d_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)^2 \quad (5)$$

2 Calculate partial derivatives of loss function:

$$\frac{\delta loss}{\delta x_i} = 2 \cdot \left(d_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) \cdot -\frac{x_i - x_j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \quad (6)$$

$$\frac{\delta loss}{\delta y_i} = 2 \cdot \left(d_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) \cdot -\frac{y_i - y_j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \quad (7)$$

3 Define Step size = slope x learning rate:

$$Step\_size = \delta loss \cdot Learning\_rate \quad (8)$$

4 Calculate new coordinates as:

$$\begin{cases} x_i = x_i - Step\_size_{x_i} \\ y_i = y_i - Step\_size_{y_i} \end{cases} \quad (9)$$

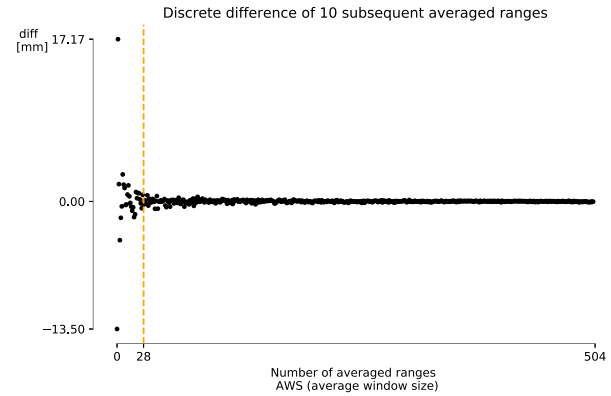
To achieve better and more robust results, we designed and implemented three different strategies to apply to the ranges that are then fed to Algorithm 1 to calculate the position of the anchors. These are using multiple ranges as opposed to a single one (averaging out), machine learning correction and prediction and finally adaptive settings for each anchor pair to ensure that the best settings are always used.

3.2 Averaging over multiple ranges

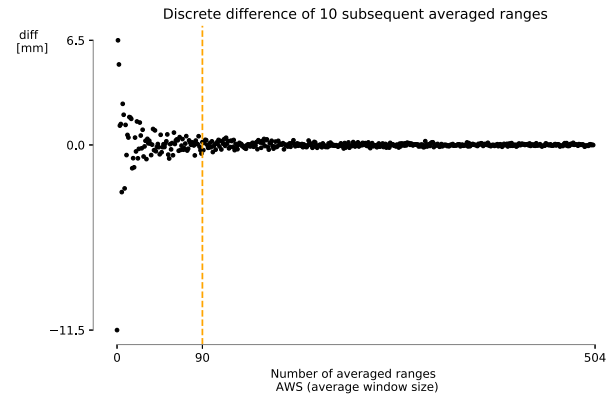
The first strategy we propose has low complexity and it considers taking the average of multiple ranges between anchors as opposed to use a single range measurement in each iteration step of Algorithm 1. Supposedly, this can mitigate the ranging error between the two nodes while maintaining low the computational cost. More importantly, it is supposed to ensure that the gradient descent converges faster. In order to choose a proper value to average the distances, we used experimental data from one of our research facility [28]. The setup and experiments will also be explained in the next section. **The goal is to find the number of averaged ranges that minimizes the fluctuations around the average range.** Basically we want to know after how many ranges, it is not beneficial anymore to average among multiple measurements. From the original set containing all the ranges between different nodes, we combine them in a way that the first element of the new set A contains the average of 1 range, the second the average of 2 ranges and so on till the end of the set (10), which is around 500 ranges per set of UWB settings (channel, bitrate, preamble length, PRF and transmission gain).

$$A(i) = \frac{\sum_{n=1}^{i} (range_n)}{i} \quad (5)$$

We study the discrete differences of 10 subsequent ranges for this new set A. We select the average window size (AWS), when the difference is lower than a specific threshold, which depends on the magnitude of the discrete difference itself. In Fig. 1, the x-axis represents the number of ranges that were used in the average calculation. This is done for two cases with the same UWB settings. The anchors are operating in LOS in Fig. 1a and in NLOS in Fig. 1b. This step is important to understand how a simple technique impacts the quality of the self-calibration procedure. The NLOS case reports more than three times the amount of ranges before the averaging actually becomes stable, i.e. 90 vs 28 of for the LOS scenario. This is explained with an initial larger spread of the measurements (Fig. 1b). Since it is important to make sure that our assumption is relevant [29], we extended the calculation to a bigger subset of measurement but this time without filtering settings and/or LOS conditions. More precisely, we look at 104 different measurements carried out for a single node. The distribution of the AWS is shown in Fig. 2, where it is confirmed than 50% of measurements are below an AWS of 51, which is a relatively high number from our experience. The 95th percentile is even larger, with 123 ranges needed to stabilize the range around the mean value. In the rest of the paper, when applying this technique we use both AWS of 51 and 123 ranges to compare the effect



(a) Discrete differences of subsequent averaged ranges when operating in LOS anchors



(b) Discrete differences of subsequent averaged ranges when operating in NLOS anchors

Fig. 1 **a** Discrete differences for LOS scenario and **b** Discrete differences for NLOS conditions. The NLOS differences have a larger spread, resulting in more than three times larger AWS than the LOS case, i.e. Average Window Size (AWS) = 90 (NLOS) and AWS = 28 (LOS)

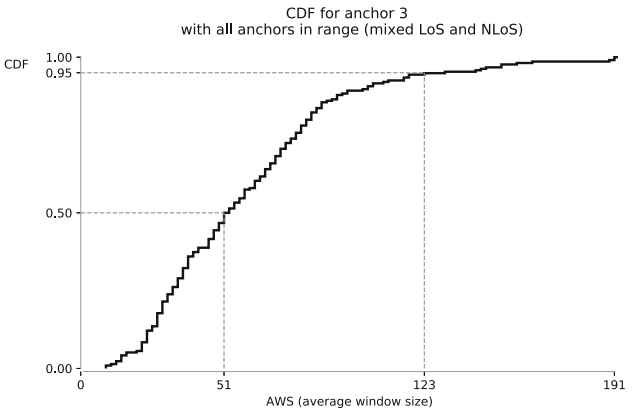


Fig. 2 Cumulative distribution function of the AWS (average window size) indexes for 104 experiments on a single anchor with different settings and conditions

of a larger number of ranges in both LOS and NLOS conditions. While using multiple averaged ranges should ensure a faster convergence rate of the optimization algorithm, it also implicates higher UWB latency since more ranges are required to be used as input of Algorithm 1. In fact, a single range is typically executed within a few ms, depending on the PHY settings and modulation scheme. Therefore, using 51 ranges can take up to 250 ms or even more per anchor, which can be a long time in some scenarios, for example when steering an autonomous flying drone [30].

In Sect. 4 we will evaluate the possible benefits of using 51 ranges instead of a single one in the self-calibration procedure.

3.3 Machine learning

In the second strategy, we apply the gradient descent algorithm to ranges (obtained from ToF) that are firstly processed by our machine learning techniques. This results in a higher complexity, although with its small footprint (<570000 parameters) it should still remain possible to implement the model on embedded devices [31]. First, the ranges are corrected using a trained model specialized for this task. Second, these improved ranges are used in conjunction with Algorithm 1.

The ranging correction model is a convolutional neural network (CNN) and is obtained by the following steps:.

1. The architecture of the CNN is illustrated in Fig. 3. The model incorporates three convolutional layers followed by 5 fully connected layers and one output layer. This design enables automatic feature extraction at the upper layers and will finally result in a single output

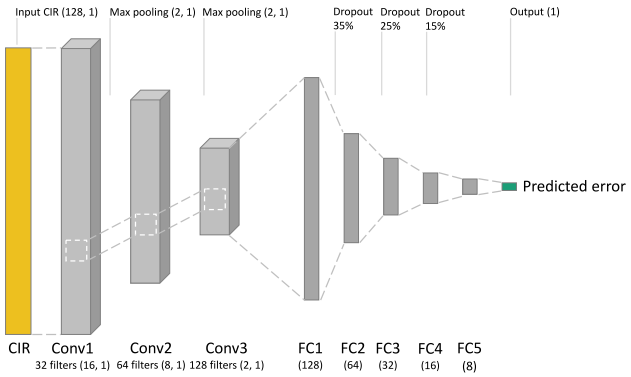


Fig. 3 Architectural overview of convolutional neural network indicating the machine learning steps from Channel Impulse Response (CIR) as input data through hidden layers until the output achieves a predicted error. The hidden layers include three different Convolutional layers (Conv), used for spatial feature extraction, and five fully connected layers (FC)

neuron. The output value is an estimation of the ranging error. At the opposite side of the neural network, the input layers expects the CIR. Dropout layers are added in the fully connected layers to prevent overfitting of the model. Overfitting can cause the model to only perform well in the environment on which it was trained. This undesirable behaviour leads to inadequate generalization towards other environments. A rectified linear unit (ReLU) activation function, 1D max-pooling and kernel regularizers are used, which produced the most accurate model in our use-case. Finally, this architecture results in a model with 568929 trainable parameters.

2. The model is trained using 134 000 CIR samples, from 11 positions in an office environment, using anchors A5 to A15 as it can be seen in Fig. 7b. This excludes the positions that are used as validation data in the Sect. 5, which are anchors A1 to A4. The complex valued CIR is first cut into 128 values around the first path index. This part of the CIR contains the most valuable information for ranging error correction. Next, the absolute value of the complex values are extracted to end up with the received signal strength of the CIR. These values are provided to the model together with the ground truth error forming a supervised dataset. The Adam optimizer trains the model with a learning rate of 0.01, a mean absolute error loss function and a batch size of 256. TensorFlow with the Keras high-level API is used to implement the model [32].
3. Once the model is trained, it can easily be used to perform ranging error correction on different positions. As such, this model is further validated on different locations throughout the Officelab testbed, which is described in Sect. 4.2.

The CNN architecture expects a 1D input of 128 CIR values. A CIR logged by the Decawave chip typically counts 1016 samples. However, only a small part of this is useful for the machine learning algorithm. Particularly, we focus the algorithm to detect multi-path components, which only happen after the first path of the signal. The first path index indicates when a signal in the CIR above the noise threshold level has been detected. As illustrated in Fig. 4, we take 51 samples before and 77 after the first path index and arrive at a total length of 128 samples. This also ensures that the machine learning expects the detected first path at the same position, which is not always the case in the original CIR. This induces the first path index as a feature and allows the machine learning model to detect wrong first path estimations that lead to ranging errors."

Using this approach we primarily aim to minimize and correct the effect of NLOS communication that are happening between unlocalized anchor nodes during the self-calibration procedure.

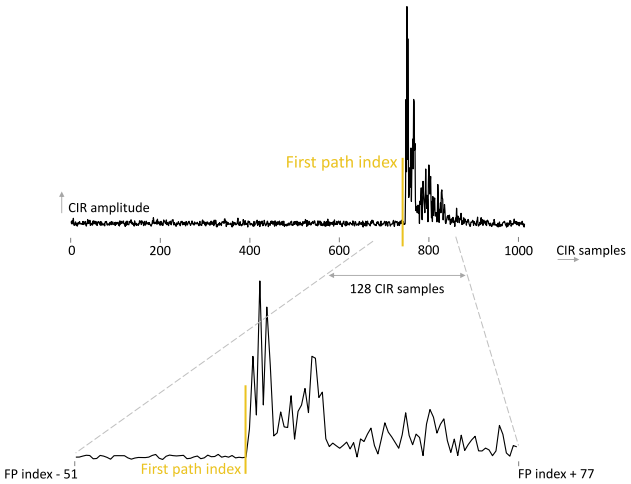


Fig. 4 The CNN input consists of 128 CIR samples, selected around the first path index

3.4 Adaptive UWB settings

UWB performs very well in open spaces and LOS conditions by estimating distances between devices with cm level accuracy and high precision [33]. However, performance can quickly deteriorate when the technology is used in heavily populated environments such as warehouses and industrial sites [34]. Nevertheless, good accuracy can be achieved in these challenging conditions too [35]. Adopting UWB in these special circumstances may require specific actions to be taken, e.g., use of specific modulations to mitigate the NLOS propagation of the UWB signal. When ranging is done between anchors, chances are high that the operating conditions are changing for each anchor-anchor set even within a single environment, e.g., two anchors on ground level will not have the same conditions as one anchor on the ceiling and one on the ground with perhaps metal racks in between them. As the causes of error in UWB systems are multiple and diverse, Table 2 reports the most common ones for which solution to mitigate the problems are referenced. Errors can be due to intrinsic UWB signal properties such as low power and large bandwidth or due to the environment in which UWB systems are deployed, e.g., presence of NLOS conditions, scalability needs and dilution of precision (DOP) problems.

Generally speaking, the parameters that play the largest roles in UWB performance are the following: channel, pulse repetition frequency (PRF), bitrate, and preamble length. Moreover, the transmit power (here adjusted by changing the transmission gain) could influence the overall self-calibration procedure. In fact, higher TX power means longer range and more anchors to communicate with.

Table 2 Most common sources of errors in UWB localization systems

Cause of error	Possible mitigation approach
Clock drift	Double sided TWR algorithm
Range bias	Model to correct bias, e.g., on RX power
NLOS conditions	Including CIR in the range calculation
Latency	TDOA algorithm
Scalability	Scheduled access MAC protocols
Dilution of precision (DOP)	Optimization of anchors placements

Although, these higher gains could also introduce errors in the signal and break the limitations imposed by the ETSI harmonised standard [20]. Changing parameters must be well designed and carefully implemented. An attempt to enable runtime adaptation of PHY settings is presented in [36]. The authors of [36] aim to improve the energy efficiency and robustness of UWB. After an extensive study to quantify the effect of each physical layer settings, they use the CIRs to measure the link quality and derive information about the deployment area, e.g., presence of destructive interference. Their final runtime solution is experimentally validated with two devices at 5 m from each other and they show improved communication performance in terms or robustness and energy efficiency. In our solution, the scheme we adopt to evaluate the impact of having different settings per anchors set, is to select the setting combination that has (i) highest reliability among all anchors, i.e. success rate and (ii) least error by simply using the available ground truth to find the best settings for each specific anchor pair. This is done because as we have described, errors come from many sources and mitigating them is not strictly the focus of this paper.

Table 3 lists an overview of the three main strategies we presented in this section. For each strategy we indicate their potential improvement in terms of accuracy, their complexity and the convergence rate when applied to self-calibration.

Table 3 Overview of the advanced techniques in terms of improvement magnitude, complexity and convergence rate

Strategy	Improvement	Complexity	Convergence rate
Averaging	Low	Low	Slow
ML	High	High	Fast
Adaptive PHY	High	Medium	Fast

4 Experimental evaluation

In this section we evaluate Algorithm 1 and the impact of the improvements on accuracy and convergence rate. Firstly, we describe the environments where we conducted the experiments and how we performed the measurements. Secondly, we discuss and analyze the findings.

4.1 Industrial IoT lab

The Industrial IoT (IIoT) lab of imec - IDLab - Ghent University [28] offers the ideal test environment (30 × 11 m^2) to replicate typical industrial conditions, including an open space with robot arms and a large area with metal racks as in a small scale warehouse, Fig. 5.

Moreover, a MOCAP system to track objects with mm accuracy is installed in the open area. This represents the perfect ground truth for our tests [37]. Therefore, we focus in IIoT on the open space area of around 5x11x3 m^3, which is the volume covered by the MOCAP. In this space, 4 anchor nodes are installed, all operating in LOS conditions.

4.2 Office lab

In the same building of the IIoT lab, the IDLab research group also offers a test office environment called OfficeLab

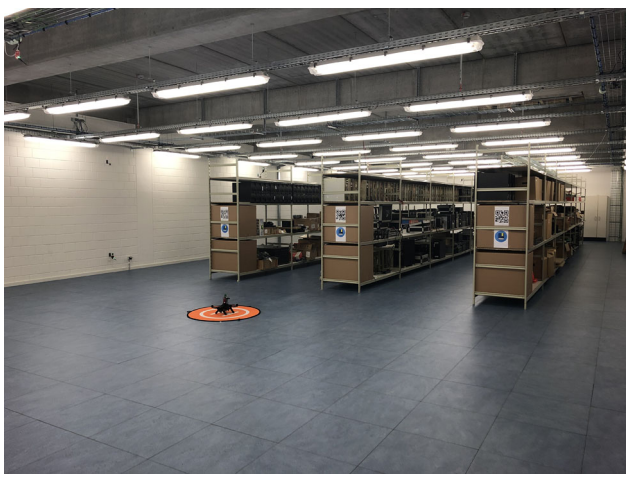


Fig. 5 Industrial IoT laboratory of imec—IDLab—Ghent University

Fig. 6 NUCs installed on the ceiling of OfficeLab imec—IDLab—Ghent University

[28], which includes 3 floors equipped with 40 Intel NUC nodes, supporting several WiFi and sensor technologies including UWB, see Fig. 6.

For our tests, we focus on a single floor for a total area of around 41 $\times$ 26 m^2 and 15 anchor nodes in corridors, meeting rooms, and offices. The anchors are placed at the same height, i.e. around 2.6 m from the floor. Both IIoT and OfficeLab setups are rendered in Fig. 7. The dimensions of the spaces are also shown and the anchors are labelled.

4.3 Methodology

We used the same hardware, software, and settings in both environments described in Sects. 4.1 and 4.2. More precisely, we used Wi-PoS: a Low-Cost UWB Hardware Platform with Long Range Sub-GHz Backbone [38]. The MAC protocol that regulates the communication and ranging between multiple devices is explained in detail in [30]. On UWB side, the ranging method used by the devices is two way ranging (TWR) [39]. In this bidirectional communication, both the transmitter and the receiver log their raw responses, which are then combined in a more readable format file, e.g., csv. Each row of the experiment file corresponds to one range between two anchors and contains all the information needed to evaluate how good the distance estimation was between the two nodes, i.e., power levels, channel impulse responses (CIRs) and so on.

The UWB settings used in the experiments are reported in Table 4. We used all the combinations and let the devices range with each other for a total of 500 range attempts per combination. For the large OfficeLab scenario, this approach resulted in more than 605 thousands ranges overall. This is due to the fact that on average one anchor could only range with 7 other anchors out of the total 15 that were installed, because of walls between different offices and rooms. For the IIoT campaign where the 4 anchors were all in LOS (225 thousands ranges collected), the only errors are due to software issues that sometimes occur, e.g. a sub-GHz beacon may get lost resulting, in synchronization problems and loss of a superframe [30].

The next section compares the results of the self-calibration algorithm with and without the improvements described in Sect. 5. The following results will be analyzed and compared in terms of mean absolute error (MAE) in relative distances among the anchors and convergence time:

- Gradient Descent with 'raw' single UWB range per iteration (GD)
- Gradient descent with averaged multiple UWB ranges (Sect. 3.2) (GD + AVG)
- Gradient descent with ML correction (Sect. 3.3) (GD + ML)
- Gradient descent with optimized UWB settings (Sect. 3.4) (GD + ADP)

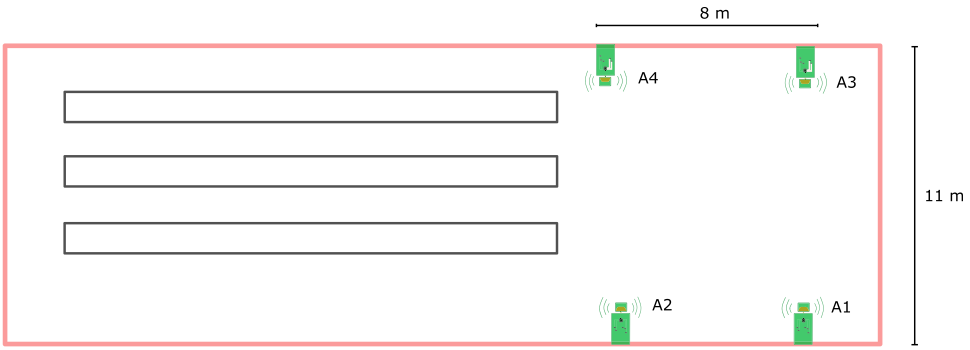
Initially, we select a single setting to analyze the data:

- Channel: 3
- Bitrate: 110 kbps
- Preamble: 4096 symbols
- Pulse Repetition Frequency (PRF): 64 MHz
- TX gain: Default (11.5 dB)

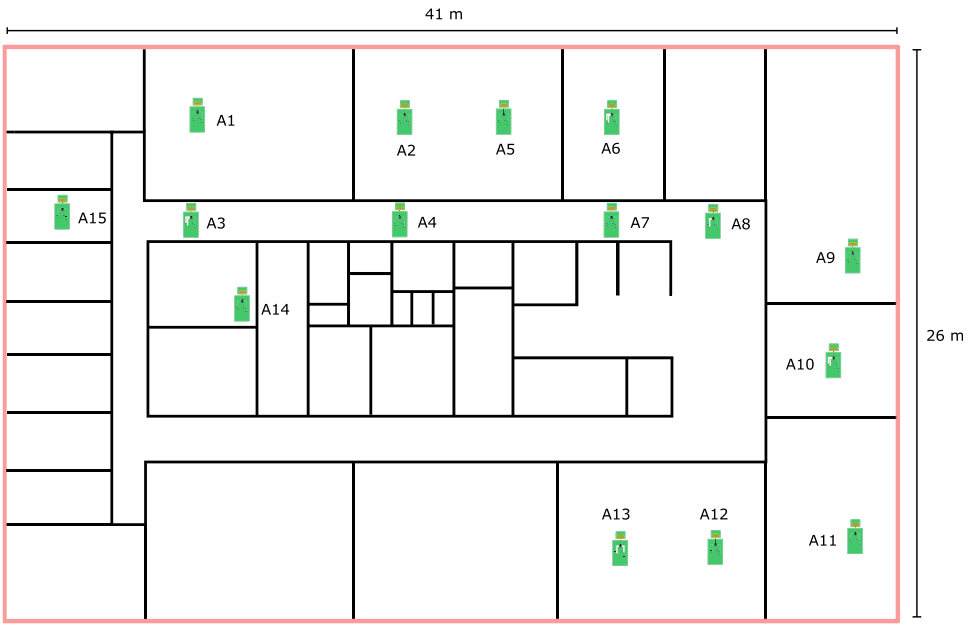
We choose these settings as these resulted in the most complete and reliable set with 100 % success rate for each anchor, i.e., no errors were found for this combination in both campaigns in IIoT and OfficeLab. Eventually, each pair of anchor will have slightly different settings since they can be optimized depending on the operating conditions, e.g., LOS or NLOS.

5 Results

Similar to the approach used by the authors in Batstone *et al.* [13], we first evaluate our self-calibration algorithm in the IIoT room with the MOCAP system and then we move the the more complex environment, i.e., the OfficeLab. We do so in order to evaluate first the performance in optimal conditions, i.e. LOS, and to have the best possible



(a) Experiment setup with 4 anchor nodes (A1 to A4) scattered over the empty space of the IIoT lab at imec - IDLab - Ghent University



(b) Experiment setup with 15 anchor nodes (A1 to A15) scattered over the 9th floor of the OfficeLab at imec - IDLab - Ghent University

Fig. 7 Test environments for the self-calibration measurements, with a controlled environment with LOS communication **a** and a large office space **b** with many anchors in mixed LOS and NLOS conditions

Table 4 UWB settings used throughout the tests to collect data for the self-calibration algorithm

Setting	Values
Channel	3, 5, 7
Bitrate	110 kbps and 6.8 Mbps
Preamble length	128, 1024 and 4096 symbols
Pulse Repetition Frequency (PRF)	16 and 64 MHz
TX gain	Default Decawave [40] and 0 dB gain

accuracy thanks to the camera of the MOCAP ground truth. Afterwards, the same evaluation will be carried out for the OfficeLab tests.

5.1 IIoT results

As explained in Sect. 4.1, all four anchors operate in LOS conditions and their exact location is computed with a very accurate MOCAP system [37].

In Fig. 8, the distributions of the relative distance errors between each pair of anchors are shown. We listed a total of 9 combinations from the 4 approaches listed in Sect. 4.3:

- standard gradient descent (GD)
- averaging (AVG)
- machine learning (ML)
- adaptive settings (ADP)

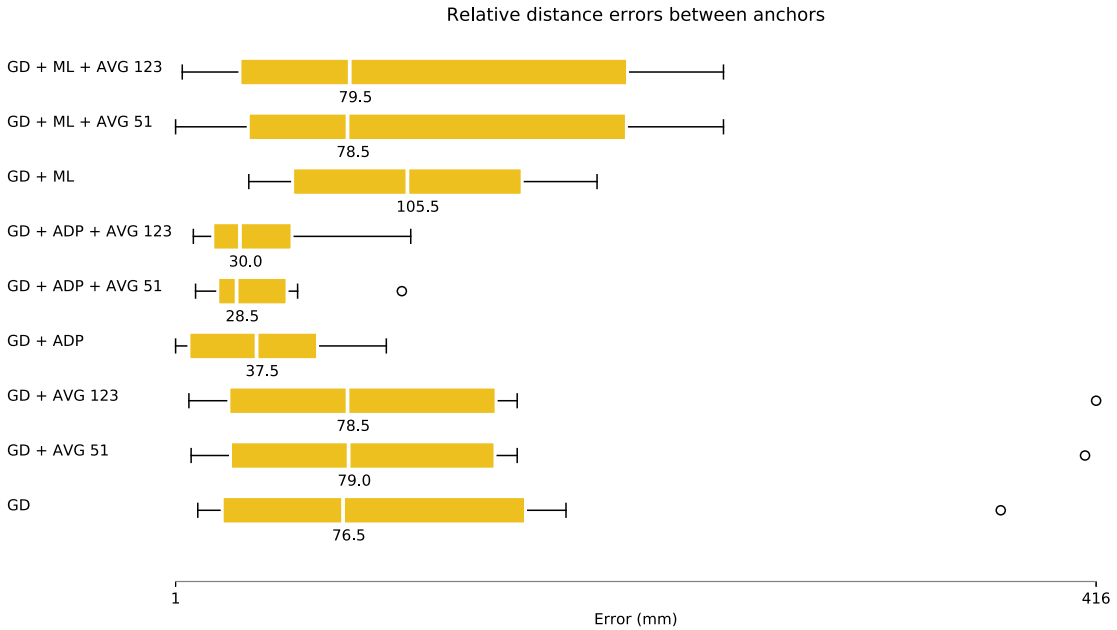


Fig. 8 Boxplot comparing self-calibration accuracy for different algorithmic combinations (GD: Gradient Descent, AVG: Averaging, ADP: Adaptive settings, ML: Machine Learning). In the **IIoT lab**, gradient descent and adaptive settings (GD + ADP) is the most accurate combination

Moreover, the averaging is done for two different AWS of 51 and 123 (Sect. 3.2), respectively. In addition, we combined multiple improvements to evaluate which one has the best accuracy overall. We start from the basic gradient descent (GD), which uses a single 'raw' range estimation per anchor pair and this resulted in a median accuracy of 76.5 mm (MAE of 121.2 mm) with a relatively large interquartile range (IQR). When averaging over multiple ranges (51 or 123) as opposed to a single one, the overall accuracy does not improve as it can be seen for 'GD + AVG 51' and 'GD + AVG 123'. The difference is really minimal with the median values differing within 3 mm from each other. This is expected as in LOS scenarios the fluctuations of ranges are limited, e.g., less outliers in the distance calculations than in NLOS conditions. Moreover, AVG 51 and 123 do not differ a lot, which is again in line with the environment since these numbers were actually modeled on mixed operating conditions, which is not the case for IIoT lab. We obtained the largest gain in accuracy by using different settings for each anchor pair (GD + ADP in Fig. 8). This resulted in a ranging error of 37.5 mm with lower spread than the GD option and MAE of 40.4 mm. That is a 67% improvement to the baseline, i.e., GD case in Fig. 8. In the GD + ADP case, if adaptive settings are used and the self-calibration is performed with the average of multiple ranges, the median error decreases of 9 and 7.5 mm for the 51 AWS and 123 cases respectively. This can be explained because the adaptive settings had higher fluctuations (larger standard deviation) than the standard setting used for GD approach. Thus, averaging out multiple measurements results in increased accuracy when we used adaptive PHY settings. This is not the case for the last improvements we applied to the IIoT anchors, i.e., machine learning. It can be seen that feeding the GD algorithm with a single corrected range (the result of machine learning correction) as opposed to the 'raw' single range (GD), worsens the average ranging accuracy (105.5 mm median error, Fig. 8). As for the adaptive case, when considering the average of multiple corrected ranges, the median error decrease, even though the mean absolute error (MAE) slightly increases. However, both GD + ML + AVG cases are comparable to the GD + AVG ones, which means that in this LOS conditions the benefits of correcting the ranges is very limited. The machine learning model was trained on the OfficeLab (LOS/NLOS) environment and this could play the largest role on the complete LOS scenario such as IIoT. Moreover, in LOS environments, there is generally less to correct (and learn), mainly because the first path index (see Fig. 4) is correctly estimated. Therefore, in LOS conditions, correction is not the best choice, especially when the model is trained on a different environment. This is also the reason that we do not show the last possible combination, i.e. machine learning for the adaptive settings averaged over multiple ranges. We found out that there was no significant difference in the average ranging error after correction compared to the non-corrected adaptive scenario, which is already extremely accurate (average ranging error of 47.4 mm among all four anchors).

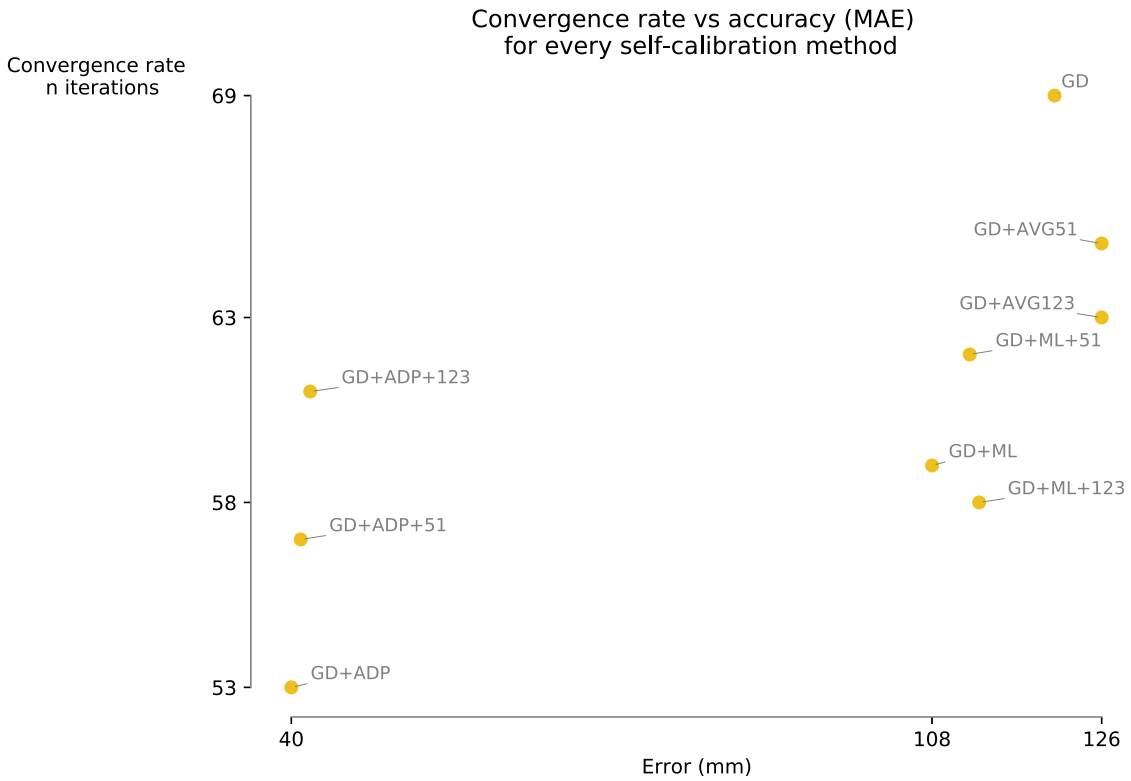


Fig. 9 Comparing convergence rate and Mean Absolute Error (MAE). For the **LOS IIoT** tests, the lowest convergence rate (GD + ADP) is also the most accurate method

In Fig. 9, we report the convergence rate of the algorithm as function of the accuracy (MAE), for each combination we tested. Interestingly, averaging over multiple ranges does not necessarily mean lower convergence rate. The GD + ADP combination, i.e., the most accurate in terms of mean absolute error converged after 53 iterations, whilst the averaged versions required almost 10 more iterations when AWS was 123. The opposite is true in the simple GD scenario. In this case, averaging 51 ranges reduced the number of iterations of the self-calibration algorithm and even more when an AWS of 123 was used. The last case with ML corrected ranges had an improvement in terms of convergence rate with AWS of 123 but the number of iterations was larger when using the average of 51 ranges. Overall, averaging never benefited the MAE of the solution, it always is slightly lower regarding the accuracy, with no significant difference between 51 and 123 AWS cases.

5.2 OfficeLab results

To better compare the results of a more complex environment such as OfficeLab with the more open space tests carried out in IIoT lab, we decided to initially limit the number of anchors to 4 but using anchors both in LOS and NLOS to test the self-calibration algorithm in mixed conditions. To be precise, the anchors that are used for this single-hop evaluation in the OfficeLab are anchors 1, 2, 3 and 4 of Fig. 7b. Differently from the previous case, in this tests the ground truth was obtained using a laser meter, which still has mm accuracy although this is more sensitive to human errors than the MOCAP system.

Figure 10 shows the distribution of the relative distance errors between anchors for each approach. In these mixed LOS/NLOS condition the baseline error is around 10 cm larger than for the LOS IIoT configuration, with median value for the GD case of 172.5 mm (MAE of 188.3 mm). As for the IIoT case, averaging 51 or 123 ranges does not improve accuracy, it makes the self-calibration less accurate, which we were not expecting the OfficeLab is much more complex environment than the IIoT. However, using the average multiple ranges improves the algorithm when adaptive PHY settings are used (GD + ADP) with the best median error of 76.5 mm and MAE 113.1 mm in the GD + ADP + AVG 123 case. Nevertheless, the corrected ranges with machine learning using adaptive settings (GD + ML + ADP) score the best calibration results with a median error of only 11 mm and MAE of 24.2 mm, which is an improvement of 87% compared to the GD case. When introducing NLOS ranges in the process, the ability to predict and correct erroneous ones makes the system more accurate than in any other case. In fact, GD + ML + ADP

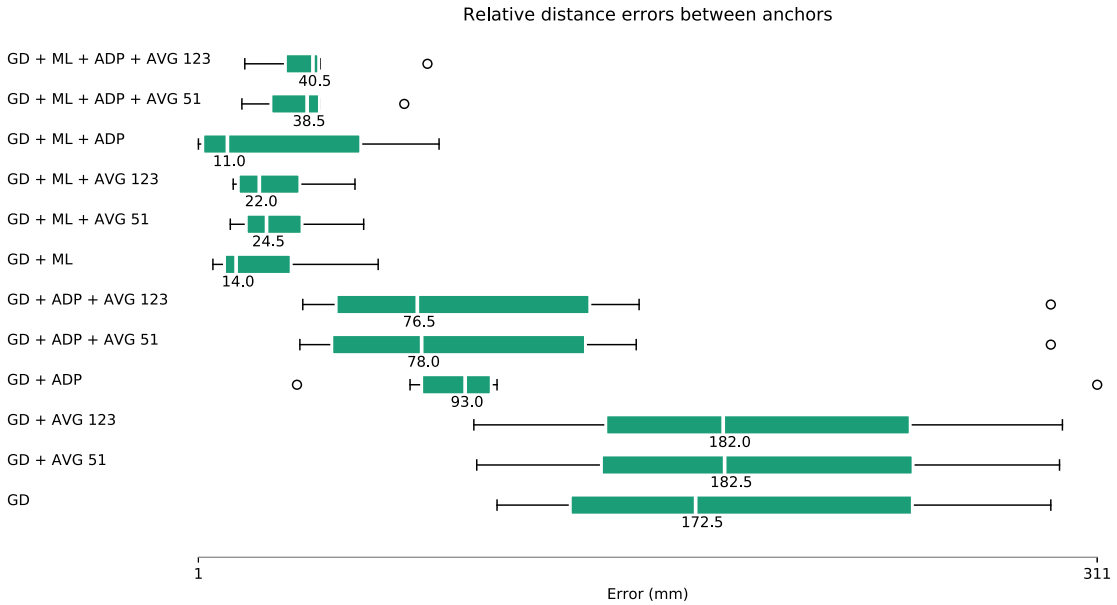


Fig. 10 Boxplot comparing self-calibration accuracy for different algorithmic combinations (GD: Gradient Descent, AVG: Averaging, ADP: Adaptive settings, ML: Machine Learning). In the **OfficeLab**, gradient descent and machine learning (GD + ML) is the most accurate combination

outperforms the GD + ADP best case of the IIoT lab in Fig. 8 by 40%. In the OfficeLab scenario, the averaged version of the GD + ML + ADP approach decreases the spread of the error while increasing the median errors. The same trend occurs when using averaged range for the pure machine learning case (GD + ML), which is again good in terms of accuracy with 14 mm median fault. We also noticed that in this case, the number of averaged ranges had a limited impact, although an AWS of 123 ranges was almost always better than AWS of 51 in terms of median error, which was expected as the choice of 123 was taken by investigating at similar LOS/NLOS mixed conditions.

Figure 11 shows each tested combination in terms of mean absolute error and number of iterations (convergence rate). It is clear that the most accurate solutions are those that use corrected ranges of the machine learning algorithm. Moreover, most accurate also means more iterations needed, i.e., more than 62. The gradient descent using adaptive settings (GD + ADP) is also exceptionally high in iterations (80), which are significantly lower when using the average of 51 or 123 ranges with around 60 iterations. However, the averaging does not improve accuracy nor convergence rate in the 'raw' GD form, which is also the fastest solution to converge with only 36 iterations, as opposed to the 51 and 61 iterations of the correspondent averaged versions. If we compare the OfficeLab results with the previous findings of the IIoT tests (Fig. 9) in terms of absolute figures, the maximum number of iterations in IIoT is 69 while for OfficeLab this goes up to 80 iterations. However, for the best combination in each environment, the OfficeLab is then faster to converge with only 36 iterations needed versus 53 in the IIoT case. Interestingly, the trend of convergence vs accuracy is the exactly the opposite in the two scenarios. For the IIoT results, the best accuracy has also the fastest convergence rate. This behaviour can be attributed to the characteristics operating conditions in which tests are carried out. It seems that the self-calibration requires more effort to converge as the environments gets more challenging. However, the improvement with respect to the baseline (GD in our cases) is bigger than in 'ideal' LOS conditions. Moreover, we know that UWB is quite mature and reliable in guaranteeing high level accuracy in clear LOS and this could explain the gap with the NLOS algorithm performance.

We can conclude that using machine learning to correct ranges in mixed LOS/NLOS conditions drastically improve the accuracy of our self-calibration algorithm by an astonishing 87%, with the disadvantage of requiring more iterations to achieve the optimal solution. GD + ML lowers the MAE from 188.3 mm of the GD case to only 24.2 mm but it took double the number of iterations to converge. Whenever accuracy is the main requirement, machine learning is recommended. On the opposite, if latency and convergence speed are essential, simple gradient descent can provide solutions with half computational load.

6 Future work

As we described in Table 1, existing papers focus only on single hop with a limited number of anchors. However, in realistic conditions the environments in which UWB operates are more complex and diverse. Extending single hop scenarios to multi-hop opens new research questions and future studies to scale up an important mechanism such as self-calibration, which we have shown to be be very accurate, even in real life conditions. However, scaling-up our gradient descent approach may be computationally expensive and the algorithm itself could be stuck at a local minimum. Therefore, further research on extending this solution is needed. In a multi-hop situation, it is crucial to optimize the deployment of the anchors to endure the lowest possible DOP. Moreover, other strategies can also improve the process, e.g. increasing the range by allowing higher transmission gains, could ensure the ability to reach a higher number of anchor thus increasing the chances to have better anchor topology for the self-calibration algorithm. Most importantly, strategies to mitigate the propagation of the error between hops become crucial.

Alongside scaling up self-calibration algorithms, the importance of making these solutions as less application-dependent as possible, is also an important aspect. Many applications are constantly being defined and as such UWB must be able to cope with numerous requirements, e.g., energy efficiency, accuracy and scalability. Self-calibration plays an important role in making UWB more readily available to everybody's life. However, attention must be paid to ensure a flexible and adaptive solution, that can work in any sort of conditions. We have shown that adaptive PHY can improve the overall accuracy error. Nevertheless, a more in-depth analysis of all the implications of using different modulations is needed to understand the impact on each parameter (energy efficiency, scalability, etc.) and not only on pure accuracy.

7 Conclusions

In order to accurately localize assets in indoor environments, UWB provides high accuracy even in challenging conditions such as industrial or office spaces. However, being the majority of deployment anchor-based, the accuracy is strictly related to how accurate the anchor nodes are calibrated, i.e., how good is their location known. Moreover, in some cases such as emergency rescue activities or in historical buildings, the calibration procedure is much more complicated or even not possible. Therefore, we design a self-calibration method based on gradient descent to allow the anchors to self-calibrate their position purely

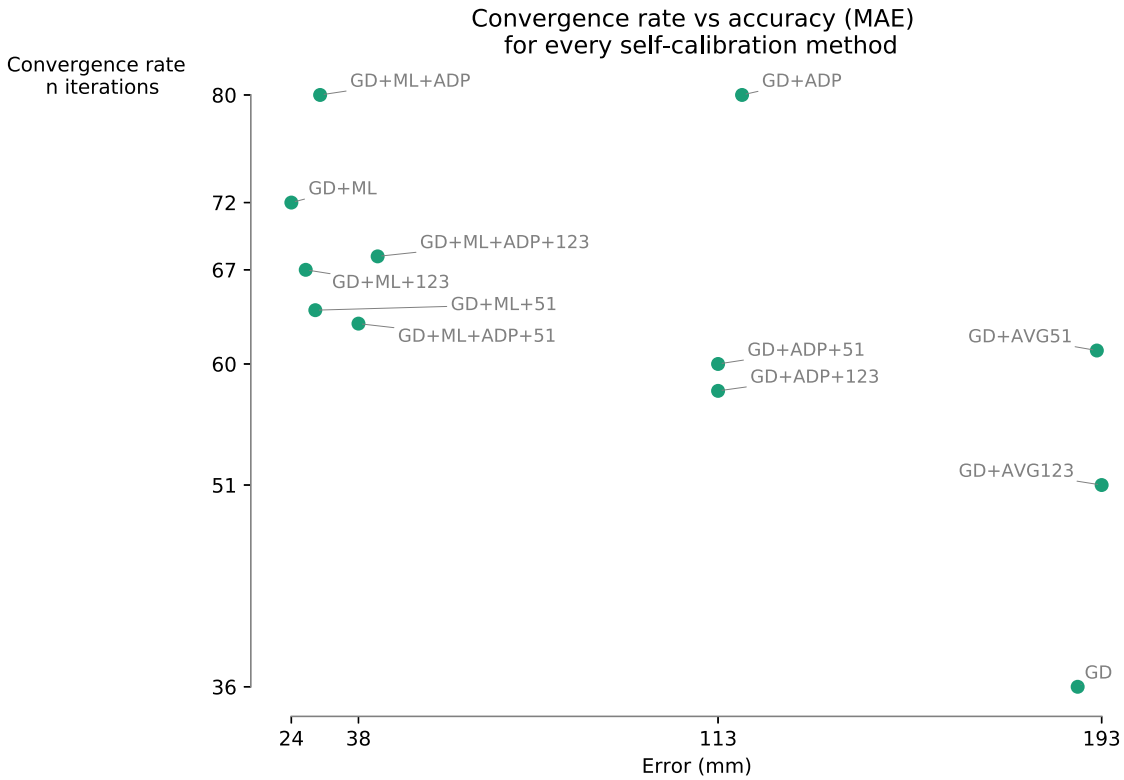


Fig. 11 Comparing convergence rate and Mean Absolute Error (MAE). For the **OfficeLab** results, most accurate solution means also slowest to converge. To halve the accuracy, the algorithm requires double the iterations

based on the UWB messages exchange among them, i.e., without need of external hardware or tools (laser meters, surveyor, or other sensors). Additionally, we exploit some of the UWB characteristics to further improve the accuracy of the automated calibration. These are (i) the use of multiple averaged ranges opposed to a single range (ii) the adaptive PHY settings that change for each communication and (iii) machine learning algorithms to study and correct the signal that propagates between anchors. We tested the algorithm in two realistic environments with a single hop setup. Both in LOS and NLOS conditions, the self-calibration algorithm can achieve very high accuracy in the order of few centimeters. We also found that the improvement techniques are crucial to ensure that obstacles and obstructions have the least effect on the procedure, especially in mixed LOS/NLOS conditions. In this regards, machine learning appears to be the most effective technique to minimize errors in NLoS conditions, with an improvement of 87% with respect to the basic gradient descent version. This, at the cost of doubling the amount of iterations that the algorithm needs to converge. Further research on scaling up self-calibration algorithms using purely UWB messages is needed to assess the accuracy of this procedure in more complex environments with tens of more anchors. This is closely related to a broader topic of automated UWB network planners for optimal installation of UWB nodes (combined with self-calibration) with energy, localization and interference constraints.

Acknowledgements This work has partially been funded by the following projects: VLAIO proeftuinproject SmartConnectivity and the imec.icon InWareDrones.

References

1. Basri, C., & El Khadimi, A. (2016). Survey on indoor localization system and recent advances of WIFI fingerprinting technique. In: *Proceedings of the 2016 5th international conference on multimedia computing and systems (ICMCS)* (IEEE), pp. 253–259.
2. Shi, G., & Ming, Y. (2016). Survey of indoor positioning systems based on ultra-wideband (UWB) technology. In: *Wireless communications, networking and applications* (pp. 1269–1278). New York: Springer
3. Xiao, J., Zhou, Z., Yi, Y., & Ni, L. M. (2016). A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys (CSUR), 49*(2), 1.
4. Wolf, P. R. (2002). wolf2002surveying. *Journal of Surveying Engineering, 128*(3), 79.
5. Ridolfi, M., Vandermeeren, S., Defraye, J., Steendam, H., Gerlo, J., De Clercq, D., et al. (2018). Edge inference for UWB ranging error correction using autoencoders. *Sensors, 18*(1), 168.
6. Witrisal, K., Hinteregger, S., Kulmer, J., Leitinger, E., & Meissner, P. (2016). High-accuracy positioning for indoor applications: RFID, UWB, 5G, and beyond. In: *Proceedings of the 2016 IEEE international conference on RFID (RFID)* (IEEE), pp. 1–7.
7. Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A., & Al-Khalifa, H. S. (2016). Ultra wideband indoor positioning technologies: analysis and recent advances. *Sensors, 16*(5), 707.
8. Ruiz, A. R. J., & Granja, F. S. (2017). Comparing ubisense, bespoon, and decawave uwb location systems: indoor performance analysis. *IEEE Transactions on Instrumentation and Measurement, 66*(8), 2106.
9. Andrews, J.R. (2003). UWB signal sources, antennas and propagation. In: *Proceedings of the 2003 IEEE topical conference on wireless communication technology* (IEEE), pp. 439–440.
10. Moses, R. L., Krishnamurthy, D., & Patterson, R. M. (2003). A self-localization method for wireless sensor networks. *EURASIP Journal on Advances in Signal Processing, 2003*(4), 839843.
11. Xu, L., Yao, L., He, J., Wang, P., Long, K., & Wang, Q. (2018). Collaborative geolocation based on imprecise initial coordinates for internet of things. *IEEE Access, 6,* 48850.
12. Gualda, D., Ureña, J., Alcalá, J., & Santos, C. (2019). Calibration of beacons for indoor environments based on a digital map and heuristic information. *Sensors, 19*(3), 670.
13. Batstone, K., Oskarsson, M., & Åström, K. (2017). Towards real-time time-of-arrival self-calibration using ultra-wideband anchors. In: *Proceedings of the 2017 international conference on indoor positioning and indoor navigation (IPIN)* (IEEE), pp. 1–8.
14. Hamer, M., & D'Andrea, R. (2018). Self-calibrating ultra-wideband network supporting multi-robot localization. *IEEE Access, 6,* 22292.
15. Vashistha, A., Gupta, A., & Law, C.L. (2018) Self calibration of the anchor nodes for UWB-IR TDOA based indoor positioning system. In: *Proceedings of the 2018 IEEE 4th world forum on internet of things (WF-IoT)* (IEEE), pp. 688–693.
16. Shi, Q., Zhao, S., Oui, X., Lu, M., & Jia, M. (2019). Anchor self-localization algorithm based on UWB ranging and inertial measurements. *Tsinghua Science and Technology, 24*(6), 728.
17. De Preter, A., Goysensn G., Anthonis, J., Swevers, J., & Pipeleers, G. (2019) Range bias modeling and autocalibration of an UWB positioning system. In: *Proceedings of the 2019 international conference on indoor positioning and indoor navigation (IPIN)* (IEEE), pp. 1–8.
18. Yu, W., Zhao, X., & Sun, G. (2017) Self-calibration of anchor positions for indoor localization. In: *Proceedings of the 2017 IEEE international conference on robotics and biomimetics (ROBIO)* (IEEE), pp. 581–586. https://doi.org/10.1109/ROBIO.2017.8324479. http://ieeexplore.ieee.org/document/8324479/.
19. Almansa, C. M., Shule, W, Queralta, J. P., & Westerlund, T. (2020). Autocalibration of a mobile UWB localization system for ad-hoc multi-robot deployments in GNSS-denied environments. arXiv preprint arXiv:2004.06762.
20. European Telecommunications Standards Institute, Short Range Devices (SRD) using Ultra Wide Band technology (UWB); Harmonised Standard covering the essential requirements of article 3.2 of the Directive 2014/53/EU; Part 2: Requirements for UWB location tracking. Standard, European Telecommunications Standards Institute (2016)
21. Muqaibel, A., Safaai-Jazi, A., Woerner, B., & Riad, S. (2002). UWB channel impulse response characterization using deconvolution techniques. In: *The 2002 45th midwest symposium on circuits and systems, 2002. MWSCAS-2002.*, (IEEE), vol. 3, pp. III–605.
22. Krishnan, S., Xenia Mendoza Santos, R., Ranier Yap, E., & Thu Zin, M. (2018). Improving UWB based indoor positioning in industrial environments through machine learning. In: *Proceedings of the 2018 15th international conference on control, automation, robotics and vision (ICARCV)*, pp. 1484–1488. https://doi.org/10.1109/ICARCV.2018.8581305.

23. Kristensen, J. B., Massanet Ginard, M., Jensen, O. K., & Shen, M. (2019). Non-line-of-sight identification for UWB indoor positioning systems using support vector machines. In: *Proceedings of the 2019 IEEE MTT-S International Wireless Symposium (IWS)*, pp. 1–3. https://doi.org/10.1109/IEEE-IWS.2019.8804072.
24. Zeng, Z., Liu, S., & Wang, L. (2019). UWB NLOS identification with feature combination selection based on genetic algorithm. In: *Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5. https://doi.org/10.1109/ICCE.2019.8662065.
25. Tiemann, J., Pillmann, J., & Wietfeld, C. (2017). Ultra-wideband antenna-induced error prediction using deep learning on channel response data. In: *Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–5. https://doi.org/10.1109/VTCSpring.2017.8108571.
26. Li, Weijie, Zhang, Tingting, & Zhang, Qinyu (2013). Experimental researches on an UWB NLOS identification method based on machine learning. In: *Proceedings of the 2013 15th IEEE international conference on communication technology*, pp. 473–477. https://doi.org/10.1109/ICCT.2013.6820422.
27. Mao, C., Lin, K., Yu, T., & Shen, Y. (2018). A probabilistic learning approach to UWB ranging error mitigation. In: *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. https://doi.org/10.1109/GLOCOM.2018.8647602.
28. Ghent university - idlab research infrastructure. https://www.ugent.be/ea/idlab/en/research/research-infrastructure
29. De Poorter, E., Van Haute, T., Laermans, E., & Moerman, I. (2017). Benchmarking of localization solutions: guidelines for the selection of evaluation points. *Ad Hoc Networks, 59,* 86.
30. Macoir, N., Bauwens, J., Jooris, B., Van Herbruggen, B., Rossey, J., Hoebeke, J., & De Poorter, E. (2019). Uwb localization with battery-powered wireless backbone for drone-based inventory management. *Sensors, 19*(3), 467.
31. Fontaine, J., Ridolfi, M., Van Herbruggen, B., Shahid, A., & De Poorter, E. (2020). Edge inference for UWB ranging error correction using autoencoders. *IEEE Access, 8,* 139143.
32. The Functional API | TensorFlow Core. https://www.tensorflow.org/guide/keras/functional
33. Malajner, M., Planinšič, P., & Gleich, D. (2015). UWB ranging accuracy. In: *Proceedings of the 2015 International Conference on Systems, Signals and Image Processing (IWSSIP)* (IEEE), pp. 61–64.
34. Silva, B., & Hancke, G. P. (2016). IR-UWB-based non-line-of-sight identification in harsh environments: principles and challenges. *IEEE Transactions on Industrial Informatics, 12*(3), 1188.
35. Silva, B., Pang, Z., Åkerberg, J., Neander, J., & Hancke, G. (2014). Experimental study of UWB-based high precision localization for industrial applications. In: *Proceedings of the 2014 IEEE International Conference on Ultra-WideBand (ICUWB)* (IEEE), pp. 280–285.
36. Großwindhager, B., Boano, C. A., Rath, M., & Römer, K. (2018). Enabling runtime adaptation of physical layer settings for dependable uwb communications. In: *Proceedings of the 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)* (IEEE), pp. 01–11.
37. Qualisys motion capture systems. http://www.qualisys.com/
38. Van Herbruggen, B., Jooris, B., Rossey, J., Ridolfi, M., Macoir, N., Van den Brande, Q., et al. (2019). Wi-PoS: a low-cost, open source ultra-wideband (UWB) hardware platform with long range sub-GHz backbone. *Sensors, 19*(7), 1548.
39. Kreiser, D., Martynenko, D., Klymenko, O., & Fischer, G. (2015). Simple and efficient localization method for IR-UWB systems based on two-way ranging. In: *Proceedings of the 2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)* (IEEE), pp. 1–4.
40. Decawave ltd. - transmit power calibration and management. https://www.decawave.com/application-notes/

Matteo Ridolfi received the M.Sc. degree in telecommunication engineering from the University of Bologna, Italy, in 2016. He is currently a Ph.D. Researcher with the IDLab Research Group, imec, Ghent University. He is the author and coauthor of various publications on Ultra-wideband localization solutions. His research focus is on indoor localization systems, mainly targeting solutions with Ultra-wideband technology. His interests include localization algorithms, scalability and accuracy analysis, self-calibrating networks, and collaborative localization.

Jaron Fontaine is a Ph.D. researcher at IDLab in Ghent University. He received his M.Sc. degree in Information Engineering Technology from Ghent University with honor in 2017. He is a recipient of the FWO-SB grant and funded by the Scientific Research Flanders (Belgium, FWO-Vlaanderen). His research interests include machine learning, big data and data analytics, wireless networks, transfer learning, semi-supervised learning, generative adversarial networks, embedded AI etc. Jaron has already published and co-authored numerous papers on these topics in journals and presented his work at conferences.

Ben Van Herbruggen was born in Antwerp in 1995. He received the M.Sc. degree in electrical engineering from Ghent University, Belgium, in July 2018. In September 2018, he started as Research Assistant with the Department of Information Technology (INTEC) at Ghent University in the IDLab research group towards a PhD degree. His scientific work is focused on accurate indoor localization system based on Ultra wideband technology and the use of energy harvesters for wireless networking.

Wout Joseph (M'05) was born in Ostend, Belgium on October 21, 1977. He received the M.Sc. degree in electrical engineering from Ghent University (Belgium) in July 2000. From September 2000 to March 2005, he was a research assistant at the Department of Information Technology (INTEC) of the same university. During this period, his scientific work was focused on electromagnetic exposure assessment. His research work dealt with measuring and modelling of electromagnetic fields around base stations for mobile communications related to the health effects of the exposure to electromagnetic radiation. This work led to a Ph.D. degree in March 2005. Since October 2007, he is a Post-Doctoral Fellow of the FWO-V (Research Foundation—Flanders). Since October 2009 he is professor in the domain of "Experimental Characterization of wireless communication systems." He is IMEC PI since 2017. His professional interests are electromagnetic field exposure assessment, propagation for wireless communication systems, antennas and calibration. Furthermore, he specializes in wireless performance analysis and Quality of Experience.

Jeroen Hoebeke is an associate professor in the Internet Technology and Data Science Lab of Ghent University and imec. He is conducting and coordinating research on wireless (IoT) connectivity, embedded communication stacks, deterministic wireless communication and wireless network management. This expertise has been applied in a variety of application domains such as logistics, Industry 4.0, building automation, healthcare and animal monitoring. He is particularly active in national funded projects as well as in defining, executing and managing such projects. He has also been involved in several EU research funded projects and is author or co-author of more than 150 publications in international journals or conference proceedings.

Eli De Poorter is associate professor at the IDLab research group from imec and Ghent University (https://idlab.technology). His team performs research on wireless communication technologies such as (indoor) localization solutions, wireless IoT solutions and machine learning for wireless systems. He performs both fundamental and applied research. For his fundamental research he is currently the coordinator of several research projects (SBO, FWO, GOA, etc.) and has over 180 publications in international journals or in the proceedings of international conferences. For his applied research, he collaborates with (Flemish) industry partners to transfer research results to industrial applications as well as to solve challenging industrial research problems. Prof. De Poorter is also co-founder of the lopos spin-off company (https://lopos.be) which offers privacy-aware UWB wearables for safety and social distancing.