IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ADVANCES ON HIGH PERFORMANCE WIRELESS NETWORKS FOR AUTOMATION AND IIOT

Received 11 February 2023, accepted 13 March 2023, date of publication 20 March 2023, date of current version 24 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259320

RESEARCH ARTICLE

FlexTDOA: Robust and Scalable Time-Difference of Arrival Localization Using Ultra-Wideband Devices

GEORGE-CRISTIAN PĂTRU[1], LAURA FLUERATORU[1,2], IULIU VASILESCU[3], DRAGOŞ NICULESCU[1], AND DANIEL ROSNER[1]

[1]Department of Computer Science, University Politehnica of Bucharest, 060042 Bucharest, Romania
[2]Electrical Engineering Unit, Tampere University, 33720 Tampere, Finland
[3]TIA RESEARCH, 820169 Tulcea, Romania

Corresponding author: George-Cristian Pătru (cristi.patru@upb.ro)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska Curie Grant, A-WEAR: A network for dynamic wearable applications with privacy constraints, under Agreement 813278; and in part by the Ministry of European Investments and Projects through the Human Capital Sectoral Operational Program 2014-2020, under Contract 62461/03.06.2022, SMIS code 153735. The work of George-Cristian Pătru was supported by NXP Romania.

ABSTRACT In this paper, we propose FlexTDOA, an indoor localization method using ultra-wideband (UWB) radios, and we demonstrate its performance in a functional system. Our method uses time-difference of arrival (TDOA) localization so that the user device remains passive and is able to compute its location simply by listening to the communication between the fixed anchors, ensuring the scalability of the system. The anchors communicate using a custom and flexible time-division multiple-access (TDMA) scheme in which time is divided in slots. In each time slot, one anchor interrogates one or more anchors which respond in the same slot. The anchors do not need to have their clocks synchronized. We implemented FlexTDOA on in-house designed hardware using a commercial UWB module. We evaluate the localization accuracy of FlexTDOA with different system parameters such as the number of responses, the order of responses, and the number of anchors. We simulate and evaluate the effect of the physical speed of the tag on the choice of optimum system parameters. We also compare FlexTDOA against the classic TDOA approach and range-based localization in a deployment of ten anchors and one tag, both with and without obstructions. Results show that FlexTDOA achieves the highest localization accuracy in most of the scenarios, with up to 38% reduction in the localization error compared to the classic approach.

INDEX TERMS Ultra-wideband, indoor localization, time-difference of arrival, Internet of Things (IoT).

I. INTRODUCTION

Ultra-wideband (UWB) technology has experienced a revival during the past few years, mainly for its high-accuracy ranging and localization capabilities. It is estimated that more than 1 billion UWB devices will be shipped by 2025, and that over the next 5–10 years all smartphones will have UWB capabilities [1]. With the growing number of users, localization systems will face high scalability requirements to satisfy network demands with acceptable location update rates [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Hans-Peter Bernhard.

UWB-based localization systems usually consist of a mobile node that needs to be localized, called *tag*, and several fixed nodes with known locations, called *anchors*, which communicate with the tag and aid the localization process. Range-based localization is arguably the most popular localization technique since it provides the highest accuracy [3]. In range-based localization, the location is computed based on distances between each anchor and the tag using multilateration. To avoid synchronizing the transmitter (TX) and the receiver (RX) [4], [5], at least two message exchanges between the tag and each anchor are needed to compute *one* distance, technique known as TWR. Because of the high

number of messages, range-based localization (or multilateration) scales poorly with the growing number of tags and anchors.

Time-difference of arrival (TDOA) is an alternative localization method which uses the difference between the arrival times of two packets (usually, exchanged by the tag and two anchors) [6]. For each difference, the tag can be located on a hyperbola focused at the anchors (in a 2D coordinate system). By computing the TDOA for more anchor pairs, the tag's location can be found at the intersection of multiple hyperbolas [3]. One TDOA variant frequently called downlink (DL) TDOA [7] has gained popularity for its scalability properties. In this setup, only the anchors transmit periodic messages, while the tag records their arrival times and localizes itself. In DL TDOA, the tag can remain *passive*, i.e., does not need to transmit any uplink messages. Using passive tags, since there is no need for the tags to share the channel, DL TDOA can scale to an unlimited number of users [8], [9], [10], [11].

The main drawback of TDOA localization is that the anchors need to be synchronized, usually by estimating the clock offsets of each anchor relative to a reference clock [11]. A convenient way to obtain the reference clock is to designate a reference anchor that periodically broadcasts a synchronization beacon [12]. The rest of the anchors respond to this beacon and track their clock offsets with respect to the reference anchor. The disadvantage of this approach is that, if the link between the reference anchor and the tag is obstructed, the tag will receive the synchronization beacon with a delay and *all* subsequent TDOAs pertaining to that beacon will be corrupted. This method is therefore not reliable in the presence of obstructions.

In this paper, we propose an alternative TDMA scheduling scheme for TDOA localization called FlexTDOA. In FlexTDOA, there is no single reference anchor. Instead, all the anchors in the system can be configured to take turns in transmitting the synchronization beacon. Similarly, the order of the anchors that respond to the beacon changes in a round-robin manner. Therefore, depending on the needs of the system, less anchors than the maximum available can respond to a beacon, which reduces clock drift errors caused by the delay between the first and the last response while allowing all anchors to participate in the localization process. FlexTDOA therefore exploits the full channel diversity of the environment, is not subject to single-link failures, and can maintain small errors even in large networks.

We implemented FlexTDOA in a localization system based on the Qorvo DW3000 UWB chipset [13]. We compare the proposed system against the classic TDOA approach and the standard range-based multilateration algorithm in a deployment of ten anchors and one tag in an office environment, in both line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. We also evaluate the impact of several parameters on the ranging and localization accuracy, such as: the number of responses for each synchronization beacon for different system update rates, the number of anchors in the system, and the impact of changing the initiator and/or the order of responses.

Besides providing an increased robustness to harsh conditions, FlexTDOA distinguishes itself from previous approaches by not using tracking filters to estimate clock parameters, such as in [10], [11]. Instead, each receiver uses carrier frequency offset (CFO) estimation to locally correct the relative time skew between its clock and the transmitter's clock. This allows administrators to easily add new anchors to the system without increasing its complexity. Moreover, the flexible scheduling scheme facilitates the deployment of the localization system in larger spaces where the anchors might be split over multiple rooms and only a subset of anchors should respond to an initiating message. The proposed scheme also preserves the location privacy of the user, since the user device remains passive in the localization process [7]. Therefore, no information of the user presence or their location is leaked to the infrastructure.

We make all the measurements available open-source[1], in order to facilitate the evaluation of other network parameters than those covered in the paper. The dataset can also be used to prototype NLOS error-mitigation techniques in the future.

The rest of the paper is organized as follows. Section III introduces the basics of the localization methods used throughout the paper and the particularities of the proposed scheduling scheme. Section IV presents the setup of the localization system, such as the hardware used for evaluation, the environment, and the anchor placement. Section V evaluates the most important parameters of the localization system, while Section VI compares range-based localization, the classic TDOA, and FlexTDOA. Section II reviews related work and highlights differences from previous approaches. Finally, Section VII sums up our contributions.

II. RELATED WORK

In the following, we will review the most important previous works on TDOA localization, with a focus on *DL* TDOA schemes which offer the best multi-user scalability.

A. SCALABLE UWB LOCALIZATION

In [10], a DL TDOA localization system which implements a clock synchronization protocol with a reference anchor is proposed. The authors mention that the system does not scale to large anchor networks. In a setup of eight anchors, the system obtained a 2D localization root-mean squared error (RMSE) of 14 cm and a 3D RMSE of 28 cm. In a comparable setup of seven anchors in LOS, FlexTDOA obtained a 2D RMSE of 16.2 cm and a 3D RMSE of 23.5 cm (averaged over all considered locations), so comparable to the ones in [10]. In [11], a similar approach to [10] is proposed, in which the pairwise clock error is tracked using a Kalman filter that can handle fluctuating reception periods.

[1] https://zenodo.org/record/7619764

In [14], the authors propose a DL TDOA scheme in which the anchors respond only to the previously-transmitted message instead of responding to a single synchronization beacon, as in our case. The mean and maximum localization errors obtained with a configuration of four anchors were 31 cm and 81 cm, respectively.

Although named concurrent ranging, the works in [15], [16] essentially implement the classic DL TDOA scheme. However, here the focus is on processing all the responses within a single reception period by exploiting the multipath information from the channel impulse response of the signal.

In [17], a TDOA localization system implemented using UWB devices called ATLAS is introduced. Although the system also uses an initiating anchor for synchronization, it is not clear if the tag is completely passive since it is mentioned that only whitelisted tags are processed at a localization server. Therefore, ATLAS relies on centralized processing to correct the TDOAs, whereas in FlexTDOA the localization is offloaded to the user device, preserving its location privacy. Iterations of ATLAS have been introduced in [18], [19]; however, in these works, the tag is active.

A localization system named VULoc that follows the principles of DL TDOA has recently been proposed in [20]. However, in VULoc, the initiator sends one additional message after all of the anchors have responded, which means that the tag has to listen to one extra message compared to FlexTDOA. Perhaps the most significant difference between VULoc and FlexTDOA is that we also propose a flexible, highly-configurable TDMA scheme for anchor transmissions, whereas in [20] it is mentioned that VULoc does not need a scheduling protocol because tags are passive. However, we argue that in large-scale building deployments there is a need to easily add or remove anchors from the system and to schedule their transmissions. As a result, we could easily evaluate the performance of FlexTDOA with up to ten anchors, whereas VULoc was evaluated only with five anchors. While VULoc also uses changing initiators, it does not evaluate the impact brought by the added channel diversity, as we do. In a laboratory setting, VULoc obtained a median error of 15.5 cm and a 90 % error of 23.6 cm. In a similar experiment in which the tag was placed in the center of the room, FlexTDOA obtained a median error of 15.4 cm and a 90 % error of 22.2 cm, so comparable to the errors achieved by VULoc.

Although it does not implement a DL TDOA scheme, the work in [21] proposes a scalable UL TDOA localization scheme called TALLA. The high-precision synchronization necessary for TDOA localization is maintained by a server, which can compute the clock parameters to synchronize to any reference anchor in the system. While this approach provides great flexibility in the case of large-scale deployments, it poses more privacy concerns since the network has information about the location of all tags in the system. In our approach, since the tag localizes itself, the network does not have any information about the users' locations.

Another important contribution of our work is that we evaluated the performance of a DL TDOA localization system in NLOS conditions *experimentally*, since most previous works either consider only LOS scenarios or base their observations on simulated data. In [20], the proposed TDOA system is also evaluated in NLOS conditions and an anchor selection method based on an empirically-chosen confidence parameter is proposed. However, in [20], the purpose of the evaluation is to demonstrate the effectiveness of the anchor selection method, whereas we quantified the effect of the channel diversity brought by the scheduling method proposed in FlexTDOA. In [22], the authors propose a sensor placement strategy for cluttered environments that is validated through experimental data. A UL TDOA localization system that takes into account NLOS conditions has been proposed and evaluated experimentally in [23]. In [24], the authors propose an algorithm to select anchor pairs in a UL TDOA by taking into account errors caused by NLOS propagation.

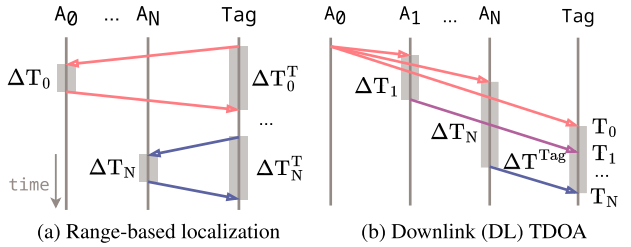
In [25], the authors propose a framework designed for scalable indoor localization and implement it using UWB radios. The work in [25] is focused on the software, which enables cooperation between both fixed and mobile nodes in order to achieve seamless localization. In comparison, we focus on the specific TDOA localization algorithm, which enables scalable and accurate localization.

B. CLOCK OFFSET CORRECTION

In our TDOA scheme, we avoid tracking the clock parameters using Kalman filters like in previous works [10], [11]. Instead, we correct the relative clock offset between two devices directly at the receiver using the CFO estimation feature of the DW3000 chipset. The method has been described in [26] and the systematic error has been derived for single-sided two-way ranging (SS-TWR), A-TDOA, and SS-TWR with A-TDOA extension. The method has been evaluated experimentally but only for TWR schemes. A similar CFO correction is evaluated for a TDOA scheme in [27]. However, the proposed TDOA scheme is based on the alternative double-sided TWR (AltDS-TWR) method, in which the tag is active, which is different from the DL TDOA schemes evaluated in our work. A CFO-assisted synchronization algorithm for TDOA has been proposed in [28], in which the TOA at each receiver is compensated using the estimated CFO w.r.t. the reference (master) node. Although the TDOA scheme from [28] is different from the DL TDOA scheme used in our work, the correction method is similar. In addition, we evaluate the feasibility of the correction method through experiments with commercial UWB systems, whereas in the cited work the method was evaluated in simulations and using software-defined radios.

III. BACKGROUND

In this section, we introduce principles related to the proposed localization methods. For more in-depth details on UWB ranging and localization, we refer the reader to the



(a) Range-based localization (b) Downlink (DL) TDOA

FIGURE 1. Localization based on TWR or on DL TDOA (the time periods are not to scale).

papers [29], [30]. In Section III-A and III-B we explain how distance measurements and, respectively, TDOA measurements are obtained using UWB devices. In Section III-C, we describe the two approaches we use to solve the system of equations in order to estimate the user's location: either least-squares minimization between the measured and the calculated ranges or an extended Kalman filter (EKF). In Section III-D, we describe the scheduling scheme implemented in FlexTDOA.

A. TWR

Range-based localization uses distances between a mobile target, called tag, and anchors with known locations to compute the location of the tag. The tag is found at the intersection of circles (in a 2D space) or spheres (in a 3D space) with a radius equal to the anchor–tag distances and centered at the anchors.

We use the following notations:

- $\vec{X}_{A_1}, \vec{X}_{A_2}, \ldots, \vec{X}_{A_N}$ are the locations of anchors $A_1, A_2, \ldots, A_N$, respectively;
- $\vec{X}_T(x, y, z)$ is the location of the tag;
- $d_{\vec{X}\vec{Y}}$ is the *true* distance between the nodes with locations $\vec{X}, \vec{Y} \in \{\vec{X}_T, \vec{X}_{A_1}, \ldots, \vec{X}_{A_N}\}$;
- $\tilde{d}_{XY}$ is the *measured* distance between the nodes $X, Y \in \{T, A_1, \ldots, A_N\}$.

The true distance between the tag and anchor A_i is the Euclidean distance between their locations:

$$d_{\vec{X}_T\vec{X}_{A_i}} = \|\vec{X}_T - \vec{X}_{A_i}\|. \tag{1}$$

The measured distance between a tag and an anchor A_i can be written as:

$$\tilde{d}_{TA_i} = d_{\vec{X}_T\vec{X}_{A_i}} + \omega_{TA_i}, \ i = 1, \ldots, N, \tag{2}$$

where ω_{TA_i} is the measurement noise, which is modeled as a zero-mean Gaussian random variable with variance σ_i^2.

To avoid synchronizing the anchors and the tag, the distances are usually obtained using TWR by exchanging at least two messages between the tag and each anchor [29]. We implemented the SS-TWR variant which uses two message exchanges between each anchor and the tag, illustrated in Fig. 1. One distance is obtained as:

$$\tilde{d}_{TA_i} = c \cdot \frac{\Delta T_i^T - \Delta T_i}{2}, \tag{3}$$

where c is the speed of light. ΔT_i is the time between the arrival of the tag's request at anchor $\mathrm{A_i}$ and the anchor's transmission of the response message, as measured by the anchor. Similarly, ΔT_i^T is the time between the tag's transmission of the request and the arrival of the anchor's response, as measured by the tag.

Because each device measures its own processing time, there is no need to synchronize the clocks of neither the anchors nor of the anchors and the tag. However, due to the fact that ΔT_i^T is measured by the tag while ΔT_i is measured on the anchor, the relative clock skew of the two nodes compounded with the significant value of ΔT_i can introduce large errors in the measurement. To counter this effect, the DS-TWR was proposed [31], in which the tag transmits a third message which is used by the anchor to measure the clock skew. Modern radios, such as the one we are using, can directly measure the relative clock skew by analyzing the carrier frequency of the received packet. Research shows [32] that DS-TWR and SS-TWR attain almost the same precision with clock skew estimation.

SS-TWR based localization (which we will alternatively call TWR localization) is attractive because it enables centimeter-level localization and does not need any synchronization between the devices. However, it does not scale well when increasing the number of anchors and tags, since it needs pair-wise message exchanges between each anchor and each tag in the system. DS-TWR is even less scalable than SS-TWR because it needs an extra message for the ranging. Moreover, because of the asymmetry, in DS-TWR the distance is computed by the responder (anchor), not the initiator (tag). Therefore, for navigation, the distance would need to be sent back to the user in another message. Given these drawbacks, we will use SS-TWR as a comparison baseline for FlexTDOA.

B. TDOA

An alternative localization technique uses the time *difference* between the arrival of two messages either at one device or at multiple clock-synchronized devices [5]. We define the true *distance* difference between anchors A_i and A_j relative to the tag as:

$$d_{\vec{X}_T\vec{X}_{A_i}\vec{X}_{A_j}} = d_{\vec{X}_T\vec{X}_{A_j}} - d_{\vec{X}_T\vec{X}_{A_i}} \tag{4}$$

$$= \|\vec{X}_T - \vec{X}_{A_j}\| - \|\vec{X}_T - \vec{X}_{A_i}\|. \tag{5}$$

The *measured* distance difference between anchors A_i and A_j relative to the tag is:

$$\tilde{d}_{TA_iA_j} = d_{\vec{X}_T\vec{X}_{A_i}\vec{X}_{A_j}} + \omega_{TA_j} - \omega_{TA_i}, \tag{6}$$

where, similarly to the TWR case, ω_{TA_k} is the measurement noise between the tag and anchor $A_k, k \in \{1, \ldots, N\}$, modeled as zero-mean Gaussian noise with variance σ_k^2.

In the noiseless case, any $N - 1$ distance difference measurements that form a minimum spanning subtree are sufficient for TDOA localization [33]. However, more redundant measurements can be used to improve the resilience to noise

in a realistic setup. In TDOA localization, the tag's position is found at the intersection of hyperbolae. This makes TDOA localization more sensitive to noise than range-based (or TOA) localization [3].

The *time* difference of arrival can be derived from Eq. (6) by dividing the distance difference by the speed of light:

$$t_{TA_iA_j} = \frac{d_{\vec{X}_T\vec{X}_{A_i}\vec{X}_{A_j}}}{c} = \frac{d_{\vec{X}_T\vec{X}_{A_j}} - d_{\vec{X}_T\vec{X}_{A_i}}}{c}. \tag{7}$$

The time differences can be computed either by a *passive* tag when the anchors transmit simultaneously their messages or by the anchors (or a central entity) when an *active* tag transmits a broadcast message which is received by all anchors. We will, respectively, call the two methods downlink (DL) and uplink (UL) TDOA.

To avoid synchronizing the anchors, we use a DL TDOA variant with reference and responding anchors previously used in [10], [11], [15], [16] The scheme is illustrated in Fig. 1b. Anchor $\mathtt{A}_0$ is the initiator and transmits a broadcast message, received by the tag at time T_0. Anchors $\mathtt{A}_1$ to $\mathtt{A}_N$ receive the message and then wait a period ΔT_i which includes the processing time and a delay necessary to avoid overlapping transmissions from successive anchors. The tag receives the responses at times T_1 to T_N.

For the general case in which $\mathtt{A}_i$ is the initiator and $\mathtt{A}_j$ is the responder, let us denote by $\Delta T_{ij} \triangleq T_j - T_i$ the difference between the time at which the tag receives the response (T_j) and the time at which the tag receives the request (T_i). In order to obtain only the TDOA from Eq. (7), we need to subtract the processing time ΔT_j and the TOF between anchors $\mathtt{A}_i$ and $\mathtt{A}_j$ (denoted by t_{ij}) from the timestamp difference ΔT_{ij}:

$$\tilde{t}_{TA_iA_j} = T_j - T_i - \Delta T_j - t_{ij}, \tag{8}$$

where $\tilde{t}_{TA_iA_j}$ is the *estimated* TDOA between the tag and the anchors $\mathtt{A}_i$ and $\mathtt{A}_j$. The TOF t_{ij} is usually known because the anchors are placed at fixed, known locations.

Because ΔT_j is measured by $\mathtt{A}_j$ but subtracted from the timestamp difference ΔT_{ij} measured by the tag, the TDOA will contain an error due to the relative clock skew between anchor $\mathtt{A}_j$ and the tag. Because in TDOA localization multiple anchors respond to the same synchronization message, the processing times are longer for this scheme than in TWR localization. It is therefore crucial to correct the relative clock skew errors in TDOA localization [15]. To eliminate the errors, we leverage the capability of the DW3000 chipset to estimate the carrier frequency offset (CFO) between the local receiver and the remote transmitter [26]. The interval measured by the tag will contain an error due to the tag running at a different frequency from an "ideal" nominal frequency [13]:

$$\Delta\hat{T}_{ij} = k_T\,\Delta T_{ij} = (1 - e_T)\Delta T_{ij}, \tag{9}$$

where k_T is the multiplicative and e_T the additive error of the tag's clock. Similarly, the processing time measured by the anchor $\mathtt{A}_j$ will contain an error:

$$\Delta\hat{T}_j = k_j\Delta T_j = (1 - e_j)\Delta T_j, \tag{10}$$

where k_j and e_j are the multiplicative, respectively, the additive errors of the anchor's clock.

The relative CFO between the clocks of the tag and $\mathtt{A}_j$ as measured by the tag is [26]:

$$\kappa_{Tj} = \frac{k_T}{k_j} = 1 - \epsilon_{Tj}, \tag{11}$$

where κ_{Tj} and ϵ_{Tj} are the multiplicative, respectively, the additive *relative* clock frequency offsets. The DW3000 chipset estimates ϵ_{Tj} (expressed in ppm), which we use to correct the processing time[2]:

$$\Delta T_j^{corr} = \Delta T_j(1 - \epsilon_{Tj}). \tag{12}$$

C. LOCALIZATION ALGORITHMS

So far, we have discussed the basic principles to obtain the ranges or the range differences between the anchors and the tag. In order to estimate the user's location, we need to solve a system of equations based on Eq. (1) and (5). We implemented two localization algorithms, each capable of operating with either TWR or TDOA data, each suiting different needs.

The first algorithm, AlgMin, solves the localization problem for a series of consecutive measurements using squared error minimization. This algorithm does not track the user's location nor does it smooth the location estimates, and it is therefore suitable to evaluate the impact of several parameters (e.g., the number of responses or anchors) on the localization accuracy.

The second algorithm, AlgEKF, solves the localization problem using an EKF, by incrementally updating the location with each additional available measurement. This approach is advantageous because we do not need to wait for the minimum number of measurements (four in the case of TWR localization and five for TDOA localization) in order to update the tag's location. However, it smooths the location estimates and hides the impact of noisy measurements. Therefore, we use it only when we compare several setups that generate a different number of equations per time slot in Section V-C.

Both algorithms start with the known anchor positions $\vec{X}_{A_1}, \vec{X}_{A_2}, \ldots, \vec{X}_{A_N}$ and estimate the tag's location $\hat{\vec{X}}_T$. In our experiments, the anchors' positions are determined using a self-localization algorithm, which is a variant of AlgEKF using TWR measurements between each pair of anchors, run over a period of several minutes. The locations of the anchors are determined once, at the beginning of the experiments, and kept fixed thereafter.

[2]The conventions for the additive clock offset are opposite for the DW1000 and DW3000 chipsets. For the DW1000 chipset, which uses the conventions from [26], if the additive clock offset is positive, then the receiver's clock is running at a faster rate than the transmitter clock [34], while for DW3000 the reverse is true [35]. In this paper, we used the conventions for the DW3000 chipset.

1) AlgMin

We use least square error minimization between the measured and the calculated distances. Therefore, for the TWR measurements:

$$\hat{\vec{X}}_T = \arg\min_{\vec{X}} \sum_i \left(d_{\vec{X}\vec{X}_{A_i}} - \tilde{d}_{TA_i}\right)^2, \tag{13}$$

where $d_{\vec{X}\vec{X}_{A_i}}$ is the calculated Euclidean distance between the 3D locations $\vec{X}$ and $\vec{X}_{A_i}$ and $\tilde{d}_{TA_i}$ is the TWR measured distance between the tag T and anchor A_i (Fig. 1).

For TDOA measurements, the equivalent minimization problem is as follows:

$$\hat{\vec{X}}_T = \arg\min_{\vec{X}} \sum_{i,j} \left(d_{\vec{X}\vec{X}_{A_j}} - d_{\vec{X}\vec{X}_{A_i}} - \tilde{d}_{TA_iA_j}\right)^2, \tag{14}$$

where $\tilde{d}_{TA_iA_j} = c \cdot \tilde{t}_{TA_iA_j}$ and $\tilde{t}_{TA_iA_j}$ is the TDOA measurement performed by the tag T while listening to the two-way communication between nodes A_i and A_j (Fig. 1b). The TDOA measurement $\tilde{t}_{TA_iA_j}$ is computed using Eq. (8), where t_{ij} is obtained from the payload of the packets transmitted by the anchors.

The algorithm needs a good initialization to avoid converging to a local minimum of the function that is optimized [36]. In practice, we first initialize the algorithm using the starting position of the tag and for subsequent initializations we use the position of the tag estimated by AlgMin in the previous iteration.

2) AlgEKF

The location of the moving tag is computed iteratively using an EKF. Again, we implemented two filters one for the TWR and one for the TDOA. In both cases, the state of the filter is constituted by the position of the tag. The EKF assumes a system that can be described by the following general equations:

$$\begin{aligned} \vec{X}_k &= f(\vec{X}_{k-1}, U_k) + w_k \\ z_k &= h_k(\vec{X}_k) + v_k, \end{aligned} \tag{15}$$

where:

- $\vec{X}_k$ is the position of the tag estimated at moment k;
- $f(\vec{X}_{k-1}, U_k)$ is the motion model for the tag dependent on input command U_k and previous state;
- w_k is Gaussian noise $\mathcal{N}(0, Q_k)$ due to the uncertainty in the motion model, where Q_k is the covariance matrix of the motion model;
- z_k is the vector of measurements (either TWR or TDOA);
- $h_k(\vec{x})$ is a function that computes the expected value of the measurements given the state (position) $\vec{X}$ of the tag;
- $v_k \sim \mathcal{N}(0, R_k)$ is Gaussian noise modeling the uncertainty in the measurements, where R_k is the covariance matrix quantifying the uncertainty in our measurements.

In our case, we do not assume a known motion model, so $f(\vec{X}_{k-1}, U_k) = \vec{X}_{k-1}$, corresponding to a static tag. We choose Q_k, the uncertainty in the model, large enough to cover the motion of the tag. In IV-B, we detail the choice of values for the covariance matrices using in the EKF. The choice of using a static model was deliberate, in order to evaluate the raw performance of our UWB localization method. In a real system, any knowledge about the actual movement of the tag can be input into f to improve the accuracy of the system.

The function h and the vector z depend on the type of measurement performed. In the case of TWR:

$$\underbrace{\begin{pmatrix} \tilde{d}_{TA_i} \\ \tilde{d}_{TA_j} \\ \cdots \end{pmatrix}}_{z_k} = \underbrace{\begin{pmatrix} d_{\vec{X}_k\vec{X}_{A_i}} \\ d_{\vec{X}_k\vec{X}_{A_j}} \\ \cdots \end{pmatrix}}_{h_k}. \tag{16}$$

In the case of TDOA:

$$\underbrace{\begin{pmatrix} \tilde{d}_{TA_{i_1}A_{j_1}} \\ \tilde{d}_{TA_{i_2}A_{j_2}} \\ \cdots \end{pmatrix}}_{z_k} = \underbrace{\begin{pmatrix} d_{\vec{X}_k\vec{X}_{A_{j_1}}} - d_{\vec{X}_k\vec{X}_{A_{i_1}}} \\ d_{\vec{X}_k\vec{X}_{A_{j_2}}} - d_{\vec{X}_k\vec{X}_{A_{i_2}}} \\ \cdots \end{pmatrix}}_{h_k}. \tag{17}$$

In both cases, we use the classical prediction and update steps of the EKF filter to compute the tag's current position $\vec{X}_k$, starting from the tag's previous position estimate $\vec{X}_{k-1}$ and the new measurements:

$$\vec{X'}_k = \vec{X}_{k-1} \tag{18}$$

$$\begin{aligned} P'_k &= P_{k-1} + Q_k \\ K_k &= P'_k H_k^T (H_k P'_k H_k^T + R_k)^{-1} \\ \vec{X}_k &= \vec{X'}_k + K_k(z_k - h_k(\vec{X'}_k)) \\ P_k &= (I - K_k H_k) P'_k, \end{aligned} \tag{19}$$

where P_k is the covariance matrix quantifying the uncertainty in our estimation and H_k is the Jacobian of the function h. The rest of the notations are defined above. Compared to the general EKF case for a system of equations, ours is simpler given that f is the identity function and its Jacobian is the unit matrix I.

The EKF update step can be done with a single measurement at a time (z_k is a single distance) or with multiple measurements at a time. To minimize the latency, we update the tag's position with every incoming measurement.

For the self-localization of the anchors, we use AlgEKF with TWR measurements. The algorithm works similarly to the tag's localization, with the following differences: (1) the state variable $\vec{X}_k$ contains the 3D location of all anchors, (2) we use a much smaller covariance Q_k to account for the fact that the anchors are fixed, and (3) we perform an update using multiple ranges at a time (instead of updating the location for every incoming measurement). For simplicity, we align the resulted coordinate frame such that anchor A_0 has the location $(0, 0, 0)$, anchor A_1 is on the O_y axis, anchor A_2 is located in the O_{xy} plane with a positive x, and A_3 has a negative z coordinate.

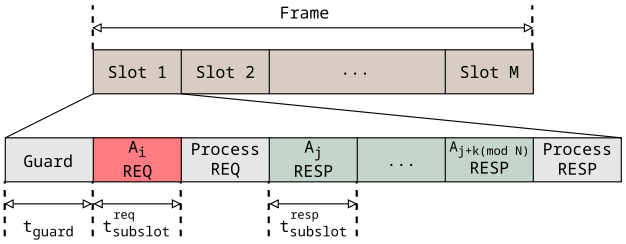


FIGURE 2. TDMA schedule used for both TDOA and TWR localization. Each slot in a frame belongs to a node which is the initiator in that slot and decides which K nodes to interrogate (depending on the current scheme).

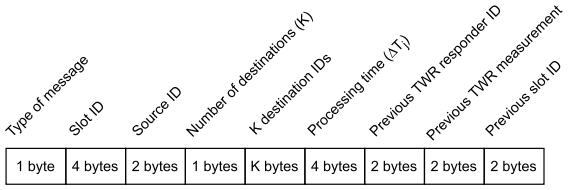


FIGURE 3. Localization packet format used by the anchors and the tag.

D. SCHEDULING

In the "classic" TDOA approach, there is a single designated reference anchor which broadcasts the synchronization message. The rest of the anchors respond to the broadcast in a predefined order. Instead, we propose, implement, and evaluate a *flexible* TDOA scheduling scheme in which all the anchors in the system can play the role of the initiator and the order of responses can also change.

We propose and implement a time-division multiple access (TDMA), scheme shown in Fig. 2, which can be configured for either TWR or TDOA localization. At this point, we do not differentiate between anchors and tags and instead consider all of them equally-participating nodes. The distinction will be made according to the implemented localization method.

The TDMA scheme is organized in *time slots*, which are comprised of a broadcast message sent by an initiating node, which we will call a *request*, and K responses from other nodes, where $K < N$ and N is the number of nodes in the system. Each response will provide a TWR measurement between the initiator node and the responding node and a TDOA measurement for each of the other nodes listening to the exchange. Inside a time slot, each transmission by a node occurs in a *subslot* with duration t_{subslot}. At the beginning of a time slot there is a guard time, followed by the request of the initiating node. The request includes the number of nodes that will respond, their ID, and the order of their response. All the listening nodes in the system process the request. If the initiator requested a response from the listening node in the subslot with index $k \in \{1, \ldots, K\}$, the node will wait a period of $(k-1) \times t_{\text{subslot}}$ and then answer. During the last part of a time slot, the initiator processes the responses.

The time slots are organized in frames (Fig. 2). Each frame contains M time slots, each of them assigned to one of the nodes (anchors or tags). For instance, if slot i is assigned to node j, then, in all frames, node j will be the initiator in slot i, and will decide which K nodes to interrogate based on the currently-selected scheme. More than one slot can be assigned to the same node and there may be nodes that have no slot assigned (e.g., a tag that is passively listening and only using TDOA measurements for localization).

The network starts with the default programmed parameters. The nodes first listen for a few seconds before transmitting. From any received message, they can determine the current slot and synchronize with the TDMA schedule. All the TDMA parameters are decided by the system administrator and can be changed at runtime: M, K, the slot assignment, the list of nodes that are allowed to be interrogated (for example, to avoid the interrogation of a passive tag). To change the parameters, the system administrator connects via USB to any of the nodes, uploads the new configuration into the node, and requests the node to broadcast it. The node broadcasts the information using a special packet format. All the nodes that hear the new configuration will rebroadcast it for several seconds. In a real system, we could envision an algorithm that makes the decisions instead of the administrator.

All localization messages exchanged by the anchors or the tag have the format shown in Fig. 3. It consists of: the message type (request or response), the source ID, the list of interrogated anchors (in case of a request), and a payload in which the sender can include a previously-acquired TWR measurement. Each packet contains the current slot ID, which is a 32-bit counter incremented continuously. This counter is used by the nodes to determine the current slot in the frame (modulo M) and to stay synchronized with the TDMA schedule.

In the current implementation, the node IDs, K, and M are 8-bit packet fields and variables, which implies a maximum of 256 devices and 255 slots in a frame. A larger network can be accommodated by choosing larger packet fields. At the moment, a slot can only be assigned to one node. For a network spread over a larger area, this should be changed so that nodes that are sufficiently far apart (i.e., not in radio range) can reuse the same slot.

To configure the TDMA scheme to perform TWR localization, the tag will be set as initiator in all slots and the anchors will be the responders. In one time slot with K responses, the tag obtains K raw distance measurements which are input to the multilateration system to estimate the tag's location.

To perform DL TDOA localization, only the anchors will be initiators, interrogating other anchors, while the tag will be a passive listener. Depending on how we choose the initiators and the responders, we can derive four variants of TDOA localization:

- *Fixed initiator, fixed responders* (FI-FR), or the "classic" TDOA, with a designated reference anchor (or initiator), where the other anchors in the system respond in a fixed order according to their index. In this case, all slots in a frame are assigned to a single anchor (the initiator). In every slot, the initiator interrogates the same list of (responding) anchors.

- *Fixed initiator, changing responders* (FI-CR) with a designated reference anchor, where the responding anchors change in a round-robin (RR) manner. In this case, in the first slot, the initiator will pick K anchors to interrogate (from the allowed list), in the next slot the next K, and so on, wrapping around.
- *Changing initiator, fixed responders* (CI-FR), in which the initiator changes every time slot in a RR manner and the rest of the anchors respond in a (fixed) ascending order of their index. In this case, there are multiple initiators (anchors that have slots assigned) and each of them interrogate the same fixed list of responders.
- *Changing initiator, changing responders* (CI-CR), in which both the initiator and the responder order change every time slot in a RR manner. This is the scheme used in FlexTDOA. Multiple initiators interrogate K other nodes at a time from the pool of allowed responders.

We note that the TDMA scheme also allows the anchors to localize *themselves*. The initiator measures the distance between itself and each responder (other anchors) using TWR. These distances are then transmitted as piggyback payloads of future messages (requests or responses) of the initiator. As a result, simply by listening to the message exchanges, the localization engine can compute the location of each anchor in the local coordinate system using the self-localization algorithm described in Section III-C.

IV. EVALUATION SYSTEM

In this section, we present the localization system used to evaluate the TWR and TDOA algorithms. In Section IV-A, we describe the hardware used; in Section IV-B, we present the settings used for the UWB radio, scheduling algorithm, and EKF. In Section IV-C, we describe the environment in which we acquired the measurements and the placement of the anchors.

A. HARDWARE

For the experimental evaluation, we designed and fabricated our own UWB node, shown in Fig. 4. We used the Qorvo DWM3000 wireless transceiver [13], which implements the IEEE 802.15.4 standard [37].

At the core of the UWB node is an Arm Cortex-M4 based STM32F429ZIT6 microcontroller with 2 MB of Flash memory, 256 KB of SRAM memory and a frequency of 168 MHz [38]. All the ranging and scheduling algorithms are running on the on-board MCU and the ranging results are transmitted over an USB 2.0 port. However, the board can be configured to transmit the information over any other interface, like I^2C, SPI or UART. The entire software stack was written in-house.

The UWB node is powered by a single Li-ion rechargeable battery with a capacity of 6Wh, providing over 15h of autonomy. This allowed us to easily place the nodes independently of available power sources.

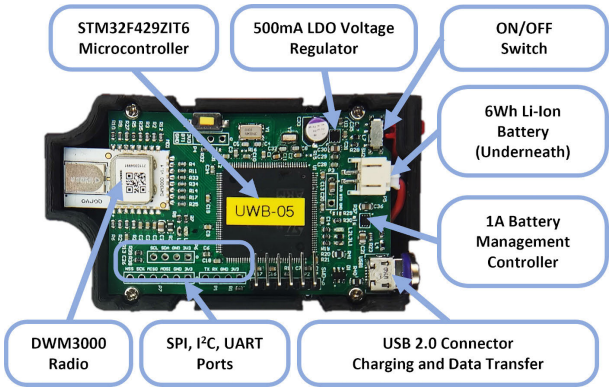


FIGURE 4. Custom build UWB Node: a completed hardware and software UWB node which is battery powered and capable to expose ranging information over multiple serial communication interfaces.

The power management is implemented using an MCP73830 IC, which is a 1 A Single-Cell Li-Ion battery charge management controller. This allows easy charging over the USB port. The MCU and the radio are powered through a 500 mA LDO voltage regulator, NCV8705.

B. SYSTEM SETTINGS

We configured the UWB transceiver to operate on channel 5 (6.5 GHz) with a preamble length of 128 symbols, a 6.8 Mb/s data rate, and a pulse repetition frequency of 64 MHz.

The duration of one time slot in the TDMA scheme shown in Fig. 2 is computed as:

$$t_{\text{TS}} = t_{\text{guard}} + t^{\text{req}}_{\text{subslot}} + t^{\text{req}}_{\text{process}} + K \cdot t^{\text{resp}}_{\text{subslot}} + t^{\text{resp}}_{\text{process}} \quad (20)$$

where:

- $t_{\text{guard}} = 250\,\mu$s is the guard time at the beginning of the time slot necessary to wait for the responders to go into the receive mode;
- $t^{\text{req}}_{\text{subslot}} = 2000\,\mu$s is the period it takes the initiator to send a request;
- $t^{\text{req}}_{\text{process}} = 250\,\mu$s is the period during which the initiator enters the receive mode and the responders process the request message;
- K is the number of responses;
- $t^{\text{resp}}_{\text{subslot}} = 250\,\mu$s is the duration of one response in a subslot;
- $t^{\text{resp}}_{\text{process}} = K \cdot 600\,\mu$s is the time allocated for the initiator to process the responses.

The subslots have different durations for sending a synchronization beacon or a response because of additional time needed to prepare request message sent it over UWB radio and send measurement data over serial port. Table 1 shows the duration of one time slot for each number of responders.

For both EKF filters (based on TWR and TDOA measurements) we chose a variance of the model uncertainty of $\sigma_Q^2 = 100\,\text{cm}^2$, which accounts for the motion of the tag between measurements and assumes a maximum speed of the tag of 10 cm/s. We chose a variance of $\sigma_R^2 = 10\,\text{cm}^2$ for

TABLE 1. Duration of one time slot (t_{TS}) for each number of responders K.

K	1	2	3	4	5	6	7	8	9
t_{TS}(ms)	3.35	4.20	5.05	5.90	6.75	7.60	8.45	9.30	10.15

FIGURE 5. Office setup.

the measurement noise, which was based on the measurement noise we obtained during experiments.

For the EKF filter used for the self-localization of the anchors, we used $\sigma_Q^2 = 1\,\text{cm}^2$ because the anchors are static. The location of the anchors is determined once, at the beginning of the experiments and kept fixed thereafter.

C. ENVIRONMENT AND ANCHOR PLACEMENT

We evaluate the localization systems in the office shown in Fig. 5. The 3D anchor placement is shown more clearly in Fig. 6. Five of the anchors ($\mathtt{A}_0$ to $\mathtt{A}_3$ and $\mathtt{A}_9$) are fixed on the ceiling using metallic bars, while the rest of the anchors are either placed on the ground ($\mathtt{A}_4$) or on tables ($\mathtt{A}_5$ to $\mathtt{A}_8$). The location of the anchors is determined using the self-localization algorithm described in Section III-C. We validated the resulting locations using a laser level and a laser rangefinder, both with mm-level precision.

To accurately measure the ground truth (GT) of the tag, we have built a custom electronic linear actuator shown in Fig. 7. We used a 140 cm-long aluminum rail, a stepper motor, a timing belt, an aluminum trolley carrying the tag, and a driver connected via USB to the computer. The speed of actuator is 10 cm/s and the positioning resolution is 0.1 mm.

The actuator continuously sends the current position over the USB serial port. To measure the GT of the node in our positioning system, we placed the actuator at a known location relative to the anchor $\mathtt{A}_0$, which is the origin of the coordinate system. Unless stated, the actuator moves only along the x axis of the coordinate system, between the x axis coordinates of approx. $(110, 270)$ cm, as shown in Fig. 6. We have two experiments in which the actuator is placed in a 3D orientation.

In all of the experiments, we precisely align the actuator with the local coordinate system using a laser rangefinder and a laser level.

V. EVALUATION OF SYSTEM PARAMETERS

In this section, we evaluate the impact of several factors on the localization accuracy: the order of response of an anchor, the number of responses in a time slot for the maximum update rate of the localization system and for lower update rates, and the number of anchors available. We evaluate these parameters for localization algorithms that use distance measurements (obtained using TWR), which we call "TWR localization," and for the proposed FlexTDOA system, called simply "TDOA localization." The goal of the comparison between TWR and TDOA localization is to evaluate the impact of system parameters of both distance and TDOA measurements.

Unless explicitly mentioned, we use the AlgMin algorithm described in Section III-C to estimate the user's location. The localization error is computed as the Euclidean distance between the 3D GT location and the estimated location:

$$e = \sqrt{(x-\hat{x})^2 + (y-\hat{y})^2 + (z-\hat{z})^2}, \tag{21}$$

where (x, y, z) and $(\hat{x}, \hat{y}, \hat{z})$ are the Cartesian coordinates of the GT, respectively, the estimated location.

We use boxplots to illustrate the error distributions. In a boxplot, such as the ones in Fig. 8, the box is drawn from the first to the third quartiles (or, respectively, the 25$^{\text{th}}$ and the 75$^{\text{th}}$ percentiles), which is also known as the interquartile range (IQR). Boxplots drawn for samples that can take negative and positive values (e.g., the distance and TDOA errors) have whiskers that extend from the 5$^{\text{th}}$ to the 95$^{\text{th}}$ percentiles. For strictly positive errors (for instance, the localization errors which are computed as the Euclidean distance between the estimated and the ground truth locations), the whiskers extend from the 0$^{\text{th}}$ to the 95$^{\text{th}}$ percentiles. The reasoning is that, when we plot the distribution of *absolute* errors, we are interested in the minimum value of the error. We omit plotting the outliers for simplicity. We will frequently report the 95$^{\text{th}}$ percentile, which we will alternatively call the 95% error (or P_{95}) for short, which represents the value below which 95% of the errors are found.

A. ORDER OF RESPONSE

First, we investigate how the TWR or TDOA *measurement* error changes depending on the order of the response in a time slot. In Section III-B, we mentioned that it is important to correct the processing times to account for the relative clock skew between a transmitter and a receiver. With a longer time period elapsed between the initiator's request and an anchor's response, the error in a TWR or a TDOA measurement will increase because the additive relative clock offset ϵ_{Tj} will be multiplied by a longer processing time ΔT_j in Eq. (12). Although we compensate for these errors using CFO estimation, the correction is imperfect. Therefore, we expect to see

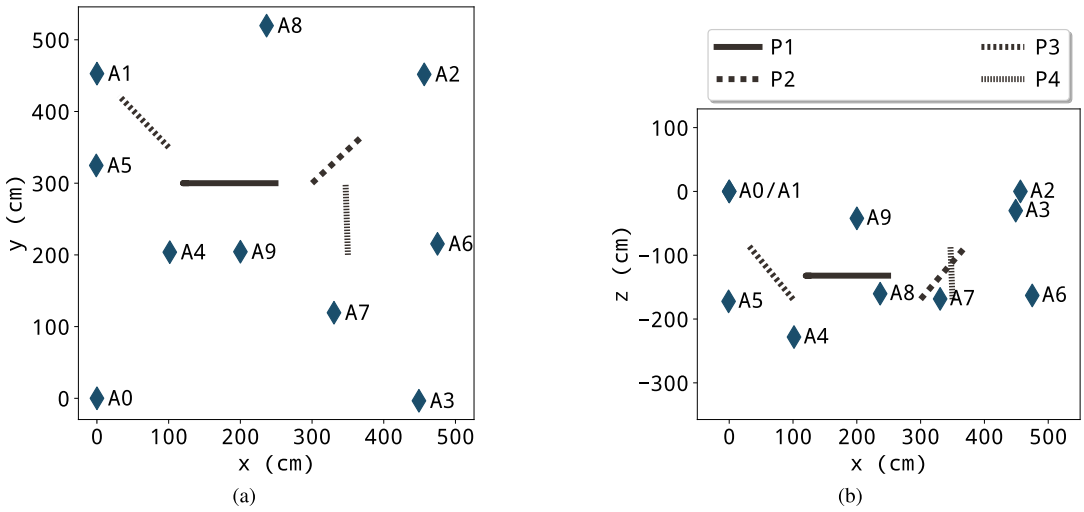


FIGURE 6. Setup of the anchors and the tag in the (a) *xy* and (b) *xz* planes. The anchors are denoted by A_0 to A_9. We evaluate the localization accuracy at four positions of the linear actuator, along which the tag moves, denoted by P1 to P4.

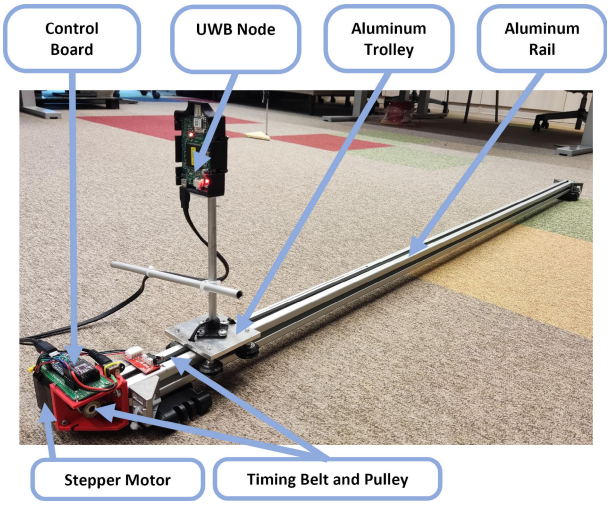


FIGURE 7. Ground truth linear actuator: An aluminum trolley that carries the tag and returns over USB the position of the tag relatively to the zero point of the actuator.

larger TWR and TDOA errors for higher response indexes in a time slot.

To evaluate the magnitude of the errors, we perform an experiment in which the tag is kept unmoved, in order to avoid any accuracy loss due to the movement of the tag. We configure the system to compute either the distance (using TWR) between each anchor and the tag or the TDOA between the tag and each pair of anchors. We use the maximum number of anchors ($N = 10$) and of responses ($K = 9$ for TDOA and $K = 10$ for TWR). To compute the distances, the tag is always the initiator and the anchors respond to its broadcast. The order in which the anchors respond changes every time slot using round-robin (RR) scheduling. To compute the TDOA, the tag only listens to the messages and both the initiating anchor and the responding nodes rotate every timeslot in a RR manner (CI-CR scheduling). We run each experiment for 3 min and compute the TWR and TDOA errors for each possible index of response.

Fig. 8 shows the TWR and TDOA error distributions for each order of response. Table 2 shows the statistics of the errors in the first and last response: the mean, standard deviation (σ), IQR, and whisker spread ($P_{95} - P_5$). The whisker spread of TDOA errors increases with 14.7 cm between the last and the first in the list of responses. TWR errors are less affected by the order of response than TDOA errors: the whisker spread of TWR errors increases with only 4.1 cm between the last and first order of response. The results are in line with the theory: with a longer waiting time between the initiator's message and the response, the error due to the relative clock skew increases and the measurements are corrupted by noise. Over many measurements, the mean error remains small, in absolute terms, regardless of order of response. What increases significantly is the noise in each measurement. This suggests that, for a relatively static tag, a larger number of responses and averaging would increase the accuracy (since it uses the time more efficiently), while for a fast moving tag, where averaging is not possible without a motion model, a smaller number of responses would perform better. We further explore this trade-off in Section V-C.

B. NUMBER OF RESPONSES

Since the TWR and TDOA *measurement* error increases with the order of response in the time slot, we investigate to what extent the *localization* accuracy changes with the number of responses in a time slot. In schemes with a fixed order of responses (FI-FR and CI-FR), the number of responding anchors is given by the number of anchors in the system (so that all anchors can participate in the localization process). In schemes with a changing order of response, however,

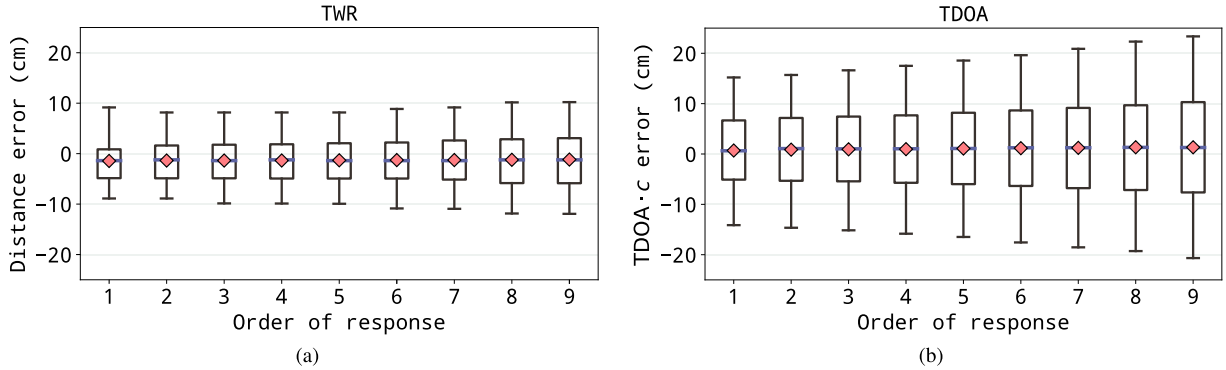


FIGURE 8. The distribution of (a) TWR and (b) TDOA errors (expressed in cm using the speed of light) against the order of response aggregated over all anchors.

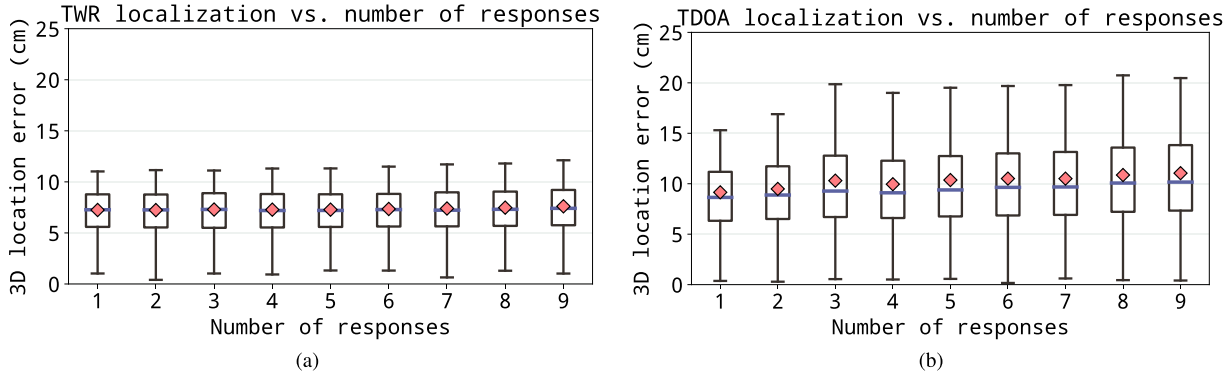


FIGURE 9. Localization error of TWR and TDOA localization depending on the number of responses in a slot. The error increases for more responses due to the longer period between the first and the last response, which increases the effect of clock drift estimation error.

TABLE 2. Statistics of raw measurement errors in the first and last (9^{th}) response: mean, standard deviation (σ), IQR, and difference between the 95^{th} (P_{95}) and 5^{th} (P_5) percentiles.

Measurement	Response index	Mean (cm)	σ (cm)	IQR (cm)	$P_{95} - P_5$ (cm)
TWR	1	−1.5	4.8	5.7	18.1
	9	−1.2	6.9	8.9	22.1
TDOA	1	0.7	9.0	11.8	29.3
	9	1.3	13.5	17.9	44.0

we can decrease the number of responses to be smaller than the number of anchors in the system and still have all the anchors participate, only in different time slots.

For this evaluation, we keep the same setup as in Section V-A, so using $N = 10$ anchors, but we vary the number of responses $K \in \{1, \ldots, 9\}$ and let the tag move on the trolley.

Fig. 9 shows the localization error of TWR and TDOA localization with a varying number of responses. As expected, the localization error is the smallest for the minimum number of responses. However, the increase in the mean and IQR of the localization error with a higher number of responses is almost negligible: less than 3 cm between the maximum and the minimum number of responses for both TDOA and TWR localization. The 95% TDOA localization error has a slightly larger increase than the IQR with nine vs. one responses, of 5.2 cm.

The results seem counter-intuitive: more responses in a slot yield more measurements per unit of time, which should decrease the error. This is not the case because of several reasons. (1) Even in the worst case (one response in a slot), the system generates about 300 measurements per second. This is enough to approach the maximum theoretical performance given the relatively slow speed of the tag of 10 cm/s. So, for a slow tag, we get a better performance with few precise measurements than with many noisy measurements. (2) Even with a lower number of responses in a slot, all anchors remain engaged as initiators and pick their responders round-robin from all other anchors, thus preserving the diversity. We further investigate the effect of K, under additional constraints, in the next section.

C. NUMBER OF RESPONSES FOR DIFFERENT TAG SPEEDS

Although using only one response per time slot yields the smallest error spread, this configuration has at least two disadvantages. First, over a fixed time period, the number of TDOA measurements decreases with the number of responses per time slot, because of the overhead added by the initiator's request. This trend is illustrated in Fig. 10,

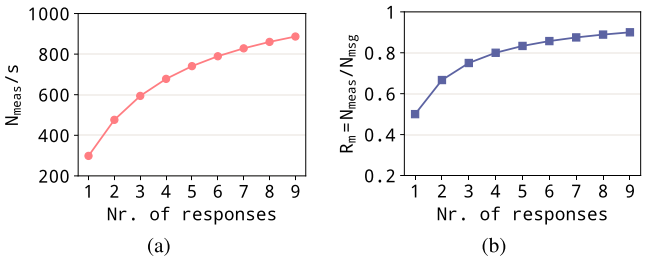


FIGURE 10. (a) The number of TWR/TDOA measurements (N_{meas}) obtained per second vs. the number of responses (K) in a slot and (b) the ratio between N_{meas} and the number of messages (N_{msg}) obtained per second vs. K.

which shows the number of distance or TDOA measurements (N_{meas}) obtained per second for $K \in 1, \ldots, 9$ responses. With $K = 9$, the number of measurements per second is approx. $3\times$ larger than with $K = 1$. Second, because of the same reason, the energy consumed by the tag to receive a certain number of TDOAs increases as the number of responses decreases. We can compute the ratio between the number of measurements and the number of exchanged messages (N_{msg}) over the same time period, which is an indicator of the efficiency of the tag. This ratio (denoted by R_m) is illustrated in Fig. 10b. With $K = 4$, the efficiency is $1.6\times$ higher than with $K = 1$, but the slope declines as K increases.

In a real system, the system administrator will face the question of choosing K to optimize the location accuracy under the specific conditions: available power to the system, available air time for transmissions, and maximum speed of the tag. To give an insight on the trade-offs involved, we evaluate how the 3D localization error changes with K when we keep constant over the same time period either the *total transmission time* (T_{TX}) or the *number of exchanged messages* (N_{msg}).

Additionally, to measure the effect on errors of a tag that is moving faster than our ground truth trolley, we deliberately slow down our system by introducing some idle time. To achieve all this, we group one or more time slots plus some the necessary idle time in a frame which has a fixed length. We call this the repetition period (Fig. 11). In our experiments, it takes the values $T_{rep} \in \{0.02, 0.5, 1\}$ s. By increasing the repetition period with additional idle time, we simulate the scenario in which the tag is listening for localization messages at a lower rate or, equivalently, the situation in which the tag is moving at a higher speed.

In both experiments, we used $N = 10$ anchors and varied the number of responses $K \in \{1, 4, 9\}$. We therefore want to find the *repetition time* (or tag speed) for which more TDOA measurements compensate for the clock drift error incurred by a higher number of responses either when we have a fixed time budget (T_{TX}) or a fixed energy budget (N_{msg}). We consider that the number of received messages is proportional to the energy consumed by the tag.

For the evaluation, we perform the localization using the AlgEKF algorithm from Section III-C, which updates the location for every incoming measurement. We prefer using the AlgEKF over the AlgMin because the latter needs a minimum of four TDOA equations for one location update. If that were the case, would have to update each experiment at different rates, which can bias the results.

1) SAME TRANSMISSION TIME

We first evaluate how the 3D localization error changes when the total transmission time (T_{TX}) is constant and the number of responses varies. In all three experiments, during each repetition period, we have approximately 10.0 ms of transmission time (1 slot with 9 responses, 2 slots with 4 responses, or 3 slots with 1 response). The rest is idle time. Given our tag's speed of 10 cm/s and the repetition period of 20 ms, 0.5 s, and 1.0 s, we essentially simulate tag speeds of 20 cm/s, 5 m/s, and 10 m/s, respectively.

Fig. 12 shows the error distributions for all combinations of number of responses and number of TDOA measurements per repetition period ((K, N_{meas})) and for all repetition periods (T_{rep}). When T_{rep} is the lowest, the highest localization accuracy is obtained for the minimum number of responses, $K = 1$, as in Section V-B. However, as T_{rep} increases, so does the average localization error for $K = 1$. At $T_{rep} = 1$ s, for $K = 1$, the mean error is 10 cm/s higher and the spread is almost double compared to $K = 9$.

The error is so high for $K = 1$ because the number of TDOAs obtained every second is lower than the minimum number of TDOAs needed to obtain a location. Because the EKF updates the tag's location for every incoming measurement, we do not need to wait for the minimum number of TDOAs. However, the location estimate suffers as a result.

It is interesting to note that, for $T_{rep} = 0.5$ s, the localization error is lower for $K = 4$ and $N_{meas} = 8$ than for $K = N_{meas} = 9$. This means that one extra TDOA cannot compensate for the higher clock drift errors of $K = 9$ responses. However, at $T_{rep} = 1$ s the update rate becomes low enough such that the mean 3D error is approximately equal for $K = 4$ and $K = 9$. In this case, the error spread is actually smaller for the highest number of responses. Therefore, at a high tag speed (low update rate), it is preferable to use a high number of responses.

2) SAME NUMBER OF MESSAGES

We consider that the energy consumed by the tag is proportional to the number of received messages[3] denoted by N_{msg}. Therefore, we want to find out which configuration is optimal for a fixed number of messages (or energy consumption). Note that, although the number of messages is fixed, the

[3]Although the transmission time T_{TX} is higher for the "Same N_{msg}" configuration than for the others in Fig. 11, the transmission time also includes the time during which the responding anchors process the initiator's request and the guard time between the anchors' responses. Because this time is known, we put the tag in the idle mode between the reception of two successive responses, such that the time during which the tag is in the receive mode is shorter than T_{TX}.

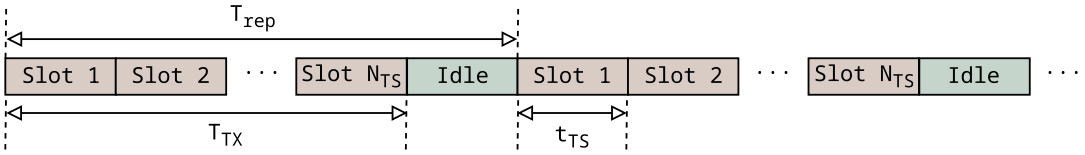


FIGURE 11. To evaluate the optimal K under various simulated situations (time constraints, energy constraints, and tag speed), we create localization frames of fixed duration T_{rep}, containing N_{TS} time slots and the required idle time. Each time slot contains one request and K responses. We do two experiments: (1) we vary N_{TS} and K while keeping the total air time ($T_{TX} = N_{TS} \cdot t_{TS}$) constant, thus simulating time constraints, and (2) we vary N_{TS} and K but we keep the total number of messages exchanged ($N_{msg} = N_{TS} \cdot (K+1)$) constant, thus simulating energy constraints. In both cases, we vary the idle time to simulate a tag moving at various speeds.

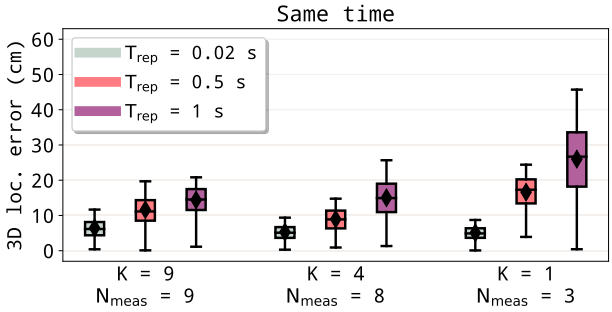


FIGURE 12. Error distributions for the same TX time (T_{TX}). By increasing T_{rep}, we simulate a higher speed of the tag.

TABLE 3. Setup for experiments with approximately the same transmission time (T_{TX}).

K	N_{TS}	N_{meas}	T_{TX} (ms)
9	1	9	10.15
4	2	8	11.80
1	3	3	10.05

TABLE 4. Setup for experiments with the same number of messages (N_{msg}).

K	N_{TS}	N_{meas}	N_{msg}
9	1	9	10
4	2	8	10
1	5	5	10

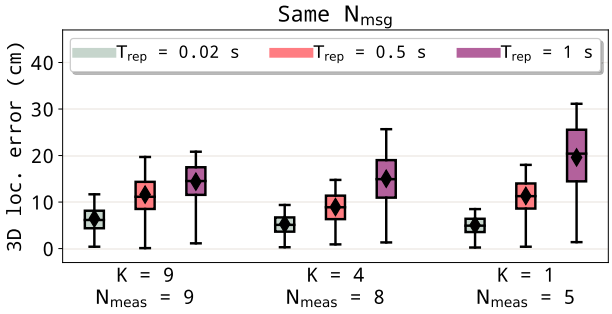


FIGURE 13. Error distributions for the same number of transmitted messages (N_{msg}).

number of TDOAs that can be extracted from these messages increases with the number of responses in one time slot.

We keep the number of messages (N_{msg}) fixed during each repetition period and we vary the repetition time as in the previous experiment. Fig. 13 shows the error distributions for a fixed number of exchanged messages. Similar to the previous case, for a high update rate, $K = 1$ is the optimal number of responses. However, as T_{rep} increases, it is more beneficial to have more TDOAs than to minimize the clock drift error. At $T_{rep} = 0.5$ s, the mean error and the error spread in all three configurations are approximately equal. Beyond this T_{rep} value, it pays off to maximize the number of responses per time slot.

In a real deployment, the number of responses cannot be adjusted based on each user's needs, since the mobile devices are passive and the localization system may not even be aware of their presence. Therefore, we should aim to find the optimal number of responses that covers a wide range of user movement patterns, update rates, and accuracy constraints. Based on our data, we suggest that using up to $K = 9$ responses is preferable to using the minimum number of responses for several reasons. First, at the maximum rate, the decrease in the localization accuracy is only 3 cm in the mean and 8 cm in the spread for $K = 9$ compared to $K = 1$. However, at $T_{rep} = 1.0$ s, the mean localization error is 10 cm lower and the IQR twice as small for $K = 9$ compared to $K = 1$. Therefore, it is preferable to use the maximum number of responses in most cases.

D. NUMBER OF ANCHORS

During the previous experiments, we kept the highest number of anchors available ($N = 10$). However, depending on the deployment, it might be unfeasible to have that many anchors available in the tracking area. Therefore, we evaluate the 3D localization error (computed as in Eq. (21)) when varying the number of anchors participating in the localization between 5 and 10.

As illustrated in the anchor setup from Fig. 6, the first four anchors (which are used in the smallest configuration) are placed close the ceiling and the fifth anchor on the ground, to ensure enough spread on the z axis. The first four anchors are placed on the perimeter of the tracking area and the fifth anchor approximately in its center. For the configurations with more anchors, we add one anchor in the order of its index (e.g., the setup with 6 anchors uses the anchors A_0 to A_5).

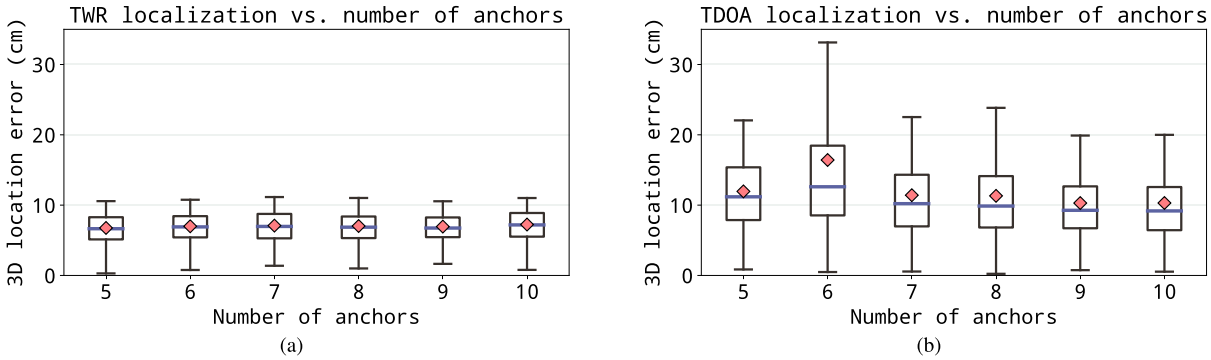


FIGURE 14. The 3D localization error of range-based localization depending on the number of devices.

In order to eliminate the influence of the number of responses on the localization error, we set the number of responses to $K = 4$ (which is the minimum number of responses for the minimum number of anchors $N = 5$).

Fig. 14 shows the 3D localization error for TWR and TDOA localization when varying the number of anchors. The general trend is that the localization error decreases for more anchors. However, the decrease is much smaller for TWR than for TDOA localization. For TDOA localization, the mean localization error with 10 anchors is 1.7 cm smaller than with 5 anchors, while the 95% error decreases with 2.1 cm. For TWR localization, the gains in the localization accuracy are even smaller.

There is an anomaly in TDOA localization, in which the errors are the highest for $N = 6$ anchors, and not for the minimum number of anchors, as one would expect. These errors are caused by the unfavorable geometry of this configuration, since anchors $\mathtt{A}_4$ and $\mathtt{A}_5$ are placed quite close to each other. This configuration results in large localization errors when the tag is furthest away from these anchors. We also noticed that TDOA localization is more sensitive to a poor anchor placement than TWR localization. This means that the deployment should be done according to a simulation of the expected localization errors for different configurations. Strategies for optimal anchor placement are outside the scope of the paper, but we refer the reader to [22], [39] for analyses of sensor placement in TDOA localization.

The improvement brought by using more anchors might seem modest, since we could perhaps expect the localization accuracy to improve more dramatically with more anchors. However, the improvement is bounded by the accuracy of the *technology*, which is reflected in the distance/TDOA errors from Fig. 8. Since the measurements for different numbers of anchors were acquired in ideal conditions, i.e., with LOS between each pair of nodes, adding more devices cannot improve the localization accuracy beyond the capabilities of the technology itself. However, having more anchors than the minimum necessary is more beneficial in challenging conditions, where part of the nodes are in NLOS with each other. We will evaluate this scenario in Section VI-B.

VI. COMPARISON OF LOCALIZATION METHODS

In this section, we compare the localization accuracy of the considered localization methods. In Section VI-A, we first compare the four variants of TDOA localization presented in Section III-D: FI-FR (or the classic TDOA), FI-CR, CI-FR, and CI-CR (or FlexTDOA). The goal is to evaluate the improvement brought by changing only the initiator, only the list of responders, or both. In Section VI-B, we compare only the classic TDOA, FlexTDOA, and TWR localization in a NLOS scenario. Throughout this section, we use the AlgMin algorithm to estimate the user's location.

A. FIXED VS. CHANGING INITIATOR AND/OR RESPONSE ORDER

In the classic (FI-FR) TDOA scheme, there is a designated reference anchor (analogous to our initiator) which broadcasts a message. All the anchors in the system respond in a predefined order with a certain time delay in order to avoid overlapping answers at the receiver. This approach is not ideal because it does not fully exploit the channel diversity of all anchor pairs. Moreover, if the tag does not have a good link to the reference anchor, all the measured TDOAs will be biased or noisy.

As we mentioned in Section III-D, we can derive three other TDOA schemes from the classic approach (denoted by FI-FR): with a fixed initiator but changing the order in which anchors respond (FI-CR), changing the initiating anchor every time slot but keeping the order of the responding anchors fixed (CI-FR), and changing both the initiator and the order of responses every time slot (CI-CR, implemented in FlexTDOA). In the FI schemes, anchor $\mathtt{A}_1$ will be the initiator. The FI-FR experiences the lowest and CI-CR the highest channel diversity. Therefore, we expect these schemes to have the worst and, respectively, the best localization accuracy, assuming that all anchors have equally good propagation conditions to the tag. However, in case the link between some anchors and the tag leads to higher errors, the CI-CR will be at a disadvantage because those anchors can become initiators and corrupt all the TDOAs in one time slot.

In this part, we evaluate to what extent the channel diversity improves the localization accuracy in LOS

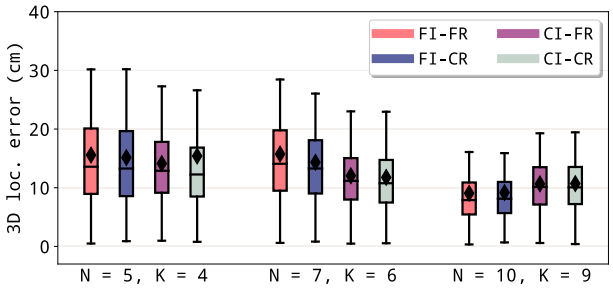


FIGURE 15. Comparison between FI-FR (classic TDOA), FI-CR, CI-FR, and CI-CR (FlexTDOA) in LOS at location P1.

conditions. We evaluate the localization errors for $(N, K) \in \{(5, 4), (7, 6), (10, 9)\}$[4] at three positions of the rail on which the tag moves, denoted by P1, P2, and P3 in Fig. 6. Position P1 is in the center of the room, parallel to the XY plane, where we should have the highest accuracy. Position P2 and P3 are inclined relative to the XY plane, so that we can evaluate the errors at multiple tag heights.

Each experiment, i.e., for each location, for every method and for each (N, K) combination, lasts one minute and a half, which gives us approx. 7000 location estimates obtained with AlgMin. When aggregating the measurements over the 3 locations of the rail, we obtain approx. 21,000 location estimates in LOS for each method and (N, K) combination. The number of actual TDOA measurements in each experiment depends on the (N, K) configuration; for instance, for $(N, K) = (10, 9)$, we obtain approx. 80,000 TDOAs in one run of the experiment.

Fig. 15 compares the four TDOA variants at location P1. The FI schemes have the highest localization errors for five and seven anchors, but the lowest for ten anchors. One explanation for the reverse trend in the case of ten anchors is the fact that, in this case, some of the anchors on the ground or on the tables will become initiators and they do not have ideal propagation conditions to all other anchors. Therefore, in this scenario, the localization errors will be slightly higher for the CI schemes than for the FI schemes, where the initiating anchor is placed at an ideal location.

We observe that changing the order of responders does not bring a significant improvement to the localization accuracy. Instead, the *initiator* plays a crucial role in the localization process because all the TDOAs in a time slot are computed with respect to its time frame. Any error in timestamping the initiator's message at the tag will affect all the TDOAs in that time slot. For this reason, from now on, we will consider only the FI-FR and CI-CR schemes, which we will alternatively call the classic TDOA and FlexTDOA, respectively. FlexTDOA is our proposed method for improving TDOA localization.

Table 5 shows the localization errors of the classic TDOA (FI-FR) and FlexTDOA (CI-CR) at the three considered locations: P1, P2, and P3. Fig. 16 shows the distributions of the localization errors for the same methods, aggregated over all considered locations. In LOS, where the channel between all anchors and the tag should be "ideal" (i.e., without obstructions), the FlexTDOA yields a similar or slightly better accuracy compared to the classic TDOA.

[4]In each case, $K = N - 1$ so that, even for a fixed order of responses, all anchors get to participate in the localization process.

TABLE 5. Median (P_{50}) and 95th percentile (P_{95}) of the 3D localization error for all combinations of changing/fixed initiators and order of responses. The errors are presented for the positions P1, P2, and P3, shown in Fig. 6.

		N = 5, K = 4		N = 7, K = 6		N = 10, K = 9	
Pos.	Method	P_{50} (cm)	P_{95} (cm)	P_{50} (cm)	P_{95} (cm)	P_{50} (cm)	P_{95} (cm)
P1	FI-FR	13.6	30.2	14.1	28.5	7.9	16.1
	CI-CR	12.3	26.6	10.8	23.0	10.1	19.4
P2	FI-FR	23.0	71.0	20.8	59.9	17.6	46.9
	CI-CR	19.1	69.4	15.2	41.4	13.8	30.6
P3	FI-FR	29.0	79.3	18.4	67.3	11.4	29.1
	CI-CR	25.2	78.1	19.1	62.3	15.6	38.2

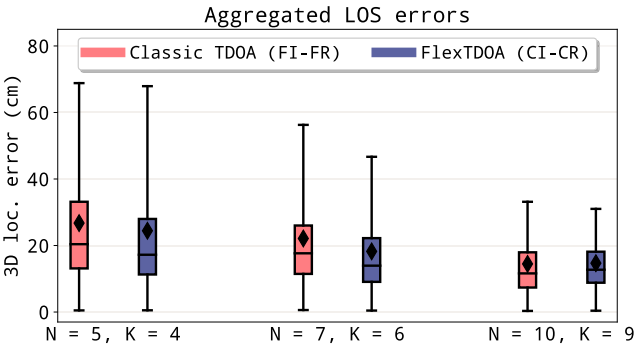
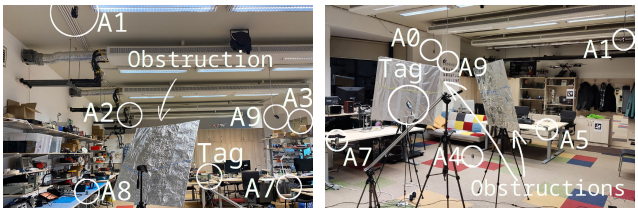


FIGURE 16. Distributions of the localization errors of classic TDOA (FI-FR) and FlexTDOA (CI-CR), in LOS, aggregated over all the evaluated positions (P1, P2, P3).



(a) (b)

FIGURE 17. Photos of the setups used to obtain NLOS measurements. The setup in Fig. 17a, which corresponds to the actuator position P1 from Fig. 6, includes one aluminum panel placed as an obstruction between anchor A_1 and the tag. The setup in Fig. 17a, which corresponds to the actuator position P4 from Fig. 6, includes two aluminum panels placed as obstructions. The rightmost aluminum panel blocks the LOS between the tag and A_1 and partially A_5. The leftmost aluminum panel blocks A_0 and A_9. There is significant interference due to multipath propagation for A_4, A_7 and A_8.

B. NLOS PROPAGATION

Obstacles between the nodes of a localization system are common in real-life scenarios. In this part, we evaluate

TABLE 6. Median (P_{50}) and 95th percentile (P_{95}) of the 3D localization error in the FI-CR and CI-CR TDOA schemes, in NLOS, in positions P1 and P4 (from Fig. 6), using the obstructions shown in Fig. 17a and Fig. 17b, respectively.

Pos.	Method	$N = 5$		$N = 7$		$N = 10$	
		P_{50} (cm)	P_{95} (cm)	P_{50} (cm)	P_{95} (cm)	P_{50} (cm)	P_{95} (cm)
P1	TWR	15.5	55.4	11.8	45.5	8.1	29.4
	Classic TDOA	16.5	45.1	17.3	48.1	14.5	48.8
	FlexTDOA	15.5	44.6	11.8	35.4	10.6	32.4
P4	TWR	26.9	64.4	24.0	60.7	23.7	44.4
	Classic TDOA	55.5	167.7	32.5	89.6	32.6	106.9
	FlexTDOA	40.3	176.5	22.5	79.3	22.3	59.4

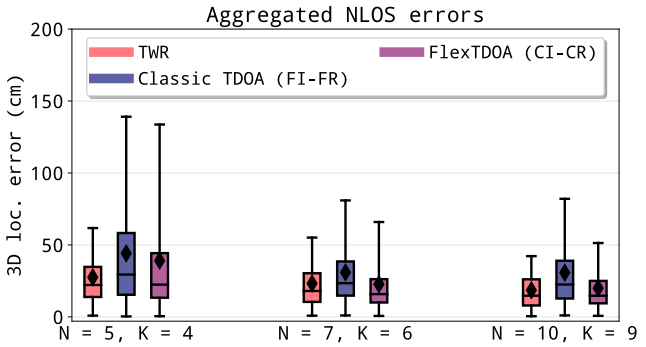


FIGURE 18. Distributions of localization errors in NLOS using TWR, classic TDOA (FI-FR), and FlexTDOA (CI-CR), aggregated over both NLOS scenarios (at location P1, with one obstruction, and at location P4, with two obstructions).

the NLOS performance of the three localization approaches considered so far: based on TWR, the classic TDOA, and FlexTDOA.

We performed measurements at two positions of the rail on which the tag moves. The positions are denoted by P1 and P4 in Fig. 6. At P1, we placed a panel covered in aluminum foil between the anchor $\mathtt{A}_1$ and the tag, shown in Fig. 17a. At P4, we placed two such panels, shown in Fig. 17b. The panel on the left can obstruct multiple anchors depending on the tag's position on the actuator. The panel on the right obstructs the direct path to anchor $\mathtt{A}_1$ at all times.

We perform the experiments for $N \in \{5, 7, 10\}$ anchors. For TDOA localization, we use $K = N - 1$ responses. For TWR, we use $K = N$ responses. For each algorithm and each (N, K) combination, we have approx. 14,000 location estimates in NLOS (7,000 at each position of the rail).

At the moment, we do not implement any NLOS mitigation procedure and we take into account all the measurements.

Table 6 shows the median and the 95th percentile (P_{95}) 3D localization errors of each method at the positions P1 and P4. Fig. 18 shows the distribution of 3D localization errors aggregated over both locations. FlexTDOA achieves lower errors than the classic TDOA in all NLOS scenarios. With five anchors, FlexTDOA has only a modest improvement of 5 cm to 7 cm in the median and P_{95} errors compared to the classic TDOA. However, the improvement is more evident for seven and ten anchors, where FlexTDOA reduces the P_{95} error by 19% and 38%, respectively, compared to the classic TDOA.

As expected, TWR has the smallest errors in all scenarios. This happens because, in TWR-based localization, an obstacle placed between an anchor and the tag introduces an error only in the distance between them. Therefore, the bias in the location estimate using TWR can be reduced if the rest of the distance measurements have small errors.

On the other hand, in TDOA localization, if the obstacle is between the initiating anchor and the tag, it incurs an error in *all* the TDOAs from that time slot. This is where FlexTDOA is more advantageous than the classic TDOA: by changing the initiating anchor, we ensure enough channel diversity to improve the robustness of the location estimate if the initiating anchor is obstructed.

Even though TWR-based localization provides the most accurate location estimates, it scales poorly with the increasing number of tags. For $N = 9$ anchors, the minimum duration of a time slot needed to obtain measurements between all anchors and the tag is approx. 10 ms (see Table 3). Therefore, with TWR, at most 100 tags could be localized per second. In fact, the number of tags would likely be smaller given that such a scheme would need a multiple access protocol to synchronize their access to the channel. TDOA-based localization, on the other hand, can scale to an unlimited number of tags. Using the same example, each tag could obtain 100 measurements per second independent of the number of tags in the area.

VII. CONCLUSION

In this paper, we presented a new flexible TDMA scheduling scheme for TDOA localization that fully exploits the channel diversity in the environment. We compared FlexTDOA, the proposed method, against the classic TDOA implementation with a fixed reference anchor and responder list and against range-based localization in a deployment of up to ten anchors and one tag in an office environment.

FlexTDOA achieves lower localization errors than the classic TDOA in most scenarios, with and without obstructions. In LOS, the improvement in the median accuracy brought by FlexTDOA compared to the classic TDOA is modest (2 cm to 3 cm) because the initiator in the classic TDOA already has a good link to the tag. However, the robustness brought by the increased diversity is evident in NLOS. In NLOS, FlexTDOA reduces 95th percentile of the localization error with up to 38% compared to the classic TDOA. Overall, FlexTDOA achieves a median localization error of 13 cm to 17 cm in LOS and 15 cm to 22 cm when one or more anchors are in NLOS with the tag (the error depends on how many anchors are used).

While TWR localization yields the highest accuracy among all methods, it has poor scalability with a growing

number of anchors and responders. In contrast, FlexTDOA can scale to an unlimited number of tags.

In the future, we will scale up the proposed system to a multi-room or building environment. This will require several issues to be addressed: pairs of anchors that are not in communication range, system calibration (self-localization) for the sparsely connected network, and an efficient TDMA scheme which reuses slots for out-of-range nodes.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of NXP Romania for providing valuable insights regarding trends for the IoT solutions.

REFERENCES

[1] *Unleashing the Potential of UWB: Regulatory Considerations*, FiRa Consortium, Beaverton, OR, USA, Aug. 2022.

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

[3] A. Urruela, J. Sala, and J. Riba, "Average performance analysis of circular and hyperbolic geolocation," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 52–66, Jan. 2006.

[4] 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, May 2022.

[5] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009.

[6] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 6, Apr. 2003, p. 553.

[7] L. Santoro, M. Nardello, D. Brunelli, and D. Fontanelli, "Scale up to infinity: The UWB indoor global positioning system," in *Proc. IEEE Int. Symp. Robotic Sensors Environ. (ROSE)*, Oct. 2021, pp. 1–8.

[8] J. Tiemann, F. Eckermann, and C. Wietfeld, "Multi-user interference and wireless clock synchronization in TDOA-based UWB localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–6.

[9] J. Friedrich, J. Tiemann, and C. Wietfeld, "Accurate multi-zone UWB TDOA localization utilizing cascaded wireless clock synchronization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Nov. 2021, pp. 1–8.

[10] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3131–3137.

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

[12] J. Tiemann and C. Wietfeld, "Scalability, real-time capabilities, and energy efficiency in ultra-wideband localization," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6313–6321, Dec. 2019.

[13] Qorvo. *DWM3000 6.5 & 8.0 GHz Ultra-Wideband (UWB) Module*. Accessed: Nov. 15, 2022. [Online]. Available: https://www.qorvo.com/products/p/DWM3000

[14] M. Pelka and H. Hellbrück, "S-TDoA—Sequential time difference of arrival—A scalable and synchronization free approach forl positioning," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.

[15] B. Großwindhager, M. Stocker, M. Rath, C. A. Boano, and K. Römer, "SnapLoc: An ultra-fast UWB-based indoor localization system for an unlimited number of tags," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 61–72.

[16] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 133–144.

[17] J. Tiemann, F. Eckermann, and C. Wietfeld, "ATLAS—An open-source TDOA-based ultra-wideband localization system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–6.

[18] J. Tiemann and C. Wietfeld, "Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–7.

[19] J. Tiemann, Y. Elmasry, L. Koring, and C. Wietfeld, "ATLAS FaST: Fast and simple scheduled TDOA for reliable ultra-wideband localization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2554–2560.

[20] J. Yang, B. Dong, and J. Wang, "VULoc: Accurate UWB localization for countless targets without synchronization," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 3, pp. 1–25, Sep. 2022.

[21] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco, "TALLA: Large-scale TDoA localization with ultra-wideband radios," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.

[22] W. Zhao, A. Goudar, and A. P. Schoellig, "Finding the right place: Sensor placement for UWB time difference of arrival localization in cluttered indoor environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6075–6082, Jul. 2022.

[23] A. Prorok, P. Tomé, and A. Martinoli, "Accommodation of NLOS for ultra-wideband TDOA localization in single- and multi-robot systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Sep. 2011, pp. 1–9.

[24] B. Van Herbruggen, J. Fontaine, and E. D. Poorter, "Anchor pair selection for error correction in time difference of arrival (TDoA) ultra wideband (UWB) positioning systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Nov. 2021, pp. 1–8.

[25] B. Teague, Z. Liu, F. Meyer, A. Conti, and M. Z. Win, "Network localization and navigation with scalable inference and efficient operation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2072–2087, Jun. 2022.

[26] I. Dotlic, A. Connell, and M. McLaughlin, "Ranging methods utilizing carrier frequency offset estimation," in *Proc. 15th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2018, pp. 1–6.

[27] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, and U. Hugentobler, "Error corrections for ultrawideband ranging," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 11, pp. 9037–9047, Nov. 2020.

[28] Z. Ebadi, C. Hannotier, H. Steendam, F. Horlin, and F. Quitin, "An over-the-air CFO-assisted synchronization algorithm for TDOA-based localization systems," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, Nov. 2020, pp. 1–5.

[29] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[30] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 679–689, Apr. 2000.

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

[32] J. Tiemann, J. Friedrich, and C. Wietfeld, "Experimental evaluation of IEEE 802.15.4z UWB ranging performance under interference," *Sensors*, vol. 22, no. 4, p. 1643, Feb. 2022.

[33] J. O. Smith and J. S. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 12, pp. 1661–1669, Dec. 1987.

[34] *DW1000 Device Driver Application Programming Interface (API) Guide: Using API Functions to Configure and Program the DW1000 Transceiver Version 2.7*, Decawave, Burlingame, CA, USA, 2021.

[35] *DW3xxx Device Driver Application Programming Interface (API) Guide: Using API Functions to Configure and Program the DW3000 and QM33120 UWB Transceiver Version 2.2*, Decawave, Burlingame, CA, USA, 2021.

[36] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005.

[37] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2020 (Revision of IEEE Standard 802.15.4-2015), 2020, pp. 1–800.

[38] ST Microelectronics. *STM32F429ZI—High-Performance Advanced Line, Arm Cortex-M4 Core With DSP and FPU, 2 Mbytes of Flash Memory, 180 MHz CPU, Art Accelerator, ChromART Accelerator, FMC With SDRAM, TFT*. Accessed: Nov. 15, 2022. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f429zi.html

[39] J. T. Isaacs, D. J. Klein, and J. P. Hespanha, "Optimal sensor placement for time difference of arrival localization," in *Proc. 48th IEEE Conf. Decis. Control (CDC), 28th Chin. Control Conf.*, Dec. 2009, pp. 7878–7884.

GEORGE-CRISTIAN PĂTRU is currently pursuing the Ph.D. degree in the IIoT and robotics with the Automatic Control and Computer Science Faculty, University Politehnica of Bucharest. He has gained experience over the last four years in hardware development while working with several hardware-centric start-ups. In his role as a technical mentor, he helped them quickly build and scale live prototypes. Such prototypes include LoRaWAN-based IIoT sensors, custom-built sensors with high autonomy, the IoT-based monitoring in smart buildings, specialized wearable devices, autonomous drones, and terrestrial robots.

LAURA FLUERATORU received the M.Sc. degree in electrical engineering from ETH Zürich, Switzerland, in 2019. She is currently pursuing the double Ph.D. degree with the University Politehnica of Bucharest, Romania, and Tampere University, Finland, as a Marie Skłodowska-Curie Fellow in the A-WEAR project. During her studies, she gained experience in both industry and research from internships with EPF Lausanne, Schindler Group, and NXP Semiconductors. Her research interests include indoor localization, ultrawideband technology, wireless and mobile communications, embedded systems, and machine learning.

IULIU VASILESCU was born in Tulcea, Romania. He received the Ph.D. degree in computer science and robotics from the Massachusetts Institute of Technology (MIT), Boston, MA, USA, in 2009. He worked on underwater autonomous robots, underwater optical communications, underwater imaging, and underwater acoustic communication and localization. He pioneered the use of underwater robots in synergy with underwater sensor networks.

DRAGOŞ NICULESCU received the Ph.D. degree in computer science from Rutgers University, NJ, USA, in 2004, with a thesis on sensor networks routing and positioning. He spent five years as a Researcher with the NEC Laboratories America. Princeton, NJ, working on the simulation and implementation of mesh networks, VoIP, and Wi-Fi-related protocols. He is currently with the University Politehnica of Bucharest teaching courses in mobile computing and services for mobile networking and also researching mobile protocols, UWB, and 802.11 networking.

DANIEL ROSNER received the Ph.D. degree in computer science and engineering, on assisted living technologies for ubiquitous health, with a focus on IoT devices. He has ample experience in developing the IoT and IIoT (Industrial Internet of Things) hardware and software systems while working with start-ups, aiming to develop their prototypes or validate their technology assumptions. He is an Associate Professor with the Automatic Control and Computer Science Faculty, University Politehnica of Bucharest. His latest research projects featured IoT infrastructure deployment solutions, energy monitoring for smart homes, and embedded software solutions for automotive-grade microcontrollers.