Anchor-oriented Time and Phase-based Concurrent Self-localization using UWB Radios

Nour Smaoui
University of Houston
nour@cs.uh.edu

Milad Heydariaan
University of Houston
milad@cs.uh.edu

Omprakash Gnawali
University of Houston
gnawali@cs.uh.edu

***Abstract*—Positioning plays an important role in many IoT applications. Ultra-wideband (UWB)-based positioning is an alternative to GPS in indoor environments due to its multipath resilience and its accuracy and precision. In the presence of a large number of targets to localize, conventional UWB localization fails to provide a practical location update rate. UWB concurrency in conjunction with self-localization has been used in both time-based and phase-based localization making the targets to localize (tags), a relatively passive device with the sole role of receiving wireless packets and calculating its own location. More recently, phase-based concurrent angle estimation also offloaded the hardware complexity and cost to the anchors instead of the tags. In this work, our anchor-oriented approach combines inter-anchor and intra-anchor concurrency for phase-based localization but also allows time-based localization. Our experimental evaluation on a testbed consisting of Decawave platform shows that our technique is not only practical but also performant.**

***Index Terms*—UWB, concurrency, localization, time synchronization, TDOA, AOA**

I. Introduction

Location-based applications are being widely used to optimize different aspects of our life especially with the continuous increase of mobile connected objects ranging from vehicle navigation to asset tracking in construction, commercial and industrial environments [1], [2].

While Global Navigation Satellite Systems (GNSS) work adequately for most outdoor localization applications requirements, they often work poorly in indoor environments due to path loss through materials and the impact of multipath components on the wireless signal resulting from the buildings structures. Localization in indoor environments requires position estimation with a higher accuracy especially in small environments. Recently, many wireless technologies like WiFi, BLE, RFID etc, were presented as an alternative to GNSS in indoor environments. However, they suffer significantly in harsh environments with the presence of non-line-of-sight (NLOS) conditions. Ultra-wideband (UWB) is a good candidate for indoor UWB localization due to its high accuracy, precision, and NLOS resilience. UWB radios (like Decawave DW1000 chip) are impulse radios with wide bandwidths (>500 MHz) resulting in high capabilities in accurately and precisely estimating the packet time of arrival (TOA) with a timestamping resolution of $\sim 15.6ps$ (equivalent to 0.5 cm) [3]. This allowed the design of multiple time-based localization approaches like Two-way Ranging (TWR) and Time Difference of Arrival (TDOA) which are time of flight (TOF) based techniques. UWB radios also enable accurate phase-based localization methods like Angle of arrival (AOA) or Angle difference of arrival (ADOA). The wideband in UWB is not only useful to obtain accurate TOA but it is also necessary to receive accurate phase information. With narrowband technologies like WiFi or BLE, the impact of multipath components on the phase information can result in wrong phase measurements [4].

The localization techniques used in UWB require packet exchanges between the static Anchors and the mobile localization targets (tags). To cover multiple targets, conventional MAC protocols like ALOHA or TDMA can be used. However, with a large number of tags, these MAC protocols do not scale and usually result in a reduced location update rate. As a solution, concurrent transmissions were introduced. By quasi-simultaneously transmitting packets from multiple sources, UWB can detect the time of arrival of each packet from the channel impulse response (CIR) information. Concurrent localization was also combined with a GPS-like self-localization model to scale the system even further [5]–[9].

However, with the introduction of concurrency, we face multiple problems for both time-based and phase-based localization:

- **Time-based localization problems:** time-based localization loses considerably in accuracy when concurrency is applied. The main sources of errors are the CIR resolution and the TX scheduling uncertainty problem.
- **Phase-based localization problems:** Initially, concurrent phase-based localization required the tags to have dual-antennas resulting in an increased cost and complexity [10]. Later work proposed an architecture that uses dual-antenna anchors and single-antenna tags [11]. However, it only investigated the concurrency within the dual-antenna transmitter and did not consider the concurrency between multiple transmitters.

In this work, we propose a concurrent anchor-oriented time of arrival and angle estimation system. Our objective is to design a concurrent localization system that combines both inter-anchor and intra-anchor concurrency for phase-based localization (AOA) but also allows time-based localization (TDOA) with improved accuracy offloading therefore all complexity (protocol communication, design, cost, energy

consumption) to the anchors. This approach results not only in simpler tags but also in improved accuracy, efficiency, and scalability. Furthermore, our system can maintain a good accuracy in the presence of large transmission delays.

We implement and evaluate our system on Decawave platform. Our results suggest that our system can simultaneously estimate TDOA and AOA. AOA estimation presents an accuracy ($\pm 8°$) comparable to the state-of-the-art systems that can estimate AOA seperately while TDOA estimation can achieve a precision reaching a maximum error of $\pm 1.2\ m$ which is comparable to TDOA systems that have large transmission delays.

In this paper, we make these contributions:

- Present a TX Scheduling Uncertainty Mitigation protocol and present analytical results to quantify its contribution in error reduction.
- Present the first design of anchor-oriented concurrent UWB design that can perform both distance difference (TDOA) and angle estimation (AOA) and present analytical results to quantify its contribution in accurate angle and time of arrival estimation.
- Evaluate our system in an apartment and an academic building under multiple ground truth angles and distances.

II. Related Work

Wireless interference is one of the main factors affecting the performance of wireless systems. There is a body of research on improving the performance of wireless systems under interference, including studies on avoiding interference by TDMA schemes, ALOHA, or combining localization and communication traffic [12]; mitigating interference [13], [14]; and exploiting interference [15]–[17]. More recently, researchers used (quasi-)simultaneous transmissions with UWB radios for indoor localization to increase air time efficiency by simultaneously extracting time of arrival (TOA) or phase of arrival (POA) of multiple transmitters from the combined from the channel impulse response (CIR) of the concurrently received packets [5]–[11], [18].

A. UWB Concurrent Time-based Localization

Extracting TOA/TDOA from the CIR of concurrently received packets is possible by finding separate peaks from the CIR amplitude belonging to multiple transmitters. Traditional ranging systems make use of the ability of UWB radios to measure TOA of the arriving packet. Concurrency-based ranging systems measure range with multiple UWB transceivers by measuring the difference between TOA of later arriving packets and TOA of the first arriving packet [5], [7], [9], [18]. TOA-based solutions are not scalable in terms of the number of tags as they require two-way message passing between anchors and tags, and CIR can only accommodate a limited number of concurrent signals. Concurrency-based TDOA systems measure TDOA of later arriving packets with respect to the first arriving packet and build GPS-like indoor self-localization systems [6], [8]. Concurrent TOA/TDOA systems suffer from lower accuracy caused by TX scheduling uncertainty in UWB radios [5], [7], [9]. Solutions for TX scheduling uncertainty problem either assume certain implementation (wired correction) or have one-packet cycle delay (wireless correction) [8]; or do not support moderately large transmission delays due to clock drift [19].

B. UWB Concurrent Phase-based Localization

Extracting POA or phase difference of arrival (PDOA) from the CIR of concurrently received packets is possible with two UWB radios running on a single clock as first introduced by Decawave [4]. Concurrency-based AOA or angle difference of arrival (ADOA) localization systems measure the phases of (quasi-)simultaneously arriving packets at the receiver. AnguLoc [10] builds an ADOA-based self-localization system by measuring PDOA of concurrently arriving packets at tags with two UWB radios running on the same clock. Further, another study introduces intra-anchor concurrency (packets concurrently transmitted by multiple radios on the same anchor node) and proposes to reduce the platform cost by measuring the PDOA of arriving signals from two UWB radios running on the same clock and builds a single-antenna AOA estimation system [11]. Our work explores the possibility of combining the idea of intra-anchor concurrency with inter-anchor concurrency (packets concurrently transmitted from multiple anchors) to measure AOA from multiple dual-UWB radio anchors on single-antenna tags. Another research study combines AOA with ranging (TOA) to build a single-anchor localization system [18]. However, this solution is not scalable in terms of the number of tags as it requires two-way message passing between anchors and tags, and CIR can only accommodate a limited number of concurrent signals.

III. System Design

A. Background

1) Transmission synchronization in concurrent UWB: In concurrent localization systems, it is mandatory to have some level of synchronization to transmit the concurrent packets correctly. The transmitters can be synchronized based on a common event. In this case, the event is represented by the reception of $Sync$ packet from an independent transmitter that we call the initiator. Transmissions can be further synchronized by performing clock drift corrections. There are two approaches to correct the clock drift α using either single packet transmission or two packets transmissions [5], [20]. We opt to use the single packet method that extracts α from the carrier recovery integrator register as it considerably reduces the packet transmission overhead.

2) Conventional UWB angle estimation: The angle between a transmitter and a receiver in UWB can be estimated by calculating the path difference of a received signal on two antennas (UWB chips) of the same receiver. It has been established in the literature that the path difference can be calculated based on the phase difference extracted from the CIR of each receiving chip. Further details can be found in [10], [11], [21].

B. Overview

To make our system as scalable and efficient as possible, we considered the **concurrent self-localization architecture** where the tags act as receivers and anchors synchronize themselves and transmit concurrently. Such **inter-anchor** concurrency allows the deployment of an "unlimited" number of tags (receivers only) and an increase in the number of anchors while keeping an efficient air-usage. Another architectural feature of our system is having **dual-antenna anchor and single-antenna tags**. By having single-antenna tags, we can reduce the overall cost of the system since the number of tags is typically larger than the number of anchors. Also, due to the restrictions on the antenna separation, if tags had dual antennas, they can become bulky and consume more energy.

C. AOA estimation using intra-anchor concurrency

To perform angle estimation under the dual-antenna anchor and single-antenna tag architecture, we adopt the solution proposed by SA-AOA [11]. As opposed to the conventional AOA estimation system, in SA-AOA, the dual-antenna anchor transmits concurrently two separate packets, each from one of its antennas. Since both chips share the same clock, transmissions are already synchronized and do not require further improvement. The transmitted packets are separated by a small delay in the order of nanoseconds. The delay should not exceed the duration of CIR ($1016\ ns$), also called the concurrency window. When the tag receives the concurrent packets, it identifies the peaks of the concurrent packets from the CIR, extracts the phase information of each peak, and calculates the $PDOA$ that will be used for angle estimation. In this system, an intra-anchor delay is added between the transmission to ensure the reception of both concurrent peaks. Although it is possible to increase the intra-anchor delay to make the packet reception non-concurrent, the delay will be large enough to be impacted by clock drift and results in wrong phase measurements.

D. TX precision improvement in inter-anchor concurrency

1) Transmission Scheduling Uncertainty Mitigation (TSUM) Protocol: We first explain the TX scheduling uncertainty problem in conventional concurrent localization. Concurrent transmissions are scheduled based on the reception timestamp of the $Sync$ message from the initiator (a 40-bit number representing the clock ticks). The timestamp resolution is $\sim 15.6\ ps$ (1 tick). The transmission timestamp of each anchor is computed using the reception timestamp of the received $Sync$ message. As a result, the calculated TX timestamps have the same resolution as the RX timestamps. However, the DW1000 chip can only support a transmission timestamp resolution of $8\ ns$. This means that the resolution of the calculated TX timestamps needs to be dropped to match the transmission resolution. This drop is done by removing the 9 lower bits (512 ticks) of the calculated TX timestamp. This results in a precision loss of $\pm 8\ ns$. As an example, if the first transmitting anchor has 511 ticks in its 9 lower bits and the second anchor has 0 ticks in its 9 lower bits, then the second peak will show on the CIR after $\sim 8\ ns$ from its expected position since the first packets will be sent earlier by 511 ticks.

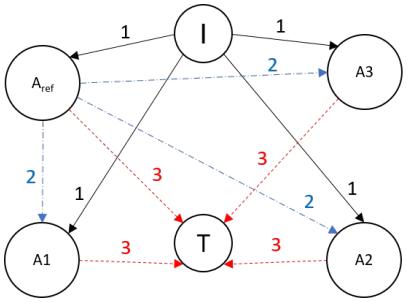

Fig. 1: Concurrent transmissions protocol with TSUM. The initiator I sends a $Sync$ message (#1). Anchor A_{ref} sends its scheduled transmission timestamp to all other anchors (#2). Anchors A_{ref} and $A1 - A3$ calculate their timestamps after correction and transmit their concurrent packets .

In this work, we aim to make the 9 lower bits of all the anchors to be similar to each other as much as possible. We propose the TSUM solution where we fix a delay for one of the anchors and try to match it on the remaining anchors. Figure 1 shows the packet exchange in our correction method. First, the initiator sends the $Sync$ message to all anchors (#1). We select one of the anchors to be a reference anchor. This anchor calculates its TX timestamp using Equation 1 where TX_{ref} is the scheduled TX timestamp of the reference, RX_{ref} is the $Sync$ reception timestamp by the reference, and Δ_{ref} is a fixed delay assigned to the reference anchor. Then the calculated TX_{ref} is sent to the neighboring anchors (#2).

$$TX_{ref} = RX_{ref} + \Delta_{ref} \tag{1}$$

After reception of packet #2, the remaining anchors are aware of TX_{ref} and the value of its 9 lower bits. They also estimate α_i, the clock drift between the reference anchor and their own clock where i is the anchor number. Then using Equation 2 they calculate their TX timestamps TX_i where RX_i is the $Sync$ reception timestamp by anchor i and Δ_i is a selected candidate delay specific to anchor i that we calculate based on our algorithm that we explain in the following sections.

$$TX_i = RX_i + \Delta_i \times (1 + \alpha_i) \tag{2}$$

2) CIR Repetition Exploitation: In conventional concurrent localization, each anchor is assigned a fixed delay to schedule its transmission. This is when the problem of TX scheduling uncertainty arises. We aim in our correction to find multiple delays that ensure detecting the packet at its designated index in the CIR. Delays at the level of each anchor are calculated based on equation 3. ε_i is a small delay $< 1016\ ns$ that guarantees the detection of the peaks within the same CIR. It is usually set to be at least twice the maximum distance in the localization environment between a tag and anchor to avoid overlapping peaks of different anchors.

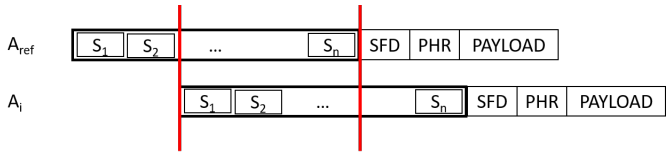

Fig. 2: Delayed concurrent transmissions by m preamble repetitions.

$$\Delta_i = \Delta_{ref} + i \times \varepsilon_i \tag{3}$$

At the same time, CIR is the result of correlation and accumulation of multiple preamble symbols. A concurrent CIR is built based on overlapping symbols from multiple anchors. So even when the preamble symbols of the concurrent transmissions are delayed by one or more preamble symbol duration(s) (PSD) (also known as preamble repetition), we can still detect them on the CIR. As Figure 2 shows, anchor A_i can transmit its packet after m ($m < preamble\ size$) skipped preamble symbols (S_i) and both A_{ref} and A_i preamble transmissions are still overlapping (area between two red bars). This allows us to generate a set of candidate timestamps for each anchor according to equation 4 where $n \in [0..preamble\ length]$. The PSD in the configurations that we use is typically $\sim 1016\ ns$ (65024 ticks). We optimistically search among the candidate timestamps for one that reduces the difference between the 9 lower bits of the reference timestamp and the 9 lower bits of each anchor i timestamp.

$$\Delta_i^n = \Delta_i + i \times \varepsilon_i + n \times PSD \tag{4}$$

3) Rule-based Correction: We optimize further the error reduction by applying a rule-based correction method. This method guarantees that the error does not exceed $4\ ns$ of error. The 9 lower bits of each timestamp represent the precision loss during the transmission of any packet. Let the 9 lower bits loss in the reference timestamp be E_{ref} and the 9 lower bits loss in anchor i timestamp be E_i. We can reduce the pairwise loss by modifying the 10^{th} bit of anchor i timestamp according to algorithm 1 which is equivalent to adding or subtracting 512 ticks to the timestamp. The rule-based correction is only valid when the difference between the 9 lower bits of anchor i and the reference anchor is larger than 256 ticks. We can identify two cases. First, when the 9 lower bits of the reference anchor are lower than 256 ticks. This means that the 9 lower bits of anchor i are larger than 256 ticks, and therefore, by adding a few ticks to the timestamp, the resulting error is lower than 256 ticks ($\sim 4\ ns$). Similarly, in the second case, when the 9 lower bits of the reference anchor are larger than 256 ticks, we reduce the 10^{th} bit of anchor i timestamp.

We summarize the entire procedure to find the best TX timestamp in algorithm 2. We generate candidate timestamps for each candidate delay using CIR repetitions. We correct each candidate timestamp using our rule-based method. Then, we select the timestamp that minimizes the error.

Algorithm 1 TX Timestamp Correction

Input: TX_{ref}, TX_i
Output: CTX_i, CE
  function CORRECTTS(TX_{ref}, TX_i)
    $E_{ref} \leftarrow Get9LowerBits(TX_{ref})$
    $E_i \leftarrow Get9LowerBits(TX_i)$
    if $|E_{ref} - E_i| > 256$ **then**
      if $E_{ref} > 256$ **then**
        $CTX_i \leftarrow DecrementBit10(TX_i)$
      else
        $CTX_i \leftarrow IncrementBit10(TX_i)$
      end if
      $CE \leftarrow CalculateError()$
    end if
    return CTX_i, CE_i
  end function

Algorithm 2 TSUM algorithm

Input: TX_{ref}, RX_i, i
Output: CTX_i
  function FINDBESTTIMESTAMP(TX_{ref}, i)
    $EMax \leftarrow 512$
    for $n \leftarrow 0$ to $PreambleLength$ **do**
      $\Delta_i \leftarrow \Delta_{ref} + i \times \varepsilon_i + n \times PSD$
      $TX \leftarrow RX_i + \Delta_i$
      $TX, E \leftarrow CorrectTs(TX_{ref}, TX)$
      if $E < EMax$ **then**
        $EMax \leftarrow E$
        $CTX_i \leftarrow TX$
      end if
    end for
    return CTX_i
  end function

E. Overall architecture

We describe in this section the combination of our system architectures to be able to perform both angle and distance difference estimations concurrently. We first describe the communication protocol running on the anchors. Then, we detail the tag estimation process.

1) Communication protocol:
In our system, we have 4 different entities that interact with each other. Figure 3 shows the interactions between these entities. The initiator I sends the broadcast message to synchronize all the anchors (#1). We define A_i^j as the transmitting chip/antenna i in the anchor j. The synchronization message is received by A_1^j for each anchor, and the reception timestamp is recorded. Then the reference anchor calculates its TX timestamp for its first chip A_1^{ref} and transmits it to all the anchors (#2). the reference anchor also schedules the TX timestamp for the second chip A_2^{ref} by adding the intra-anchor delay. Once received, the anchors find the optimal TX timestamp for their first chip A_1^j and similarly calculate the TX timestamp for the second chip A_2^j by adding the intra-anchor

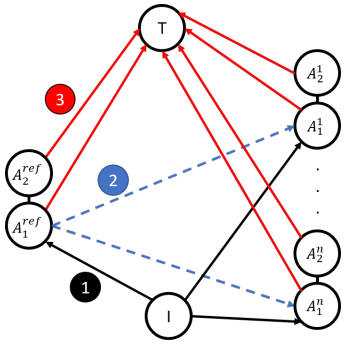

Fig. 3: Combined inter-anchor and intra-anchor concurrency.

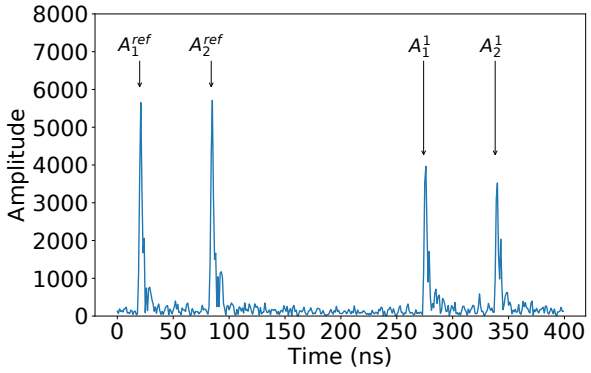

Fig. 4: CIR with 4 concurrent transmitters. The first pair of peaks represent intra-concurrent transmissions by the reference anchor and the second pair of peaks represent another intra-anchor concurrency by anchor 1.

delay. The schedule packets are then transmitted concurrently (#3). The intra-anchor delay is set to be a multiple of 8 ns to keep the same 9 lower bits similar for both A_1^j and A_2^j. Figure 4 shows an example of the CIR with both inter-anchor and intra-anchor concurrency with 2 anchors. As annotated in the figure, the first pair of peaks represent the concurrent packets transmitted by the reference anchor separated by a 64 ns intra-anchor delay. The second pair represents the second anchor. We can also notice that the first and third peaks are separated by the inter-anchor delay (256 ns in this example). According to previous work, the minimum value of intra-anchor that we can consider is 64 ns to be able to accurately estimate angles, especially in the presence of strong multipath components. While the inter-anchor delay depends strongly on the dimensions of the localization space. The delay between the second peak and the third peak needs to be at least twice the largest distance in our localization space. For example in this case, we have 192 ns $(256 - 64)$ of delay, making the maximum distance to be 28.8 m (96 ns).

2) Distance difference and angle estimation: When the tag receives the concurrent CIR (example in Figure 4), it extracts two sets of information: Angle of arrival measurements and pair-wise anchor distance difference.

Our system starts by identifying the indexes of each peak in the CIR. We extracted the CIR pulse shape by collecting 50 CIR samples. The pulse shape is used as a template that we correlate with the CIR to extract the indexes with the highest correlation.

Distance difference measurements can be extracted by calculating the time difference between the first peaks of each anchor and then removing the inter-anchor delay. Since 1 ns represents 30 cm, we can calculate the distance difference (in cm) by simply multiplying the time difference by 30.

As for the angle of arrival measurements, for each intra-concurrent pair of peaks, we extract the phase at their respective detected indexes, and we calculate the phase difference that we can covert to angle measurements as explained earlier. Further details about the angle of arrival measurements can be found in [11].

IV. Evaluation

A. Experimental setup

In our experiments, we used Decawave DWM1002 as our dual-antenna anchor nodes [21] and TREK1000 [20] as our tag and initiator. We conducted the experiments in a hallway in a university building. We placed all the nodes on tripods at a height of 1.5 m. We used frequency channel 5, with preamble lengths of 128, a data rate of 6.8 $Mbps$, and PG_DELAY of 0xC0, which is a recommended configuration for concurrency-based studies in the literature. We studied our system using four anchors, one initiator, and one tag.

B. Metrics

We use three primary metrics in our study of transmission synchronization errors and their correction.

RX timestamp estimation error: The difference between the time when a CIR peak is expected if the transmitter had no error and when the peak is detected.

Time precision loss: Difference between 9 lower bits of reference anchor timestamp and anchor A_i timestamp. This quantifies the jitter resulting from the TX scheduling uncertainty. The maximum Time precision loss in our setting is 256 ticks, i.e., ± 4 ns.

Total non-precision time error: The error in the estimate of CIR peak time resulting from factors other than Time precision loss. E.g., CIR resolution, Clock drift, platform jitter.

Thus, RX timestamp estimation error = Time precision loss + Total non-precision time error

C. Impact of TSUM on RX Timestamp Estimation

First, we investigate the overall impact of TSUM on timestamp estimation and compare it to the case with no correction. Figure 5 shows the CDF of timestamp estimation error. We can see that overall, TSUM reduces the amount of error (improvement: 43% at 90^{th} percentile, 33% in average).

Next, we analyze this result in further detail to understand the impact of each component in our algorithm on the timestamp correction. We look at the contributions by the combinations of mechanisms comprising TSUM as follows:

	RX timestamp estimation error (ns)			
Technique	**Avg**	**Std**	**50th**	**90th**
RB	2.17	1.31	2.43	3.76
REP	2.00	1.19	2.13	3.54
RB + REP	2.51	1.11	2.64	3.80
NRI	1.84	1.21	1.75	3.60

TABLE I: Breakdown of RX timestamp estimation errors with different techniques used by TSUM

- **RB:** Best timestamp candidate found using only the rule-based correction
- **REP:** Best timestamp candidate found using only the CIR repetitions
- **REP+RB:** Best timestamp candidate found using a combination of rule-based correction and CIR repetitions

Note that at a small fraction of times, TSUM correction does not change the estimate if it is already optimal and there is no room for improvement (**NRI**)

As a matter of fact, table I shows that for all the techniques used by TSUM, the average RX timestamp estimation is $\sim$ 2 ns and the 90^{th} percentile lower than 4 ns. The table also shows that the different components contributed largely equally towards the overall performance of TSUM.

D. Total Non-precision Time Error

Our correction method proposed in algorithm 2 attempts to reduce the time precision loss caused by the TX scheduling uncertainty. The algorithm chooses the best candidate time that reduces the time precision loss in the 9 lower bit between the reference timestamp and the anchor timestamp. To verify that our method works correctly, we ran our system with a preamble size of 128. Our anchors are set up to be equidistant from both the initiator and the tag. For each round of concurrent transmissions, we collected the expected time precision loss from each anchor after selection of the best timestamp. We also collected the concurrent CIR and we detected the locations of the peaks on the CIR. Figure 6 shows the total non-precision time error being the difference between the time precision loss and the RX timestamp estimation error. We notice that, on average, the total non-precision time error does not exceed 1.5 ns. However, in some cases, it can go up to 3 ns. The sources of these inaccuracies can be attributed to three main factors:

- **CIR resolution:** the CIR has a resolution of 1 ns. a single error in CIR calculation can result in ± 1 ns of error.
- **Clock drift estimation:** Concurrent transmissions in our system are scheduled initially by adding a fixed delay of 10 ms to the blink timestamp. A delay of 10 ms translates to around 6×10^8 ticks while the clock drift estimation is maxed to 20×10^{-5}. Since the clock drift value changes over time, even a slight change in the clock drift during 10 ms can lead to a few hundreds of ticks in error.
- **Platform jitter:** Further jitter could be introduced in the transmission process from mechanical components like the antenna. This jitter is in the order of few hundreds of picoseconds, and therefore, it can reach tens of ticks.

E. Impact of Preamble Repetitions on Timestamp Estimation

We utilize the preamble symbol repetitions as a main component of our system. Each preamble size represents, in theory, the maximum number of preamble repetitions to skip before losing the concurrent aspect of transmissions. We limit the number of skipped repetitions to 90.

We found that if we expand our timestamps candidate range beyond 90 repetitions, it can result in the failure of our system since once the reference anchor finishes transmitting its packet, it switches to reception mode, and instead of receiving the $Sync$ message from the initiator, it starts receiving some of the other anchors' preambles.

We also investigated the impact of skipping preamble repetitions on the overall performance of our system. We collected multiple rounds of concurrent CIR from the tag and we also collected the best timestamp candidate information from each node containing the repetition it matches.

Figure 7 shows the mean and standard deviation of RX timestamp estimation error for each range of chosen preamble repetition. First, we observe that our algorithm does not consider repetitions between 1 and 20. The reason is that with low clock drift, the impact of delaying the transmissions with a small number of repetitions on the candidate timestamp does not considerably change the 9 lower bits of the timestamp. As an example, if the clock drift is in the order of 10^{-5}, skipping 20 repetitions is equivalent to adding 13 ticks to the timestamp. Second, we observe that overall, even with a high number of skipped repetitions (40 to 90), we can achieve RX timestamp estimation errors in the order of 3 ns in average while keeping a maximum error of 4.5 ns

F. Transmission delay impact on distance difference estimation

Our distance difference correction method is expected to reduce the time precision error for systems that require large transmission delays (Δ_{ref}). To validate our claim, we designed an experiment where we put the initiator and tag at an equidistant distance from a reference anchor and a regular anchor. We expect therefore that the index of the first path of the regular anchor should be around the inter-anchor delay ε_i (200 ns in this case). We tested multiple values of Δ_{ref} ranging between $[10\ ms, 500\ ms]$ and we collected 100 time difference measurements for each setting. Figure 8 shows that between 10 ms and 40 ms the distance difference error has an average mean of -14 cm (< 1 ns) and an average standard deviation of 58 cm (~ 2 ns). However, starting from 50 ms of transmission delay, we noticed that the errors increase considerably, reaching a mean of -105 cm and a large standard deviation of 117 cm. We confirmed this limitation further for Δ_{ref} values of $100, 200, 300, 400$ and 500 ms and we noticed that although the average mean of errors is around -70 cm, the standard deviation of the errors reaches around 196 cm. The increase of error standard deviation results from the failure of clock drift correction method beyond 50 ms since the value of the clock drift already changed after a very large delay. Therefore, we can confirm that our method

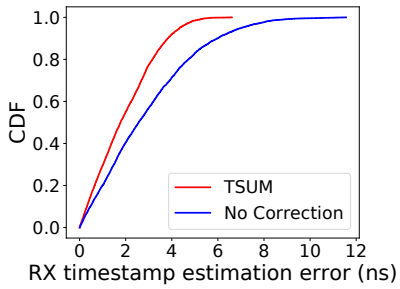

Fig. 5: CDF of RX timestamp estimation errors using TSUM and detailed impact of each technique on the error.

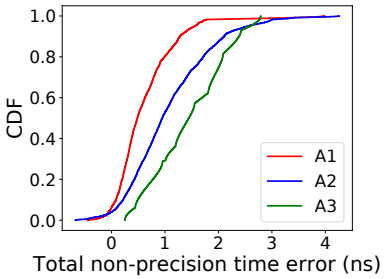

Fig. 6: Total non-precision time error CDF per transmitting anchor.

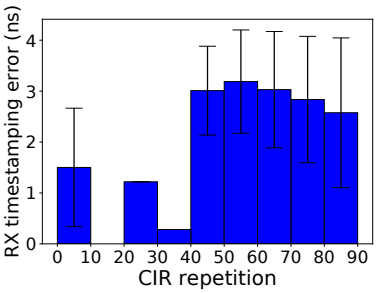

Fig. 7: RX timestamp estimation errors per number of skipped preamble repetitions.

performs properly up to $40\ ms$ of transmission delays as opposed to previous work where only small delays of few hundreds of μs are supported.

G. Impact of Preamble Length on Anchors Scalability

We considered three preamble sizes: $128, 256,\ and\ 512$. These longer preamble sizes allow us to have enough repetitions to consider in our algorithm while keeping the size of packets small enough to avoid having large transmission delays. For analysis, We use a measure similar to the peak-to-average ratio used in OFDM that we call the peak detection indicator (PDI). PDI is the ratio between the maximum sample and the average of all samples in the CIR.

We noticed that a peak is considered detectable if the PDI is greater than 5. We collected around 300 concurrent transmissions for each preamble size. We found that for almost all the preamble sizes, the 2^{nd} and 3^{rd} peaks have a high detection ratio, while the 4^{th} peak has a lower detection ratio reaching around 70% for preamble size 128. We suspect that the drop in detection ratio is due to the fact that at the level of the 4^{th} peak, four preamble symbols transmissions overlap with each other. Although UWB uses impulse radios, it is possible that a preamble symbol may not be detected if too much interference is present.

H. Impact of distance on Timestamp Estimation

Our earlier experiments demonstrated that our system works when the tag is equidistant to all anchors. Next, we evaluate how the system works when the tag is at different distances from the anchors. For this purpose, we designed a scenario where initially the anchors are placed at approximately a distance of $\sim 1\ m$ from both the initiator and the tag. We consider this scenario our baseline to evaluate the impact of distance on timestamp estimation. We then designed 3 experiments where each time we place one of the anchors (except the reference anchor) at a distance of $4\ m$ from the initiator.

In each scenario, we collect concurrent CIR data and report the RX timestamp estimation errors in figure 9. The baseline boxplots (red) represent the RX timestamp estimation errors for each anchor in the baseline scenario. The right boxplots (blue) in the three groups represent the RX timestamp estimation errors of the peak associated to anchor A_i for the scenario where the anchor was placed at a distance of $4\ m$ from its initial location. For all the anchors, we observe a similar error range between the baseline scenario and the other scenarios confirming that distance does not impact TSUM's performance. We can also observe that although, in theory, we should not get errors larger than $4\ ns$, they can reach up to $5.5\ ns$. The source of these excessive errors can be due to the factors detailed in section IV-D. Therefore, we can conclude that our system can reduce the error even with the tags at different distances from the anchors.

I. AOA estimation accuracy with different angles

We examine the impact of ground truth angle on concurrent angle estimation for two anchors. We placed the tag at the same distance from two anchors. Then we changed the tag location such that we change the angle between the tag and the anchors while keeping the tag equidistant to both anchors. The results for angle estimation are shown in Figure 10, with the absolute value of angles shown on x axis. Although A2 angle estimations show larger errors compared to A1, all errors are below $8°$ and we can see no significant change in error when we move the tag and change the ground truth angle. The results for distance differences estimation are shown in Figure 11. Δ_d is the distance difference between anchor pairs. All errors are approximately below 1.4 m and we can see no significant change in error when we move the tag and change ground truth angles.

V. Discussion

Although the results demonstrate the capability to estimate TDOA and AOA simultaneously are promising, we still fundamentally suffer from the $1\ ns$ granularity issue for CIR sampling. There are recent work that try to address this problem, but this issue cannot be eliminated and will continue to dictate the floor of performance.

Scaling concurrent localization is not trivial as we have to disambiguate multiple arrivals on the CIR. As a result, our system also has similar scaling limitations as other concurrent systems. For example, these systems scaled to 2-3 anchors for

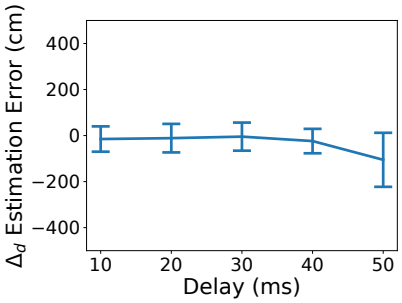

Fig. 8: Distance difference mean and standard deviation under different TX delays.

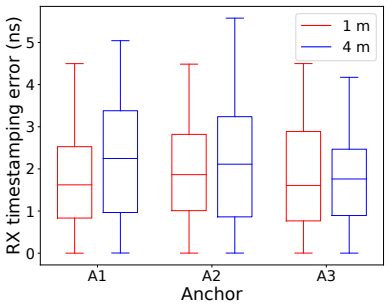

Fig. 9: RX Timestamp estimation errors for the baseline (red) and rest (blue) of the scenarios.

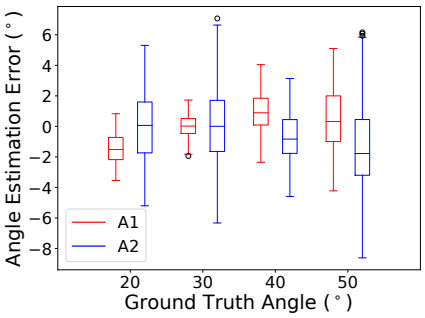

Fig. 10: Impact of different angles on angle estimation error for two anchors.

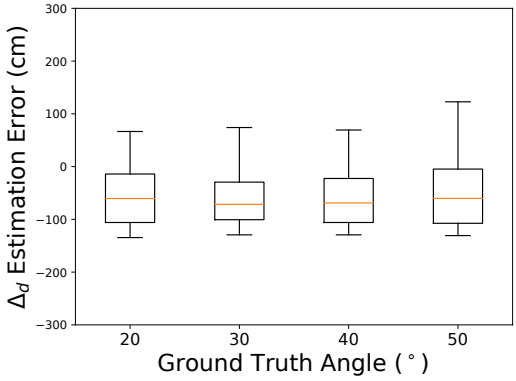

Fig. 11: Impact of different angles on distance difference estimation error for two anchors.

angle and 4-5 anchors for time estimations but certainly not dozens of concurrent arrivals at the receiver. Fortunately, in many deployments, strategic placement of anchors may result in scenarios where just a handful of anchors are visible from given locations, thus limiting the concurrency to a few arrivals at the receiver.

VI. Conclusions

In this paper, we presented the design of a UWB-based concurrent localization system that can estimate both TDOA and AOA simultaneously. Furthermore, this system is the first demonstration of how to utilize both inter-anchor concurrency and intra-anchor concurrency in a concurrent localization setting. Our experiments on a testbed built using Decawave platform suggest that such a system is not only practical but also performant.

References

[1] A. Ansaripour, M. Heydariaan, and O. Gnawali, "Viper: Vehicle pose estimation using ultra-wideband radios," in *IEEE DCOSS 2020*.

[2] N. Smaoui, K. Kim, O. Gnawali, Y.-J. Le, and W. Suh, "Respirable dust monitoring in construction sites and visualization in building information modeling using real-time sensor data," *Sensors and Materials*, vol. 30, no. 8, Dec. 2018.

[3] *DW1000 Datasheet*, Decawave, 2019.

[4] I. Dotlic, A. Connell, H. Ma, J. Clancy, and M. McLaughlin, "Angle of arrival estimation using decawave dw1000 integrated circuits," in *WPNC 2017*.

[5] M. Heydariaan, H. Mohammadmoradi, and O. Gnawali, "R3: Reflection resilient concurrent ranging with ultra-wideband radios," in *IEEE DCOSS 2019*.

[6] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: Uwb concurrent transmissions for gps-like passive localization of countless targets," in *ACM/IEEE IPSN 2019*.

[7] P. Corbalán and G. P. Picco, "Concurrent ranging in ultra-wideband radios: Experimental evidence, challenges, and opportunities," in *EWSN 2018*.

[8] 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 *ACM/IEEE IPSN 2019*.

[9] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer, "Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution," in *IEEE ICDCS 2018*.

[10] M. Heydariaan, H. Dabirian, and O. Gnawali, "Anguloc: Concurrent angle of arrival estimation for indoor localization with uwb radios," in *IEEE DCOSS 2020*.

[11] N. Smaoui, M. Heydariaan, and O. Gnawali, "Single-antenna aoa estimation with uwb radios," in *IEEE WCNC 2021*.

[12] H. Mohammadmoradi, M. Heydariaan, and O. Gnawali, "Srac: Simultaneous ranging and communication in uwb networks," in *IEEE DCOSS 2019*.

[13] D. K. Rout and S. Das, "Multiple narrowband interference mitigation using hybrid hermite pulses for body surface to external communications in uwb body area networks," *Wireless Networks*, vol. 23, no. 2, pp. 387–402, 2017.

[14] H. Mohammadmoradi and O. Gnawali, "Study and mitigation of non-cooperative uwb interference on ranging," in *EWSN*, 2019, pp. 142–153.

[15] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *ACM/IEEE IPSN 2011*.

[16] B. Kempke, P. Pannuto, and P. Dutta, "Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization: Demo abstract," in *ACM SenSys 2016*.

[17] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco, "Playing with Fire: Exploring Concurrent Transmissions in Ultra-wideband Radios," in *IEEE SECON 2019*.

[18] T. Wang, H. Zhao, and Y. Shen, "An efficient single-anchor localization method using ultra-wide bandwidth systems," *Applied Sciences*, vol. 10, no. 1, p. 57, 2020.

[19] P. Corbalán and G. P. Picco, "Ultra-wideband concurrent ranging," *ACM Trans. Sen. Netw.*, vol. 16, no. 4, Sep. 2020.

[20] *TREK1000 User Manual*, Decawave, 2018.

[21] *PDOA Primer - Calculating position directly from range and phase difference of arrival at two antennas*, Decawave, 2018.