

U-TDOA Synchronization Based on Edge Computing for UWB Localization Systems

W. F. F. Santos, P. J. S. Costa, R. H. P. Silva, H. L. N. Silva and C. B. Carvalho

Abstract—The accuracy of ultra-wideband (UWB) real-time locating systems (RTLSSs) critically depends on synchronization among anchors, particularly in uplink time-difference-of-arrival (U-TDOA) topologies. This paper proposes a distributed architecture in which time correction is performed at the edge, directly on each anchor’s microcontroller. The method combines an affine clock model with an exponential moving average (EMA) applied in firmware to smooth the clock-drift estimate. The EMA-based firmware configuration reduced the 95th percentile (P95) of simultaneous alignment from 751 ps to 484 ps and the median from 360 ps to 238 ps; moreover, 99.1% of the individual residuals were below 500 ps. Its overall effect on localization was small and geometry-dependent, underscoring that synchronization is the primary contribution.

Keywords—UWB, Uplink TDOA, Edge Computing, Synchronization, Clock Skew, EMA Filter, Gauss-Newton.

I. INTRODUCTION

Continuous monitoring applications for assets and mobile robots in industrial and logistics environments have driven the adoption of Ultra-Wideband (UWB) due to its high temporal resolution and good immunity to narrowband interference [1], [5], [13]. Among the available topologies, U-TDOA (*Uplink Time Difference of Arrival*) stands out for its low energy demand on tags, favoring extended operation with small batteries [6].

The main challenge of this topology is synchronizing all anchors with nanosecond precision. Small oscillator deviations can produce positioning errors on the order of meters [2]. Recent work explores both cascaded synchronization [9] and distributed estimation [3], although their architectures and metrics differ from those considered here. In this work, affine correction and EMA filtering are executed in the firmware of each anchor before the *timestamps* are sent to the server.

This paper shifts the synchronization logic to the anchors. Its main contribution is distributed clock synchronization at the edge; localization is evaluated only as a secondary effect of time quality. In the experiments, the EMA-based configuration reduced simultaneous-alignment P95 from 751 to 484 ps and the median from 360 to 238 ps. The global positioning means, 11.4 cm with EMA and 11.1 cm without EMA, were close, highlighting the influence of anchor geometry.

The remainder of this paper is organized as follows. Section II discusses related work; Section III describes the architecture; Section IV presents the models and algorithms; Section V reports the experiments and results; and Section VI concludes the paper, discusses its limitations, and outlines future work.

W. F. F. Santos, P. J. S. Costa, R. H. P. Silva, H. L. N. Silva and C. B. Carvalho are with the HUB I4.0 Laboratory, Universidade do Estado do Amazonas (UEA), Manaus-AM, Brazil. This work was partially funded by TPV do Brasil under the R&D Program.

II. BACKGROUND AND RELATED WORK

A. Hardware Challenges and Dynamic Correction

Rinaldi et al. [2] experimentally characterize the thermal drift of UWB oscillators, whereas Ferrari et al. [8] obtain picosecond resolution using radio counters and dedicated single-board computers. Senevirathna et al. [3] address gradient-based synchronization, and Zhang et al. [9] employ the Clock Skew Ratio (CSR) with interpolation and cascaded synchronization. In contrast, we retain the CSR formulation but execute correction and EMA filtering on each anchor’s microcontroller. The metrics and topologies of these studies are not directly equivalent to those of our experiment; a standardized quantitative comparison remains future work.

Architecturally, the main distinction is where the correction is performed. Reference [8] uses external computing resources, while [3] distributes estimation through interactions among nodes. Our proposal uses one master anchor in *broadcast* mode and estimates *offset* and *skew* on each STM32. This comparison is functional because the available platforms and metrics do not support a claim of numerical superiority among the methods.

B. Localization Algorithms and Industrial Environments

Several approaches solve the TDOA equations [15], from asynchronous methods to two-stage algorithms such as Chan-Taylor [4], [7], [16]. VULoc [10] prioritizes scalability by relaxing strict synchronization; here, synchronization remains explicit. In industrial environments, non-line-of-sight (NLOS) operation remains critical and motivates future extensions for NLOS mitigation and fusion between an inertial measurement unit (IMU) and UWB [14], [17].

III. SYSTEM MODEL

The proposed system consists of a network of intelligent anchors at known positions. The experiments consider a planar configuration, with $\mathbf{a}_i = (x_i, y_i)^T$; extending the system to three dimensions requires the vertical coordinate and anchors arranged in a noncoplanar geometry.

A. Edge Architecture and Timing Hardware

Each anchor combines a UWB DW1000 transceiver with an STM32F429VIT6 microcontroller unit (MCU), based on an ARM Cortex-M4 with floating-point support. The MCU executes the EMA filter and *skew* correction. The update uses fixed-size state, so its cost per packet does not grow with the sample history; processor cycles and energy consumption were not instrumented. The radio provides the time base: the DW1000 derives a 499.2 MHz internal clock from the 38.4 MHz reference, and its timestamp unit is $1/(499.2 \text{ MHz} \times 128) \approx 15.65 \text{ ps}$. Transmission (TX) and reception (RX)

instants are read from the 40-bit registers and stored in 64-bit variables; their differences use 40-bit modular arithmetic before sign extension, and RX events are handled through an interrupt request (IRQ).

B. Communication Protocol and Data Flow

The implemented flow adapts periodic synchronization from [9] and asynchronous *blinks* from [17]. Its architectural contribution is to normalize time at each anchor before the Central Localization Engine (CLE). Corrected records use a Transmission Control Protocol/Internet Protocol (TCP/IP) backhaul. The three concurrent layers are illustrated in Fig. 1.

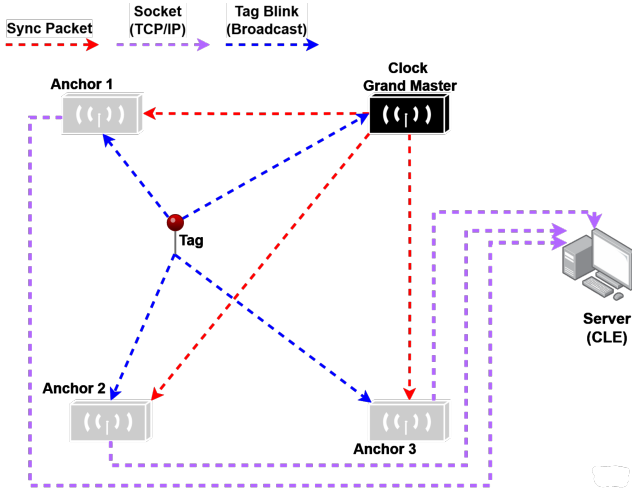


Fig. 1. U-TDOA topology: master anchor synchronizes the network (red), tag emits asynchronous *blinks* (blue), anchors forward corrected *timestamps* to the CLE server via TCP/IP (purple).

- 1) **Synchronization (Sync):** The master anchor emits calibration packets in *broadcast* every 150 ms. Adopted from [9], this value was adequate in the evaluated scenario, although its sensitivity to thermal drift was not investigated.
- 2) **Positioning (Tag Blink):** The tag emits asynchronous *blinks* with random delays in an ALOHA-like scheme [17]. At high density, contention, packet losses, and fading may prevent the CLE from receiving complete timestamp sets; that regime was not evaluated here.
- 3) **Edge Processing and Backhaul:** The anchor corrects the *timestamp* and, in the analyzed implementation, sends the CLE a 57-byte application frame per anchor/*blink*, containing identifiers, state, telemetry, and $\hat{t}_{global}$. No comparative traffic or latency trace was obtained against a centralized implementation.

IV. METHODOLOGY AND MATHEMATICAL FORMULATION

A. Local Clock Model

The local clock $C_i(t)$ of the i -th anchor is described by an affine relationship between real and measured time, following Zhang et al. [9] and Senevirathna et al. [3]:

$$C_i(t) = (1 + \delta_i) \cdot t + \theta_i + n_i(t) \quad (1)$$

where δ_i is the crystal frequency error (*skew*), θ_i is the initial offset, and n_i is the oscillator phase noise. To synchronize the time bases, the time of flight (*ToF*) of the P_{sync} packet must

be subtracted. Here, d is the known distance to the master anchor and c is the speed of light, giving a propagation time of $T_{prop} = d/c$.

B. Edge Correction Algorithm and EMA Filter

The anchor's firmware stores a state vector containing the clock updates based on the 40-bit *timestamps* read directly from the transceiver's registers. Instead of relying on a constant predefined nominal period for the update rate, the system extracts the exact interval between events using the transmission timestamps sent by the Master anchor.

Upon receiving the k -th synchronization packet, the slave anchor computes the instantaneous clock *Offset* $O[k]$, subtracting the Master's transmission time (T_{tx}) and the Time of Flight (ToF) from the local reception time (T_{rx}). The instantaneous drift (*measured skew*) is not computed over a fixed interval; instead, it is derived from the *Offset* variation (ΔO) divided by the actual elapsed time as measured by the Master (Δt_{master}):

$$skew_{meas}[k] = \frac{O[k] - O[k-1]}{\Delta t_{master}} \quad (2)$$

where $\Delta t_{master} = (T_{tx}[k] - T_{tx}[k-1]) \times 15.65$ ps. Due to the statistical noise in UWB reception, the metric $skew_{meas}$ exhibits high-frequency fluctuations. To stabilize the estimate without introducing *buffer* latency, an Exponential Moving Average (EMA) filter is applied directly on the MCU:

$$skew_{ema}[k] = (1 - \alpha) \cdot skew_{ema}[k-1] + \alpha \cdot skew_{meas}[k] \quad (3)$$

The parameter α tunes the filter's sensitivity to crystal drift. We adopted $\alpha = 0.05$, which, for $\Delta t = 150$ ms between *sync* packets, corresponds to $\tau \approx -\Delta t / \ln(1 - \alpha) \approx 2.9$ s. This choice smooths rapid fluctuations while preserving the tracking of slow drift; a controlled sweep of α is still required to quantify this tradeoff.

When a tag *blink* arrives at the local instant t_{tag} , the anchor computes the drift correction proportional to the time elapsed since the last *sync* (Δt_s) and projects the *timestamp* onto the global time domain ($\hat{t}_{global}$) with drift compensation:

$$\hat{t}_{global} = t_{tag} - O[k] - (skew_{ema}[k] \cdot \Delta t_s) \quad (4)$$

In Eq. (4), all variables (t_{tag} , $O[k]$, Δt_s , and $\hat{t}_{global}$) are expressed in the same time unit, with values obtained in DW1000 *ticks* and converted to seconds using the resolution of 15.65 ps per count. Algorithm 1 organizes the procedure into three stages: it estimates $O[k]$ and $skew$ after each *sync*; rejects spurious residuals and updates the lock state; and, upon a *blink*, projects its time onto the global domain. Subtractions involving the 40-bit counters, which wrap around every 17.2 s, use modular arithmetic (2^{40}) before sign extension to 64 bits.

The 2048-tick gate prevents a large residual from updating the EMA, whereas the 64-tick lock window requires eight consecutive in-range residuals before synchronization is declared. The dynamic loss-of-sync timeout follows the observed inter-sync period with a 5% margin. These prototype engineering settings were not varied in this campaign.

C. Solving the U-TDOA Equations (Central Solver)

The server receives the normalized global times ($\hat{t}_{global}$) from a set of anchors. Considering a reference anchor (e.g.,

Algorithm 1 EMA with Spurious-Sample Rejection, Clock-Skew Projection, and Lock Detection

Input: 40-bit DW1000 timestamps (15.65 ps/count), known propagation distance (ToF)

```

1: Init:  $skew_{ema} \leftarrow 0$ ,  $initialized \leftarrow false$ ,  $\alpha \leftarrow 0.05$ ,  $lockCount \leftarrow 0$ ,  $status \leftarrow UNSYNCED$ 
2: loop
3:   if RX IRQ triggered then
4:      $T_{rx}[k] \leftarrow$  40-bit reading from radio (subtractions in modular arithmetic  $2^{40}$  and sign-extended to 64-bit)
5:     if Packet is Master Sync then
6:       Extract  $T_{tx}[k]$  from payload
7:        $O[k] \leftarrow T_{rx}[k] - T_{tx}[k] - ToF$ 
8:       if  $initialized = false$  then
9:          $O[k-1] \leftarrow O[k]$ ;  $T_{tx}[k-1] \leftarrow T_{tx}[k]$ ;  $T_{rx\_sync} \leftarrow T_{rx}[k]$ ;  $initialized \leftarrow true$ ; continue
10:      end if
11:       $\Delta t_m \leftarrow (T_{tx}[k] - T_{tx}[k-1]) \times 15.65$  ps
12:       $skew_{meas} \leftarrow (O[k] - O[k-1]) / \Delta t_m$ 
13:       $O_{pred} \leftarrow O[k-1] + skew_{ema} \cdot \Delta t_m$ 
14:       $residual \leftarrow O[k] - O_{pred}$ 
15:      if  $|residual| > 2048$  ticks ( $\approx 32$  ns, i.e.  $\sim 9.6$  m) then
16:         $O[k] \leftarrow O_{pred}$ 
17:         $lockCount \leftarrow 0$ ;  $status \leftarrow SYNCING$ 
18:      else
19:         $skew_{ema} \leftarrow (1-\alpha) \cdot skew_{ema} + \alpha \cdot skew_{meas}$ 
20:        if  $|residual| \leq 64$  ticks ( $\approx 1$  ns) then
21:           $lockCount \leftarrow lockCount + 1$ 
22:        else
23:           $lockCount \leftarrow 0$ 
24:        end if
25:         $status \leftarrow LOCKED$  if  $lockCount \geq 8$ , else  $SYNCING$ 
26:      end if
27:      Update  $O[k-1]$ ,  $T_{tx}[k-1]$ ,  $T_{rx\_sync}$ ; reset dynamic timeout
28:    else if Packet is Tag Positioning Signal then
29:       $\Delta t_s \leftarrow (T_{rx}[k] - T_{rx\_sync}) \times 15.65$  ps
30:       $\hat{t}_{global} \leftarrow T_{rx}[k] - O[k] - skew_{ema} \cdot \Delta t_s$ 
31:      Send:  $\{ID_{tag}, \hat{t}_{global}, status\}$  to Server
32:    end if
33:  end if
34: end loop

```

$i = 1$), the Time Difference of Arrival (TDOA) for the j -th anchor is obtained by:

$$\Delta t_{1,j} = \hat{t}_{1,global} - \hat{t}_{j,global} \quad (5)$$

The two-dimensional tag position $\mathbf{p} = (x, y)^T$ is obtained by solving the hyperbolic system of distance differences ($k_{1,j}$), where c is the speed of light:

$$k_{1,j} = c \cdot \Delta t_{1,j} = \|\mathbf{p} - \mathbf{a}_1\| - \|\mathbf{p} - \mathbf{a}_j\| \quad (6)$$

Algorithm 2 details the Gauss–Newton method executed by the CLE. At each iteration, it computes the distances, forms the residual $\mathbf{F}$ and Jacobian J , applies the correction, and tests $\|\mathbf{F}\| < \epsilon$. For the documented reference configuration, each solution is initialized at the center of the monitored region, $\mathbf{p}_0 = [2.0, 1.0]^T$ m, with $\epsilon = 0.05$ m and $MaxIter = 20$. The central initialization avoids coincidence with an anchor, while the tolerance and iteration cap provide an engineering compromise between the observed noise scale and bounded processing time. Choosing ϵ below the distance-equivalent noise floor may increase the iteration count without a measurable accuracy gain, whereas an excessively small $MaxIter$ may terminate the solver before convergence. Convergence also depends on the initial guess and on the conditioning of

$J^T J$, especially with nearly collinear anchors. In 2D, each iteration processes $N - 1$ differences and costs $\mathcal{O}(N)$; the total cost is $\mathcal{O}(N \cdot MaxIter)$. Sensitivity to ϵ , $MaxIter$, anchor count, and geometric conditioning was not swept in this experiment.

Algorithm 2 TDOA Solver via the Gauss–Newton Method

Input: Time vector $\mathbf{t} = [t_1, t_2, \dots, t_N]$, Anchor positions $\mathbf{a}$, Initial guess $\mathbf{p}_0 = [x_0, y_0]^T$, Tolerance ϵ , $MaxIter$

Output: Estimated tag coordinate $\hat{\mathbf{p}}$

```

1: Compute  $dt_{1,j} = t_1 - t_j$  for  $j \in \{2, \dots, N\}$ 
2: Convert to distance:  $k_{1,j} = c \cdot dt_{1,j}$ 
3: Initialize  $\hat{\mathbf{p}} \leftarrow \mathbf{p}_0$ 
4: for  $iter = 1$  to  $MaxIter$  do
5:   Compute distances  $d_i = \|\hat{\mathbf{p}} - \mathbf{a}_i\|$  for every anchor  $i$ 
6:   Build the residual vector  $\mathbf{F}$ 
7:    $f_j \leftarrow (d_1 - d_j) - k_{1,j}$  for  $j \in \{2, \dots, N\}$ 
8:   if  $\|\mathbf{F}\| < \epsilon$  then
9:     break
10:  end if
11:  Compute the Jacobian matrix  $J$ 
12:     $J_{(j-1),1} = \frac{x-x_1}{d_1} - \frac{x-x_j}{d_j}$ 
13:     $J_{(j-1),2} = \frac{y-y_1}{d_1} - \frac{y-y_j}{d_j}$ 
14:  Compute the Gauss–Newton step
15:     $\Delta \mathbf{p} = (J^T J)^{-1} J^T \mathbf{F}$ 
16:  Update estimate:  $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} - \Delta \mathbf{p}$ 
17: end for
18: Return  $\hat{\mathbf{p}}$ 

```

V. PERFORMANCE EVALUATION AND RESULTS

A. Laboratory Experimental Setup

The end-to-end experiments used one tag and four anchors in a 4×2 m laboratory, without obstacles and under line-of-sight (LoS) conditions. The master transmitted *sync* packets every 150 ms. The positions, measured with a tape, were: master $A_1 = (2.00, 0.00)$, slave 1 $A_2 = (0.00, 0.00)$, slave 2 $A_3 = (4.00, 0.00)$, and slave 3 $A_4 = (2.00, 2.00)$.

For each complete synchronization event, simultaneous alignment is defined as the span of the three slave residuals, $S[k] = \max_i r_i[k] - \min_i r_i[k]$, for $i \in \{A_2, A_3, A_4\}$.

B. Synchronization without EMA Smoothing

Two firmware configurations were compared. In the baseline (“without EMA”), EMA smoothing and the associated firmware filters were disabled, while instantaneous *skew* estimation and *offset* correction were retained. In the proposed configuration (“with EMA”), EMA smoothing and the residual/lock filters in Algorithm 1 were enabled. Thus, the comparison evaluates the complete EMA-based filtering configuration rather than the EMA recursion alone. Approximately 2024 events per anchor were recorded, yielding 6072 residuals. Three raw diagnostic events above 100 ns, associated with chain reinitializations, were excluded offline under the same validity rule used for both configurations, leaving 6069 valid residuals and 2019 complete alignments.

Fig. 2(a) shows that 100% of the valid residuals were below 1 ns, with P95 values of 517 ps (A_2), 483 ps (A_3), and 455 ps (A_4). In Fig. 2(b), simultaneous alignment had a median of 360 ps and a P95 of 751 ps, with 27.5% of the events above 500 ps.

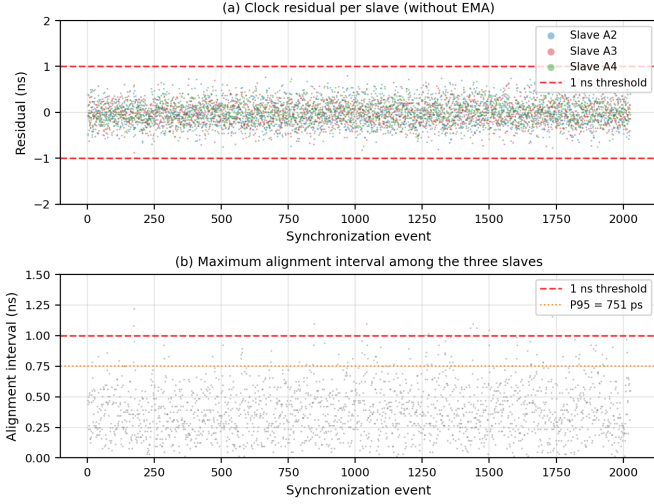


Fig. 2. Synchronization evaluation **without** the EMA filter. (a) Residual per anchor: 100% of the 6069 valid samples below 1 ns (P95: 517 ps, 483 ps, and 455 ps for A2, A3, and A4). (b) Simultaneous alignment of the 2019 complete events: median 360 ps and P95 751 ps.

C. Synchronization with EMA Compensation

With the EMA-based filtering configuration enabled (Algorithm 1), approximately 2032 events per anchor were recorded, yielding 6096 residuals. After the same offline exclusion of three raw diagnostic events above 100 ns associated with chain reinitializations, 6093 valid residuals remained. For alignment, the analysis retained the 2015 complete events containing records from all three slave anchors.

Fig. 3(a) shows that 100% of the valid residuals were below 1 ns and 99.1% below 500 ps; P95 values were 422 ps (A2), 364 ps (A3), and 329 ps (A4). In Fig. 3(b), simultaneous alignment had a median of 238 ps and a P95 of 484 ps; 96.0% of the events were below 500 ps, and all 2015 events were below 1 ns.

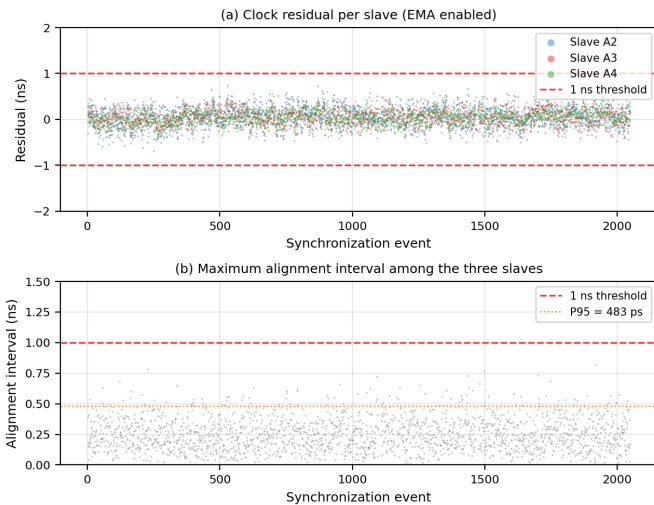


Fig. 3. Synchronization evaluation **with** the embedded EMA filter. (a) Residual per anchor: 100% of the 6093 valid samples below 1 ns (P95: 422 ps, 364 ps, and 329 ps for A2, A3, and A4). (b) Simultaneous alignment of the 2015 complete events: median 238 ps and P95 484 ps.

D. Comparison: With vs. Without EMA Compensation

Table I compares the two conditions. With filtering, **simultaneous alignment** P95 decreased from 751 to 484 ps (−36%), the median from 360 to 238 ps (−34%), and the standard deviation from 201 to 129 ps.

TABLE I
SYNCHRONIZATION COMPARISON WITH/WITHOUT EMA AFTER
DISCARDING THE THREE REINITIALIZATIONS

Metric	Without EMA	With EMA
<i>Individual clock residual per anchor</i>		
P95 — A2	517 ps	422 ps
P95 — A3	483 ps	364 ps
P95 — A4	455 ps	329 ps
Samples <500 ps	95.5%	99.1%
Samples <1 ns	100%	100%
<i>Simultaneous alignment among the three slaves</i>		
Median	360 ps	238 ps
Standard deviation	201 ps	129 ps
P95	751 ps	484 ps
P99	940 ps	607 ps
Events <500 ps	72.5%	96.0%
Events <1 ns	99.4%	100%

E. Evaluation at Multiple Tag Positions (9-Point Grid)

Nine measurement points were defined in the 4×2 m environment, as shown in Fig. 4.

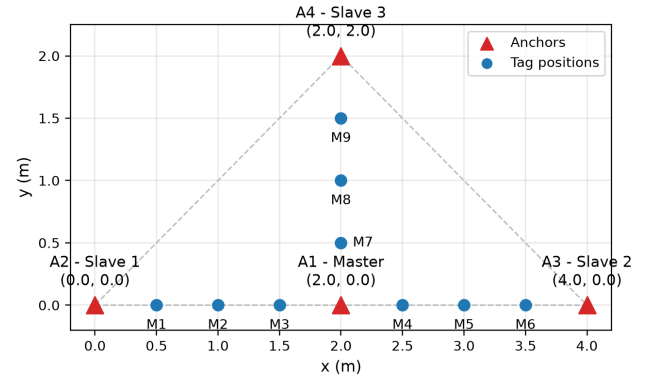


Fig. 4. Experimental geometry. Triangles: master $A_1 = (2.0, 0.0)$ and slaves $A_2 = (0.0, 0.0)$, $A_3 = (4.0, 0.0)$, and $A_4 = (2.0, 2.0)$. Circles: nine positions of the same tag.

A total of 2048 estimates were collected per position and condition, yielding 18432 samples with EMA and 18432 without EMA. Table II shows a reduction in mean error at M_1 – M_3 , particularly at M_1 (from 17.0 to 6.1 cm), but not at M_4 – M_9 . Three anchors on $y = 0$ produce high geometric dilution of precision (GDOP), so geometry can amplify error and mask time-domain gains. Fig. 5 shows this spatially heterogeneous effect.

Fig. 5(b) confirms similar global results. With EMA, the mean error was 11.4 cm, the median 9.6 cm, and P95 27.6 cm; 52.8% of the estimates were below 10 cm and 87.6% below 20 cm. Without EMA, the mean was 11.1 cm, the median 9.0 cm, and P95 24.7 cm, with proportions of 57.0% and 90.6%, respectively. Treating the nine positions, rather than the 18432 correlated estimates, as paired units, the mean difference (with–without EMA) was +0.36 cm, with a 95% confidence interval of $[-3.51, 4.23]$ cm; the paired test yielded $t(8) = 0.215$ and $p \approx 0.84$, providing no evidence of a

TABLE II

 POSITIONING ERROR PER POINT: COMPARISON WITH/WITHOUT EMA
(2048 SAMPLES EACH)

Point	With EMA (cm)		Without EMA (cm)	
	Mean	P95	Mean	P95
M_1	6.1	13.2	17.0	30.3
M_2	7.0	14.3	9.6	15.8
M_3	7.8	13.7	9.4	18.0
M_4	12.7	19.8	9.3	17.5
M_5	16.6	25.0	11.6	20.4
M_6	26.8	39.8	20.9	37.8
M_7	7.8	12.7	6.8	13.2
M_8	8.0	14.2	6.2	11.2
M_9	9.9	16.6	8.6	15.2

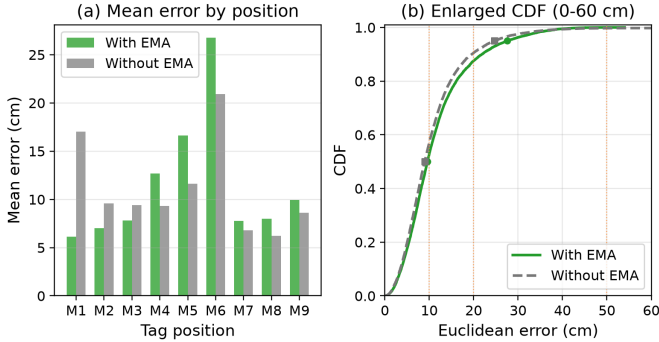


Fig. 5. Positioning with/without EMA. (a) Mean error by position. (b) Cumulative distribution function (CDF) of the 18432 estimates per condition, enlarged to 60 cm; percentiles use the complete sample.

global gain. Because only one campaign was conducted per condition, independent repetitions are still required.

VI. CONCLUSIONS

This work validated a distributed synchronization architecture for U-TDOA UWB, with EMA-based *clock drift* correction on each anchor's microcontroller. The main contribution is to show that time normalization can be performed at the edge before data are sent to the CLE. Because an equivalent centralized baseline was not implemented, no quantitative traffic, latency, or processing gain is attributed to the prototype.

At the synchronization layer, the EMA-based filtering configuration reduced simultaneous-alignment P95 from 751 to 484ps, the median from 360 to 238ps, and the standard deviation from 201 to 129ps. In addition, 96.0% of simultaneous alignments and 99.1% of individual residuals were below 500ps, showing lower time dispersion in the evaluated configuration. In localization, the reduction occurred at M_1 – M_3 , but the global mean difference was +0.36cm (with–without EMA), with a 95% confidence interval containing zero. Thus, the data support a synchronization gain, but not an overall localization improvement; the positioning effect depends on geometry and GDOP.

The experimental scope is limited to one 4×2 m LoS campaign with one tag, four anchors, partially collinear geometry, and no individual calibration of antenna delays [11]. The results therefore cannot yet be generalized to NLOS conditions, industrial environments, or dense networks, nor can they quantify traffic, latency, and resource-use gains over a centralized solution.

Future work will extend the experiments to NLOS and industrial scenarios, with 2D and 3D geometries, more anchors, and multiple tags. Scalability will be assessed under

increasing ALOHA load, including packet losses, fading, collisions, and a comparison between random and coordinated access. We will also investigate α , the synchronization period, ϵ , and $MaxIter$, as well as traffic, end-to-end latency, processor cycles, computational cost, and energy consumption, using a centralized implementation as a baseline. New campaigns with independent repetitions will include antenna-delay calibration and IMU+UWB fusion using Kalman filters [14], [17].

ACKNOWLEDGMENTS

The authors thank TPV do Brasil for the investment and support in the development of this research under the R&D Program tied to the Manaus Free Trade Zone Information Technology Law (Brazilian Federal Law n° 8.387/1991, regulated by SUFRAMA). They also thank the Universidade do Estado do Amazonas (UEA) and the HUB I4.0 Laboratory for the infrastructure and space provided for the experiments. The experimental plots were generated with the *Matplotlib* library (Python). The review and textual refinement of this article were assisted by the artificial intelligence assistant *GitHub Copilot* (Claude Sonnet 4.7 model, Anthropic/Microsoft).

REFERENCES

- [1] S. Bottiglieri et al., "A Low-Cost Indoor Real-Time Locating System Based on TDOA Estimation of UWB Pulse Sequences," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [2] S. Rinaldi et al., "An Experimental Characterization of Time Synchronization in Multiple UWB Location Cells," *Proc. IEEE ISPCS*, 2022.
- [3] N. M. Senevirathna et al., "Asymptotic Gradient Clock Synchronization in Wireless Sensor Networks for UWB Localization," *IEEE Sensors J.*, vol. 22, no. 24, pp. 24578–24592, 2022.
- [4] D. Chiasson et al., "Asynchronous Hyperbolic UWB Source-Localization and Self-Localization for Indoor Tracking," *IEEE Internet Things J.*, vol. 10, no. 13, 2023.
- [5] L. Flueratoru et al., "Challenges in platform-independent UWB ranging and localization systems," *Proc. ACM WiNTECH*, pp. 9–15, 2022.
- [6] M. Rostami and K. Sundaresan, "Enabling high accuracy pervasive tracking with ultra low power UWB tags," *Proc. ACM MobiCom*, 2022.
- [7] J. Leng et al., "Improved TDOA Two-Stage UWB Localization Algorithm For Indoor Mobile Robot," *Proc. IEEE RASSE*, 2021.
- [8] P. Ferrari et al., "Resilient Time Synchronization Opportunistically Exploiting UWB RTLS Infrastructure," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
- [9] F. Zhang et al., "Ultra-Wideband-based real-time positioning with cascaded wireless clock synchronization method," *IEEE Internet Things J.*, vol. 11, no. 9, 2024.
- [10] J. Yang et al., "VULoc: Accurate UWB Localization for Countless Targets without Synchronization," *Proc. ACM IMWUT*, vol. 6, no. 3, 2022.
- [11] M. Garcia-Fernandez, "Handling systematic node biases in TDOA positioning systems," *IEEE Trans. Veh. Technol.*, 2026.
- [12] M. Martalò et al., "Improved UWB TDoA-Based Positioning Using a Single Hotspot for Industrial IoT Applications," *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, 2022.
- [13] Y. Cao and A. Dhekne, "ITTrackU: Tracking a Pen-like Instrument via UWB-IMU Fusion," *Proc. ACM MobiSys*, 2021.
- [14] J. Liu et al., "Performance analysis and comparison for TDOA and DTDOA," *Proc. Int. Conf. Inf. Technol. Comput. Commun.*, 2021.
- [15] H. Xu et al., "Research on the Algorithm of High-precision Ultra-Wide Band Indoor Positioning Based on CHAN-TAYLOR Cascaded Positioning," *Proc. ICFEICT*, 2021.
- [16] N. M. Senevirathna et al., "Tightly Coupled UWB-INS Positioning With Passive Synchronization Using Continuous Clock Phase Tracking," *IEEE Internet Things J.*, vol. 12, no. 8, pp. 9716–9728, 2025.