Successive Threshold-Based Multipath Mitigation Aided by Neural Network for UWB Ranging

Alireza Sheikh[§], Bo Yang[†], Mohieddine El Soussi[§], Amirashkan Farsaei[§], and Peng Zhang[§]
[§]IMEC, Eindhoven, The Netherlands,[†]Eindhoven University of Technology

***Abstract*—Despite the recent advances in UWB systems especially after releasing the IEEE 802.15.4z standard, achieving the "cm" level ranging accuracy and precision using UWB in a general multipath environment is very challenging. In this paper, we introduce a successive threshold-based multipath mitigation algorithm (STM) that improves the ranging performance in multipath conditions. We further improve the ranging performance of STM in multipath conditions by aiding the STM with STMnet as a neural network that estimates the ranging error of the STM. Using a UWB packet simulator, we show that the (mean, Q95) of ranging error for the STM aided by STMnet achieves up to (-0.07cm, 0.63cm), (-0.49cm, 21.3cm), and (0.06cm, 39cm) for LOS environment, LOS residential and LOS office multipath environments, respectively. The improved performance using STMnet is at the cost of $330KB$ of required memory and $232K$ of floating point operations.**

***Index Terms*—Channel impulse response, leading-edge, multipath, neural network, range estimation, UWB**

I. Introduction

The high temporal resolution of the Ultra-WideBand (UWB) signal yields a more accurate and precise range estimation compared to other Internet of Things (IoT) alternate systems such as Bluetooth and Zigbee. This enables a myriad of localization-critic applications such as asset tracking for warehouses or life stocks [1], [2], service robots [3], and passive entry and start for smart cars/buildings [4]. UWB system is capable of joint communication and sensing, where the sensing is based on estimated Channel Impulse Response (CIR) via exchanging the UWB packets which can carry the data as well. The range estimation in UWB is based on Time of Flight (ToF) estimation. The ToF can be estimated by finding the time of arrival of the peak of the first path, which is referred to as Line-of-Sight (LOS) path in the estimated CIR [5].

Leading Edge (LE) algorithms [6], [7] has been proposed to find the lead of the first detectable path, which is used to estimate the peak of the first path. This method has an accuracy limited by the sampling rate of the receiver [7]. Maximum-likelihood-based peak estimation can further improves the accuracy of the first peak estimation in the LOS channel by more than one order of magnitude with negligible computational complexity [8]. Nevertheless, the accuracy of range estimation in UWB is still limited by the Non-LOS (NLOS) paths, in the so-called multipath conditions. Recently, Neural Network (NN)s are employed to improve the UWB ranging performance in the multipath condition. In particular, [9] uses all the CIR samples as the input of 2D-Convolutional Neural Networks (CNN) to estimate the range, which in turn requires a large memory and computational complexity, i.e., $4MB$ of memory and $1.4M$ floating point operations (FLOPs). In [10], 1D-CNN is trained based on a measured data set to estimate the ranging error of the LE-based range estimation. The required memory and computational complexity of the trained model is in the order of $1MB$ and $1M$FLOPs and it is not clear how the trained NN performs in the general multipath environment models.

In this paper, we focus on the problem of range estimation in mainly multipath environments using the estimated CIRs from exchanging the IEEE 802.15.4z-compliant packets between two nodes. To make the proposed methods general, we focus on the LOS channel (denoted by CM0), IEEE channel model for the LOS residential environment (known as CM1 in [11]), and IEEE channel for the LOS office environment (known as CM3 in [11]). We first propose a successive multipath mitigation algorithm which we referred to as STM. STM successively removes the impact of dominant multipath components in the estimated CIR. At the last stage of the algorithm, STM finds the peak of the LOS path in the updated CIR after multipath mitigation. We further improve the ranging performance of the STM algorithm using STMnet as a neural network (NN) which estimates the ranging error of the STM. We show that as STMnet requires only 61 samples of CIR around the STM estimated value, the required memory and computational complexity is limited, i.e., $330KB$ of memory and $232K$ FLOPs. We show that Q95 of ranging error as a measure of ranging precision for the STM aided by STMnet achieves 0.63cm, 21.3cm, and 39cm for CM0, CM1, and CM3, respectively[1].

A. Preliminaries

The range between two UWB devices (also referred to as initiator and reflector) in IEEE 802.15.4z is estimated using Single-Sided Two-Way Ranging (SS-TWR) or Double-Sided Two-Way Ranging (DS-TWR) [5]. In Fig. 1(left), SS-TWR for range estimation between two device [5] is shown using

[1] Notation: Boldface letters are used for vectors e.g., $\boldsymbol{x}$. $\boldsymbol{x}^2$ stands for the element-wise squared of $\boldsymbol{x}$. $\boldsymbol{x}(v)$, $\boldsymbol{x}(v_1 : v_2)$, $\boldsymbol{x}(\boldsymbol{x} > 0)$, and $\boldsymbol{x}(\boldsymbol{x} < 0)$ are the v-th component of $\boldsymbol{x}$, a slice from v_1-th to v_2-th components of $\boldsymbol{x}$, positive components of $\boldsymbol{x}$, and negative components of $\boldsymbol{x}$, respectively. $max(\boldsymbol{x})$ gives the component of $\boldsymbol{x}$ with maximum value. "$==$", "&", "$|$", and "$\tilde{}$" denote equality, AND, OR, and NOT logical operands, respectively. $\lfloor x \rfloor$ and $|x|$ stands for the floor value and absolute value of x, respectively. KB, MB, and K denote the kilobytes, megabytes, and kilo, respectively. Q95 of $\boldsymbol{x}$ is defined as the difference between 97.5%-quantile and 2.5%-quantile of $\boldsymbol{x}$.

978-8-3503-8544-1/24/$31.00 ©2024 IEEE

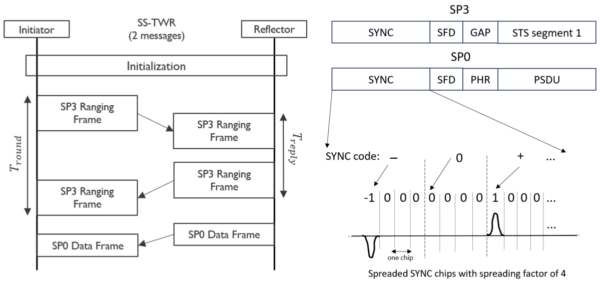

Fig. 1. SS-TWR procedure (left) and the SP3/SP0 packet structures with an example about the spreaded SYNC sequence (right).

Scrambled Time Sequence (STS) Packet Configuration 3 (SP3) ranging frame, and STS Packet Configuration 0 (SP0) data frame. SP3 packet consists of Synchronization (SYNC), Start-of-Frame Delimiter (SFD) and STS. SP0 packet replaces the STS by Physical Layer Header (PHR) and Physical Layer Service Data Unit (PSDU). The SYNC comprises of repeated symbols where each symbol consists of a spreaded code with periodically ideal cross correlation properties. An example for the spreaded code and the corresponding pulse shaped signal is shown in Fig. 1 (right). The SFD is used to identify the end of the SYNC. Pseudo randomized pulses in STS improves the security of the ranging [5]. In SS-TWR, the ToF can be computed using the T_{round} and T_{reply} duration captured at the initiator and reflector, i.e., ToF=$(T_{round} - T_{reply})/2$, where T_{reply} value is communicated to the initiator using SP0 data packet. In fact in SS-TWR, SP3 and SP0 packets are used for sensing and communication purposes, respectively. To compute T_{round} and T_{reply}, both devices shall capture the Time of Departure (ToD) and Time of Arrival (ToA) timestamps of the transmitted/received packets. The value of T_{reply} can be sent using the PSDU section of the SP0 packet to the initiator. T_{round} can be calculated as $T_{round} = ToA_{init} - ToD_{init} - \tau_{cal} - \tau_c$. ToA_{init} and ToD_{init} are the timestamps in which the UWB pulse after SFD is at the initiator's receiver and transmitter, respectively, τ_{cal} is the total transmitter and receiver delays including the antenna delays (to be calibrated in advance), and τ_c is the time correction value of the detected path (which is used for time capturing) in the initiator estimated CIR from the LOS path. We highlight that the detected path at the initiator and reflector can be LOS or NLOS, hence, the time correction value is essential for the ToF computation. Without loss of generality, in the following, we focus on the essential step in Two-Way Ranging (TWR), in which it is required to estimate the peak position of the LOS path from the estimated CIR.

II. System model

The UWB system model consists of transmitter (TX), LOS/NLOS channels and receiver (RX). At the transmitter, the IEEE 802.15.4z-complaint UWB packet (SP3 or SP0) is generated in which the TX sequence is pulse shaped and then amplified by the power amplifier. Before RX processing, the impact of hardware nonidealities are added. First, due to the possible crystal offset between the TX and RX, frequency and time errors are added. Then, the signal passes a sequence of low-pass filters and then the impact of front-end gain and thermal noise including the noise figure of the receiver chain is added. The resulting signal is then quantized using analog-to-digital converter (ADC). The hardware models used in this paper corresponds to the design presented in [12]. The receiver system model is shown in Fig. 2. The Rx processing block detects the packet and performs frequency and time synchronization. The synchronization is done by computing the cross correlation between the reference transmitted sequence and the received signal [12]. After the synchronization, the receiver performs CIR estimation and starts searching for the SFD. After SFD detection, the receiver either estimates another CIR using the STS sequence or decodes the payload [5]. The output of the RX processing in Fig. 2 for the v-th sample in the NLOS channel can be modeled as [8]

$$\sum_{i=1}^{Y}(a_i g\left(v-\tau_i\right) e^{j\theta_i(v)}) + z(v), \tag{1}$$

where Y, a_i, τ_i, $\theta_i(v)$, $z(v)$ are the number of NLOS components, i-th complex channel coefficient, i-th channel delay, phase shift due to i-th channel delay in addition to the unknown phase of the i-th path due to residual Carrier Frequency Offset (CFO) and hardware nonidealities, and correlated noise due to cross correlation operation at the RX, respectively. Furthermore, $g(v)$ is the cross correlation of the TX and RX impulse responses, i.e., $g(v) = \int_{-\infty}^{+\infty} s_{TX}(t) s_{RX}(t-v)dt$, where $s_{TX}(t)$ and $s_{RX}(t)$ are the TX and RX impulse responses, respectively. In Fig. 2, the absolute-squared of the CIR is computed which removes phases. We referred to the absolute-squared CIR as $\boldsymbol{h}^0$. If the delay of two paths are separated by at least T_s (sampling time), the signal after the absolute-squared operation in Fig. 2 can be approximated as

$$\boldsymbol{h}^0(v) \approx \sum_{i=1}^{Y}(|a_i|^2 g^2\left(v-\tau_i\right)) + \tilde{z}(v), \tag{2}$$

where $\tilde{z}(v)$ is modeled as a real-valued correlated Gaussian with the covariance given in [8, Proposition 1]. This approximation is valid due to narrow width of $g(v)$ for pulse shapes used in UWB [5], in which the other cross terms values due to absolute-squared operation become negligible. As can be seen from Fig. 2, LE algorithm is executed which estimates the noise variance of CIR (N_{est}) and uses N_{est} to find the rising edge of the first path of $\boldsymbol{h}^0$, referred to as LE path (see e.g., [6], [7] for details about the LE algorithm). Then, the local peak after LE path is found, which we referred to as $idx_{LE-PK}^{\boldsymbol{h}^0}$. The range estimation using $idx_{LE-PK}^{\boldsymbol{h}^0}$ is called LE-only in this paper. We refer to the list of dominant paths in $\boldsymbol{h}^0$ as $\boldsymbol{L}_p$. Note that $\boldsymbol{L}_p$ can be found by setting a noise dependent threshold on the $\boldsymbol{h}^0$ samples. In [8], an Maximum Likelihood (ML)-based interpolation block is proposed which improves the accuracy of LE-only scheme with small additional computational complexity. ML interpolation block first over-samples $\boldsymbol{h}^0$ around the $idx_{LE-PK}^{\boldsymbol{h}^0}$ index, then, it applies an ML-based algorithm to probabilistically finds the peak position. The scheme proposed in [8] corresponds to the lower branch of Fig. 2, which we refer to it as ML

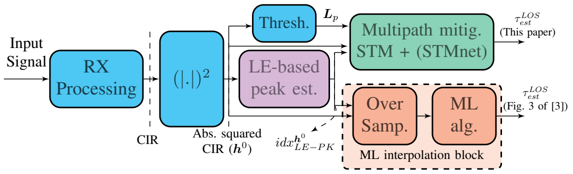
Input Signal
RX Processing
CIR
$(|.|)^2$
Abs. squared CIR ($\boldsymbol{h}^0$)
Thresh.
$\boldsymbol{L}_p$
LE-based peak est.
$idx^{\boldsymbol{h}^0}_{LE-PK}$
Multipath mitig. STM + (STMnet)
τ^{LOS}_{est} (This paper)
Over Samp.
ML alg.
ML interpolation block
τ^{LOS}_{est} (Fig. 3 of [3])

Fig. 2. The receiver system model for range estimation using the proposed method in this paper and the method presented in [8].

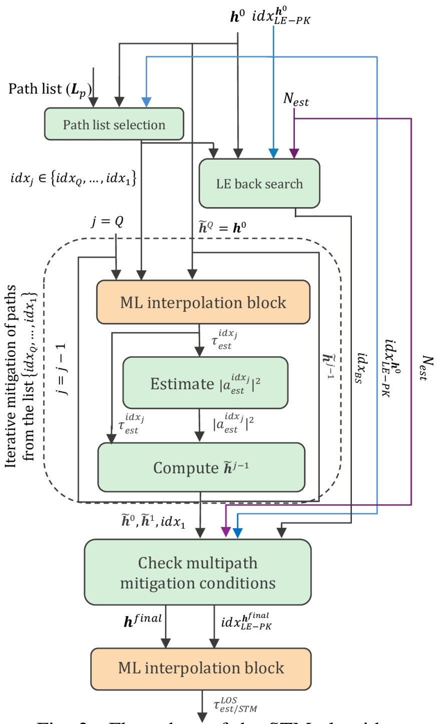
$\boldsymbol{h}^0$ $idx^{\boldsymbol{h}^0}_{LE-PK}$
Path list ($\boldsymbol{L}_p$)
Path list selection
N_{est}
LE back search
$idx_j \in \{idx_Q, \ldots, idx_1\}$
$j = Q$
$\tilde{\boldsymbol{h}}^Q = \boldsymbol{h}^0$
Iterative mitigation of paths from the list $\{idx_Q, \ldots, idx_1\}$
$j = j - 1$
ML interpolation block
$\tau^{idx_j}_{est}$
Estimate $|a^{idx_j}_{est}|^2$
$\tau^{idx_j}_{est}$
$|a^{idx_j}_{est}|^2$
Compute $\tilde{\boldsymbol{h}}^{j-1}$
$\tilde{\boldsymbol{h}}^{j-1}$
idx_{BS}
$idx^{\boldsymbol{h}^0}_{LE-PK}$
N_{est}
$\tilde{\boldsymbol{h}}^0, \tilde{\boldsymbol{h}}^1, idx_1$
Check multipath mitigation conditions
$\boldsymbol{h}^{final}$
$idx^{\boldsymbol{h}^{final}}_{LE-PK}$
ML interpolation block
$\tau^{LOS}_{est/STM}$

Fig. 3. Flow chart of the STM algorithm.

interp.-only in this paper. In the upper branch of Fig. 2, we propose ranging in multipath, which employs the ML-based interpolation, $\boldsymbol{L}_p$, $\boldsymbol{h}^0$, $idx^{\boldsymbol{h}^0}_{LE-PK}$, and N_{est} to improve the ranging accuracy in the NLOS channel. In the next section, we first propose a successive threshold-based multipath mitigation algorithm, referred to as STM. We then improves the ranging performance of STM using STMnet.

III. Ranging in Multipath Environment

A. STM algorithm

The flow chart of STM algorithm is shown in Fig. 3. In what follows, we explain details of STM.

1) Input/output: The inputs of STM are $\boldsymbol{L}_p, \boldsymbol{h}^0, idx^{\boldsymbol{h}^0}_{LE-PK}$, and N_{est}. As explained in Sec. II, $\boldsymbol{L}_p$ is the dominant path list of the estimated CIR and $\boldsymbol{h}^0$ is the absolute-squared of estimated CIR. Furthermore, $idx^{\boldsymbol{h}^0}_{LE-PK}$ and N_{est} are given by applying the LE algorithm (e.g., [6], [7]) based on the input of $\boldsymbol{h}^0$. The output of STM is the LOS delay estimate which is proportional to the range, and we referred to as $\tau^{LOS}_{est/STM}$.

2) Path list selection/LE back search: As only close NLOS paths to the LOS path can impact the accuracy of LOS peak estimation, STM first filters $\boldsymbol{L}_p$ such that only close paths to the $idx^{\boldsymbol{h}^0}_{LE-PK}$ remains [2]. Furthermore, as the filtered paths may belong to one NLOS path due to the width of pulse used in pulse shaping, only paths in this filtered list which corresponds to the local peaks of $\boldsymbol{h}^0$ are considered as the candidate paths for multipath mitigation. We assume that after such path list selection, the candidate path list indices $\{idx_Q, \cdots, idx_1\}$ will be used as the input for the iterative step of the STM, where idx_Q is the farthest path from the $idx^{\boldsymbol{h}^0}_{LE-PK}$ path. The LE back search algorithm is given in Alg. 1. Alg. 1 is based on the intuition that if the local peak at $idx^{\boldsymbol{h}^0}_{LE-PK}$ index is due to the superposition of LOS path with strong NLOS path, the values of $\boldsymbol{h}^0$ which are only few samples apart from $idx^{\boldsymbol{h}^0}_{LE-PK}$ should be much larger than N_{est}. In this situation LOS path is hided by the NLOS path, hence, it is necessary to perform the LE algorithm again after mitigating of all paths in $\{idx_Q, \cdots, idx_1\}$ list. The output of Alg. 1 is the index of back search (idx_{BS}) will be used in Alg. 2. Furthermore, in Alg. 1 W_{BS} is the back search window, P_1 and P_2 are selected thresholds.

```
Algorithm 1: LE back search
Input: $\boldsymbol{h}^0$, $idx^{\boldsymbol{h}^0}_{LE-PK}$, $N_{est}$
Output: $idx_{BS}$
Initialization: $idx_{BS} = 0$, select $W_{BS}$, $P_1$, and $P_2$
if $(\boldsymbol{h}^0(idx^{\boldsymbol{h}^0}_{LE-PK} - W_{BS}) > P_1 N_{est}$ & $\boldsymbol{h}^0(idx^{\boldsymbol{h}^0}_{LE-PK} - W_{BS} - 1) > P_2 N_{est})$ | $(idx^{\boldsymbol{h}^0}_{LE-PK} < idx_1)$ then
    $idx_{BS} = 1$
end if
```

3) Iterative mitigation of paths: For the j-th iteration where $j \in \{Q, \cdots, 1\}$, we use the ML interpolation block (see [8] for details) to estimate $\tau^{\text{est}}_{idx_j}$ as the peak position of path index idx_j in the j-th absolute-squared CIR profile, referred to as $\tilde{\boldsymbol{h}}^j(v)$. We remark that iteration starts such that $\tilde{\boldsymbol{h}}^Q(v) = \boldsymbol{h}^0(v)$. Using $\tau^{\text{est}}_{idx_j}$ and (2), amplitude-squared of the path index of idx_j can be approximated as[3]

$$|a^{\text{est}}_{idx_j}|^2 \approx \frac{\tilde{\boldsymbol{h}}^j(\tau^{\text{est}}_{idx_j})}{g^2(\tau^{\text{est}}_{idx_j} - \tau_{idx_j})}, \tag{3}$$

where the value of $g^2(\tau^{\text{est}}_{idx_j} - \tau_{idx_j})$ can be computed by interpolating of $g^2(v)$. Given $\tau^{\text{est}}_{idx_j}$ and $|a^{\text{est}}_{idx_j}|^2$, the path index idx_j is mitigated from $\tilde{\boldsymbol{h}}^j(v)$, which gives the updated absolute-squared CIR profile, i.e.,

$$\tilde{\boldsymbol{h}}^{j-1}(v) = \tilde{\boldsymbol{h}}^j(v) - |a^{\text{est}}_{idx_j}|^2 g^2\left(v - \tau^{\text{est}}_{idx_j}\right). \tag{4}$$

4) Check multipath mitigation conditions: This stage of STM is given in Alg. 2, where (5)-(7) are conditions used in Alg. 2. In Alg. 2, W_t is a window size, and T_1, D_1-D_4 are the selected thresholds. From a high-level perspective, Alg. 2 verifies some threshold-based conditions in the output of iterative stage of STM to ensure that $\boldsymbol{h}^{final}$ can be used for

[2] Assuming 1GHz sampling rate, we use the paths within 8ns from the path index $idx^{\boldsymbol{h}^0}_{LE-PK}$.

[3] An ML-based estimation problem for $|a^{\text{est}}_{idx_j}|^2$ can also be formulated, however, we did not achieve any noticeable ranging performance improvement compared to the approximation given in (3).

Algorithm 2: Check multipath mitigation conditions

```
Input: $\tilde{\boldsymbol{h}}^0$, $\tilde{\boldsymbol{h}}^1$, $idx_1$, $N_{est}$, $idx^{\boldsymbol{h}^0}_{LE-PK}$, $idx_{BS}$,
Output: $\boldsymbol{h}^{final}$, $idx^{\boldsymbol{h}^{final}}_{LE-PK}$
Initialization: Select $W_t$, $T_1$, $D_1$, $D_2$, $D_3$, $D_4$
 1: $\tilde{\boldsymbol{h}}^0_{trim} \triangleq \tilde{\boldsymbol{h}}^0(idx_1 - W_t : idx_1 + W_t)$;
 2: Find ratio of the sum of negative to positive values of $\tilde{\boldsymbol{h}}^0_{trim}$:
    $R^{\tilde{\boldsymbol{h}}^0_{trim}} \triangleq |\sum(\tilde{\boldsymbol{h}}^0_{trim}(\tilde{\boldsymbol{h}}^0_{trim} < 0))|/\sum(\tilde{\boldsymbol{h}}^0_{trim}(\tilde{\boldsymbol{h}}^0_{trim} > 0))$
 3: Run the LE algorithm with input $\tilde{\boldsymbol{h}}^0$ and find the first peak
    after LE index, referred to as $idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}$
 4: $\tilde{\boldsymbol{h}}^{LE}_{trim} \triangleq \tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK} - W_t : idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK} + W_t)$;
 5: Find ratio of the sum of negative to positive values of $\tilde{\boldsymbol{h}}^{LE}_{trim}$:
    $R^{\tilde{\boldsymbol{h}}^{LE}_{trim}} \triangleq |\sum(\tilde{\boldsymbol{h}}^{LE}_{trim}(\tilde{\boldsymbol{h}}^{LE}_{trim} < 0))|/\sum(\tilde{\boldsymbol{h}}^{LE}_{trim}(\tilde{\boldsymbol{h}}^{LE}_{trim} > 0))$
 6: if $C_1$ & $C_2$ & $C_3$ then
 7:     if $C_4$ then
 8:         $\boldsymbol{h}^{final} = \tilde{\boldsymbol{h}}^1$, $idx^{\boldsymbol{h}^{final}}_{LE-PK} = idx^{\boldsymbol{h}^0}_{LE-PK}$
 9:     else
10:         $\boldsymbol{h}^{final} = \tilde{\boldsymbol{h}}^0$,
11:         Run the LE algorithm with input $\boldsymbol{h}^{final}$ and find the first
            peak after LE index, referred to as $idx^{\boldsymbol{h}^{final}}_{LE-PK}$
12:     end if
13: else if $C_5$ then
14:     $\boldsymbol{h}^{final} = \tilde{\boldsymbol{h}}^0$,
15:     if $idx_{BS} == 1$ then
16:         Run the LE algorithm with input $\boldsymbol{h}^{final}$ and find the first
            peak after LE index, referred to as $idx^{\boldsymbol{h}^{final}}_{LE-PK}$
17:     else
18:         $idx^{\boldsymbol{h}^{final}}_{LE-PK} = idx^{\boldsymbol{h}^0}_{LE-PK}$
19:     end if
20: else
21:     $\boldsymbol{h}^{final} = \tilde{\boldsymbol{h}}^1$, $idx^{\boldsymbol{h}^{final}}_{LE-PK} = idx^{\boldsymbol{h}^0}_{LE-PK}$
22: end if
```

range estimation. These conditions aims to prevent undesired situations in multipath mitigation such as removing the LOS path. As an example, in Alg. 2, lines 1-5 define $R^{\tilde{\boldsymbol{h}}^0_{trim}}$ and $R^{\tilde{\boldsymbol{h}}^{LE}_{trim}}$ as metrics to evaluate the local behavior of $\tilde{\boldsymbol{h}}^0$ around the last mitigated path (i.e., idx_1) and the local peak after the LE index of $\tilde{\boldsymbol{h}}^0$ (i.e., $idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}$). As the values of $\tilde{\boldsymbol{h}}^0$ should be positive (due to absolute-squared operation), if the last mitigated path was a LOS path, there should be large negative values in $\tilde{\boldsymbol{h}}^0$, hence, the $R^{\tilde{\boldsymbol{h}}^0_{trim}}$ and/or $R^{\tilde{\boldsymbol{h}}^{LE}_{trim}}$ are large values. By setting a threshold on these metrics, STM prevents mitigating of the LOS path. We highlight that in Alg. 2, $C_1 \& C_2 \& C_3 \& \bar{C}_4$ or $\overline{(C_1 \& C_2 \& C_3)} \& C_5$ are the necessary and sufficient conditions for accepting $\tilde{\boldsymbol{h}}^0$ for range estimation. Furthermore, (5)-(7) are empirically designed by analyzing the range estimation performance using $\tilde{\boldsymbol{h}}^0$ when the UWB packet simulator based on the IEEE channel models [11] is employed.

5) ML interpolation block: In this final stage of STM, ML interpolation block (see Sec. II and [8] for details) will be applied based on inputs of $\boldsymbol{h}^{final}$ and $idx^{\boldsymbol{h}^{final}}_{LE-PK}$, which gives the final LOS delay estimate $\tau^{LOS}_{est/STM}$ that corresponds to the estimated peak of the LOS.

B. STMnet

In the STM algorithm, there are some thresholds (see (5)-(7)) which depends on the multipath environment, system parameters and nonidealities[4]. For a given set of propagation channels and system parameters, as we will show in Sec. IV, the ranging performance has a limit. This limitation is due to the fact that the best thresholds for a given channel are not necessarily the best for other channels.

To improve the ranging performance of the STM, we assist the STM by a NNs, which we referred to as STMnet. The block diagram of the ranging architecture using STMnet is shown in Fig. 4. STMnet estimates the ranging error of the STM. As it can be seen, first $\boldsymbol{h}^0$ is trimmed around $\tau^{LOS}_{est/STM}$. In particular, we refer to the trimmed CIR of length 61 samples as $\boldsymbol{h}^t \triangleq \boldsymbol{h}^0(\lfloor idx^{LOS}_{est/STM} \rfloor - 20 : \lfloor idx^{LOS}_{est/STM} \rfloor + 40)$, where $idx^{LOS}_{est/STM}$ stands for the index of $\tau^{LOS}_{est/STM}$ [5]. This choice is made as the $\boldsymbol{h}^0$ samples before and after $\lfloor idx^{LOS}_{est/STM} \rfloor$ can provide information about the type of the multipath channel. Furthermore, more samples after $\lfloor idx^{LOS}_{est/STM} \rfloor$ index is considered, as multipath paths arrives later than LOS path. The trained NN estimates the range error τ_e. Finally, the estimated LOS delay is given as $\tau^{LOS}_{est/STM+STMnet} = \tau^{LOS}_{est/STM} - \tau_e$. STMnet employs the STM algorithm as a pre-processing stage, which determines the region in $\boldsymbol{h}^0$ where the LOS path is positioned. To make the STMnet trainable for all channels, different power levels, and possible hardware nonideality impacts, we found that the input of the STMnet should be normalized. We used the 'max' normalization as used also in [9]. We refer to the normalized trimmed CIR as $\boldsymbol{h}^t_n = \boldsymbol{h}^t / max(\boldsymbol{h}^t)$.

The architecture of the STNnet is shown in Fig. 5. In Fig. 5, we show the input size of the STMnet as well as the output size of each hidden layer. $\boldsymbol{h}^t_n$ is first padded to have the length 64, which facilitates the dimension changing in the upcomming 2D convolution layers. As it can be seen, STMnet contains two 1D convolution layers, three 2D convolution layers, four fully connected layers, and one output layer, which outputs τ_e. Each 1D/2D convolution layers are followed by a batch normalization layer and the ReLu function is used as the activation of the fully connected layers. We use the notation "filters (n, m, s)" for convolution layers, where "(n, m)" are the size of the filter and "s" stands for the stride size of the convolution operation. The term "same" in Fig. 5 stands for selecting the padding such that the first dimension in 1D convolution layer and the first two dimensions in 2D convolution layers does not change. The output of the second 1D convolution layer is transformed into an $8 \times 8 \times 8$ tensor. This transformation is done by taking the 64 values along the first dimension of the last 1D convolution layer and arrange them into a 8×8 matrix. Furthermore, the output of the third 2D convolution layer is transformed to an FC layer of size 512. The use of 1D convolution layers can be motivated as the TX signal is convolved with the channel response and received by RX, hence, these layers can extract some temporal features about the LOS and other closed by NLOS paths. Furthermore,

[4] Parameters such as low-pass filter responses. Nonidealities such as residual uncompensated CFO/time slip after synchronization.

[5] Note that $idx^{LOS}_{est/STM}$ is not necessarily an integer index due to employing the ML interpolation block, hence, the floor operation is used.

$$C_1 = (R^{\tilde{\boldsymbol{h}}^0_{trim}} < T_1)|(R^{\tilde{\boldsymbol{h}}^{LE}_{trim}} < T_1)|(\tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}) > D_1\tilde{\boldsymbol{h}}^1(idx_1)),\ C_2 = (\tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}) > D_2N_{est})\&((\tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}-1) > D_3N_{est})|(\tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}+1) > D_3N_{est})) \quad (5)$$

$$C_3 = (\tilde{\boldsymbol{h}}^0(idx_1) > -D_4N_{est})\&((\tilde{\boldsymbol{h}}^0(idx_1-2) > -D_4N_{est})|(\tilde{\boldsymbol{h}}^0(idx_1+2) > -D_4N_{est})),\ C_4 = (\tilde{\boldsymbol{h}}^0(idx^{\tilde{\boldsymbol{h}}^0}_{LE-PK}) < D_1\tilde{\boldsymbol{h}}^1(idx_1))\&((R^{\tilde{\boldsymbol{h}}^0_{trim}} > T_1)|(R^{\tilde{\boldsymbol{h}}^{LE}_{trim}} > T_1)) \quad (6)$$

$$C_5 = ((R^{\tilde{\boldsymbol{h}}^0_{trim}} < T_1)\&(\tilde{\boldsymbol{h}}^0(idx_1) \geq 0)\&(\tilde{\boldsymbol{h}}^0(idx_1-1) \geq 0)\&(\tilde{\boldsymbol{h}}^0(idx_1) > \tilde{\boldsymbol{h}}^0(idx_1-1))\&(\tilde{\boldsymbol{h}}^0(idx_1) > \tilde{\boldsymbol{h}}^0(idx_1+1)))|(idx_{BS} == 1) \quad (7)$$

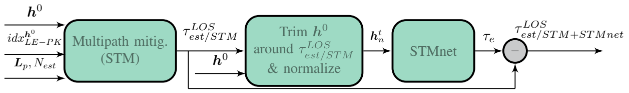
$\boldsymbol{h}^0$
$idx^{\boldsymbol{h}^0}_{LE-PK}$
$\boldsymbol{L}_p, N_{est}$
Multipath mitig. (STM)
$\tau^{LOS}_{est/STM}$
$\boldsymbol{h}^0$
Trim $\boldsymbol{h}^0$ around $\tau^{LOS}_{est/STM}$ & normalize
$\boldsymbol{h}^t_n$
STMnet
τ_e
$-$
$\tau^{LOS}_{est/STM+STMnet}$

Fig. 4. The block diagram of the ranging using STM aided by STMnet.

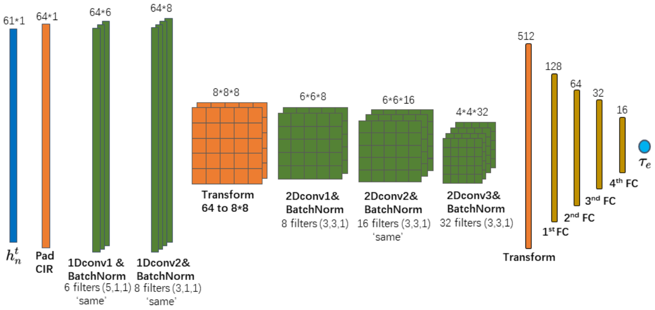
61*1
64*1
64*6
64*8
8*8*8
6*6*8
6*6*16
4*4*32
512
128
64
32
16
h^t_n
Pad CIR
1Dconv1 & BatchNorm 6 filters (5,1,1) 'same'
1Dconv2& BatchNorm 8 filters (3,1,1) 'same'
Transform 64 to 8*8
2Dconv1& BatchNorm 8 filters (3,3,1)
2Dconv2& BatchNorm 16 filters (3,3,1) 'same'
2Dconv3& BatchNorm 32 filters (3,3,1)
Transform
1st FC
2nd FC
3nd FC
4th FC
τ_e

Fig. 5. The proposed STMnet.

the use of 2D convolution layers can be motivated as they can provide some high level features about the channel type and the paths that are far apart. Finally, the fully connected layers estimates the range estimation error of the STM from such extracted features.

In [10] and [9], NNs are used for estimating the range from the unprocessed CIR. For the sake of comparison, in Sec. IV, we use these networks as a substitute to STMnet in the ranging architecture given in Fig. 4. We refer to such ranging schemes as "STM+(CN1D [10])" and "STM+(CN2D [9])", respectively. Note the we slightly adapted the input size of the networks in [10] and [9] such that $\boldsymbol{h}^t_n$ can be used as the input where the output is τ_e. As an ablation study, we also substitute the STMnet in Fig. 4 by a fully connected network, which we refer them as "STM+FC1". FC1 contains 9 hidden layers with Relu activation, where the number of neurons for each hidden layer is 64, 128, 128, 256, 50, 50, 32, 16, 16. We highlight that the $\boldsymbol{h}^t_n$ and τ_e are also the input and output of FC1. In Table I, we compare the complexity of STMnet and all other NNs used as a substitute of STMnet in Fig. 4 in terms of floating point operations (FLOPs) and the required memory. The complexity analysis is based on [13] and FC1 is designed with the same required memory as that of STMnet.

IV. Simulation results

We consider the mandatory Base Pulse Repetition Frequency (BPRF) mode [5] in the simulation results. We also use CM0, CM1, and CM3 [11] for performance evaluation. The parameters of STM algorithm (see Algorithms 1&2) used in the simulation results are $W_{BS} = 3$ $P_1 = 60000$, $P_2 = 30000$, $W_t = 3$, $T_1 = 0.13$, $D_1 = 0.1$, $D_2 = 10000$, $D_3 = 2000$, and $D_4 = 2$. For the sake of paper compactness, we only show the results of range estimation using SYNC CIR, however, the same methodology can be applied for range estimation using the STS CIR. We use a UWB packet simulator which emulates all the TX and RX hardware components of the taped-out chip [12]. 1GHz sampling rate, 40ppm Xtal offset, and 7-bit ADC resolution is used in the simulations. We use the Mean and Q95 of range error as the measure of range estimation accuracy and precision, respectively. The training process involved utilizing 20000 estimated CIR per received power (in dBm) and per channel (CM0, CM1, CM3). For each estimated CIR, the true range (ground truth), noise in the system, and channel realization are randomly and independently chosen. By applying the STM algorithm and using the true range τ_e is computed which is used for training. We use Adam optimizer with a learning rate of 0.0001, Mean Squared Error Loss function, and a batch size of 2048. For the simulation results, we use 5000 independent channel realizations per each power, where different NNs under our study are not trained for.

Table I
Complexity comparison of NNs used to assist STM in Fig. 4.

Neural Network	FLOPs	Memory
STMnet	232K	330KB
CN1D [10] / CN2D [9]	1.873M / 118K	2.61MB / 330KB
FC1	82K	330KB

In Fig. 6, we compare the mean and Q95 of range estimation error for STM, STM aidded by STMnet (STM+STMnet), and other schemes. As it can be seen, although the ML interp.-only improves the ranging performance compared to LE-only scheme in LOS channel (CM0), it gives a large ranging error in NLOS channels (CM1 and CM3). The performance of the ML interp.-only scheme is significantly improved using the STM algorithm. In particular, the (mean, Q95) of ranging error for the STM for -45dBm of RX power achieves up to (0.38cm, 0.9cm), (-0.79cm, 63.85cm), and (7cm, 105.9cm) for CM0, CM1, and CM3, respectively. Furthermore, the ranging error performance can be further improved using STM with different NNs in the ranging architecture of Fig. 4. In particular, the best performance in all channels are achieved by using the proposed STMnet. In particular, the (mean, Q95) of ranging error for the STM+STMnet for -45dBm of RX power achieves up to (-0.07cm, 0.63cm), (-0.49cm, 21.3cm), and (0.06cm, 39cm) for CM0, CM1, and CM3, respectively. From Fig. 6 and Table. I, one can conclude that in multipath environment the ranging architecture given in Fig. 4 with employing STMnet provides the best performance-complexity trade-off.

V. Conclusions

STM as a successive threshold-based multipath mitigation algorithm is proposed. It is shown that STM improves the ranging performance in multipath conditions compared to the LE-based scheme. The ranging performance of STM in multipath scenarios is improved by employing STMnet which estimates the range estimation error of the STM. The STM

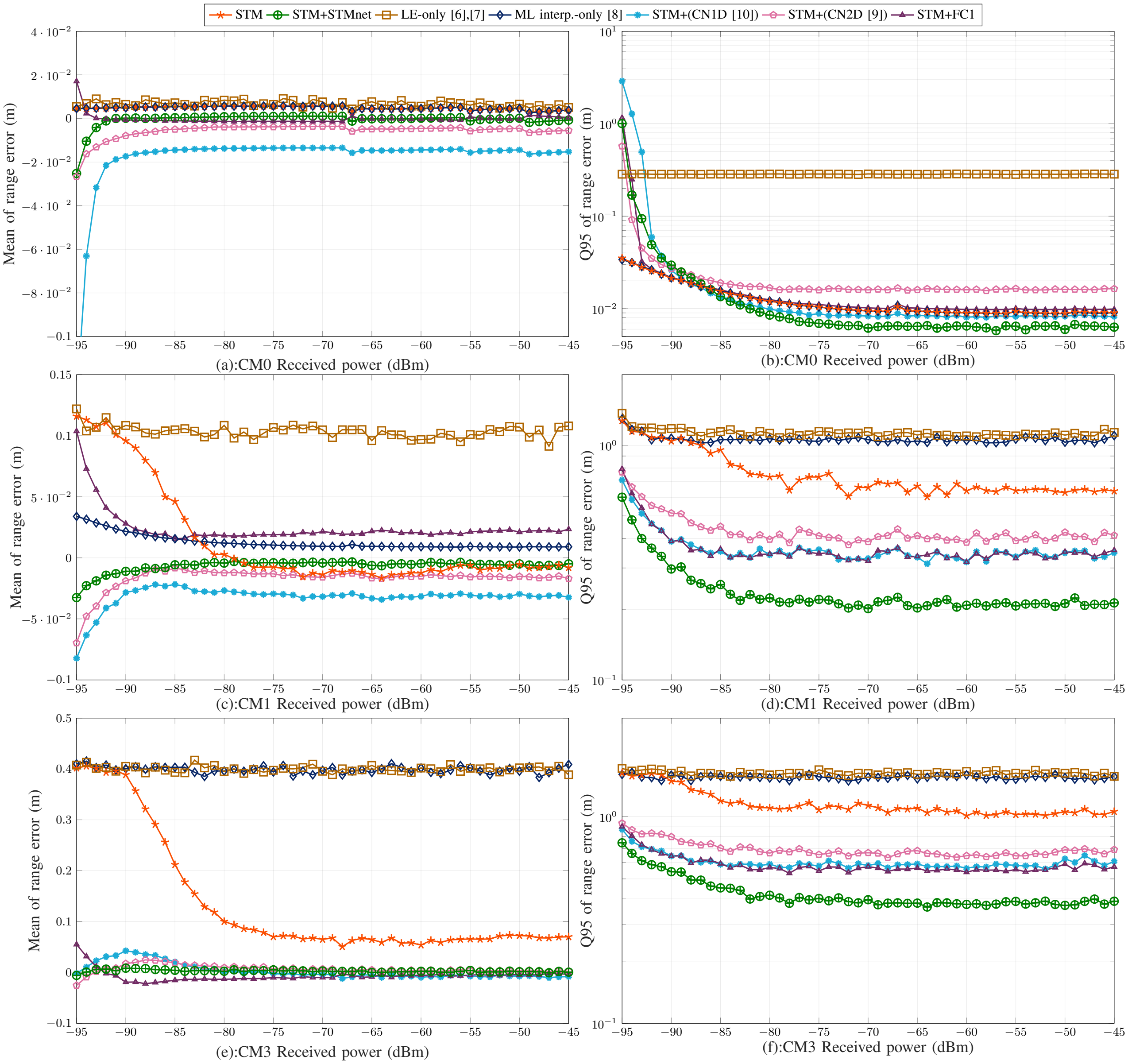

Fig. 6. Comparing the mean (left) and Q95 (right) of the range error for different STM, STMnet, LE-only [6], [7], ML interp.-only [8], STM+(CN1D [10]), STM+(CN2D [9]), and STM+(FC1).

algorithm acts as a feature extraction module for STMnet by providing the correct region in the CIR where the LOS lies. Simulation results confirm that STM+STMnet provides the best ranging performance in CM0, CM1, and CM3. Furthermore, we showed that the memory and computational complexity of STMnet is much smaller than other comparable NNs proposed for ranging in the literature.

References

[1] K. Zhao et.al., "Joint RFID and UWB technologies in intelligent warehousing management system," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 640–11 655, 2020.

[2] R. Keni et.al., "A sensor-fusion-system for tracking sheep location and behaviour," *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, 2020.

[3] R. S. Kulikov et.al., "Two-dimension positioning solution of high accuracy navigation and orientation for service robots," in *REEPE*, 2019.

[4] A. Kalyanaraman et.al., "CaraoKey : Car States Sensing via the Ultra-Wideband Keyless Infrastructure," in *SECON*, 2020.

[5] "IEEE standard for low-rate wireless networks–amendment 1: Enhanced ultra wideband (UWB) physical layers (PHYs) and associated ranging techniques," *IEEE Std 802.15.4z-2020*, pp. 1–174, 2020.

[6] Michael J. Kuhn et. al., "Adaptive leading-edge detection in UWB indoor localization," in *IEEE RWS*, 2010, pp. 268–271.

[7] W. Liu et. al., "TOA estimation in IR-UWB ranging with energy detection receiver using received signal characteristics," *IEEE Communications Letters*, vol. 16, no. 5, pp. 738–741, 2012.

[8] A. Sheikh et. al., "Accurate and efficient range estimation for IEEE 802.15.4z-compliant UWB system," *Accepted in WCNC*, 2024.

[9] J. Joung et. al., "CNN-based Tx–Rx distance estimation for UWB system localisation," *Electronics Letters*, vol. 55, no. 17, pp. 938–940, 2019.

[10] M. Ridolfi et. al., "UWB anchor nodes self-calibration in NLOS conditions: A machine learning and adaptive PHY error correction approach," *Wireless Networks*, vol. 27, no. 4, pp. 3007–3023, 2021.

[11] A. F. Molisch et. al., "IEEE 802.15.4a channel model - final report," Tech. Rep., Jan. 2005.

[12] E. Bechthum et. al., "A 3-10GHz 21.5mW/Channel RX and 8.9mW TX IR-UWB 802.15.4a/z 1T3R Transceiver," in *ESSCIRC*, 2022.

[13] T. Yep, "torchinfo," Mar. 2020. [Online]. Available: https://github.com/TylerYep/torchinfo