

Received September 14, 2020, accepted September 18, 2020, date of publication September 22, 2020, date of current version October 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025896

Portable Full Channel Sounder for Industrial Wireless Applications With Mobility by Using Sub-Nanosecond Wireless Time Synchronization

ÓSCAR SEIJO¹, IÑAKI VAL¹, (Senior Member, IEEE),

AND JESÚS ALBERTO LÓPEZ-FERNÁNDEZ²

¹Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), 20500 Arrasate/Mondragón, Spain

²Group of Theory of Signal and Communications, University of Oviedo, 33203 Gijón, Spain

Corresponding author: Iñaki Val (ival@ikerlan.es)

This work was supported in part by U4INDUSTRY and 5G4BRIS3 (ELKARTEK) Projects of the Basque Government, Spain, in part by the Ministerio de Ciencia, Innovación y Universidades through the Project under Grant TEC2017-86619-R (ARTEINE), and in part by the Gobierno del Principado de Asturias through the Project under Grant GRUPIN-IDI-2018-000191.

ABSTRACT Wireless communications have attracted great interest from the industry due to its lower costs and the possibility of enabling new use cases. The new use cases are commonly related to mobile robotics, such as Unmanned Aerial Vehicles or terrestrial robots. The design of wireless systems for these use cases requires deep knowledge of the channel behavior and therefore the use of channel sounders that are able to measure the phase and gain of the channel is mandatory. However, channel sounders require very precise synchronization, which is typically implemented through a wired connection between the nodes. As a result, the mobility of the nodes is constrained to the length of the synchronization wire. In this paper, we present the design and implementation of a portable 802.11-based channel sounder that exploits a sub-nanosecond wireless time synchronization scheme. Thanks to the wireless synchronization, the channel sounder can operate as a channel sounder with wired synchronization, being able to measure synchronized channel impulse response samples. From these samples, relevant channel parameters can be extracted, including the Power Delay Profile, Doppler spectrum, and channel delay. The channel sounder presents several advantages compared to a conventional channel sounder with wire synchronization, such as smaller size and weight, simpler and more flexible operation, and it does not constraint the movement of the environment or nodes. The verification of the channel sounder through a wireless channel emulator and the measurement carried out in an industrial facility confirm its feasibility for industrial wireless and other applications.

INDEX TERMS UAV, channel sounder, wireless synchronization, PTP, mobile robotics, drones, IEEE 802.11, industry 4.0., wireless channel measurements, channel emulator.

I. INTRODUCTION

Wireless communications are an emerging trend in the context of the industry 4.0, due to the reduced deployment and maintenance costs and free movement of the nodes connected by wireless. Free movement is the key enabler of several industrial applications, which includes fixed robots with moving elements, small terrestrial robots inside factories, and Unmanned Aerial Vehicles (UAVs). However, the use of wireless systems in industrial applications is not straightforward. The reasons are diverse, though two

of the top issues in industrial wireless are the harshness and unpredictability of industrial wireless channels, and that wireless solutions are not typically designed to fulfill the requirements of the industrial applications.

A common methodology for the design and implementation of industrial wireless systems is as follows. Initially, the application requirements must be translated into communication requirements (real-time, latency, data rate, etc.). In addition, deep knowledge of the deployment scenario is mandatory (i.e. the wireless channel behavior). Based on the requirements and the wireless channel behavior, an appropriate wireless solution is designed and implemented. Afterward, the system may be in-lab verified using a

The associate editor coordinating the review of this manuscript and approving it for publication was Fakhrol Alam¹.

wireless channel emulator [1]. Finally, the system is deployed in the real Radio Frequency (RF) propagation scenario.

As can be seen, the initial steps (requirement description and wireless channel knowledge) determine the wireless solution and thus they must be carefully studied. Specifically, wireless channel characterization is highly relevant in industrial wireless systems, where the RF propagation is, in general, as noted before, very harsh.

A convenient way to improve the wireless channel knowledge in a specific scenario is to perform a channel measurement campaign using a channel sounder. Unfortunately, channel measurement campaigns in industrial applications, especially in those applications that include mobility, are quite complex [2] and their availability in the literature is limited. The main reasons behind the complexity are that the wirelessly connected mobile nodes may have free and unpredictable movements, they have payload constraints in terms of size and weight, (i.e. they cannot carry large channel sounders), and the wireless channel may rapidly vary due to the movement of the nodes.

This paper presents the design and implementation of a portable channel sounder using a high-performance wireless time synchronization scheme [5]. This channel sounder has been built over a hardware (HW)-based Software Defined Radio (SDR) platform, using an 802.11g modem [4] previously implemented over the same platform. It has a bandwidth of 16.25 MHz, a channel sampling rate of 500 μ s and it operates in carrier frequencies from 2.3 to 2.6 GHz. These features provide real-time, fast, wideband measurements of wireless channels in industrial scenarios with mobility and other domains. The channel sounder has been in-lab verified over a Non-Line-of-Sight channel model with mobile conditions using a wireless channel emulator, and we have carried out a channel measurement campaign in an industrial facility. This paper extends our previous work [6] by providing a comprehensive review of the state-of-the-art in channel sounding, the definition of the synchronization scheme and its applicability to channel sounding, a detailed view of the HW/SW implementation over the ADI RF SOM platform, and a measurement campaign performed in a mechanical workshop.

The main novelty of our channel sounder resides in the use of our high-performance wireless time synchronization scheme [5]. First, thanks to the wireless time synchronization, the channel sounder is able to perform wideband, time-variant wireless channels measurements without requiring any kind of external synchronization (e.g. wired synchronization or Global Positioning System). Second, since the channel sounder does not rely on any external synchronization, its operation does not depend on the scenario and it is simpler than a traditional channel sounder which must coordinate different systems. Finally, the channel sounder is smaller and more cost-effective since it requires fewer components to operate.

The rest of the paper is organized as follows. First, Section II presents the state of the art in the implementation

of channel sounders. In Section III, the high-performance wireless time synchronization scheme is presented. The channel sounder design is presented in Section IV and its implementation is described in Section V. The verification of the channel sounder using a wireless channel emulator is presented in Section VI. A measurement campaign in an industrial workshop is reported in Section VII. Finally, Section VIII summarizes some conclusions of the work.

II. BACKGROUND AND RELATED WORK

This section reviews some conventional implementations of channel sounders designed for industrial wireless communications and other domains and highlights the common synchronization schemes used in channel sounding.

We have classified the state-of-the-art channel sounders into two classes based on whether they require synchronization or not: limited channel sounders (do not require synchronization), or full channel sounders (require synchronization). On the one hand, limited channel sounders [3] are designed to measure parameters such as the Received Power, the PER, or the system throughput. The information provided by limited channel sounders is not raw, i.e. they cannot directly measure the gain and phase of the CIR, and hence the knowledge that can be extracted from a measurement campaign is rather limited. They are commonly built with Software Defined Radio (SDR) or commercial devices [7]. These channel sounders are preferred for applications with mobility, despite their lower performance, because of their size, weight, and cost, and because they do not require any kind of synchronization. On the other hand, full channel sounders are designed to measure the gain and phase of the CIR [8]. From the CIR, relevant channel statistics, such as the Power Delay Profile (PDP), the fading distributions, or the Doppler spectrum, can be obtained. Note that the PDP and the fading distributions require the measurement of the gain of the CIR, whereas the Doppler spectrum requires both gain and phase. Thus, it is of utmost importance to have adequate phase and frequency synchronization.

The next three subsections describe the state-of-the-art in the implementation of limited and full channel sounders and summarize the synchronization techniques commonly used in full channel sounders.

A. LIMITED CHANNEL SOUNDERS

Limited channel sounders do not include any kind of synchronization in their design. Due to this, they are not able to directly measure the phase and gain of the CIR, though only some specific parameters of the channel, such as the PDP, the wireless system throughput, or PER. Limited channel sounders are usually built with low-cost COTS equipment, such as smartphones, small laptops with a wireless Network Interface Cards (LTE/WiFi), or SDR equipment, such as Universal Software Radio Peripheral (USRP) [7].

The literature about channel measurements using limited channel sounders in industrial scenarios is mainly focused on mobile robotics, such as UAV communications. For instance,

an extensive survey on channel measurements and modeling for UAV communications is presented in [2], where more than 20 measurement campaigns are analyzed using different channel sounders. Most of them are done with compact limited channel sounders. For instance, in [9] a measurement campaign using COTS 802.11 HW running in the 2.4 GHz band is presented. The measurements were carried out from air to air in an open field with different flying altitudes. The measured variables were the received power (i.e. the channel gain), the 802.11 throughput, and the PER. [10] is another example of a measurement campaign using COTS HW. In this case, the authors used LTE running at 800 MHz, and the UAV was deployed in a rural area. The measured parameters were the PER and the fading probability. As can be seen, these measurement campaigns provide some secondary parameters that may help to characterize the behavior of the channel, though they do not exactly represent the channel itself.

B. FULL CHANNEL SOUNDERS

There are several implementations of full channel sounders in the literature based on different techniques and with different features in terms of BW, maximum Doppler frequency shift, dynamic fading range, etc. Based on the technique used to measure the CIR, three main full channel sounders can be distinguished: Vector Network Analyzer (VNA)-based, time domain-based, or frequency domain-based. The common requirement of all of them is the synchronization between the master and slave.

VNA-based channel sounders use a VNA with at least two ports and two antennas connected at each port. The VNA is configured to measure the S21 parameter, which is the impulse response from port 1 to port 2 (i.e. the CIR). The measurement is performed by transmitting small BW pulses at different frequencies to cover a large BW. These channel sounders are very appropriate for the measurement of very large BW (>10 GHz), but their operation is very slow, and they cannot be used to evaluate time-variant channels. As an example of a VNA-based channel sounder, in [11] P. Pajusco *et al.* presented an extensive 3-dimension measurement campaign of the residential indoor channel using a VNA-based channel sounder. The channel sounder used in the measurement campaign had an effective BW of 10 GHz and an array antenna electronically controlled to provide beamforming capabilities. The measurement campaign took 130 hours for 32 static measurement locations, which represents the slow capture speed of VNA-based channel sounders.

Time-domain and frequency-domain channel sounders periodically transmit short signals and they are suitable to measure time-variant channels [12]. Their operation is as follows. The master and slave are perfectly synchronized, and their operation is synchronously triggered. When the master is triggered, it transmits a specific waveform $s(t)$ that traverses the communication channel and it is received by the slave antenna as the signal $r(t)$.

An appropriate waveform for these channel sounders has the next properties: low autocorrelation outside of 0 (i.e. its autocorrelation must be similar to a Delta), low Peak-to-Average Power Ratio (PAPR), flat frequency response, and a duration smaller than the channel coherence time. These properties are also desirable in the preambles of frames of wireless systems [13]. In the time domain, Pseudo-Noise (PN) sequences are commonly used [14], whereas Zadoff-Chu sequences [15] are chosen for frequency-based channel sounders because they provide minimum PAPR with flat frequency response.

Both time-domain and frequency-domain channel sounders are based on the correlation of $r(t)$ with $s(t)$ to estimate the CIR. The main difference among them is the way to implement the correlation, time-domain channel sounders perform the correlation in the time domain whereas frequency-domain channel sounders perform the correlation in the frequency domain.

As an example of a time domain-based channel sounder, a PN sequence-based channel sounder is shown in [16]. This channel sounder is built with instrumentation equipment and offers a chip rate of 12.5 GHz and a CIR sampling period T_{cs} of 0.625 ms, which enables the measurement of fast time-varying channels. As an example of a frequency domain-based channel sounder, B. T. Maharaj *et al.* presented in [12] a frequency-domain channel sounder with 100 MHz BW, f_c from 2 to 8 GHz, $T_{cs} = 3.2$ ms and Multiple-In, Multiple Out (MIMO) support. The second channel sounder is built with low-cost components and thus it offers lower capabilities than the former.

C. SYNCHRONIZATION IN FULL CHANNEL SOUNDERS

In order to measure the CIR, full channel sounders require time and frequency synchronization between the master and slave. In state-of-the-art full channel sounders, the synchronization is typically delivered from the master to the slave through one or more wired connections. The time synchronization is used to simultaneously trigger the master and slave operations, and the frequency synchronization eliminates the frequency offset of the local oscillators, enabling carrier phase measurements.

Regarding the necessity of a wired connection between master and slave nodes, there are different implementations of full channel sounders that have been classified into three main classes: permanent wired connection (A), wired connection during the initial synchronization phase (B), and totally non-wired connection (C). Table 1 classifies some relevant channel sounders in the literature based on their synchronization technique.

Some examples of the first type (permanent wired connection) are presented in [12], [16]–[18], and [19]. In [12], the authors present a frequency domain-based full channel sounder with a BW of 100 MHz and MIMO support. This channel sounder uses two rubidium clocks (one in each side), and a synchronization unit that simultaneously triggers the master and slave. The synchronization unit is connected by

TABLE 1. Summary of state-of-the-art channel sounders based on their synchronization.

Ref.	Year	Sync Class	Synchronization method	Oscillators	Channel sounder type
[12]	2008	A	Wired Trigger from master to slave	Two rubidium one at each side	Frequency-domain
[16]	2015	A	Common Synchronization unit	One Rubidium	Time-domain
[17]	2016	A	Common Synchronization unit	One Rubidium	Time-domain
[18]	2017	A	Common Synchronization unit	One Rubidium	Frequency-domain
[19]	2019	A	Common Synchronization unit	One Rubidium	Time-domain
[20]	2008	B	Sync phase and free-run phase	Two rubidium one at each side	Frequency-domain
[8]	2017	B	Sync phase and free-run phase	Two rubidium one at each side	Time-domain
[21]	2015	C	Rubidium stability	Two rubidium one at each side	Time-domain
[22]	2019	C	GPS	Two rubidium one at each side	Frequency-domain

fiber to both the master and slave. The channel sounder presented in [16] has two connections between the master and slave: one connection to share the clock and the other one to synchronously trigger the transmitter and receiver. Other full channel sounders follow similar procedures sharing the clock through other signals. For instance, in [17] the master shares its local oscillator with the slave through a wired connection. Additionally, the master sends a trigger to the slave every time that a sequence is transmitted.

In some applications, the channel sounder cannot have a permanent wired connection between the master and slave (e.g. mobility applications or long distances). In this case, a two-step procedure with a wired synchronization phase and a free-running phase is commonly used to eliminate the wired connection during the measurement campaign (Sync Class B). The RUSK channel sounder, which was used to develop the Winner II channel models [20], is an example of a Sync Class B channel sounder. The RUSK channel sounder of [20] uses a signal generated in the frequency domain with an effective BW of 640 MHz, it operates in the 5 GHz band, and it provides a maximum transmission power of 33 dBm. The channel sounder uses two rubidium clocks, one for the master and one for the slave. In the synchronization phase, the rubidium clock of the slave is connected to the rubidium clock of the master. This operation is used to synchronize the rubidium clock of the slave. Once they are stable and synchronized, the synchronization wire can be eliminated, and both clocks will slowly drift apart. The small drift of Rubidium clocks ensures that the system will be synchronized for few hours. [8] presents a B-class time-domain channel sounder though based on PN sequences. It follows a similar synchronization philosophy as [20] with one synchronization phase and one free-running phase.

Finally, some channel sounders do not require a wired connection at all (Sync Class C). For instance, The channel sounder described in [21] uses two rubidium clocks (one in the master and one in the slave) that provide good frequency stability, and it relies on the oscillator characteristics to maintain an adequate relative synchronization. On the downside, the channel sounder cannot measure the true

channel delay since the master and slave are not synchronized to a common time. In [22], it is presented a millimeter-Wave channel sounder that is able to operate in real-time. This channel sounder uses GPS as an external synchronization signal and, as a result, it does not require a wired connection between the nodes. On the downside, the channel sounder of [22] cannot be used in scenarios where the GPS signal is weak, such as indoor, underground (e.g. tunnels), and on bad weather conditions.

In summary, both limited and full channel sounders have specific drawbacks which either reduce their performance or their flexibility and operability. In contrast, the channel sounder proposed in this paper overcomes the performance issues of limited channel sounders by providing real-time raw CIR measurements, while still maintaining their characteristic advantages in terms of small form factor, weight, and flexibility. As a result, the presented channel sounder stands as a reasonable trade-off between a full channel sounder and a limited one and merges the advantages of both design philosophies.

III. HIGH-PERFORMANCE WIRELESS TIME SYNCHRONIZATION

In this section, we present the main insights of our high-performance wireless time synchronization scheme, which has been previously published in [5]. The wireless time synchronization method is based on Precision Time Protocol (PTP) combined with an enhanced timestamping method that has been demonstrated to be effective in both Line-of-Sight (LoS) and NLoS conditions. This method provides sub-nanosecond time synchronization precision in virtually any wireless condition and hence it may suitable to perform channel measurements.

A. PRECISION TIME PROTOCOL

IEEE 1588 [23], commonly known as PTP, is the *de-facto* standard protocol for time synchronization. PTP synchronization is based on a frame exchange between a master and a slave. The frame exchange comprises 3 frames, which are depicted in Fig. 1. First, the master transmits a PTP

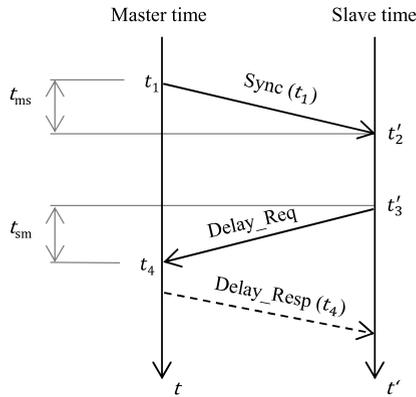


FIGURE 1. PTP frame exchange, being t the time of the master clock and t' the time of the slave clock.

sync frame, which carries the transmission timestamp of the frame (t_1). This frame is received by the slave, which takes the timestamp (t'_2). Then, the slave responds with a delay request frame and two more timestamps are taken (t'_3, t_4). Finally, the master delivers the timestamp t_4 to the slave through a delay response frame. With the timestamps, the slave calculates the time error as

$$\tilde{t}_{ms} = \frac{t'_2 - t_1 + t_4 - t'_3}{2}, \quad (1)$$

$$\tilde{t}_o = t'_2 - t_1 - \tilde{t}_{ms}, \quad (2)$$

where \tilde{t}_{ms} represents the estimated channel delay and \tilde{t}_o is the estimated clock offset between the master and slave clocks. The wireless channels are reciprocal at a specific instant and thus the delay from master to slave t_{ms} and the delay from slave to master t_{sm} can be considered approximately equal.

PTP describes the procedure to achieve perfect time synchronization, though its performance depends on the timestamping quality, which is implementation-dependent. The timestamping quality is usually improved if they are taken as close as possible to the physical layer (PHY). Even so, conventional PHY timestamps over wireless are limited by the BW of the wireless system and by the wireless channel properties. Thus, the precision provided by PTP with conventional timestamps is in the tens of nanoseconds range for 20 MHz BW [24], which is out of the synchronization required to perform channel measurements.

B. ENHANCED TIMESTAMPS

Conventional PHY timestamps are based on the detection of a pseudorandom discrete sequence $s[k]$ with length K known by the transmitter and receiver. This sequence is usually named the preamble of the frame, and it is transmitted at the start of the frame. The relation between the $s[k]$ and the received sequence $r[k]$, given that both sequences are sampled at the same frequency is the convolution of $s[k]$ and the discrete equivalent CIR $p[k]$ summed with the communication noise $n[k]$

$$r[k] = (s * p)[k] + n[k]. \quad (3)$$

The sequence $p[k]$ is the discrete-equivalent version of the time-domain CIR filtered with the pulse shaping and matched filters of the transmitter and receiver and sampled at the receiver with a sampling period T .

The detection of the frame is based on the cross-correlation between the $r[k]$ and $s[k]$. If there is a peak in the correlator above a threshold (θ), it means that a frame has been detected

$$R_{r,s}[n] = (r \star s)[n], \quad (4)$$

$$n_{ini} = \min \{n \in \mathbb{N} / |R_{r,s}[n]|^2 > \theta\}, \quad (5)$$

being n_{ini} the detection instant. It must be noted that $R_{r,s}[n]$ is an estimation of $p[k]$ given that the autocorrelation of $s[k]$ is approximately equal to a Kronecker Delta. The relation between n_{ini} and the conventional timestamp (t_{rx}) is

$$t_{rx} = T \cdot n_{ini}. \quad (6)$$

The quality of the estimation of t_{rx} depends on the timestamping resolution (given by T) and on the wireless channel properties. For instance, the time dispersion can introduce errors in the peak detection, because signal replicas received at different instants will generate different peaks in $R_{r,s}[n]$. The channel variability and its consequences in the timestamping quality are further detailed in [5] and [24].

The enhanced timestamps apply a novel correction factor $\bar{\tau}$ to the initial timestamp t_{rx} . The correction factor $\bar{\tau}$ is based on the discrete version of the mean delay spread operator $MDS\{\cdot\}$ [25] applied to $p[k]$

$$\bar{\tau} = MDS\{p[k]\} = T \frac{\sum_{k=k_s}^{k_s+K-1} |p[k]|^2 k}{\sum_{k=k_s}^{k_s+K-1} |p[k]|^2}, \quad (7)$$

$$t_{e,rx} = t_{rx} + \bar{\tau}, \quad (8)$$

being k_s the start of the sum window and K the length of the window. k_s should be equal to the position of the first non-zero element of $p[k]$, and K to its length. $p[k]$ is not known in advance, but it can be estimated ($\tilde{p}[k]$) through different procedures. For instance, it can be obtained from $R_{r,s}[n]$, or it can be estimated in the frequency domain $\tilde{P}[l]$ and then converted to the time domain. In this work, we are using the second option, because the CIR in an OFDM system is usually estimated in the frequency domain.

The mean delay spread operator $MDS\{\cdot\}$ has two relevant properties for timestamping. First, it is linearly affected by the phase difference between the transmitter and receiver sampling clocks, and hence its resolution is not constrained to T . Second, it is robust against multipath propagation, because it calculates the channel delay based on all the elements of the CIR. The enhanced timestamps greatly outperform the conventional timestamps, pushing the PTP performance to the sub-nanosecond range even in NLoS and time-variant conditions [5]. The performance range provided by this scheme may be more than enough to perform synchronization in channel sounding.

IV. CHANNEL SOUNDER DESIGN

To tackle the implementation of the channel sounder, a conventional approach would be to fully implement the time synchronization algorithm in HW. Nonetheless, a convenient procedure to reduce the implementation effort is to perform the time synchronization in a post-processing step. In this case, the HW just needs to collect the CIRs and timestamps from the frame exchange without synchronization. Afterward, the post-processing algorithm calculates the channel delay and the synchronization error. Finally, the time synchronization error is used to virtually correct the local oscillator and to align the CIRs. The HW operation and the offline time synchronization algorithm are described in the subsequent subsections.

A. CHANNEL SOUNDER OPERATION

Full channel sounders commonly use a unidirectional communication, where the master periodically transmits a sequence and the slave detects the sequence. However, this channel sounder uses a full-frame exchange, and thus a bidirectional communication is required. We have used the OFDM-based 802.11 PHY to implement the channel sounder, but the design can be generalized to other wireless standards or more efficient custom solutions.

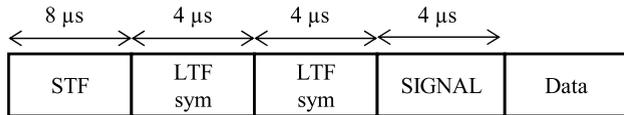


FIGURE 2. 802.11 frame format.

From the channel sounding perspective, the most relevant element of the standard is the preamble. The 802.11 legacy preamble (Fig. 2) uses a 20 MHz bandwidth and comprises two fields, the Short Training Field (STF) and the Long Training Field (LTF). The STF is mainly used for Automatic Gain Control (AGC), and Carrier Frequency Offset (CFO) correction. The LTF is used to detect the start of a frame. The LTF is divided into two OFDM symbols with a length of 4 μs each. Each symbol comprises 64 samples plus a cyclic prefix of 16 samples. In this case, we are using the 64 samples of the second LTF symbol as the $s[k]$ sequence. In the frequency domain, the 802.11 OFDM modulation has 64 subcarriers: 52 effective subcarriers, 1 dc subcarrier, and 11 guard subcarriers. Hence, its effective bandwidth is 16.25 MHz.

We have designed a modified version of the PTP frame exchange with only two frames to perform both the channel measurements and the time synchronization. This frame exchange is very convenient because it minimizes the channel sampling period (T_{cs}). The modified frame exchange is depicted in Fig. 3, being i the index of the frame exchange. The master transmits every T_{cs} a modified sync frame to the slave and takes the Tx timestamp ($t_{1,i}$). The modified sync frames are received by the slave, which takes the Rx timestamp ($t'_{2,i}$) and estimates the CIR from the modified

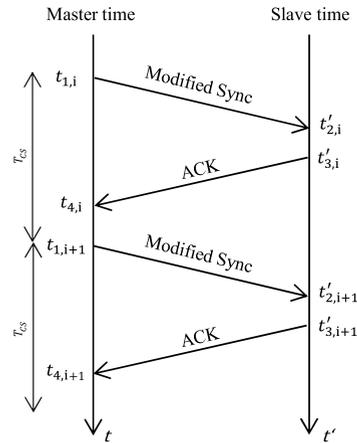


FIGURE 3. Modified PTP frame exchange with a reduced number of frames.

Sync. Then, the slave answers to the modified Sync frame with a standard 802.11 ACK and takes the Tx timestamp ($t'_{3,i}$). Afterward, the master receives the ACK, takes the Rx timestamp ($t_{4,i}$), and estimates the CIR from the ACK. The frame exchange is indefinitely executed until the end of the measurement.

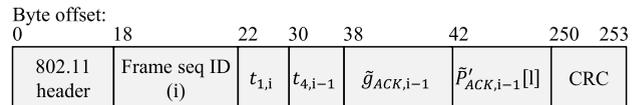


FIGURE 4. Modified Sync frame format.

The modified sync frame joins the functionalities of the PTP sync and PTP delay response frames, whereas the ACK transmitted from the slave to the master acts as the PTP delay request. The structure of the sync frame is depicted in Fig. 4. The frame comprises 254 bytes and has 5 data fields. The 802.11 header is used to be compliant with the 802.11 standard. The *Frame seq ID* field is used to identify the current frame, and it contains the channel sample index i . The Most Significant Bit (MSB) of the *Frame seq ID* field indicates if the ACK of the previous exchange was correctly received. The field $t_{1,i}$ contains the transmission timestamp of the sync frame. $t_{4,i-1}$, $\tilde{g}_{ACK,i-1}$, and $\tilde{P}'_{ACK,i-1}[I]$ are related to the previous ACK received by the master. $t_{4,i-1}$ contains the ACK timestamp, $\tilde{g}_{ACK,i-1}$ is the estimation of the channel gain in dB, which is computed as the estimated Rx power minus the Tx power in dBm, and $\tilde{P}'_{ACK,i-1}[I]$ is the estimated CIR of the ACK in the frequency domain normalized to unit energy. The ACK fields are set to 0 if the MSB bit of the *Frame seq ID* field is set to 0. $\tilde{P}'_{ACK,i-1}[I]$ is represented by its IQ components in complex format. It uses 4 bytes for each subcarrier and has a length $L = 52$. Finally, the standard 4-byte Cyclic Redundancy Check (CRC) is used to check if there is an error in the frame. At the end of each frame exchange, the slave computes the estimated CIRs in the time

domain as

$$\tilde{p}_{ACK,i}[n] = \text{IFFT} \left\{ 10^{\frac{\tilde{g}_{ACK,i}}{20}} \tilde{P}'_{ACK,i}[L] \right\}, \quad (9)$$

$$\tilde{p}_{Sync,i}[n] = \text{IFFT} \left\{ 10^{\frac{\tilde{g}_{Sync,i}}{20}} \tilde{P}'_{Sync,i}[L] \right\}, \quad (10)$$

being $\text{IFFT}\{\cdot\}$ the inverse Fourier transform, $\tilde{P}'_{Sync,i}[L]$ the estimated CIR from the Sync frame normalized to unit energy, and $\tilde{g}_{Sync,i}$ the estimation of the channel gain in dB. Then, the slave stores:

- The channel sample index, *Frame seq ID*: i .
- The four timestamps: $t_{1,i}, t'_{2,i}, t'_{3,i}, t_{4,i}$.
- The estimated CIR from the Sync: $\tilde{p}_{Sync,i}[n]$.
- The estimated CIR from the ACK: $\tilde{p}_{ACK,i}[n]$.
- A Boolean to indicate if the frame exchange was correctly performed: $exchange_{ok,i}$.
- A Boolean to indicate if the sync frame was correctly received: $SyncFrame_{ok,i}$.
- Additionally, the slave also stores the number of frame exchanges (I), the carrier frequency (f_c), and T_{CS} .

B. OFFLINE TIME SYNCHRONIZATION ALGORITHM

The offline time synchronization algorithm is used to align \tilde{p}_{Sync} to an absolute time reference and to compensate the local oscillator drift. The post-processing is divided into several steps, which may be classified into three main tasks: compute the enhanced timestamps, apply the PTP correction algorithm, and correct \tilde{p}_{Sync} . The first two tasks are summarized in Algorithm 1 and further explained below.

The algorithm output is the time synchronization error ($\tilde{t}_{err,i}$) and the conventional timestamps ($t'_{c2,i}$). The algorithm uses the enhanced timestamps combined with a Proportional Integral (PI) filter with constants (k_p, k_{int}) to minimize the time synchronization error. $t_1, t'_2, t'_3, t_4, exchange_{ok}$ are vectors with length I , and \tilde{p}_{ACK} and \tilde{p}_{Sync} are matrices with I rows and $N = 64$ columns.

In step 1., the slave timestamps $t'_{2,i}$ and $t'_{3,i}$ are corrected based on the accumulated time synchronization error ($\tilde{t}_{accum\ error}$). The result is the corrected conventional timestamp ($t'_{c,2,i}$) and the corrected transmission timestamp ($t'_{c,3,i}$) after applying the time synchronization algorithm.

In step 2., the algorithm checks if the frame exchange was successfully done. In case that $exchange_{ok} = false$, it will mean that one of the frames in the i frame exchange failed. In this case, the time synchronization error ($\tilde{t}_{o,i}$) is assumed to be 0, and $\tilde{t}_{accum\ error}$ is updated only with the integral part of the PI filter. Then, the algorithm skips the iteration.

In step 3., The enhanced timestamps ($t'_{e,2,i}, t_{e,4,i}$) are computed. To do so, the conventional timestamps $t'_{c,2,i}$ and $t_{4,i}$ are corrected by the mean delay spread ($MDS\{\cdot\}$) of $\tilde{p}_{Sync,i}[n]$ and $\tilde{p}_{ACK,i}[n]$ respectively.

In step 4., the PTP synchronization algorithm is used to compute the time synchronization error. First, $\tilde{t}_{ms,i}$ and $\tilde{t}_{o,i}$ are obtained. Then, $\tilde{t}_{o,i}$ is introduced to a PI filter with constants k_p, k_{int} and the error of this frame exchange (\tilde{t}_{err}) is computed. \tilde{t}_{err} is accumulated in $\tilde{t}_{accum\ error}$.

Algorithm 1 Offline Time Synchronization Algorithm

Input: $\tilde{p}_{ACK}, \tilde{p}_{Sync}, t_1, t'_2, t'_3, t_4, exchange_{ok}, k_p, k_{int}, I$
Output: $\tilde{t}_{err,i}, t'_{c,2}$
Initialization:
 $t_{Int} = 0.$
 $\tilde{t}_{accum\ error} = 0.$
for $i = 0$ to $I - 1$ do:
1. Correct slave clock timestamps:
1.1. $t'_{c,2,i} = t'_{2,i} - \tilde{t}_{accum\ error}.$
1.2. $t'_{c,3,i} = t'_{3,i} - \tilde{t}_{accum\ error}.$
2. if $exchange_{ok,i} = false$ do:
2.1. $\tilde{t}_{err,i} = k_{int} t_{Int}.$
2.2. $\tilde{t}_{accum\ error} = \tilde{t}_{accum\ error} + \tilde{t}_{err,i}.$
2.3. Go to the next iteration.
3. Take the enhanced timestamps:
3.1 $t'_{e,2,i} = t'_{c,2,i} + DS \{ \tilde{p}_{Sync,i}[n] \}.$
3.2 $t_{e,4,i} = t_{4,i} + DS \{ \tilde{p}_{ACK,i}[n] \}.$
4. Compute Time Sync error:
4.1. $\tilde{t}_{ms,i} = \frac{t'_{e,2,i} - t_{1,i} + t_{e,4,i} - t'_{c,3,i}}{2}.$
4.2. $\tilde{t}_{o,i} = t'_{e,2,i} - t_{1,i} - \tilde{t}_{ms,i}.$
4.3. $t_{Int} = t_{Int} + \tilde{t}_{o,i}.$
4.4. $\tilde{t}_{err,i} = k_p \tilde{t}_{o,i} + k_{int} t_{Int}.$
4.5. $\tilde{t}_{accum\ error} = \tilde{t}_{accum\ error} + \tilde{t}_{err,i}.$
end for
return

Finally, the CIRs from the Sync frames ($\tilde{p}_{Sync,i}[n]$) are time and phase-aligned based on the result of the synchronization algorithm. To do so, $\tilde{p}_{Sync,i}[n]$ must be shifted by a delay d_i using a fractional delay filter. d_i can be obtained as the time difference between the detection of the sync frames ($t'_{c,2,i}$) and its transmission ($t_{1,i}$)

$$d_i = (t'_{c,2,i} - t_{1,i}). \quad (11)$$

In addition, the CFO in each frame exchange ($\tilde{f}_{o,i}$) can be estimated from $\tilde{t}_{err,i}$

$$\tilde{f}_{o,i} = \frac{\tilde{t}_{err,i}}{T_{CS}} f_c. \quad (12)$$

Then, the relative carrier frequency phase between the master and slave ($\tilde{\phi}_i$) is

$$\tilde{\phi}_i = \tilde{f}_{o,i} \cdot T_{CS} = \tilde{t}_{err,i} \cdot f_c. \quad (13)$$

Given $\tilde{\phi}_i$ and d_i , $\tilde{p}_{Sync\ aligned,i}[k]$ is computed using a fractional delay filter as

$$\tilde{p}_{Sync\ aligned,i}[k] = e^{-j\tilde{\phi}_i} \sum_{n=-\infty}^{+\infty} \tilde{p}_{Sync,i}[n] \text{sinc} \left(k - n - \frac{1}{T_s} d_i \right). \quad (14)$$

being

$$\text{sinc}(x) = \frac{\sin(\pi x)}{x}. \quad (15)$$

Eq. (14) performs two operations. In the first place, it shifts the trigger from the frame detection instant ($t'_{c,2,i}$), which depends on the CIR and on the frame detector, to the frame transmission instant ($t_{1,i}$), and, in the second place, it eliminates the CFO to improve the estimation of the phase of the CIR. Hence, the channel samples of this channel sounder are equivalent to the operation of a full channel sounder, upon the condition of enough synchronization performance.

Finally, if a sync frame in a specific frame exchange (i_0) was not correctly received, the CIR from that sample is lost. Still, $\tilde{p}_{Sync\ aligned,i_0}[k]$ may be linearly interpolated from adjacent $\tilde{p}_{Sync\ aligned,i}[k]$, such as follows

$$\tilde{p}_{Sync\ aligned,i_0}[k] = \frac{\tilde{p}_{Sync\ aligned,i_0-1}[k] + \tilde{p}_{Sync\ aligned,i_0+1}[k]}{2} \quad (16)$$

V. IMPLEMENTATION IN AN SDR PLATFORM

The channel sounder comprises three main elements: the master, which periodically transmits the modified sync frames, the slave, which receives the modified sync frames from the master and answers with ACKs, and a computer, which receives the frame exchange information from the slave. In a channel measurement campaign, the information can be stored in the non-volatile memory of the slave.

The channel sounder could be implemented using standard COTS 802.11 circuits because it uses the standard 802.11 PHY. Unfortunately, COTS circuits do not usually provide access to HW timestamps neither to the estimation of the CIR, and thus a custom solution is required.

A fast prototyping option is the use of software-based SDR platforms. However, software-based SDR does not offer low-latency operation, which is required in the time synchronization algorithm. Hence, it is necessary to implement the channel sounder in a HW-based platform.

The platform used to implement the system is the ADI RF SOM platform (previously named Picozed SDR). The ADI RF SOM comprises a Zynq System on Chip (SoC) and an AD9361 radio chip. The Zynq includes a Field Programmable Gate Array (FPGA) and two ARM cortex A9 microcontrollers. The AD9361 chip offers a BW of 56 MHz and supports a carrier frequency from 100 MHz to 6 GHz.

The implementation of the master and slave is identical, and its behavior is configured through software. The block diagram of one node is shown in Fig. 5. The HW comprises the PHY layer of the 802.11 modem, the triggering and timestamping timer, and the interfaces with the AD9361 radio and with the software. The software performs the initial HW configuration, collects the data from the HW (CIRs, timestamps, etc.), and sends the data to the computer.

Apart from the ADI RF SOM, we have used some external extra HW. First, the internal oscillator of the ADI RF SOM has a frequency of 40 MHz and a frequency stability of 10 ppm, which is not stable enough to measure the phase of the CIR. We have replaced this oscillator

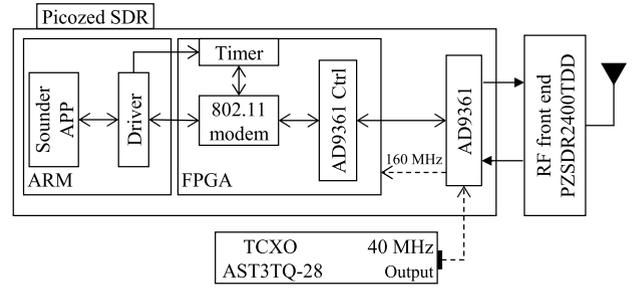


FIGURE 5. Block diagram of a node of the channel sounder.

with the mid-end AST3TQ-28 Temperature Compensated Crystal Oscillator (TCXO) from Abracon. This oscillator has a frequency stability of 0.28 ppm and a jitter of 0.4 ps, significantly better than the internal oscillator. We have also added the AD-PZSDR2400TDD-EB RF front end. The RF front end is designed to operate in the 2.4 GHz band and contains a high-gain power amplifier, a low-noise amplifier, Tx and Rx filters centered in the 2.4 GHz band, and an RF switch. With this configuration, the output power of the channel sounder is 10 dBm.

A. HW IMPLEMENTATION IN FPGA

We have used System Generator to implement the 802.11 modem. System Generator is a library for Simulink provided by Xilinx. The library contains several HW models, from simple blocks (registers, logic doors), to more complex ones, such as an FFT or a Viterbi decoder. The blocks can be combined to create complex HW, such as an 802.11 modem. The Simulink environment can be used to verify the model behavior before programming the FPGA.

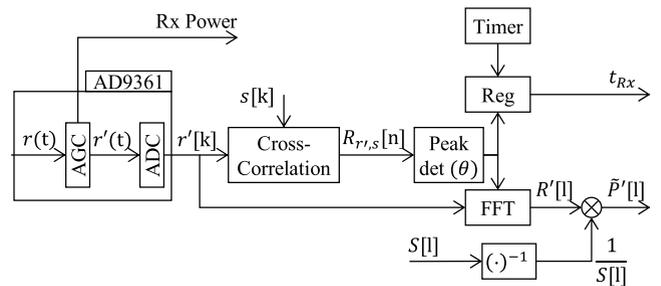


FIGURE 6. Frame detector and channel estimation.

The HW implementation of an 802.11 modem comprises several blocks with different functionalities that perform, among others: channel (de)coding, (de)scrambling, (de)interleaving, etc. However, only the frame detector and CIR estimation influence the quality of the synchronization and the channel sounding. Given that for this work the coding/decoding is not relevant, we just describe the initial steps of the Rx chain (Fig. 6). First, the AD9361 down-converts the RF signal and generates $r(t)$. $r(t)$ is the complex baseband representation of the RF signal. $r(t)$ goes through the AGC, which outputs its normalized version $r'(t)$.

Then, $r'(t)$ is sampled at a sampling period of T using an Analog to Digital Converter (ADC). The output of the ADC is a normalized version of $r[k]$ with unit energy $r'[k]$. $r'[k]$ is continuously cross-correlated with the $s[k]$ using a Finite Impulse Response (FIR) filter. When a peak exceeds θ it means (with high probability) that a frame has been detected. In this case, the peak detector generates a trigger that captures the timer value and generates the timestamp t_{rx} . The same trigger is used to start the FFT operation, which generates $R'[l]$. Afterward, the normalized estimated CIR in the frequency domain, $\tilde{P}'[l]$, is computed by dividing each of the elements of $R'[l]$ with $s[k]$ in the frequency domain ($S[l]$). Finally, t_{rx} and $\tilde{P}'[l]$ are transmitted to the registers and an interruption is triggered. The estimation of the received power used to estimate the channel gain \tilde{g} is obtained from the AD9361.

B. SW IMPLEMENTATION

The control software is implemented over a Linux operating system with kernel version 4.14. The Linux sources are provided by Analog Devices for this specific platform. The software comprises three elements: the master driver, the slave driver, and the slave application. The master driver operation performs the master initialization and manages the data from the ACK. The slave driver collects the data of the frame exchange and transmits the data to the slave application. Finally, the slave application receives the data from the slave driver and transmits it through Ethernet.

The master driver initializes the HW to send a sync frame every T_{CS} . The frame transmission is commanded in HW and the frames are deterministically sent without the intervention of the driver. After one frame is transmitted, the 802.11 modem generates an interruption. In the interruption, the driver erases the buffer data of the previous frame and writes the timestamp $t_{1,i}$ and the *Frame seq ID* for the next sync frame. If an ACK frame is received, another interruption is generated. Then, the driver extracts from the HW $\tilde{P}'_{ACK,i}[l]$, the timestamp ($t_{4,i}$), and computes $\tilde{g}_{ACK,i}$. Finally, it writes the information to the transmission buffer of the next Sync frame.

The slave driver is in idle state until an interruption is received. The interruption is generated every time that a frame is correctly received. The ACK is automatically generated by the HW without any intervention of the driver. The slave driver basically collects the information generated during the frame exchange: the *Frame seq ID* (i), the timestamps ($t_{1,i}, t_{2,i}, t_{3,i}, t_{4,i}$), $\tilde{P}'_{ACK,i}[n]$, and $\tilde{P}'_{Sync,i}[n]$. In addition, and based on the frames that have been successfully delivered, the slave calculates the $exchange_{ok,i}$ and $SyncFrame_{ok,i}$ Booleans. With this information, it creates a data structure that is sent to the application.

Finally, the slave application waits in idle state until the driver sends a data structure with the data of the frame exchange. Then, it generates an Ethernet frame, containing the data structure, that is transmitted to the computer. In a real

channel measurement campaign, the application may also store the data in the SD card of the Picozed.

VI. CHANNEL SOUNDER VERIFICATION THROUGH A CHANNEL EMULATOR

The channel sounder implementation has been in-lab verified using a wireless channel emulator. The channel emulator operates in the 2.4 GHz band and has a bandwidth of 100 MHz. It has four bidirectional ports with a delay from port to port of 910 ns, and a fixed attenuation of 44 dB. The emulator can hold a wireless channel model up to 10 taps that can be programmed with different gains and fading profiles. It supports four fading distributions: Rayleigh, Rice, Nakagami and constant; and three Doppler spectrum shapes: Jakes, Bell, and Gaussian.

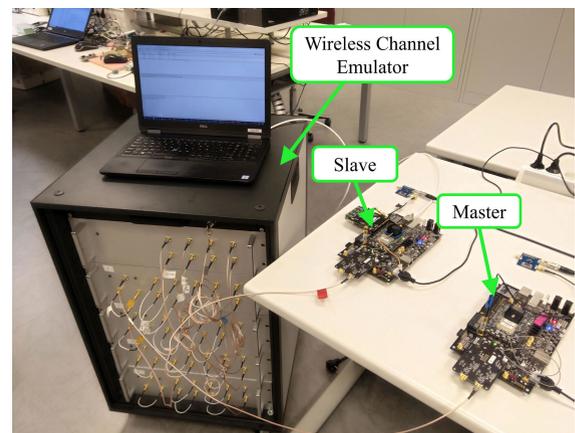


FIGURE 7. Channel sounder verification testbed.

The verification testbed is shown in Fig 7. The master RF port is connected to the first port of the channel emulator, and the slave RF port is connected to the second port. The Ethernet port of the slave board is connected to a laptop, which is used to record the data from the frame exchanges. f_c has been set to 2.48 GHz, T_{CS} to 500 μ s, and the Tx power to 10 dBm.

TABLE 2. Emulated wireless channel.

	Tap 1	Tap 2	Tap 3
Gain [dB]	-44	-54	-47
Delay [ns]	910	910 + 195.3	910 + 390.6
Fading	Rayleigh	Rayleigh	Rayleigh
Doppler spectrum	Jakes with a maximum Doppler shift of 62 Hz.		

The channel emulator is programmed with the wireless channel model described in Table 2. The wireless channel model emulates an NLoS scenario with mobile nodes. It comprises three taps with different gains and evenly spaced 195.3 ns. The fading has been set to Rayleigh (NLoS), and the Doppler spectrum has been set to Jakes with a Doppler frequency of 62 Hz. The Doppler configuration represents the movement of the nodes at a maximum relative

speed of 7.5 m/s given that $f_c = 2.48$ GHz. The channel model includes the fixed attenuation of the channel emulator (44 dB), and its delay (910 ns).

We took a record of 20 seconds, equivalent to $I = 40.000$ channel samples. The record length is enough to measure the tap position, gain, and fading properties. During the record, the wireless channel randomly varied according to the programmed configuration.

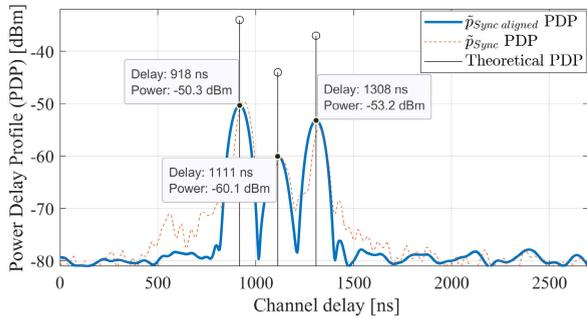


FIGURE 8. PDPs obtained from \tilde{p}_{Sync} aligned and from \tilde{p}_{Sync} .

The measurements were processed as follows. First, we applied the time synchronization algorithm, and we obtained \tilde{p}_{Sync} aligned from \tilde{p}_{Sync} . The PDPs obtained from the \tilde{p}_{Sync} aligned and from \tilde{p}_{Sync} along with the theoretical PDP are depicted in Fig. 8. The PDPs have been filtered with a Hanning window [25] to reduce their secondary lobes. \tilde{p}_{Sync} PDP has been manually aligned to \tilde{p}_{Sync} aligned PDP. The precision resulted from the time synchronization algorithm was better than 0.5 ns, which ensures the correct alignment of the CIRs.

There are significant differences between the PDPs. Regarding the \tilde{p}_{Sync} PDP, the position of the taps randomly varies around their real position in \tilde{p}_{Sync} . Due to this, the PDP has a significant tail around the taps and a considerable error in the relative gain between taps (2 dB). On the contrary, the \tilde{p}_{Sync} aligned PDP is smooth without tails, the relative gain between taps have an error below 0.5 dB, and their relative delay is very accurate. In addition, the difference between the peak and the noise floor is around 30 dB, which gives a reasonable PDP dynamic range.

From the \tilde{p}_{Sync} PDP representation, it is clear that the wireless channel presents 3 taps, which are situated at 914 ns, 1111 ns, and 1308 ns. The gain and phase of the taps randomly vary over time following the model programmed in the channel emulator (jakes with 62 Hz and Rayleigh fading). To verify that the channel sounder was able to measure the channel variations, we have obtained the Doppler spectrum and the fading distributions of the taps. Fig. 9 plots the Doppler spectrum of each tap. As can be seen, the Doppler spectrum of the three taps clearly follows a Jakes Doppler distribution with an absolute maximum Doppler shift of 65.7 Hz. The Jakes Doppler spectrum is not centered due to the lack of perfect estimation of the local oscillator drift. Still, the error is quite low. If the remaining frequency

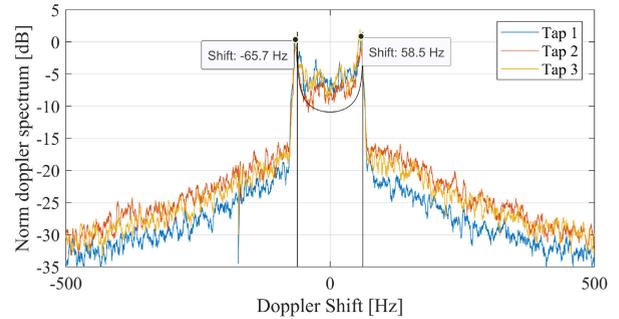


FIGURE 9. Measured Doppler spectrum of Taps 1, 2, and 3.

offset is manually compensated, the maximum Doppler shift is 62.1 Hz, very close to the Doppler shift programmed in the channel emulator. Fig. 10 represents the histogram of the fading distribution of each tap and the corresponding theoretical Rayleigh distribution. It is noticeable from the plot that the power of each tap approximately fits a Rayleigh distribution at their respective power. These results confirm the feasibility of the channel sounder to correctly measure the CIR in mobile conditions.

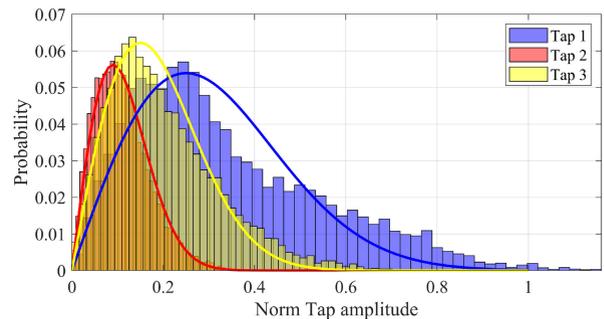


FIGURE 10. Measured fading distributions of Taps 1, 2, and 3 compared to their theoretical counterparts.

VII. MEASUREMENT CAMPAIGN IN A WORKSHOP

For the sake of illustration of the features of the channel sounder, we have performed a measurement campaign in a factory environment at Ikerlan. Due to the temporal unavailability of mobile robots in our installations, and since the channel sounder has been already validated in mobility conditions through the channel emulator, we have performed the measurements over static conditions.

The measurement campaign was carried out in a workshop at Ikerlan (Fig. 11). The nodes were placed at a distance around 13 m, and with several elements blocking the LoS. The channel sounder was programmed with $f_c = 2.58$ GHz, $T_{CS} = 500 \mu s$, and P_{Tx} was set to 10 dBm. f_c has been set outside the 2.4 GHz band to avoid interferences from nearby Wi-Fi networks. The workshop operators were working during the measurements, but there was no movement of large machines.

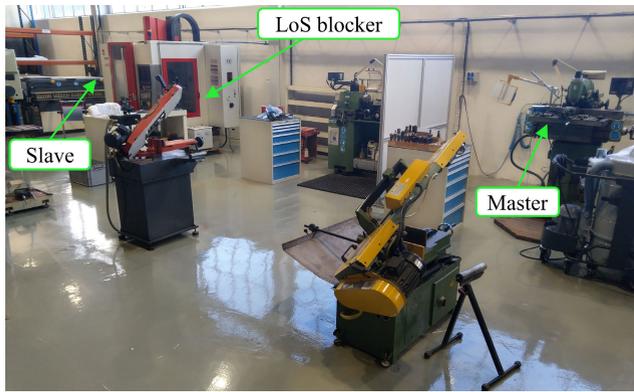


FIGURE 11. Measurement setup in the workshop at Ikerlan.

This experiment represents a quite static scenario since the nodes were placed in a fixed position and only the operators of the workshop were moving around the machines. Consequently, a wireless channel with low variations is expected. The experiment was run for a total of 5 mins and a total of $I = 600.000$ CIRs were taken. The record length is enough to measure the tap position, gain, and fading properties and capture the small variations of the wireless channel.

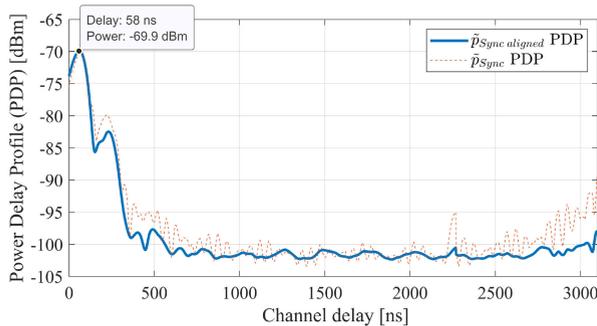


FIGURE 12. Measured PDP at the workshop.

As in the experiments with the channel emulator, the raw data obtained from the experiment was post-processed by the offline time synchronization algorithm. The PDPs obtained from the \tilde{p}_{Sync} aligned and from \tilde{p}_{Sync} are depicted in Fig. 12. Based on the PDP results, it can be concluded that the measured wireless channel presents two taps, one tap situated at a delay of 58 ns and with a peak power of -70 dBm (main tap), and a second tap situated at 233 ns with a power of -82.5 dBm. The main tap probably corresponds with the reflection of the ceil of the workshop, whereas the second tap could be the combination of reflections from several directions. The resulted delay of tap 1 is reasonable considering the radio properties of the environment. The distance between nodes is 13 m, which gives a delay of around 43 ns. However, the direct line of sight is blocked and, therefore, the wireless channel presents a slightly larger delay.

The RMS delay spread of the PDP is 72 ns, which is a quite low spreadness for industrial scenarios, which are characterized by a significant multipath. The reason of the

low spreadness is that the main tap, which is caused by the ceiling, has a significant power compared to the next tap.

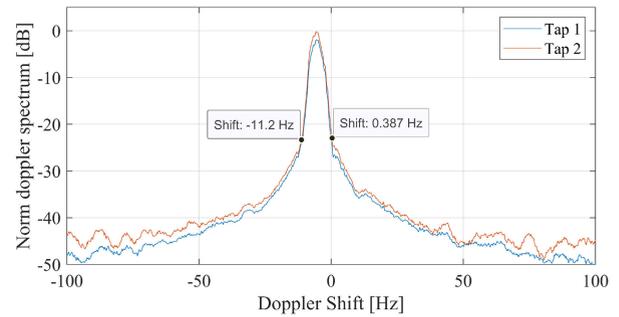


FIGURE 13. Measured Doppler spectrum at the workshop.

The Doppler spectrum distributions for both taps are depicted in Fig. 13. Note that Tap 1 is the tap at 58 ns and Tap 2 is the tap at 233 ns. The Doppler spectrum distributions present a very narrow shape, with a 6 Hz of maximum deviation from the peak with a small deviation from the center of 5 Hz due to a small imprecision of the synchronization algorithm. The Doppler spectrum shape clearly represents a channel with very low variability. If compared to the Doppler spectrum found in the experiments of Section VI, the differences are notorious. The estimated Doppler spectrum of the channel programmed in the channel emulator, which is characterized by high mobility, clearly follows a Jakes shape with a considerable maximum deviation, whereas the one of the workshop experiment has a very narrow bell shape because of the small channel variation.

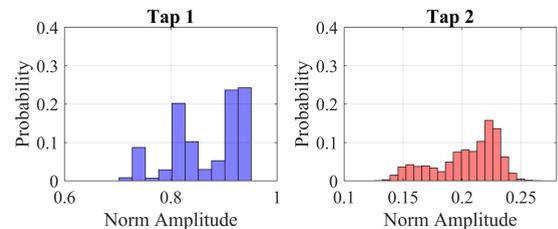


FIGURE 14. Fading distributions for tap 1 and 2 of the wireless channel measured at the workshop.

Finally, the fading distributions for both taps are represented in Fig. 14. The fading distributions plotted in Fig. 14 show a slight variation of the power over time, as expected given that the Doppler spectrum has a very narrow shape. Tap 1 has a maximum power deviation of 3 dB for the whole measurement, which is a reasonable result because it is the strongest tap and it is caused by a large reflection in the ceiling. In addition, tap 2 presents a larger variation of around 5 dB of maximum deviation because the tap comprises several signal reflections.

VIII. CONCLUSION

Channel measurements are a critical step in the design and implementation of custom wireless solutions for industrial applications, especially in mobile robotics, where the channel

variability may greatly influence the wireless solution performance.

In this paper, we exploit a sub-nanosecond wireless time synchronization scheme to create a compact portable full channel sounder that does not require any kind of wired or external synchronization (e.g. GPS). We have carried out the implementation of the channel sounder over a HW-based SDR platform, where we have built an 802.11 modem to obtain the complex baseband CIR samples and their corresponding timestamps from a modified PTP frame exchange. In addition, we have designed an offline post-processing algorithm to calculate the time synchronization error. With the time synchronization error, the CIR samples can be aligned to absolute time and the carrier frequency offset can be compensated.

We have verified the channel sounder using a wireless channel emulator under NLoS and time-variant conditions. The experiment showed that the channel sounder was able to measure the wireless channel delay, the number of taps of the channel, and their gain and relative delay. In addition, we computed the Doppler spectrum and fading distributions of each of the taps and we verified that the measured Doppler spectrum and fading distributions corresponded with the theoretical curves programmed in the channel emulator. The experiment confirms the feasibility of high-performance wireless time synchronization applied to channel sounding. Finally, and for the sake of illustration, we performed a measurement campaign in an industrial scenario and over static conditions.

In summary, we think that the proposed channel sounder solves significant challenges found in conventional channel sounders for mobile applications and that it opens up new research and implementation possibilities in this field. Still, the channel sounder has some limitations, such as small BW, or that it only uses one antenna. Therefore, the migration of the channel sounder to a more powerful HW platform could be an interesting research line to further validate the use of wireless synchronization to build channel sounders. In addition, the channel sounder design could be moved from 802.11 to a dedicated wireless solution to maximize the channel sounding capabilities, including channel sounding period, conversion gain, No. of antennas, etc. Finally, high-precision and real-time channel measurements may help to develop better adaptive algorithms for wireless systems (e.g. 5G), whose performance strongly relies on the quality of the wireless channel and the modulation scheme used in the communication. Therefore, the integration of the channel sounder in a real communication system could be an interesting future line of work.

REFERENCES

- [1] I. Val, F. Casado, P. M. Rodriguez, and A. Arriola, "FPGA-based wideband channel emulator for evaluation of wireless sensor networks in industrial environments," in *Proc. IEEE Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2014, pp. 1–7.
- [2] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, 4th Quart., 2018.
- [3] Y. Mostofi, M. Malmirchegini, and A. Ghaffarkhah, "Estimation of communication signal strength in robotic networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1946–1951.
- [4] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard 802.11-2016, 2012.
- [5] O. Seijo, J. A. Lopez-Fernandez, H.-P. Bernhard, and I. Val, "Enhanced timestamping method for subnanosecond time synchronization in IEEE 802.11 over WLAN standard conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5792–5805, Sep. 2020.
- [6] O. Seijo, I. Val, and J. A. Lopez-Fernandez, "Portable full channel sounder for mobile robotics by using sub-nanosecond time synchronization over wireless," in *Proc. 16th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Apr. 2020, pp. 1–4.
- [7] D. Maas, M. H. Firooz, J. Zhang, N. Patwari, and S. K. Kasera, "Channel sounding for the masses: Low complexity GNU 802.11b channel impulse response estimation," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 1–8, Jan. 2012.
- [8] G. R. MacCartney and T. S. Rappaport, "A flexible millimeter-wave channel sounder with absolute timing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1402–1418, Jun. 2017.
- [9] T. Liu, Z. Zhang, H. Jiang, Y. Qian, K. Liu, J. Dang, and L. Wu, "Measurement-based characterization and modeling for low-altitude UAV air-to-air channels," *IEEE Access*, vol. 7, pp. 98832–98840, 2019.
- [10] D. Hague, H. Kung, and B. Suter, "Field experimentation of cots-based UAV networking," in *Proc. MILCOM*, Oct. 2006, pp. 1–7.
- [11] P. Pajusco, N. Malhouroux-Gaffet, and G. El Zein, "Comprehensive characterization of the double directional UWB residential indoor channel," *IEEE Trans. Antennas Propag.*, vol. 63, no. 3, pp. 1129–1139, Mar. 2015.
- [12] B. T. Maharaj, J. W. Wallace, M. A. Jensen, and L. P. Linde, "A low-cost open-hardware wideband Multiple-Input–Multiple-Output (MIMO) wireless channel sounder," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 10, pp. 2283–2289, Oct. 2008.
- [13] O. Seijo, Z. Fernandez, I. Val, and J. A. Lopez-Fernandez, "SHARP: A novel hybrid architecture for industrial wireless sensor and actuator networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–10.
- [14] T. B. Welch, C. H. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2007.
- [15] E. Kassem, R. Marsalek, and J. Blumenstein, "Frequency domain zadoff-chu sounding technique for USRPs," in *Proc. 25th Int. Conf. Telecommun. (ICT)*, Jun. 2018, pp. 302–306.
- [16] J. Vychodil, A. Chandra, T. Mikulasek, A. Prokes, and V. Derbek, "UWB time domain channel sounder," in *Proc. 25th Int. Conf. Radioelektronika (RADIOELEKTRONIKA)*, Apr. 2015, pp. 268–271.
- [17] F. Talebi and T. Pratt, "Channel sounding and parameter estimation for a wideband correlation-based MIMO model," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 499–508, Feb. 2016.
- [18] B. Zhang, Z. Zhong, R. He, B. Ai, G. Dahman, M. Yang, and J. Li, "Multi-user channels with large-scale antenna arrays in a subway environment: Characterization and modeling," *IEEE Access*, vol. 5, pp. 23613–23625, 2017.
- [19] J. Moreno Garcia-Loygorri, I. Val, A. Arriola, and C. Briso, "Channel model and interference evaluation for a wireless train backbone," *IEEE Access*, vol. 7, pp. 115518–115527, 2019.
- [20] J. Meiniälä, P. Kyösti, T. Jämsä, and L. Hentilä, (Feb. 2008). *WINNER II Channel Models*. [Online]. Available: https://www.researchgate.net/publication/234055761_WINNER_II_channel_models
- [21] R. J. Weiler, M. Peter, T. Kühne, M. Wisotzki, and W. Keusgen, "Simultaneous millimeter-wave multi-band channel sounding in an urban access scenario," in *Proc. 9th Eur. Conf. Antennas Propag., EuCAP*, Apr. 2015, pp. 1–5.
- [22] C. U. Bas, R. Wang, S. Sangodoyin, D. Psychoudakis, T. Henige, R. Monroe, J. Park, C. J. Zhang, and A. F. Molisch, "Real-time millimeter-wave MIMO channel sounder for dynamic directional measurements," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8775–8789, Sep. 2019.
- [23] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. Standard 1588, 2009.

- [24] O. Seijo, I. Val, J. A. Lopez-Fernandez, and M. Velez, "IEEE 1588 clock synchronization performance over time-varying wireless channels," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun. (ISPCS)*, Sep. 2018, pp. 1–6.
- [25] F. P. Fontán and P. Mariño, *Espiñeira, Modeling the Wireless Propagation Channel: A Simulation Approach with MATLAB*. Hoboken, NJ, USA: Wiley, 2008.



ÓSCAR SEIJO received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Oviedo, Spain, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Communications Systems Group, IKERLAN, Mondragón, Spain, in collaboration with the Signal Theory and Communications (TSC) Group, University of Oviedo.

From September to December 2019, he was a Visiting Researcher with the Power Grids Group, ABB Corporate Research, Västerås, Sweden. His research interests include wireless high-performance PHY, MAC design for industrial applications, time synchronization over wireless systems, and digital signal processing.



IÑAKI VAL (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Electronics Engineering, Mondragon University, Spain, in 1998 and 2001, respectively, and the Ph.D. degree from the Department of Signals, Systems and Radio Communication, Polytechnic University of Madrid, Spain, in 2011. Since 2001, he has been with the Communications Systems Group, IKERLAN, Mondragón, Spain. He was with Fraunhofer IIS, Erlangen, Germany, as an

Invited Researcher, from 2005 to 2006. He is currently a Team Leader with the Communication Systems Group. His research interests include design and implementation of digital wireless communications systems for industrial communications and digital signal processing.



JESÚS ALBERTO LÓPEZ-FERNÁNDEZ was born in Avilés, Asturias, Spain. He received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Vigo, Spain, in 1999 and 2009, respectively. From April 2002 to March 2003, he was a Marie-Curie Visiting Fellow with the Mechanical and Manufacturing Engineering Department, Trinity College Dublin (TCD). Since October 2003, he has been with the Electrical Engineering Department, University of

Oviedo, Asturias, Spain, where he is currently an Associate Professor in digital communications and radar systems. His research interests include iterative methods and speed-up schemes, parallel algorithms, machine learning, and signal processing.

• • •