TS-MUWSN: Time Synchronization for Mobile Underwater Sensor Networks

Oriol Pallares, *Student Member, IEEE*, Pierre-Jean Bouvet, *Member, IEEE*, and Joaquin del Rio, *Member, IEEE*

***Abstract*—Time synchronization is an important, yet challenging, problem in underwater sensor networks (UWSNs). This challenge can be attributed to: 1) messaging timestamping; 2) node mobility; and 3) Doppler scale effect. To mitigate these problems, we present an acoustic-based time-synchronization algorithm for UWSN, where we compare several message time-stamping algorithms in addition to different Doppler scale estimators. A synchronization system is based on a bidirectional message exchange between a reference node and a slave one, which has to be synchronized. Therefore, we take as reference the DA-Sync-like protocol (Liu *et al.*, 2014), which takes into account node's movement by using first-order kinematic equations, which refine Doppler scale factor estimation accuracy, and result in better synchronization performance. In our study, we propose to modify both time-stamping and Doppler scale estimation procedures. Besides simulation, we also perform real tests in controlled underwater communication in a water test tank and a shallow-water test in the Mediterranean Sea.**

***Index Terms*—Acoustic communication, Doppler scale, orthogonal frequency-division multiplexing (OFDM), skew, timestamp, time synchronization, underwater sensor network (UWSN).**

I. Introduction

UNDERWATER wireless sensor networks (UWSNs) have recently become a common research field in both industry and academia. This is due to the necessity to perform distributed and collaborative sensing tasks all over our oceans. In fact, the ease of deployment and maintenance leads to wireless networking instead of cabled connections between sensors.

Underwater wireless communications can be performed by an optical link, electromagnetic waves, or acoustics. Optical communication requires perfect alignment between nodes and is sensitive to water turbidity, whereas electromagnetic waves suffer from large attenuation and are dedicated to low range applications. Acoustic signals are well adapted to the underwater medium but experience very challenging impairments such as Doppler, extensive multipaths, and low transmission speed that can nevertheless be corrected at the reception side; this makes acoustic communication very attractive and widely used in an underwater scenario [2], [3]. Several acoustic waveforms can be invoked to transmit digital data through the underwater medium, without loss of generality. In our study, we consider the orthogonal frequency-division multiplexing (OFDM) communication scheme to exchange data between wireless underwater nodes [4]. This communication link will be used among others to carry timestamp message required for network synchronization.

Time synchronization is a critical piece of infrastructure for any distributed system. UWSNs make extensive use of the synchronized time for many services provided by a distributed network [3]. In UWSNs, global positioning system (GPS) signals are not available and synchronization systems are mostly based on acoustic communication. Owing to high latency of the underwater acoustic transmission channel with respect to a cabled or radio network makes the use of conventional synchronization protocols [8] even more challenging underwater.

Many time-synchronization algorithms for UWSNs can be found in literature [1], [9]–[13], but only few of them take into account water channel challenges, such as low available bandwidth, long propagation delays, and sensor node mobility [1], [13].

In this work, to perform time synchronization over an UWSN, we apply the precision time protocol (PTP) Standard IEEE 1588, which is capable of synchronizing two clocks with a precision below hundreds of nanoseconds in a point-to-point cabled Ethernet network [14]. The synchronization procedure is based on a bidirectional message exchange between a master clock and a slave, which is the one to synchronize. In cabled synchronization systems, such as PTP, timestamps are acquired in the physical layer (PHY) to achieve maximum precision, avoiding indeterministic times such as operating system (OS) time slots or medium-access protocols [15]. Analogously, in acoustic communication [16], timestamps are extracted from a large acquisition window with time-stamped raw data, where the main signal to be processed is found.

So far, cabled synchronization could be ported directly to an acoustic one, but we will need a system capable of triggering a clock acquisition at a precise time, just when an acoustic frame enters or exits the system, in order to reproduce the same behavior as the PTP timestamping. Contrary to cable networks, the low celerity of a sound wave makes an underwater acoustic communication system very sensitive to a Doppler effect, causing nonuniform frequency scaling represented by compression or dilatation of the time axis. This frequency scaling can be induced by two factors: motion (sensor mobility, channel variation, etc.) and the clock skew receiver between a transmitter and a receiver. Actually, to address this problem, some systems

Manuscript received October 15, 2015; revised March 17, 2016 and May 24, 2016; accepted June 9, 2016. Date of publication September 16, 2016; date of current version October 11, 2016. This work was supported in part by the NeXOS and FixO3 projects from the European Unions Seventh Programme for Research, Technological Development, and Demonstration under Grants 614102 and 312463, respectively.

Guest Editor: D. Toma.

O. Pallares and J. del Rio are with the SARTI Research Group, Electronics Department, Technical University of Catalunya (UPC), Vilanova i la Geltrú 08800, Spain (e-mail: oriol.pallares@upc.edu; joaquin.del.rio@upc.edu).

P-J. Bouvet is with the Underwater Acoustics Lab, ISEN Brest, Brest cedex 2, 29228, France (e-mail: pierre-jean.bouvet@isen-bretagne.fr).

Digital Object Identifier 10.1109/JOE.2016.2581658

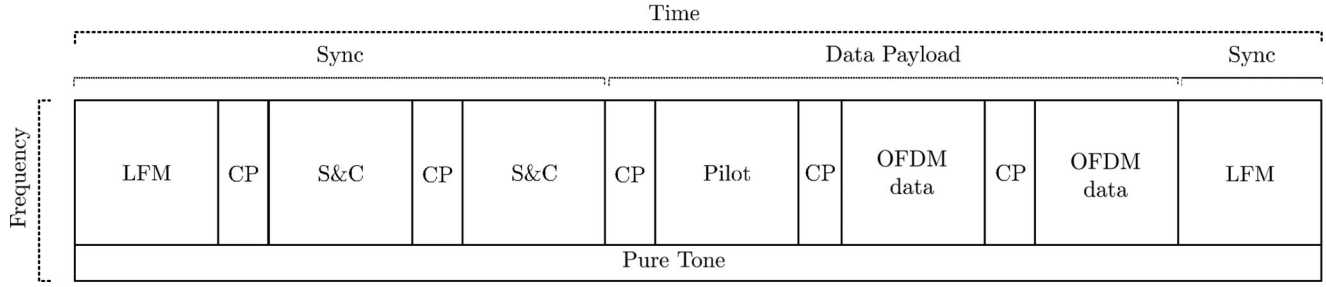

Fig. 1. Overall structure of the transmitted signal.

use expensive inertial sensors to compensate for Doppler scaling due to motion and temperature-compensated low drift clocks [17]. So it is necessary to evaluate which part is caused by motion and skew in the Doppler scaling in order to correct it.

The objective of this work is to design and experiment with a time-synchronization algorithm in real conditions, dedicated to synchronize precisely two underwater nodes by using decoder information such as Doppler scaling estimation and timestamping. We take as basis the DA-Sync time-synchronization protocol [11] designed for a mobile wireless sensor network. The first contribution of this paper consists of a study of the impact of Doppler scale estimation and timestamping on the DA-sync protocol. The second contribution of this paper lies in the experimentation of the protocol in the Mediterranean Sea and comparison with the simulation results. Finally, we propose an algorithm optimization to support a real scenario, more precisely, we show how to use frame timestamping to compute time offset between clocks and discuss how to include it in the DA-Sync protocol.

The rest of this paper is organized as follows. In Section II, we describe the OFDM acoustic modulation and the frame timestamp procedure. Then, in Section III, we detail and compare in simulation several different Doppler scale factor estimation algorithms used for both channel decoding and time synchronization. In Section IV, we detail step by step the algorithm used to synchronize a slave clock with a reference one. Finally, in Section V, we discuss our results and the maximum time accuracy obtained using the best algorithm for Doppler correction and frame timestamping. These results are obtained from real tests in a laboratory and the Mediterranean Sea, by using the expandable seafloor observatory OBSEA, placed 4 km offshore at Vilanova i la Geltrú, Spain [18].

II. Underwater Acoustic Link

A. Transmitted OFDM Frame

To implement a message exchange-based time-synchronization algorithm, it is necessary to send time information from a master to a slave and *vice versa*. This way, the slave node is capable of following the same time base as the master clock by running a synchronization algorithm. In our study, we consider acoustic communication, due to its good performance versus other underwater communication systems, as introduced above. We choose the OFDM waveform which provides good robustness against a multipath effect with a reduced complexity equalization process [5]. OFDM is a particular form of a multicarrier transmission and can be efficiently modulated by fast Fourier transform (FFT) processing. The conversion of OFDM to underwater acoustic communication has been extensively studied in the literature, namely the negative impact of the Doppler effect in an underwater scenario that has to be specifically addressed at the receiver by using the advanced Doppler compensation and tracking algorithm [4], [6], [7].

We use a convolutional cyclic prefixed OFDM (CP-OFDM) communication, where data symbols are transmitted in blocks, with the consecutive blocks separated by a cyclic prefix containing the last samples of the end of each symbol. To avoid interblock interference (IBI), CP must be no shorter than the channel impulse response (CIR) [19]. Without loss of generality, our study can be reproduced with a zero-padded OFDM (ZP-OFDM) waveform for power saving purpose [20].

The OFDM waveform is centered around f_0; the duration of each OFDM data symbol is denoted as T_{OFDM} and can be decomposed into a useful part of length $T_u = N_{\mathrm{FFT}} T_{\mathrm{sym}}$ and a CP part of duration $T_{\mathrm{CP}} = L_{\mathrm{CP}} T_{\mathrm{sym}}$ where T_{sym} denotes the duration of a complex cell. In each OFDM symbol, $K \leq N_{\mathrm{FFT}}$ active carriers are modulated by using a quadrature phase-shift keying (QPSK) constellation. The intercarrier spacing is computed as $\Delta f = (1/T_u)$ and signal bandwidth is $B_{\mathrm{OFDM}} = (K/T_u)$. The fraction of CP is set according to the maximum delay spread of the underwater acoustic channel [21]. Let us denote d_k with $k \in [1, K]$ the QPSK data cells; the time-domain-modulated OFDM signal can be expressed as

$$\tilde{x}(t) = \Re\left(\sum_{k=0}^{K-1} d_k \exp(j2\pi f_k t)\right), \qquad t \in [0, T_{\mathrm{OFDM}}] \quad (1)$$

with $f_k = f_0 + (2k - K)/(2T_u)$.

The overall structure of the frame is depicted in Fig. 1. With a duration of 2.7 s, the frame starts and ends with an identical linear-frequency-modulated (LFM) signal used for frame arrival detection and Doppler shift estimation. Given the LFM bandwidth B_{LFM} and duration T_{LFM}, the LFM signal is modeled as

$$\tilde{x}_{\mathrm{LFM}}(t) = \sin\left(2\pi f_0 t + \frac{B_{\mathrm{LFM}}}{2T_{\mathrm{LFM}}} t^2\right). \quad (2)$$

The frame also comprises two OFDM preambles used for the Schmidl and Cox (S&C) synchronization algorithm (see

Section III-B). Finally, the frame includes one pilot OFDM symbol used for channel estimation, followed by two OFDM data symbols carrying useful message information. Outside the useful spectrum, we transmit a continuous pure tone signal centered at f_{pt} used for frequency shifting detection

$$\tilde{x}_{pt}(t) = \cos(2\pi f_{pt} t). \tag{3}$$

The signals are input/output (I/O) with sampling frequency $f_s = 1/T_s$ through a data acquisition system connected to a computer running Matlab.

B. Impact of the Underwater Acoustic Channel

In an underwater acoustic (UWA) channel, the physics of the acoustic propagation gives rise to a wide range of signal distortions. Thus, it is necessary to handle it with time offsets, CIR, noise, and Doppler shifts for the recover synchronization data enclosed in the frame, in addition to the exact signal arrival and departure time from each node in order to provide it to the synchronization algorithm.

In the underwater medium, acoustic waves experience several reflections represented by P paths for which we associate a time-varying attenuation $A_p(t)$ and a time-varying delay $\tau_p(t)$ [22]. At the receiver side, the received signal $\tilde{y}(t)$ can be viewed as the sum of P attenuated and a delayed version of $\tilde{x}(t)$ plus medium noise $\tilde{w}(t)$ such as

$$\tilde{y}(t) = \sum_{p=0}^{P-1} A_p(t)\tilde{x}(t - \tau_p(t)) + \tilde{w}(t). \tag{4}$$

Let us define $x(t)$ as the complex baseband version of the transmitted signal $\tilde{x}(t)$ such as

$$\tilde{x}(t) = \Re\left[x(t)e^{j2\pi f_0 t}\right]. \tag{5}$$

The received baseband signal can be expressed as

$$y(t) = \sum_{p=0}^{P-1} A_p(t)e^{-j2\pi f_0 \tau_p(t)} x(t - \tau_p(t)) + w(t) \tag{6}$$

where $w(t)$ is the baseband version of $w(t)$. As shown in (6), varying delay $\tau_p(t)$ brings individual frequency shifting for each path leading to a Doppler spread effect for the received signal. In case of motion between the transmitter and the receiver, each $\tau_p(t)$ contains a time variation $a_m t$ identical for each path such as [23]

$$\tau_p(t) = \bar{\tau}_p + \delta\tau_p + a_m t \tag{7}$$

where $\bar{\tau}_p$ is the static delay of path p and $\delta\tau_p$ is the residual time-varying delay coming from small-scale fluctuation of a channel which is treated as being random. Finally, we have the Doppler scale factor defined as $a_m = -(v_r)/(c_w)$ where v_r is the relative motion between the transmitter and the receiver and c_w is wave celerity in water. By writing $\tau_p = \bar{\tau}_p + \delta\tau_p$, we can introduce the complex time-varying channel attenuation $h_p(t)$ as

$$h_p(t) = A_p(t)e^{-j2\pi f_0 \tau_p}. \tag{8}$$

The significant variation of $h_p(t)$ comes from the phases $2\pi f_0 \tau_p$ taken modulo 2π that can vary substantially in time and independently from one path to another, producing a Doppler spread in the received signal. With these notations, (6) can be rewritten as

$$y(t) = \sum_{p=0}^{P-1} h_p(t)e^{-j2\pi f_{d,m} t} x((1 + a_m)t - \tau_p) + w(t) \tag{9}$$

where $f_{d,m} = a_m f_0$. As shown in the previous expression, motion provides a Doppler frequency shifting $f_{d,m}$ (identical for each path) and a time dilatation (or compression) by a factor $1 + a_m$ in the received baseband signal. The Doppler scale can be compensated for before OFDM decoding by resampling the received baseband signal by a factor of $1/(1 + a_m)$ and by compensating for the phase rotation by a factor $\exp(j2\pi f_{d,m} t)$ as detailed in [6].

C. Impact of Clock Skew

Let us consider two nodes A and B. In practice, each node has its own clock that can differ from the other one. Let us assume that node A has the master clock and node B has the slave clock which has a drift of θ with respect to a master clock. The time basis in B can be written as $t[k] = k\theta/f_s$. The baseband received signal in node B from node A denoted as $\tilde{y}_{AB}(t)$ is expressed as

$$y_{AB}(t) = \sum_{p=0}^{P-1} h_p(t)e^{-j2\pi f_{d,AB} t} x((1 + a_{AB})t - \tau_p) + w(t) \tag{10}$$

where a_{AB} (respectively, a_{BA}) is the combined Doppler scale factor going from node A to node B (respectively, from node B to node A) defined as [13]

$$1 + a_{AB} = \theta(1 + a_m) \tag{11}$$

$$1 + a_{BA} = \frac{(1 + a_m)}{\theta} \tag{12}$$

and $f_{d,AB}$ (respectively, $f_{d,BA}$) is the Doppler shift in the received baseband signal $y_{AB}(t)$ (respectively, in $y_{BA}(t)$) defined as

$$f_{d,AB} = \theta f_{d,m} \tag{13}$$

$$f_{d,BA} = \frac{f_{d,m}}{\theta}. \tag{14}$$

As a result, by assuming that relative motion v_r is constant in a message exchange, the clock skew can be estimated from the estimates of the Doppler scale factor in both nodes.

III. Doppler Scale Factor Estimation

Since we are working with frequency-modulated communication, Doppler scaling will affect both communication metrics and time-stamping accuracy. In time synchronization, these two factors are what determines its performance. Communication metrics are deteriorated by the fact that a frequency shift produces a phase shift in the constellation leading to bit error rate

(BER) degradation and resulting in the loss of the timestamp information enclosed in the data message. There is no way to get a high synchronization protocol. On the other hand, the Doppler scale results in a dilatation/compression of the time basis which directly impacts the time-stamping algorithm and synchronization protocol accuracy. In the following, we consider three algorithms for the Doppler scale estimation exhibiting affordable complexity for real-time implementation. A more sophisticated algorithm based on the exhaustive search approach can be found in [24].

A. Pure Tone Doppler Shift Estimation

This approach consists of sending a pure tone (PT) signal outside the useful spectrum and studying its phase variation at the reception side estimate the Doppler shift [25]. At the reception side, after baseband conversion and narrowband filtering around f_{pt}, the expression of the baseband received PT signal can be derived from (9) as follows:

$$y_{pt}(t) = e^{-j2\pi a_m f_{pt} t} \sum_{p=0}^{P-1} A_p(t) e^{-j2\pi f_{pt}\tau_p} + w(t). \quad (15)$$

Let us define $\psi(t) = \sum_{p=0}^{P-1} A_p(t) e^{-j2\pi f_{pt}\tau_p}$, then the argument of $y_{pt}[n] = y_{pt}(n/f_s)$ can be expressed as

$$\arg y_{pt}[n] = 2\pi f_d \frac{f_{pt}}{f_0} \cdot \frac{n}{f_s} + \arg \psi[n] + \arg w[n]. \quad (16)$$

Due to the properties of $\delta\tau_p$ which can be reasonably modeled as the zero mean Gaussian process independent for each path $\psi(t)$ can be treated as an uncorrelated random process that can be mitigated by lowpass filtering. An estimation of the motion-induced Doppler shift can be formed as [25]

$$\hat{f}_d = \frac{f_s}{2\pi} \frac{f_0}{f_{pt}} \text{LPF}\left[\arg(y_{pt}[n] y_{pt}^*[n-1])\right] \quad (17)$$

where LPF $[\cdot]$ denotes the lowpass filtering operation. This algorithm allows a communication system to be independent of the Doppler shift estimation procedure, because the PT signal does not interfere with the payload bandwidth. In this study, we use an OFDM communication band centered at 30 kHz with 1.3-kHz bandwidth, and a pure tone centered at 20 kHz for the Doppler scale study. While we transmit the instrument information and synchronization framing, we are capable of adding another tone without spreading the communication time and without worsening the OFDM robustness. When separating both OFDM and pure tone signals, it is to necessary to use a sharp passband filter to avoid interferences while not cutting the Doppler frequency shifting on the sides of the pure tone signal.

B. S&C CFO Detection

Originally introduced in [26], the S&C algorithm is used to perform both time synchronization and carrier frequency offset (CFO) detection. This algorithm is based on two OFDM preambles: the first one is used for frame detection in addition to a fine CFO, and the second one is used for a coarse CFO estimation. In our case, CFO detection is to detect the Doppler shift coming from motion and/or the clock skew.

First preamble is a symmetric OFDM symbol which will suffer a phase difference between the first half and the second half

$$\phi = \pi T_{\text{sym}} \Delta f \quad (18)$$

which can be estimated as the angle resulting from the partial correlation $P(d)$ between the two halves near the best timing point [26]

$$P(d) = \sum_{m=0}^{L-1} y_{d+m}^* y_{d+m+L} \quad (19)$$

where y_n denotes the received baseband signal sampled at $1/T_{\text{sym}}$ and d is a time index corresponding to the first sample in a window of $2L$ samples. Phase variations are estimated as

$$\hat{\phi} = \arg\left(P(d)\right). \quad (20)$$

An estimation of the actual frequency shifting is given by

$$\hat{f}_d = \frac{\hat{\phi}}{\pi T_{\text{sym}}} + \frac{2\hat{g}}{T_{\text{sym}}} \quad (21)$$

where g is an integer corresponding to the coarse CFO that will be estimated in the second phase.

The second training symbol contains a pseudonoise (PN) sequence on the odd (x_1) frequencies to measure these subchannels, and another PN sequence on the even frequencies (x_2) to help determine the frequency offset. After computing the relation between the odd frequencies and the ones at the transmitter side, we will obtain a conversion factor between the pairs of frequencies, and then at the receiver side, we can use this factor to estimate $\hat{x}_2$ from x_{m1} and *vice versa*. By computing the correlation between x_2 and $\hat{x}_2$, an estimation of the g factor is found by maximizing the following correlation metric:

$$B(g) = \frac{\left|\sum_{k\epsilon x} x_{1,k+2g}^* v_k^* x_{2,k+2g}\right|^2}{2(\sum_{k\epsilon x} |x_{2,k}|^2)^2}. \quad (22)$$

C. Preamble/Postamble Doppler Scale Estimation

As mentioned before, Doppler scaling performs similarly to interpolating a signal into a different time base. Then, if we compute an analysis of time variations between two known points in a frame, we could find a relation between this time variation and Doppler scaling. As originally introduced in [28], we add LFM preamble and postamble to detect time compression or dilatation of the frame and then to estimate the Doppler scale factor. The main reason for using an LFM instead of another signal is its good robustness against the Doppler effect as well as its cross-correlation performance in environments corrupted by white Gaussian noise.

An example of cross-correlation results with LFM is plotted in Fig. 2; the Doppler scaled frame duration (denoted as t_{DS}) is estimated by computing the time difference between the two correlation peaks given by the preamble and the postamble. Then, by knowing the original frame duration t_{ideal}, the Doppler

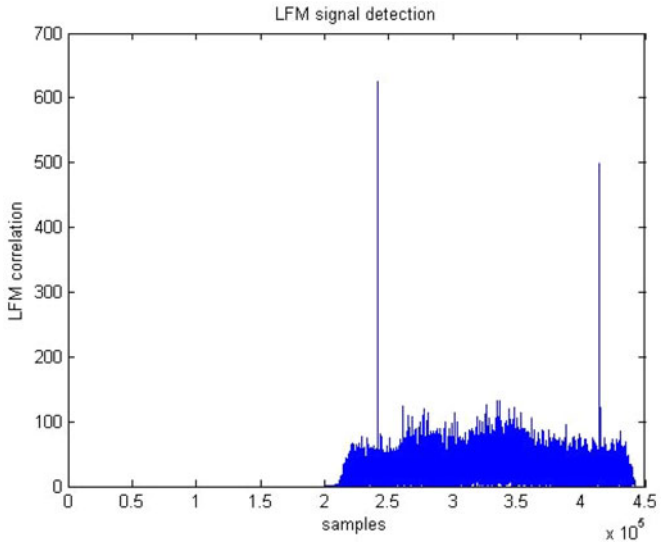

Fig. 2. Signal $|R_{xy}|$ for LFM detection.

TABLE I
COMMUNICATION PARAMETER SUMMARY (TYPICAL VALUES)

Description	Parameter	Simulation & Laboratory	OBSEA
Sampling frequency	f_s	100 kS/s	100 kS/s
OFDM frequency center	f_0	30 kHz	30 kHz
OFDM BW	B_{OFDM}	1.19 kHz	1.18 kHz
Pure tone frequency center	f_{pt}	40 kHz	40 kHz
LFM BW	B_{LFM}	5 kHz	5 kHz
OFDM symbol duration	T_{OFDM}	480 ms	168 ms
Useful part of OFDM signal	T_u	384 ms	96 ms
Cyclic prefix period	T_{CP}	96 ms	72 ms
FFT points	N_{FFT}	512	128
Active carriers	K	460	114
Signal to Noise Ratio	SNR	15 dB	15 dB
Doppler frequency	f_d	20 Hz	20 Hz

shift is estimated as

$$\hat{f}_d = f_0 \left(1 - \frac{t_{DS}}{t_{\text{ideal}}}\right). \tag{23}$$

Since the sampling frequency is a limiting factor, Doppler shift estimation accuracy can be improved significantly by using the center of gravity computation for the time difference between the correlation peaks.

D. Simulation

By simulating these three methods, we analyze the accuracy on Doppler shift detection, and how it affects frame correction and synchronization. The parameters used for simulation are summarized in Table I. First, we compute a signal-to-noise ratio (SNR) sweep 1000 times averaging the results at each SNR for avoiding aberrant errors in our results. We consider a Doppler shifting of $f_d = 20$ Hz at $f_0 = 30$ kHz, which corresponds to a relative motion of 1 m/s. Fig. 3 displays the root mean squared error (RMSE) of Doppler shift estimation defined as

$$\text{RMSE} = \sqrt{\mathbb{E}\left[|f_d - \hat{f}_d|^2\right]}. \tag{24}$$

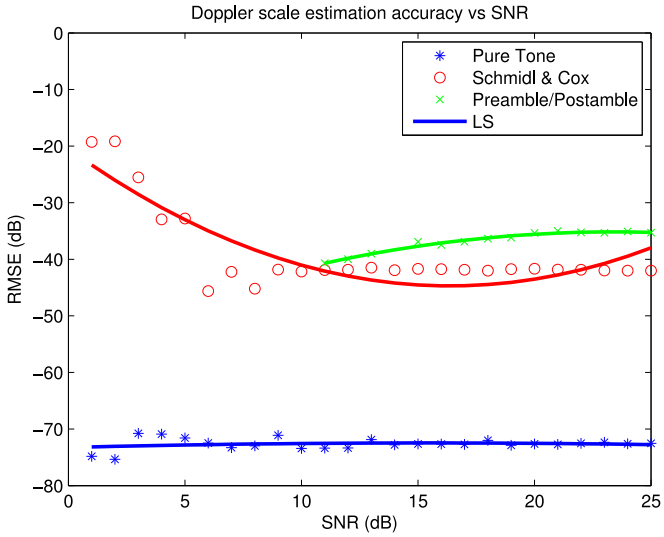

Fig. 3. Doppler scale estimation simulation $f_d = 20$ Hz at $f_0 = 30$ kHz with the SNR sweep.

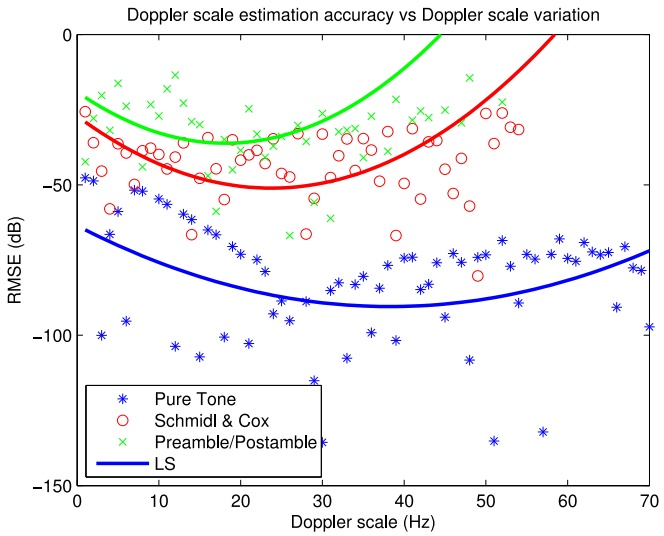

Fig. 4. Doppler scale estimation simulation versus SNR = 15 dB at $f_0 = 30$ kHz with the f_d sweep.

Then, in Fig. 4, the same simulation is repeated, but this time we keep a constant SNR of 15 dBb and the sweep is performed along f_d from 0 to 70 Hz. By using the term LS, we mean that a second-order least square regression model has been applied to make results more readable. It is just an approximation, which in some cases differs from real acquired values given by raw data points.

Hence, if we use this Doppler scale estimation for correcting frequency shifting in our communication we will be able to recover enclosed timing information. Fig. 5 displays frame mean square error (MSE), after applying Doppler scale compensation and channel equalization with each described algorithm

$$\text{RMSE} = E[|\hat{d}_k - d_k|^2] \tag{25}$$

where $\hat{d}_k$ denotes the estimation of data cell d_k after OFDM equalization.

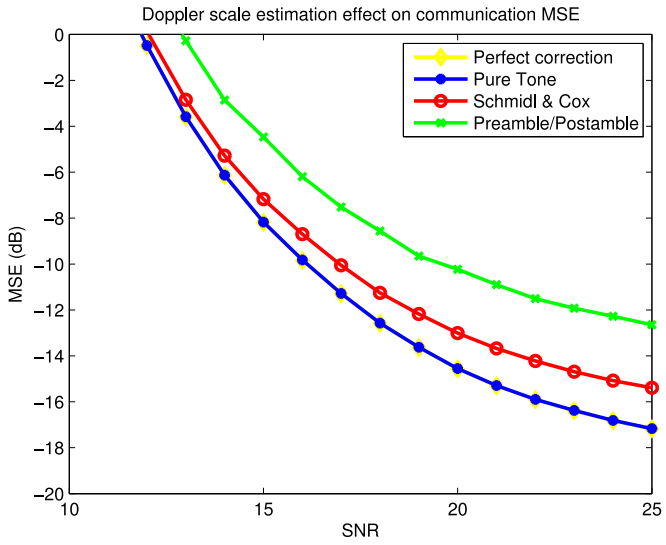

Fig. 5. Frame MSE after Doppler scale compensation on an SNR sweep with $f_d = 20$ Hz in simulation.

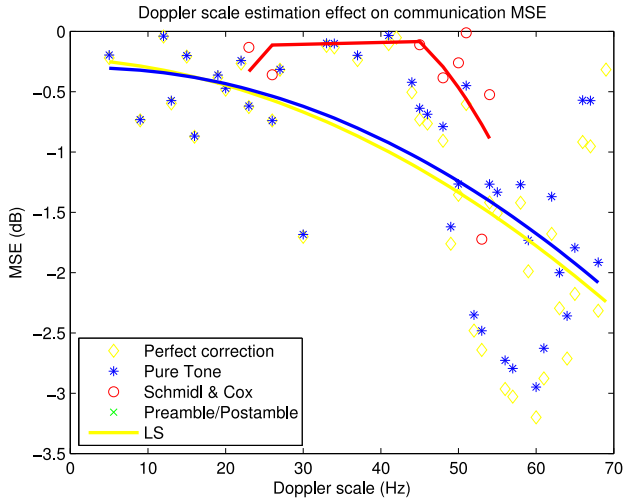

Fig. 6. Frame MSE after Doppler scale compensation on a Doppler scale frequency sweep with SNR = 15 dB in simulation.

The curve labeled perfect correction provides a lower bound on MSE performance where the Doppler shift is perfectly known and removed. Finally, in Fig. 6, we run a simulation where we keep a constant 15-dB SNR, and we perform a Doppler shift sweep.

This simulation gives us an idea of which algorithm will have better performance in an ideal scenario. The pure tone approach outperforms all the other algorithms amply. All three methods have a flat response for the Doppler frequency shift scenario, and they present a direct response with the SNR sweep. As the SNR increases, the Doppler scale estimation improves and, consequently, the MSE in the frame after correction also gets better.

IV. Time Synchronization

This study aims to synchronize two different sensor clocks by performing a bidirectional message exchange enclosing timing data. To do so, we use a widely known scheme in time-synchronization approaches [1] illustrated in Fig. 7. This bidirectional message exchange encloses data transmission and reception times, in addition to frame relative velocities. This information is used by the slave clock estimating two key points in time synchronization: the clock offset and the clock skew. Fig. 7 represents all synchronization information sources that will be described in this section.

The offset is the difference between the slave clock and the master one. This can be estimated by computing propagation times to compensate for the timestamp delay due to the message exchange. With this, it is possible to compensate for the clock error and set the slave clock to the same time as the master one. The second part, regarding the clock skew, must be also taken into account to set both clocks running with the same frequency. If this clock skew is not compensated for, the slave clock will suffer a clock drift relative to the master clock, which will not be compensated for until the next clock offset estimation.

A. Clock Offset Estimation

The slave node records the sending timestamp T_1, obtained at the physical layer (PHY), right before the message leaves. To be able to acquire a clock reading only at the PHY, we present a new approach using both the hardware timestamp with a field-programmable gate array (FPGA) to determine the acquisition start time, and then a software tunning in order to locate the starting sample of the message inside an acquisition window, where in addition to useful information there is also channel noise.

The workflow is to trigger a clock reading, and save its value in an FPGA hardware register each time a frame exits to the channel. This way we have a precise output timestamp acquired by hardware. On the other hand, for the input signals, we can also trigger a hardware reading clock when acquisition systems start to receive data, but this does not mean that the useful data start just when the first sample of the frame arrives. We will need to detect where it is placed inside the whole acquisition window.

Software detection and timestamp tunning is based on an LFM signal associated with a conventional matched filter. The main reason for using an LFM is the fact that its robustness does not decrease in high dispersive channels over time. Since we work with discretized signals, we will detect the first sample of the frame instead of the exact time of the signal arrival, which translated to time will have an accuracy equal to the inverse of the sampling rate. Therefore, the center of the gravity algorithm is applied around the detection of the first sample of the frame. With this we are able to estimate, with more accuracy, in time, the starting point of the message inside the acquisition window [29].

Upon receiving the message, the reference node also timestamps T_2, using FPGA timestamping as described above, and estimates and records the ordinary node's relative velocity v_0 with Doppler shifts as specified in Section III and message propagation time τ_0 as well. Then, after a time interval t_r, the reference node sends back a message which will generate T_3, T_4, τ_0, and v_1.

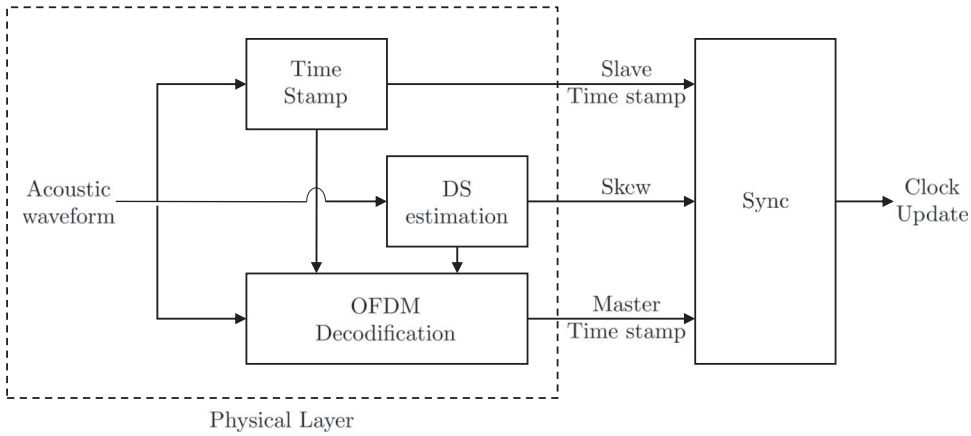

Fig. 7. Time-synchronization schema.

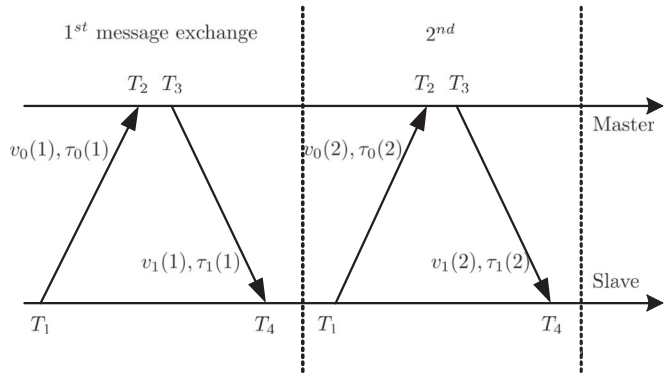

Fig. 8. Synchronization schema.

Once this bidirectional messaging procedure is completed, the system is able to compute clock offset (β) and propagation time (τ), which will allow time synchronization between the clocks. Depending on the accuracy requirement, the above message exchange process can run multiple times.

The message exchange process is shown in Fig. 8

$$\hat{\beta} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \tag{26}$$

$$\hat{\tau} = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}. \tag{27}$$

B. Clock Skew Estimation

The clock skew is extracted from a bidirectional message exchange for initialization, and after several message exchanges, it is computed by performing a weighted linear regression.

By combining (11) and (12), clock skew (θ) can be isolated from the Doppler scaling factor which has been estimated previously by one of the algorithms evaluated in Section III.

This first approach provides clock skew information, even though this is strongly affected by one frame quality. Then, by adding more information to this estimation, such as the kinematic model of the nodes and a linearization of several computations, a better performance of this estimation can be reached.

C. Data Collection

For time synchronization between the pair of clocks, DA-Sync relies on estimating the clock offset and skew, which present the relation between the time measured by two different clocks.

To do so, a bidirectional message exchange between nodes is used, as has been presented above.

D. Velocity Estimation Refinement

Since velocity will be used in our synchronization algorithm for computing propagation times, which will affect the linear regression used for skew estimation, it is necessary to estimate it as well as possible.

Then, by using kinematic equations, it is possible to refine the initial velocity estimation obtained from the DS factor calculated in Section III. Assuming a first-order kinematic model with a constant acceleration between consecutive sampling times, as described in [1], we have the dynamic equation

$$x(k+1) = F(k)x(k) + \Gamma(k)w(k) \tag{28}$$

where $w(k)$ denotes the discrete-time process noise, which is supposed to follow a Gaussian distribution. Then, as presented in [1], we have

$$x(k+1) = [v(k+1) \quad \alpha(k+1)]^T \tag{29}$$

in which $v(k+1)$ and $\alpha(k+1)$ denote velocity and acceleration, respectively. $\Delta T(k)$ can be determined in a two-point differencing procedure as

$$\Delta T(k) = \frac{\tau_0[k] + \tau_1[k]}{2} + T_3[k] - T_2[k]. \tag{30}$$

The estimation of the state $(\mathrm{k}+1)$ based on the measurement of v_0 and v_1 can be obtained with the Kalman filter, which is an optimal minimum mean square error (MMSE) state estimator under the Gaussian assumption of both the process noise and the measurement noise. Figs. 9 and 10 display the estimation of velocity and acceleration refinement by using Kalman filtering in the first kinematic model equations.

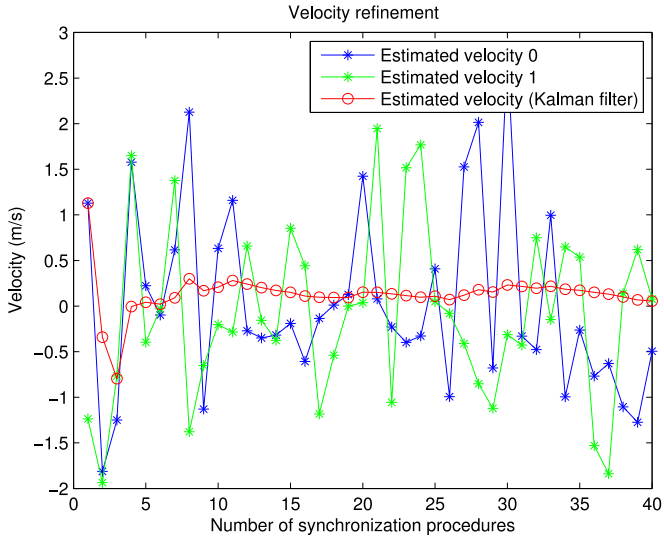

Fig. 9. Kalman filter for velocity refinement.

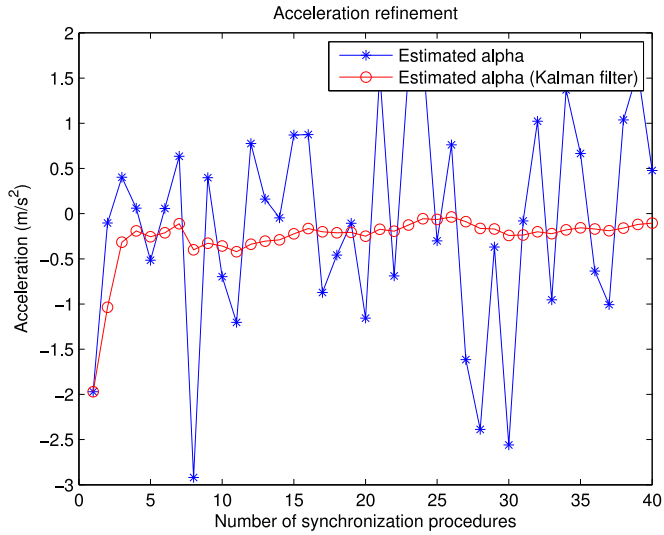

Fig. 10. Kalman filter for acceleration refinement.

Regarding the kinematic model for the relative velocity, we define

$$v = k_1 t + k_2 \tag{31}$$

where k_1 is the parameter closely related to the environmental factors such as tides and bathymetry, and k_2 is used to simulated some random factors. In our simulations, k_1 is a random variable following a normal distribution with π m/s as the variance value and 0.1π m/s as the mean.

For filter initialization, we use measured velocities in the message exchange [1]

$$\hat{v}(1) = v_0[1] \tag{32}$$

$$\hat{\alpha}(1) = \frac{v_1[1] - v_0[1]}{\Delta T[1]}. \tag{33}$$

E. Propagation Time Estimation

This phase aims to estimate the long and dynamic propagation delays.

In our synchronization schema, we have to handle τ_1 and τ_2. Then, these propagation delays will be used to compute the weighted least squares estimation (WLSE) which will result in a skew estimation by computing a linear regression slope.

To compute propagation delays, we use velocity and acceleration obtained in Sections III and IV-D. Since the acoustic waves propagation time is not linked to an individual moving pattern of any sensor, but a relative movement between two nodes, we can work directly with relative velocities obtained from the physical layer, instead of using a 3-D velocity schema.

Following the DA-Sync protocol [1], we obtain the propagation delays τ_1 and τ_2

$$\begin{cases} \tau_1 = \dfrac{(T_4 - T_1)c_w - \theta(T_3 - T_2)(c_w + \hat{v}_0) - \frac{1}{2}\hat{\alpha}(T_3 - T_2)^2}{2\theta c_w} \\ \tau_2 = \dfrac{(T_4 - T_1)c_w + \theta(T_3 - T_2)(\hat{v}_0 - c_w) + \frac{1}{2}\hat{\alpha}(T_3 - T_2)^2}{2\theta c_w}. \end{cases} \tag{34}$$

In (34), θ is needed to estimate the propagation times, and since in this algorithm step, this value is not known yet, we will set it as "1," which will be corrected in the calibration procedure.

F. Linear Regression

In this section, the clock skew (θ) is estimated by using previous timestamps and the propagation time computation (35).

Parameters β and θ are obtained by linear regression

$$T_2[k] = \theta(T_1[k] + \tau_1[k]) + \beta. \tag{35}$$

By default, in an ideal scenario, both axes of the linear regression should be identical, but they are affected by the clock skew and offset.

With linear regression, we obtain the offset from zero in the initial value of the regression line, which is the offset between clocks (β), and the slope of the line as clock skew (θ), where the regression line is defined as

$$y = \theta x + \beta. \tag{36}$$

In Fig. 11, the accuracy of the skew estimation by using three different approaches is plotted: 1) direct skew estimation at each message exchange; 2) ordinary least squares estimation (OLSE) [30]; and 3) WLSE [31]. This test has been run ten times to observe the variance of the estimations. Now, we can ensure better performance of WLSE in any scenario.

G. Calibration

For achieving this skew estimation, we have assumed in the previous section that to estimate the propagation times $\theta = 1$. So now we will need to repeat the propagation delay estimation and linear regression with the actual estimated skew values. This procedure will be repeated up to ten times or θ stabilization, as described in Fig. 12.

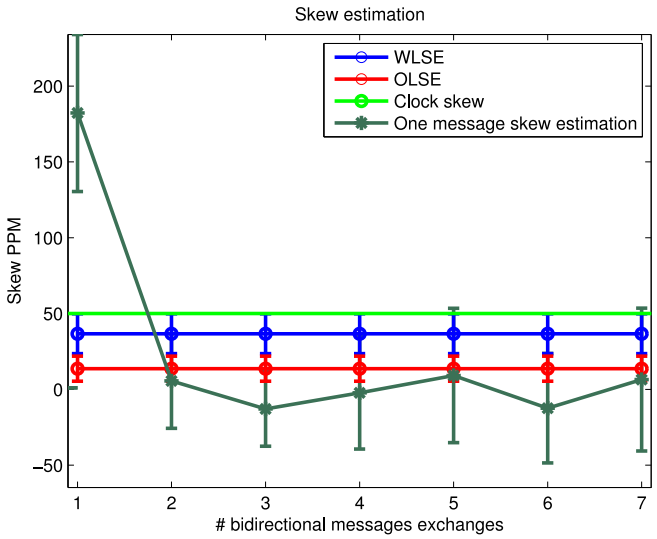

Fig. 11. Simulation of the clock skew estimation.

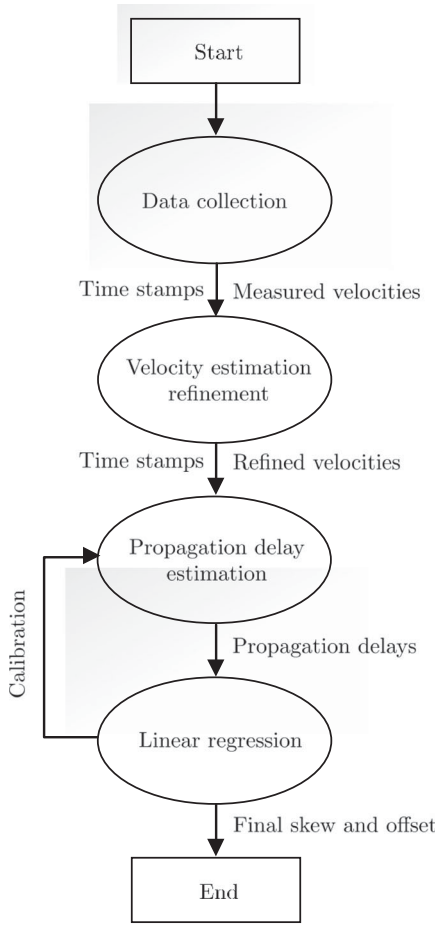

Fig. 12. Calibration procedure.

H. Simulation

After running the DA-Sync algorithm with our own frame detection procedure and Doppler scale factor estimation by using LFM and the pure tone method, respectively, we obtain the following synchronization results, after 10 s, since the last synchronization procedure described in Fig. 12. The parameters used for simulation are described in Table II. We set the inherent skew of the ordinary clock at 50 parts per Million (PPM); the clock offset is initialized as 800 μs. The response time is fixed at 1 s, and the propagation speed (c_w) is 1500 m/s.

TABLE II
TIME-SYNCHRONIZATION PARAMETER SUMMARY (TYPICAL VALUES)

Description	Parameter	Simulation	Laboratory	OBSEA
Signal to Noise Ratio	SNR	15 dB	15 dB	15 dB
Doppler frequency shift	f_d	20 Hz	20 Hz	20 Hz
Clock offset	β	800 ms	800 ms	800 ms
Clock skew	θ	50 PPM	50 PPM	50 PPM
Node relative movement	v_0, v_1	1 m/s	1 m/s	1 m/s
Sound propagation speed in sea	c_w	1500 m/s	1500 m/s	1500 m/s
Distance between nodes	d	300 m	2 m	1.5 m
Frame propagation time	$\tau_0, \tau_1, \tau_2, \tau_3$	200 ms	1.3 ms	1 ms

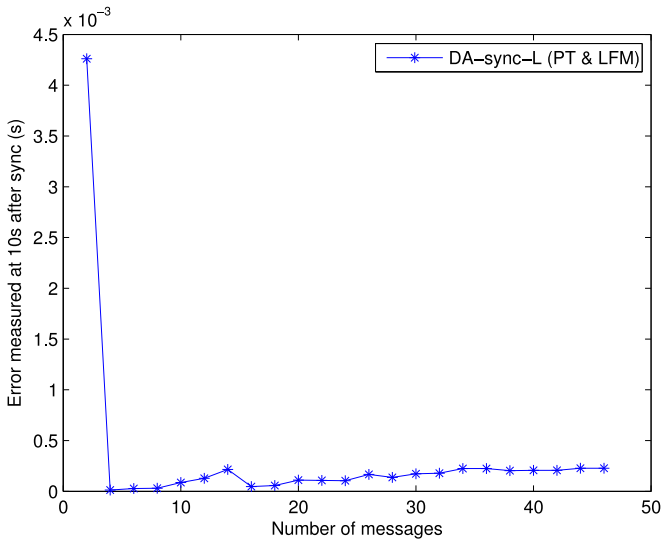

Fig. 13. Simulation of the clock offset error after 10 s versus the number of messages.

As can be observed in Fig. 13, after eight message exchanges, an accuracy below 400 μs can be reached, going below 200 μs after 22 message exchanges, due to the Kalman filter predictor enhancement with the increment of initialization points. If we analyze our system behavior after eight message exchanges, we can compute the time offset estimation error just after a synchronization procedure, and compare its performance by following Fig. 12 refinement, or directly compute the offset without any compensation. This statement is shown in Fig. 13.

By repeating this synchronization algorithm 100 times (made up of eight message exchanges), we can observe how Kalman filtering, applied in the form of a correction for a weighted linear regression, versus not applying any correction, is close to double synchronization accuracy. Fig. 14 shows the mean and the variation of the offset estimation after applying skew compensation, in an eight message exchange algorithm repeated 100 times.

On the other hand, in Fig. 15, we have the same simulation histogram but this time without the skew correction. We will

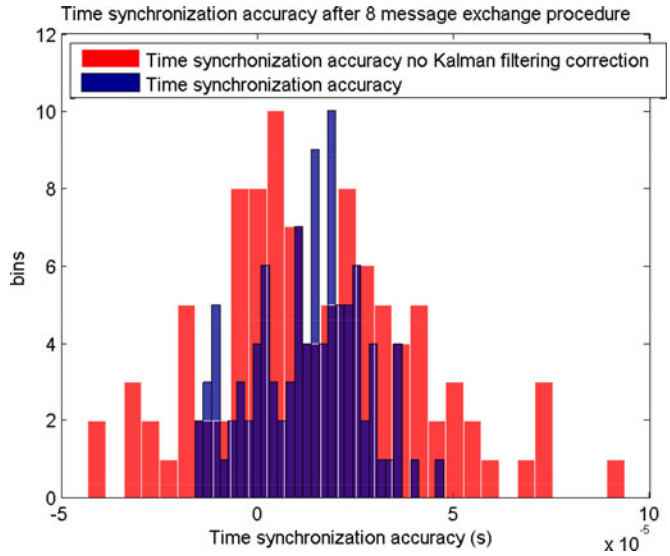

Fig. 14. Simulation of time-synchronization accuracy after the eight message exchange procedure repeated 100 times.

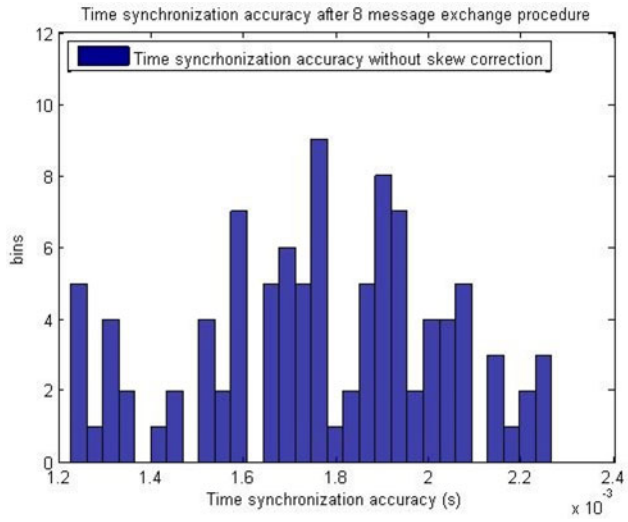

Fig. 15. Simulation of time-synchronization accuracy after the eight message exchange procedure without correcting the clock skew repeated 100 times.

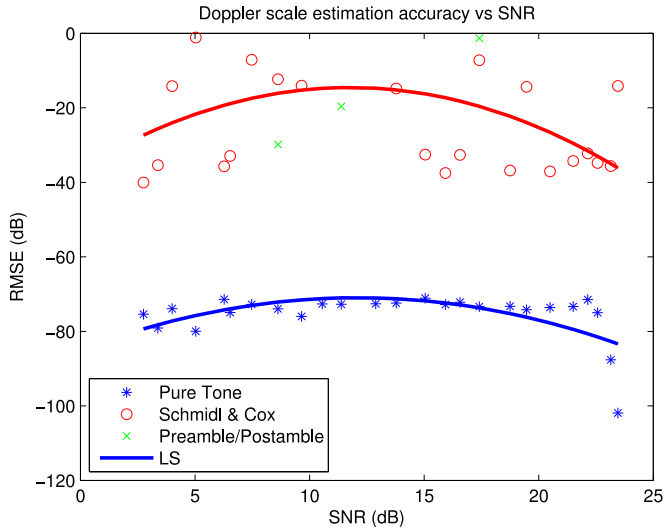

Fig. 16. Doppler scale estimation Laboratory test $f_d = 20$ Hz at $f_0 = 30$ kHz with the SNR sweep.

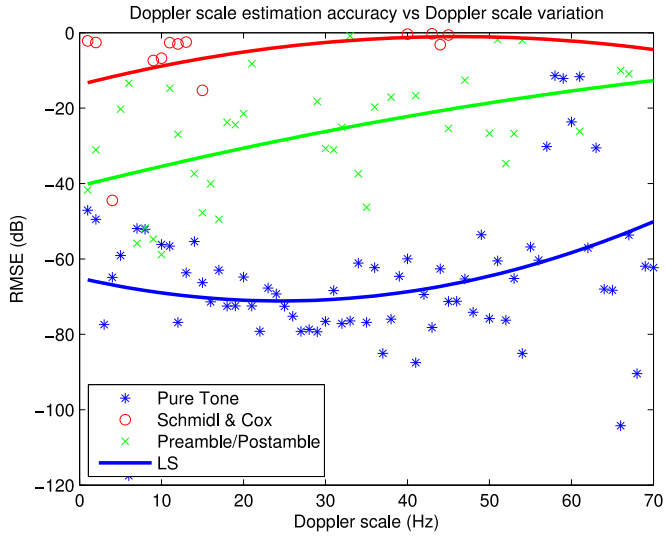

Fig. 17. Doppler scale estimation Laboratory test SNR = 15 dB at $f_0 = 30$ kHz with the f_d sweep.

have an offset correction but the skew will drift the clock during the synchronization procedure. In our study, it takes $t_{\text{sync}} = 59$ s to exchange eight bidirectional messages, so offset computation will be affected by the factor of θt_{sync}, as can be seen in the plot, where we have a histogram center shifting due to the clock skew.

V. Results

In this section, we evaluate previously simulated algorithms in real water tests, in order to verify their proper functionality in a real scenario. Parameters used for both laboratory and real tests are summarized in Tables I and II.

A. Doppler Scale Factor Estimation

1) Laboratory Test: This test was performed in a 150-cm × 40-cm × 40-cm water test tank with a separation between hydrophones of 1 m. With this experiment, we intend to verify a proper functionality of Doppler scale estimation algorithms in a controlled environment, where we do not have inherent sea currents, so that we can simulate Doppler shifting to estimate it without any additive sea factor. A simulated Doppler shift is added in the received signal by resampling the received passband signal by a factor of $1 + a$.

To compare each Doppler shift estimation algorithm, we use the same methodology as in Section III-D, but this time the signal is transmitted in the "test tank" where we have very difficult channel conditions with a strong multipath of about 75 ms, and all tests are averaged only ten times due to the real test's processor timing constraints. Figs. 16 and 17 show the RMSE performance with the SNR sweep and the Doppler shift sweep, respectively. In practice, SNR variation is performed by modifying the transmit signal power, and then the SNR displayed in

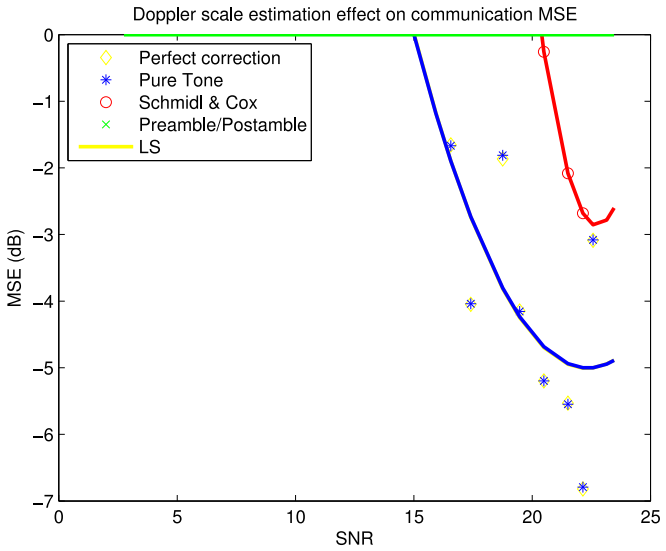

Fig. 18. Frame MSE laboratory test after Doppler scale compensation on an SNR sweep with constant $f_d = 20$ Hz.

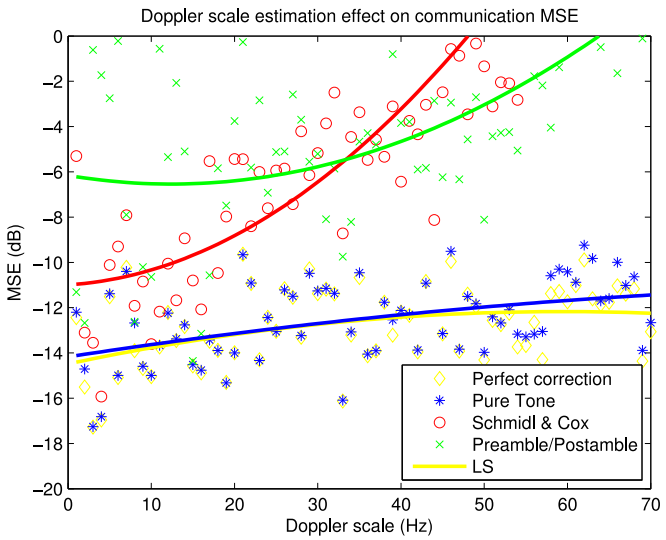

Fig. 19. Frame MSE laboratory test after Doppler scale compensation on a Doppler scale frequency sweep with constant SNR = 15 dB.

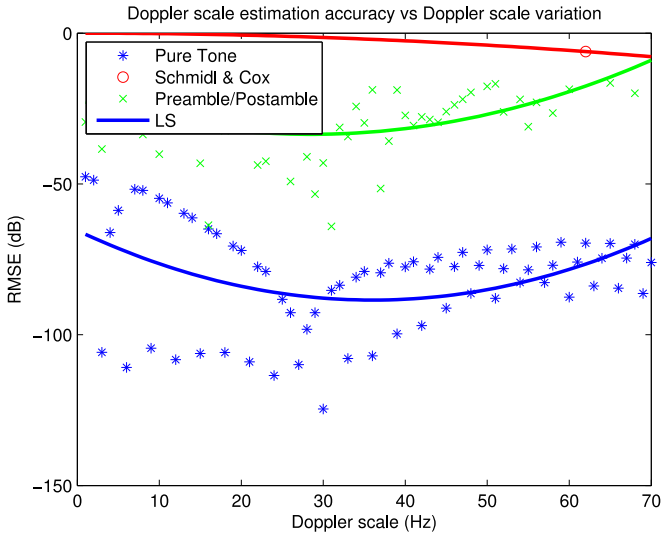

Fig. 20. Doppler scale estimation at OBSEA SNR = 15 dB at $f_0 = 30$ kHz with the f_d = sweep.

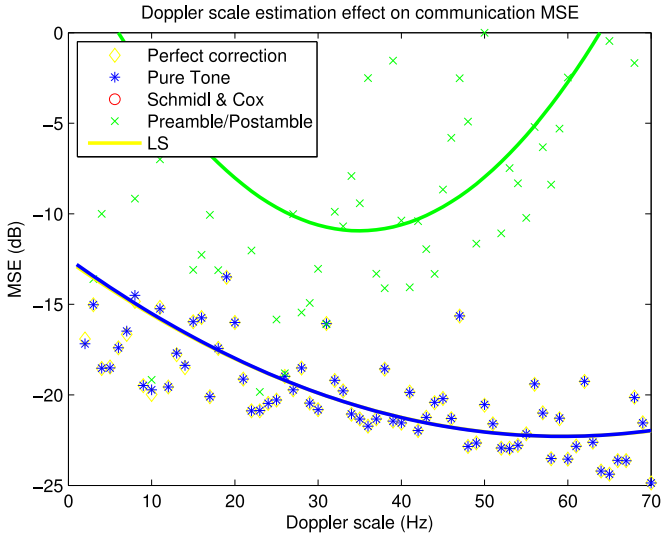

Fig. 21. Frame MSE at OBSEA after Doppler scale compensation on a Doppler scale frequency sweep with the estimated SNR = 15 dB.

the aforementioned figures is an estimated SNR defined as the ratio between the signal and the noise of each channel.

In the second step, we apply Doppler scale compensation to the OFDM frame with each frequency shifting estimation obtained in the previous stage, and we evaluate the MSE of the data frame after compensation and channel equalization. The results are presented in Figs. 18 and 19, respectively. For these laboratory tests, we repeat both SNR and Doppler frequency sweeps to observe the algorithm performance for any water channel status.

As in simulation, the PT approach outperforms the other estimation algorithm, leading to near-perfect Doppler shift correction at an SNR of 15 dB.

2) Real Test: This experiment was conducted on June 8, 2015. The same signal set as described in Section III was used. The signal was transmitted from a depth of about 20 m and received by a single hydrophone at the same depth of 1.5 m of separation of the transmitter. Both hydrophones were attached at 0.5 m of seafloor. The OFDM signals were transmitted while the hydrophones were moving free in a horizontal axis by the sea currents.

As in Section III, we first evaluate the RMSE performance of the Doppler scale algorithms, as plotted in Fig. 20, and then the MSE performance after correction and equalization is provided in Fig. 21. In this real test, we only perform a Doppler scale sweep with a constant estimated SNR fixed at 20 dB, due to hardware limitations. The power amplifier clips the signal if we need to transmit high power in the water channel, so we are not able to increase the SNR up to the same levels shown in the laboratory tests. Again, the Doppler scale is simulated by resampling the received passband signal.

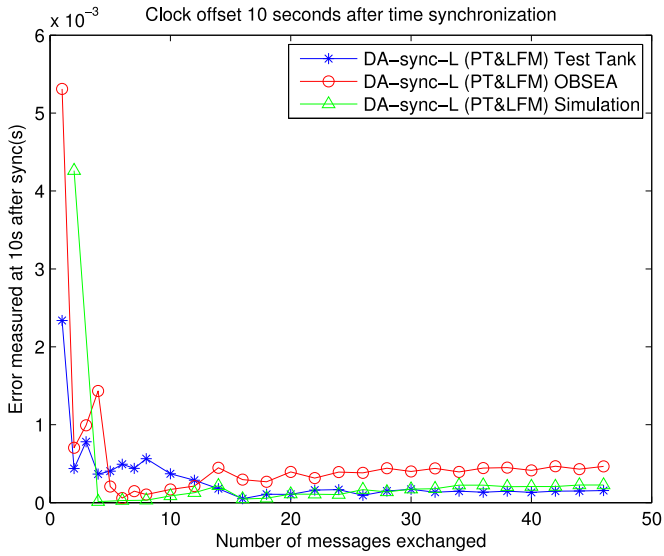

Fig. 22. Clock offset error after 10 s versus the number of messages with the clock skew set at 50 PPM and the node movement simulation of 1 m/s.

These experimental tests confirm how pure tone outperforms the S&C algorithm and the preamble/postamble LFM one, in all the possible scenarios. As in the laboratory test, Doppler scale estimation and compensation lead to an MSE quite close to a perfect Doppler shift correction. So this algorithm will be the one used for the synchronization part of this study.

B. Time Synchronization

In this section, we repeat same simulation workbench as in Section IV-H, but this time we send the signal to the cRIO module [29] instead of simulating the physical layer. Then, the signal is sent through the test tank for the laboratory test and through a real water channel in front of OBSEA for the real test. Only the water channel is modified from simulation to these real tests. Then, we expect similar performance as in simulation with some differences regarding maximum time to perform Kalman filtering and WLSE.

1) Laboratory Test: In this case, we have a controlled water environment, without water movement, so we can simulate the currents by adding Doppler scaling in our signal. Since we have simulated currents, these currents will not change during the experiment, so the result is similar time accuracy as in simulation, as shown in Fig. 22.

2) Real Test: In real tests, we should have a time-synchronization evaluation similar to the previous one. But in this case, we do not have sea current control, so they can vary in time, causing the Kalman filtering and WLSE to be limited in time to avoid extrapolating Doppler scaling from different sea current sources. As can be seen in Fig. 22, time accuracy starts to decrease after the eighth message, which means after approximately 2 min from the start of the test. So for this case, we can assume that sea currents vary significantly every 2 min. In a real application, this time should be autoadjusted in a function of time-synchronization error evolution, which is linked to the length of the Kalman estimator and WLSE.

VI. Conclusion

In this paper, we investigated the performance of time-synchronization algorithms in UWA channels with Doppler shifts. To compensate for the Doppler distortion, we have evaluated three different frequency shift estimators, which have been applied to a time-synchronization algorithm, DA-Sync [1]. The proposed work was tested in simulation, then in a water tank, and finally, in real-time shallow-water experiments. Using an SNR of 15 dB and 20 Hz of acoustic signal frequency shifting, the pure tone approach was demonstrated to outperform both preamble and postamble and S&C Doppler scale estimators.

Over the clock skew of 50 PPM, using the pure tone algorithm for the Doppler scale estimation and LFM for frame detection, since this scenario provided the best results, time-synchronization accuracy after 10 s was 179 μs in simulation, 170 μs in laboratory tests, and 400 μs in the OBSEA tests. Good performance was achieved even at high-frequency shifting (up to 70 Hz) and low SNR (less than 15 dB). Experimental results suggest that the time-synchronization algorithm decreases its performance with large message exchange procedures. Since the water channel medium changes its current characteristics during the synchronization procedure, Kalman filtering for velocity estimation is worthless, and also, at high vehicle velocity, the frame detection algorithm may decrease its performance due to LFM matched filter false peak detections.

Acknowledgment

The authors would like to thank the reviewers for their thoughtful comments. The authors would also like to thank National Instruments Spain for providing the necessary hardware used in this study. The innovative multifunctional sensors for *in situ* monitoring of marine environment and related maritime activities project (NexOS) is a collaborative project funded by the European Commission 7th Framework Programme, under the call OCEAN-2013.2—The Ocean of Tomorrow 2013. It is composed of 21 partners including SMEs, companies and scientific organizations from six European countries. Visit: www.nexosproject.eu. The Fixed point Open Ocean Observatory network (FixO3) seeks to integrate European open ocean fixed point observatories and to improve access to these key installations for the broader community. The proposal has 29 partners drawn from academia, research institutions, and SMEs. Visit: http://www.fixo3.eu/

References

[1] J. Liu *et al.*, "DA-Sync: A Doppler-assisted time-synchronization scheme for mobile underwater sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 582–595, 2014.

[2] M. Stojanovic, "Underwater acoustic communications," *Encyclopedia of Electrical and Electronics Engineering*, John G. Webster, Ed., New York, NY, USA: Wiley, 2015, pp. 1–12.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 257–279, 2005.

[4] J Ribas and M. Stojanovic, "Underwater wireless video transmission using acoustic OFDM," Ph.D. dissertation/M.S. thesis, Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[5] M. Debbah, "Short introduction to OFDM," White Paper, Mobile Communications Group, Institut Eurecom, 2004.

[6] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 198–209, 2008.
[7] J. Huang, S. Zhou, J. Huang C. R. Berger, and P. Willett, "Progressive inter-carrier interference equalization for OFDM transmission over time-varying underwater acoustic channels," *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 5, pp. 1524–1536, 2011.
[8] K. Lee, J. C. Eidson, H. Weibel, and D. Mohl, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," p. 2, Oct. 2005.
[9] H. Jiang, X. Liu, X. Wang, W. Liu, and Y. Wang, "Tri-Message: A lightweight time synchronization protocol for high latency and resource-constrained networks," in *Proc. IEEE Int. Conf. Commun.*, 2009, DOI: 10.1109/ICC.2009.5199544.
[10] A. A. Syed and J. S. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. INFOCOM*, Apr. 2006, DOI: 10.1109/INFOCOM.2006.161.
[11] N. Chirdchoo, W. S. Soh, and K. C. Chua, "MU-Sync: A time synchronization protocol for underwater mobile networks," in *Proc. 3rd ACM Int. Workshop Underwater Netw.*, Sep. 2008, pp. 35–42.
[12] F. Lu, D. Mirza, and C. Schurgers, "D-sync: Doppler-based time synchronization for mobile underwater sensor networks," in *Proc. 5th ACM Int. Workshop UnderWater Netw.*, Sep. 2010, p. 3.
[13] J. Liu *et al.*, "TSMU: A time synchronization scheme for mobile underwater sensor networks," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2011, DOI: 10.1109/GLOCOM.2011.6134447.
[14] K. Lee, J. C. Eidson, H. Weibel, and D. Mohl, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," Oct. 2005.
[15] H. Weibel and D. Bchaz, "1588—Implementation and Performance of Time Stamping Techniques," Sep. 2004.
[16] K. G. Kebkal, V. K. Kebkal, O. G. Kebkal, and R. Petroccia, "Underwater acoustic modems (S2CR Series) for synchronization of underwater acoustic network clocks during payload data exchange," *IEEE J. Ocean. Eng.*, vol. 41, no. 2, pp. 428–439, 2015, DOI: 10.1109/JOE.2015.2431531.
[17] J. Yi *et al.*, "ToA-TS: Time of arrival based joint time synchronization and tracking for mobile underwater systems," *Ad Hoc Netw.*, vol. 34, pp. 211–223, 2015, DOI: 10.1145/2532378.2532404.
[18] Expandable Seafloor Observatory OBSEA, Online. [Available]: http://www.obsea.es
[19] Z. Liu and T. C. Yang, "On overhead reduction in time-reversed OFDM underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 39, no. 4, pp. 788–800, 2014.
[20] V. Le Nir, T. van Waterschoot, J. Duplicy, and M. Moonen, "Blind coarse timing offset estimation for CP-OFDM and ZP-OFDM transmission over frequency selective channels," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, p. 44, 2009.
[21] P. J. Bouvet and A. Loussert, "An analysis of MIMO-OFDM for shallow water acoustic communications," in *Proc. OCEANS Conf.*, Sep. 2011, pp. 1–5.
[22] P. Qarabaqi and M. Stojanovic, "Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels," *IEEE J. Ocean. Eng.*, vol. 38, no. 4, pp. 701–717, 2013.
[23] M. Stovanovic and L. Freitag, "Integrated Doppler tracking and efficient resampling for phase coherent acoustic communication," *IEEE J. Ocean. Eng.*, vol. 19, no. 1, pp. 100–111, 1994.
[24] L. Wan, Z. Wang, S. Zhou, T. C. Yang, and Z. Shi, "Performance comparison of Doppler scale estimation methods for underwater acoustic OFDM," *J. Electr. Comput. Eng.*, vol. 2012, pp. 1–1, 2012.
[25] P. Kathiroli, P. P. Beaujean, and N. Xiros, "Source speed estimation using a pilot tone in a high frequency acoustic modem," in *Proc. IEEE OCEANS Conf.*, pp. 1–8, 2011.
[26] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, 1997.
[27] R. Diamant, A. Feuer, and L. Lampe, "Choosing the right signal: Doppler shift estimation for underwater acoustic signals," in *Proc. 7th ACM Int. Conf. Underwater Netw. Syst.*, Nov. 2012, p. 27.
[28] B. S. Sharif, J. Neasham, O. R. Hinton, and A. E. Adams, "A computationally efficient Doppler compensation system for underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 52–61, 2000.
[29] O. Pallares, P. J. Bouvet, and J. del Rio, "Underwater acoustic communication messaging time stamp applied to global time synchronization," in *Proc. IEEE Sensor Syst. Changing Ocean*, 2014, DOI: 10.1109/SSCO.2014.7000380.
[30] F. Hayashi, *Econometrics*, Princeton, NJ, USA: Princeton Univ. Press, 2000, ISBN: 0-691-01018-8, Ch. 1.
[31] B. R. Moulton, "Random group effects and the precision of regression estimates," *J. Econometrics*, vol. 32, no. 3, pp. 385–397, 1986.

Oriol Pallares (S'16) received the B.Sc. degree in electronics engineering (telecommunications) and the M.Sc. degree in electronics from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2012, where he is currently working toward the Ph.D. degree.

He is a Research Assistant at the research group Remote Acquisition Systems and Data Processing (SARTI), Universitat Politècnica de Catalunya. His major research interests include time synchronization, acoustic communication and deployment for underwater sensor networks.

Mr. Pallares is a member of the Open Geospacial Consortium (OGC).

Pierre-Jean Bouvet (M'10) was born in Clermont-Ferrand, France, in 1978. He received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from the National Institute of Applied Science (INSA), Rennes, France, in 2001 and 2005, respectively.

In 2005, he joined NXP semiconductors (formerly Philips), Caen, France, as Baseband Decoding Architect for digital TV demodulator products (DVB-T/T2). Since 2009, he has been an Aassociate Professor in the Department of Embedded Systems, Acoustics, and Communications in the engineering college of ISEN, Brest, France. His current research interests include underwater acoustic communications, multiple-input–multiple-output (MIMO) transmission, iterative reception, and synchronization algorithms.

Joaquin del Rio (M'07) was born in Catalonia, Spain, in 1976. He received the B.S., M.S., and Ph.D. degrees in telecommunication engineering and electronic engineering from the Technical University of Catalonia (UPC), Catalonia, Spain, in 1999, 2002, and 2011, respectively.

Since 2001, he has been a Professor in the Electronic Engineering Department, Universitat Politècnica de Catalunya, Barcelona, Spain. He is a member of the research group Remote Acquisition Systems and Data Processing (SARTI). His research focuses on electronic instrumentation, interoperability in marine sensor networks, and wireless sensor networks. He is involved in projects within the industry and publicly funded research projects and is a National Instruments Certified Instructor for teaching official LabVIEW courses.

Dr. del Rio is a member of the NIST IEEE1451.2 Working Group and the Smart Ocean Sensors Consortium and PUCK Standard Working Group (SWG).