TS-MUWSN: Time Synchronization for Mobile Underwater Sensor Networks

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

Abstract

Time synchronization is an important, yet challenging, problem in Underwater Sensor Networks (UWSN). This challenge can be attributed to: 1) Messaging time stamping; 2) Node mobility; and 3) Doppler scale effect. To mitigate these problems we present a time synchronization algorithm for UWSN, where we compare several message time stamping algorithms besides different Doppler scale estimators.

Synchronization system is based on a bidirectional message exchange between a reference node and an slave one, which has to be synchronized. Therefore, we take as reference DA-Sync-like protocol [1], that is taking into account node's movement by using first order kinematic equations, what refines Doppler scale factor estimation accuracy, and results 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 both: controlled underwater communication in a water test tank; and shallow water test in Mediterranean sea.

Index Terms

Underwater Sensor Network (UWSN), Acoustic communication, Orthogonal Frequency-Division Multiplexing (OFDM), Doppler scale, Time stamp, Time synchronization, Skew.

O. Pallares and J. del Rio are with SARTI research group [Electronics Dept] of Technical University of Catalunya (UPC), Rambla de l'Exposició 24, 08800, Vilanova i la Geltrú. (e-mail: oriol.pallares@upc.edu & joaquin.del.rio@upc.edu).

P.-J. Bouvet is with Underwater Acoustics Lab ISEN Brest, C.S. 42807, 29228 Brest cedex 2, France. (e-mail: pierre-jean.bouvet@isen-bretagne.fr).

Manuscript received April 19, 2015; revised month dd, yyyy.

I. Introduction

Underwater wireless sensor networks (UWSN) 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 multi-paths and low transmission speed that can nevertheless be corrected at the reception side, this makes acoustic communication very attractive and widely used in underwater scenario [2]. Multicarrier modulation in the form of Orthogonal Frequency-Division Multiplexing (OFDM) has prevailed in recent broadband wireless radio applications due to the low complexity of receivers required to deal with highly dispersive channels [3] [4]. Since time synchronization protocols are based in a message exchange between a reference clock and an ordinary one, OFDM acoustic communication will be used for enclosing time information with each message exchange.

Time synchronization is a critical piece of infrastructure for any distributed system. UWSN make extensive use of synchronized time for many services provided by a distributed network [5].

In UWSN, 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 cabled or radio network makes the use of conventional synchronization protocols [6] even more challenging in underwater.

Many time synchronization algorithms for Underwater Sensor Networks can be found in literature [7] [8] [9] [10] [11] [1], but only few of them take into account all the water channel challenges, such as low available bandwidth, long propagation delays and sensor node mobility [11] [1].

In this work, in order to perform time synchronization over an UWSN, we apply Precision Time Protocol (PTP) std. IEEE 1588, which is capable to synchronize two clocks with a precision below hundreds of nanoseconds in a point to point cabled Ethernet Network [12]. Synchronization procedure is based in a bidirectional message exchange between a master clock and an slave, which is the one to synchronize.

In cabled synchronization systems, such as PTP, time stamps are acquired in Medium Access Controller (MAC) in order to achieve maximum precision, avoiding indeterministic times like Operating System (OS) time slots or medium access protocols [13]. Analogously, it happens in acoustic communication, time stamps are extracted from a large acquisition window with raw data, where is found the main signal to be processed.

So far, cabled synchronization could be ported directly to an acoustic one, but we will need a system capable to trigger a clock acquisition at a precise time, just when an acoustic frame enters or exits the system, in order to reproduce same behavior than PTP time stamping.

Contrary to cable networks, the low celerity of wave sound makes underwater acoustic communications system very sensitive to Doppler effect, yielding to non-uniform frequency scaling represented by compression or dilatation of the time axis. this frequency scaling can be induced by 2 factors : motion (sensor mobility, channel variation, etc...) and clock skew receiver between transmitter and receiver. So it is necessary to evaluate in the Doppler Scaling which part is caused by the skew in order to correct.

The objective of this work is to design and to experiment in real condition a time synchronization algorithm dedicated to underwater acoustic communication able to synchronize precisely 2 underwater nodes by using decoder information such as Doppler scaling estimation and time stamping.

We take as basis the DA-Sync time synchronization protocol [11] designed for mobile wireless sensor network. A first contribution of this paper consists of a study on the impact of Doppler scale estimation and time stamping to DA-sync protocol. The second contribution of the paper lies in the experimentation of the protocol in Mediterranean sea and comparison with simulation results. Finally we propose algorithm optimization to support real scenario, more precisely, we show how to use frame time stamping to compute time offset between clocks and discuss how to include it in DA-Sync protocol.

The rest of this article is divided as follows. In section II we describe the OFDM acoustic modulation and the frame time stamp procedure. Then we study three different Doppler scale factor estimation algorithms in section III, which is the link between acoustic communication and time synchronization, since is a key point for both: recover message enclosed in the OFDM signal, and correct clock skew in the synchronization layer. We also present simulation results of the different Doppler scale factor estimation approaches in III-D. In section IV, is described the algorithm to synchronize an slave clock with a reference one. This section is divided in several subsections where is explained step by step synchronization procedure. Finally in section V we discuss our results and the maximum time accuracy obtained using the best algorithm for Doppler correction and frame time stamping. This results are obtained from real tests in laboratory and also Mediterranean sea, by using the expandable seafloor observatory OBSEA, placed 4 km offshore at Vilanova i la Geltrú, Spain [14].

II. OFDM UNDERWATER ACOUSTIC LINK

To implement a message exchange based time synchronization algorithm, it is necessary to send time information from master to slave and inversely. This way, the slave node is capable to follow same time base than the master clock by running a synchronization algorithm.

In our study we work with an acoustic communication, due to its good performance versus other communication systems, as introduced above. Recently, a worldwide convergence has occurred for the use of Orthogonal Frequency Division Multiplexing (OFDM) as an emerging technology for underwater acoustic communication since is suited for frequency selective channels and high data rates [3]. OFDM is a particular form of Multi-carrier transmission which can be easily modulated by Fast Fourier Transform (FFT) processing.

We use a convolutional cyclic prefixed OFDM (CP-OFDM) communication, where data symbols are transmitted in blocks, with consecutive blocks separated by a cyclic prefix containing last samples of the end of each symbol. To avoid interblock interference (IBI), CP must be no shorter than the channel impulse response (CIR) [15]. Without loss of generality, our study can be reproduced with a zero-padded OFDM (ZP-OFDM) waveform for power saving purpose [16].

The OFDM waveform is centered around f_0, duration of each OFDM data symbol is noted T_{ofdm} and can be decomposed into an useful part of length $T_u = N_{FFT}T_{sym}$ and CP part of duration $T_{CP} = L_{CP}T_{sym}$. In each OFDM symbol, $K \leq N_{FFT}$ active carriers are modulated by using a Quadrature phase shift keying (QPSK) constellation. The inter-carrier spacing is computed as $\Delta f = \frac{1}{T_u}$ and signal bandwidth is $B_{OFDM} = \frac{K}{T_u}$. The fraction of CP is set according to the maximum delay spread of the underwater acoustic channel [17]. Let's note d_k

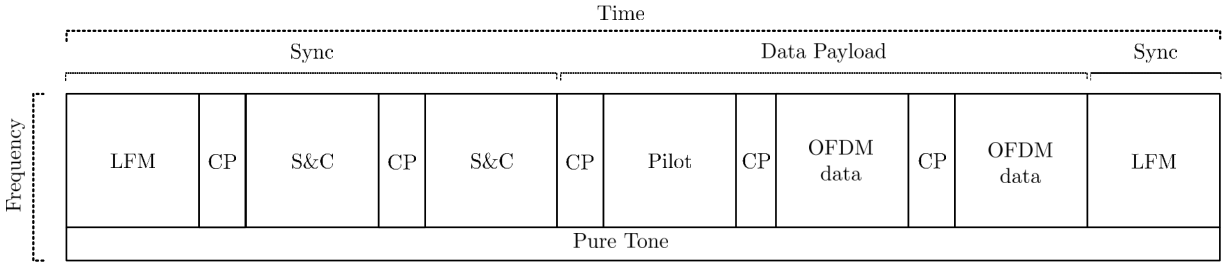

Fig. 1. Overall structure of the transmitted signal.

with $k \in [1, K]$ the QPSK data cells, the time domain modulated OFDM signal can be expressed as:

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

With $f_k = f_0 + \frac{2k-K}{2T_u}$

The overall structure of the frame is depicted in figure 1. With a duration of 2.7 seconds, 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} , angular carrier frequency f_0, and duration T_{LFM}, the LFM signal is modeled as:

$$x_{LFM}(t) = \sin\left(2\pi f_0 t + \frac{B_{LFM}}{2T_{LFM}} t^2\right) \tag{2}$$

The frame comprises also 2 OFDM preambles used for Schmidl and Cox (S&C) synchronization algorithm (see III-B). Finally the frame includes one pilot OFDM symbol used for channel estimation and followed by 2 OFDM data symbols carrying the useful message information.

Outside the useful spectrum we transmit also a continuous pure tone signal centered at f_{PT} used for frequency shifting detection. Signals are I/O with sampling frequency $f_s = 1/T_s$ through a data acquisition system connected to a computer running $Matlab^{\circledR}$.

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

Let note $\tilde{x}_{ab}(t)$ the pass-band signal described in figure 1 transmitted from node A to node B. The received pass-band signal noted $\tilde{y}_{AB}(t)$ will be affected by the UWA channel as follows :

$$\tilde{y}_{AB}(t) = \sum_{p=1}^{N_p} A_p \tilde{x}_{AB}(t - \tau_p(t)) + \tilde{\omega}(t) \tag{3}$$

Where the channel is represented by a time-varying N_p-tap linear filter with amplitude variation A_p (assumed to be constant during the frame) and varying delay $\tau_p(t)$, plus medium noise $\tilde{\omega}(t)$. The non-uniform Doppler scaling effect is induced by two factors : doppler velocity of the medium and clock drift. Both of them affect in the same way the received signal used for synchronization [18].

Let note a as the combined Doppler scaling factor formed by clock drift θ and node movement a_m. By assuming that Doppler effect is the same on each path, the time varying delay of path p may be expressed as $\tau_p(t) = \tau_p - at$ where τ_p represents the initial delay. (3) may be rewritten as [11]:

$$\tilde{y}_{AB} = \sum_{p=1}^{N_p} A_p \tilde{x}_{AB}\big((1 + a_{AB})t - \tau_p\big) + \tilde{\omega}(t) \tag{4}$$

The motion-induced Doppler scale is the same in both way transmission and is expressed as function of the relative motion between nodes v_r and wave celerity c_w as follows : $a_m = -\frac{v_r}{c_w}$

Where a_{AB} is the combined Doppler factor going from node A to node B and vice versa. If A is the master clock and B the slave one, the combined Doppler factor can be expressed as:

$$1 + a_{AB} = \theta(1 + a_m) \tag{5}$$

$$1 + a_{BA} = \frac{(1 + a_m)}{\theta} \tag{6}$$

One can easily show that the non uniform Doppler scale provides a frequency shift equal to $f_d = af_0$ into the base band received signal. Therefore, Doppler scaling yields to a sampling rate deviation from the expected one in the received signal. So, if we are capable of estimating, both together, drift and Doppler shifting due to movement, it would be possible to correct $\tilde{y}_{AB}(t)$ from f_s to a new base time f'_s without Doppler scale :

$$f'_s = \frac{1}{1 + a_{AB}} f_s \tag{7}$$

Since Doppler effect correction is performed in pass-band, phase correction is not necessary as it would be in base-band.

Typical values of the communication system are provided in Table I

III. Doppler scale factor estimation

This section describes how Doppler scaling can be estimated. Since we are using a frequency modulated communication, Doppler scaling will affect both communication metrics and time stamping accuracy. In time synchronization these two factors are what determine its performance.

Communication metrics are deteriorated by the fact that a frequency shifting produce a phase shifting in our QPSK constellation, what in some cases, depending on the FEC algorithm, can result in a Bit loss, then without global time information enclosed in a sync message, there is no way to get a high synchronization protocol.

Something similar happens with time stamping algorithms. Both LFM cross-correlation detection and S&C besides CIR equalization, are highly dependent of frequency or time base, then a Doppler scaling will modify its detection and consequently the synchronization protocol accuracy.

A. Pure Tone Doppler scale

This method is also used and introduced in a single carrier modulation communication [19].

This approach consists of sending a pure tone (PT) signal outside the useful spectrum and studying its phase variation at the reception side in order estimate the Doppler shift. Since the pilot tone is narrow-band by construction, it will not be affected by multipath effect but only by phase rotation due to Doppler effect. Let assume that the PT signal is centered around f_{pt}. At the reception side, after base band conversion and narrow-band filtering around f_{pt}, the phase of PT base band signal $y_{pt}[n]$ can be approximated to:

$$\arg y_{pt}[n] = 2\pi f_d \frac{f_{pt}}{f_0} n/f_s \tag{8}$$

By computing the angular frequency of (8), we obtain the Doppler scaling applied to the Pure Tone [19], thus an estimation of the Doppler scale applied to the useful signal can be expressed as:

$$\hat{f}_d = \frac{f_s}{2\pi}\frac{f_0}{f_{pt}}\arg\left(y_{pt}[n]y^*_{pt}[n-1]\right) \tag{9}$$

This algorithms allow communication system to be independent of Doppler scaling procedure, due to they can be centered at different frequencies and do not interfere between them. In this study we use a OFDM communication band centered at 30 kHz with 1.3 kHz Bandwidth, and a pure tone centered at 20 kHz for Doppler scale study, then while we are transmitting instrument information and synchronization framing, we are capable to add another tone without spreading communication time, and neither worsening OFDM robustness.

When separating both signals, OFDM and pure tone, is necessary to use a sharp filter for avoiding interferences, but taking into account to do not cut frequency shifting on the sides of the pure tone.

B. Schmidl & Cox

Scmidl & Cox approach [20] uses two OFDM preambles in order to detect Carrier Frequency Offset (CFO). First Preamble is used for frame detection besides a fine CFO, and the second one is used for a coarse frequency offset estimation. So Doppler scaling will be computed in two stages.

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

$$\phi = \pi T_{sym}\Delta f \tag{10}$$

Which can be estimated as the angle resulting from the partial correlation between the two halves near the best timing point [20].

Given the first preamble $P(d)$:

$$P(d) = \sum_{m=0}^{L-1} r^*_{d+m} r_{d+m+L} \tag{11}$$

Note that d is a time index corresponding to the first sample in a window of $2L$ samples. Phase variations are estimated as:

$$\hat{\phi} = \arg\left(P(d)\right) \tag{12}$$

The actual frequency shifting is:

$$\hat{f}_d = \frac{\hat{\phi}}{\pi T_{sym}} + \frac{2\hat{g}}{T_{sym}} \tag{13}$$

Where g is an integer. The second training symbol contains a Pseudo-Noise (PN) sequence on the odd (x_1) frequencies to measure these sub-channels, and another PN sequence on the even frequencies (x_2) to help determine frequency offset.

After computing the relation between odd frequencies and even ones at the transmitter side, we will obtain a conversion factor between pairs of frequencies, then at the receiver side, we can use this factor to estimate $\hat{x}_2$ from x_{m1} and vice versa. Then by computing the correlation between x_2 and $\hat{x}_2$, it will be possible to maximize the $\hat{g}$ factor which leads to a better correlation, resulting this way in a coarse CFO estimation.

$$B(g) = \frac{\left|\sum_{k\epsilon x} x^*_{1,k+2g} v^*_k x_{2,k+2g}\right|^2}{2(\sum_{k\epsilon x} |x_{2,k}|^2)^2} \tag{14}$$

C. Preamble/Postamble

As mentioned before, Doppler Scale (DS) performs similar to interpolating a signal to 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.

So in this method, we add an LFM preamble and postamble as introduced in [22] to detect time compression or expansion between them, then it is extrapolated to frequency shifting. Main reason for using an LFM instead of another signal is its good cross correlation performance in environments with white Gaussian noise, so that we will use a correlation in order to detect the exact samples where LFM preamble and LFM postamble arrive [21] [22].

In Figure 2 is computed time difference between the two correlation peaks given by the preamble and the postamble. Then by knowing the ideal separation between these two peaks, in a zero Doppler scale environment, is possible to estimate the time compression/expansion:

$$\hat{f}_d = f_0 \left(1 - \frac{t_{DS}}{t_{ideal}}\right) \tag{15}$$

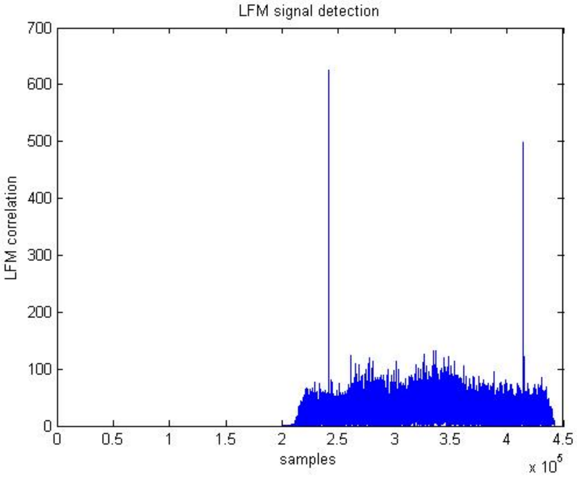

Fig. 2. Signal $|R_{xy}|$ for LFM detection

Where f_0 stands for the acoustic communication center frequency (pass-band frequency), t_{DS} is the time compressed between two LFM correlation detections when we have Doppler scaling, and t_{ideal} concerns to the same time elapse between two detections without Doppler shifting.

If center of gravity is performed, DS estimation is improved significantly, due to we avoid sampling frequency limitation when we need to compute the difference in samples between the two peaks.

D. Simulation

By using these three methods we analyze the accuracy on DS detection, and how does it affect frame correction and synchronization. First, we compute an SNR sweep 1000 times averaging the results at each SNR for avoiding aberrant errors in our results. Only one frame has been used, so all the approaches have the same Doppler shifting, noise and multipath, $f_d = 20$ Hz at $f_0 = 30$ kHz, figure 3 displays the Root Mean Squared Error of Doppler scale estimation defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((f_{di} - \hat{f}_{di})^2\right)} \tag{16}$$

Compared to a mean absolute error, RMSE amplifies large errors.

TABLE I

COMMUNICATION PARAMETERS SUMMARY [TYP. 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	$NFFT$	512	128
Active carriers	K	460	114
Signal to Noise Ratio	SNR	15 dB	15 dB
Doppler frequency	f_d	20 Hz	20 Hz

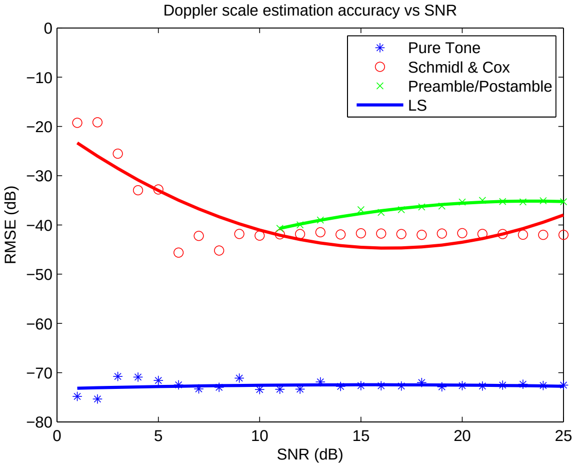

Fig. 3. Doppler scale estimation simulation $f_d = 20$ Hz at $f_0 = 30$ kHz with SNR sweep

Then, in figure 4 is repeated same simulation computing RMSE of Doppler scale estimation, but this time we keep a constant SNR of 15 dB 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 in order to make results more readable, it is just an approximation, which in some cases

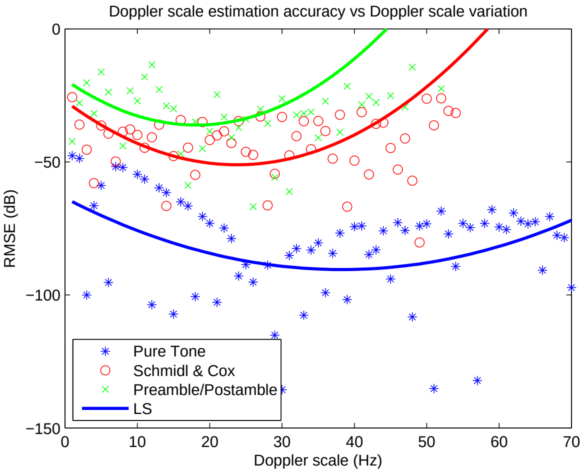

Fig. 4. Doppler scale estimation simulation vs. $\mathrm{SNR} = 15$ dB at $f_0 = 30$ kHz with f_d sweep

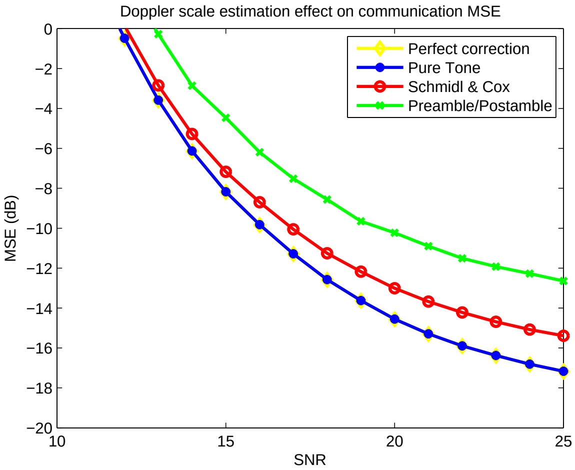

Fig. 5. Frame MSE after Doppler scale compensation on an SNR sweep with $f_d = 20$ Hz in simulation

differ 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. Figure 5 displays frame Mean Square Error (MSE) after applying Doppler scale compensation and channel equalization with each described algorithm. The curve labeled perfect correction provide a lower bound on MSE performance where the Doppler scale is perfectly known and removed.

Finally, we run a simulation where we keep a constant 15 dB SNR, and we perform a Doppler

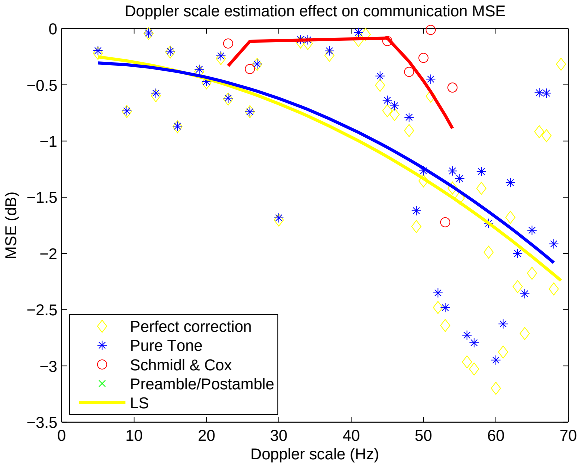

Fig. 6. Frame MSE after Doppler scale compensation on a Doppler scale frequency sweep with $\mathrm{SNR} = 15$ dB in simulation scale sweep.

This simulation gives an idea of which algorithm will have better performance in an ideal scenario. 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 increase 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. For doing so, we use a widely known scheme in time synchronization approaches [1] described in figure 8.

This bidirectional message exchange encloses data transmission and reception times, besides frame relative velocities. This information is used by the slave clock estimating two key points in time synchronization: clock offset and clock skew. Figure 7 represents all synchronization information sources that will be described in this chapter.

The offset is the difference between an slave clock and a master one, this can be estimated by computing propagation times to compensate the time stamps delay due to the message exchange. With this, it is possible to compensate clock error and set the slave clock to the same time than the master one.

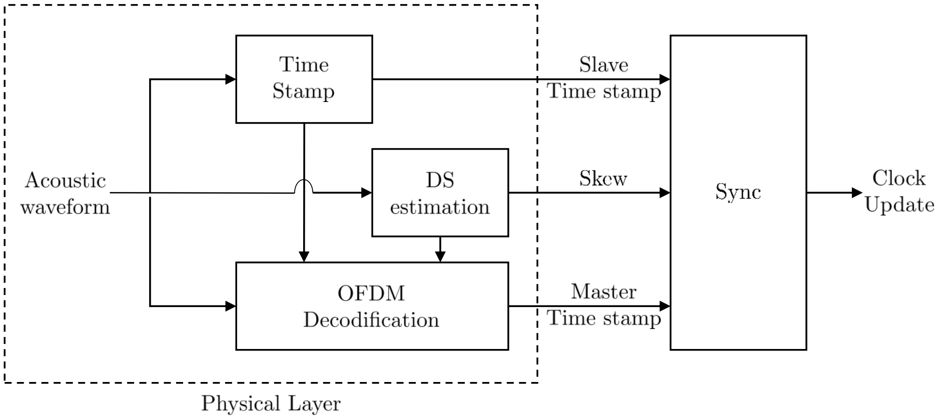

Fig. 7. Time synchronization schema

The second part, regarding clock skew, must be also taken into account to set both clocks running with the same frequency. If this clock skew, is not compensated the slave clock will suffer a clock drift relative to the master clock, which will not be compensated until next clock offset estimation.

A. Clock offset estimation

The slave node records the sending time stamp T_1, obtained at the MAC layer [23], right before the message leaves. Upon receiving the message , the reference node also time stamp T_2 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.

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

Message exchange process is shown in Figure 8

$$\hat{\beta} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \tag{17}$$

$$\hat{\tau} = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \tag{18}$$

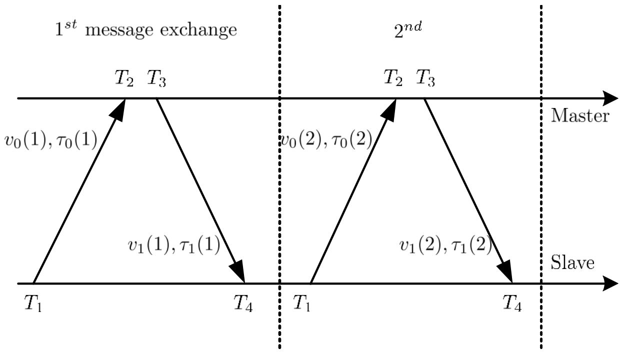

Fig. 8. Synchronization schema.

B. Clock skew estimation

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 (5) and (6), 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 kinematic model of the nodes and a linearization of several computations, can reach a better performance on this estimation.

C. Data collection

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

For doing so, a bidirectional message exchange between nodes is used, as it has been presented above.

D. Velocity estimation refinement

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

Then by using kinematic equations is possible to refine the initial velocity estimation obtained from 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

$$x(k+1) = F(k)x(k) + \Gamma(k)w(k) \tag{19}$$

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

$$x(k+1) = [v(k+1) \quad \alpha(k+1)]^T \tag{20}$$

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

$$\Delta T(k) = \frac{\tau_0[k] + \tau_1[k]}{2} + T_3[k] - T_2[k] \tag{21}$$

The estimation of the state (k+1) based on the measurement of v_0 and v_1 can be obtained with Kalman filter, which is an optimal Minimum Mean Square Error (MMSE) state estimator under the Gaussian assumption of both the process noise and measurement noise, figures 9 and 10 display the estimation of velocity and acceleration refinement by using Kalman filtering in first kinematic model equations.

Regarding the kinematic model for the relative velocity we define:

$$v = k_1 t + k_2 \tag{22}$$

Where k_1 is a parameter closely related to 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 message exchange [1]:

$$\hat{v}(1) = v_0[1] \tag{23}$$

$$\hat{\alpha}(1) = \frac{v_1[1] - v_0[1]}{\Delta T[1]} \tag{24}$$

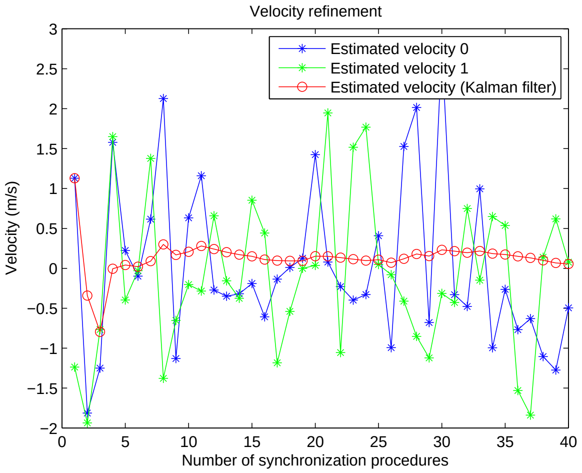

Fig. 9. Kalman filter for velocity refinement.

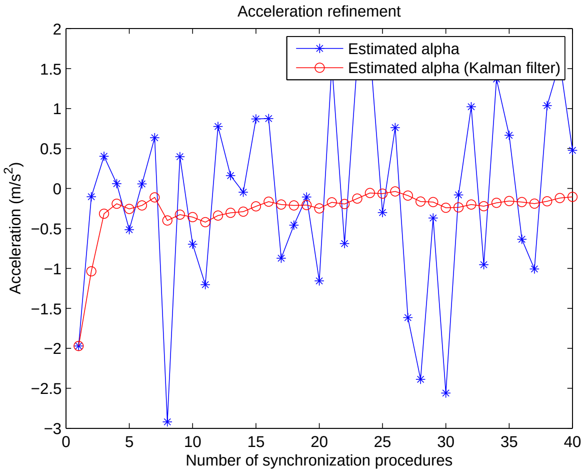

Fig. 10. Kalman filter for acceleration refinement.

E. Propagation time estimation

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

In our synchronization schema we have to handle with τ_1 and τ_2. Then these propagation delays will be used to compute Weighted Least-Squares Estimation (WLSE) what will result in an skew estimation by computing linear regression slope.

In order to compute propagation delays, we use velocity and acceleration obtained in sections III and IV-D. Since acoustic waves propagation time is not linked with individual moving pattern

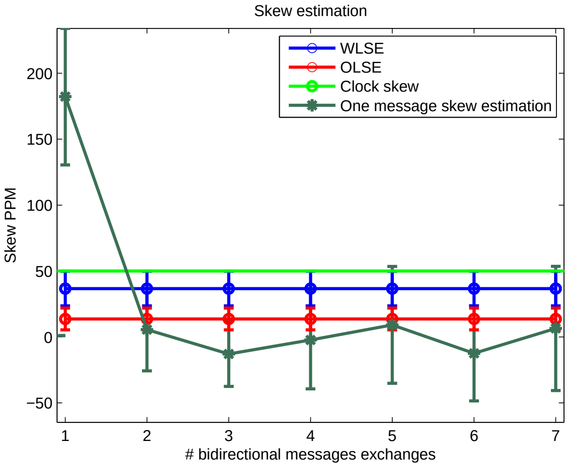

Fig. 11. Simulation of clock skew estimation.

of any sensor, but a relative movement between two nodes, we can work directly with relative velocities obtained from physical layer, instead of using a three dimensional velocity schema.

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

$$\begin{cases} \tau_1 & = & \frac{(T_4-T_1)c_w-\theta(T_3-T_2)(c_w+\hat{v}_0)-\frac{1}{2}\hat{\alpha}(T_3-T_2)^2}{2\theta c_w} \\ \\ \tau_2 & = & \frac{(T_4-T_1)c_w+\theta(T_3-T_2)(\hat{v}_0-c_w)+\frac{1}{2}\hat{\alpha}(T_3-T_2)^2}{2\theta c_w} \end{cases} \tag{25}$$

In (25), θ is needed to estimate propagation times, since in this algorithm step this value is not known yet, we will set it as '1', what will be corrected in calibration procedure.

F. Linear regression

In this section the clock skew (θ) is estimated by using previous time stamps and propagation time computation (26).

Parameter β and θ are obtained by linear regression:

$$T_2[k] = \theta(T_1[k] + \tau_1[k]) + \beta \tag{26}$$

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

With linear regression we obtain the offset from zero to 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 regression line is defined:

$$y = \theta x + \beta \tag{27}$$

In figure 11 is plotted the accuracy of skew estimation by using three different approaches: 1) Direct skew estimation at each message exchange; 2) Ordinary Least Squares Estimation (OLSE) [24]; 3) WLSE [25]. This test has been run 10 times for observing the variance of the estimations, then we can ensure a better performance of the WLSE for any scenario.

G. Calibration

For achieving this skew estimation, we have assumed in previous section $\theta = 1$ in order to estimate propagation times. So now we will need to repeat propagation delay estimation and linear regression with actual estimated skew values. This procedure will be repeated up to 10 times or θ stabilization, as described in Figure 12.

TABLE II

TIME SYNCHRONIZATION PARAMETERS SUMMARY [TYP. VALUES]

Description	Parameter	Simulation	Laboratory	OBSEA
Signal to Noise Ratio	SNR	15 dB	15 dB	15 dB
Doppler frequency	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

H. Simulation

After running DA-Sync algorithm with our own frame detection procedure and Doppler Scale factor estimation by using Linear Frequency Modulation and Pure tone method respectively, we

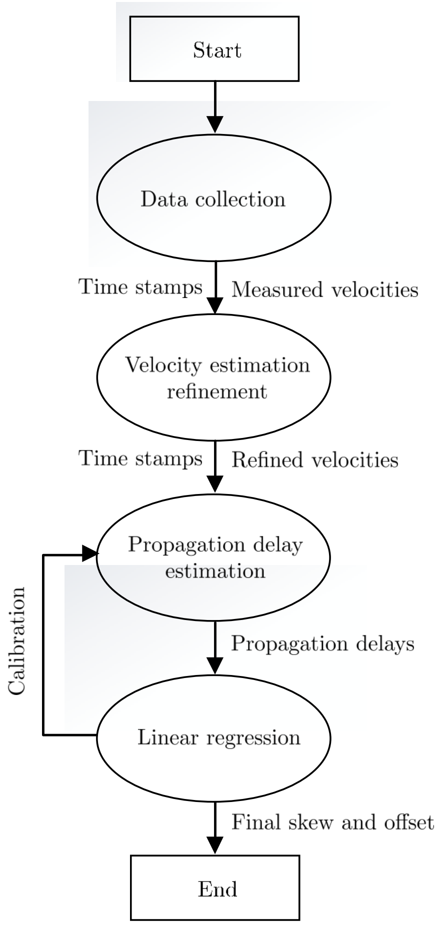

Fig. 12. Calibration procedure.

obtain following synchronization results, after 10 seconds since last synchronization procedure. Figure 13.

We set the inherent skew of the ordinary clock at 50 PPM, clock offset is initialized as 800 μs. Response time is fixed as 1 s, and the propagation speed (c_w) is 1500 m/s.

As can be observed in figure 13, after 8 messages exchanged an accuracy below 400 μs can be reached, going below of 200 μs after 22 messages exchanged due to Kalman filter predictor enhancement with the increment of initialization points.

If we analyze our system behavior after 8 messages exchanges, we can compute time offset estimation error just after a synchronization procedure and compare its performance by following figure 12 refinement, or directly compute offset without any compensation. This statement is shown in figure .

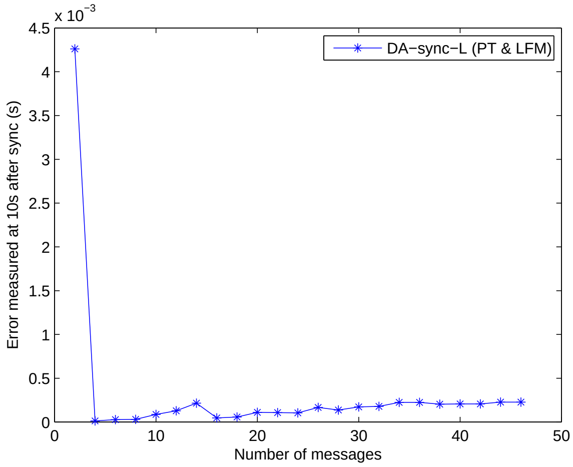

Fig. 13. Simulation of clock offset error after 10 s versus number of messages.

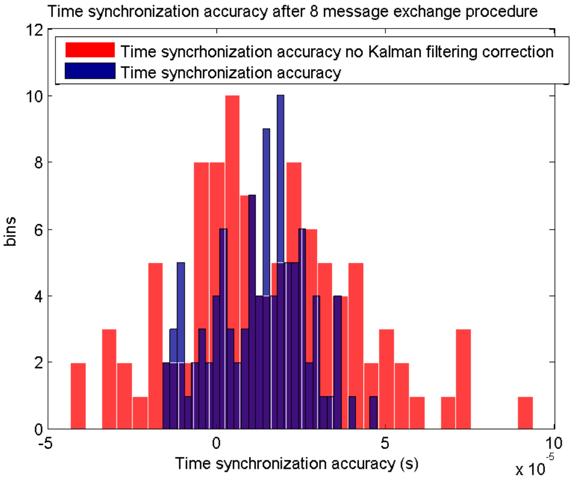

Fig. 14. Simulation of time synchronization accuracy after 8 message exchange procedure repeated 100 times.

By repeating 100 times this synchronization algorithm (made up 8 messages exchange), we can observe how Kalman filtering applied in form of correction for a weighted linear regression, is close to double synchronization accuracy versus not applying any correction. Figure 14 shows the mean and the variation of the offset estimation after applying skew compensation, in an 8 message exchange algorithm repeated 100 times.

On the other hand, in figure 15, we have the same simulation histogram but this time without skew correction. We will have an offset correction but skew will be drifting the clock during

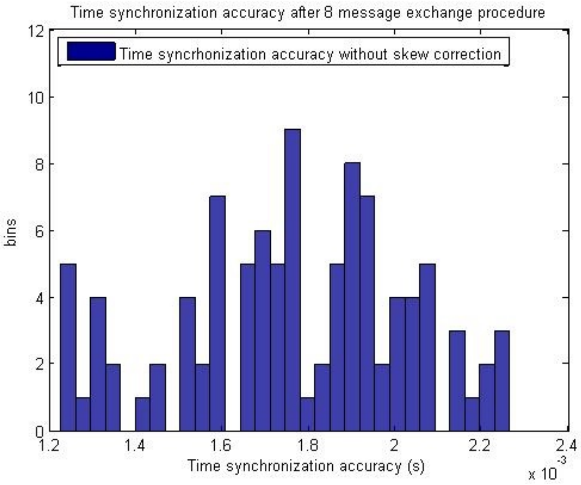

Fig. 15. Simulation of time synchronization accuracy after 8 message exchange procedure without correcting clock skew repeated 100 times.

synchronization procedure. In our study it takes $t_{sync} = 59$ seconds to exchange 8 bidirectional messages, so offset computation will be affected by the factor of θt_{sync}, as can be seen in the plot, where we have an histogram center shifting due to clock skew.

V. Results

In this section we evaluate previously simulated algorithms in real water tests, in order to verify its proper functionality in a real scenario.

A. Doppler scale factor estimation

1) Laboratory test: This test was performed in a 150 x 40 x 40 cm water test tank with a separation between hydrophones of 1 m. With this experiment we pretend to verify a proper functionality of Doppler scale estimation algorithms in a controlled environment. Where we do not have inherent sea currents, so we can simulate Doppler shifting to estimate it without any additive sea factor.

Figures 16 and 17 show same results obtained in section III-D but this time the signal is transmitted in the 'test tank' where we have a very difficult channel condition with strong multipath of about 75 ms, and all tests are only averaged 10 times due to real test's processor timing constraints. This two graphs display the performance of each one of the algorithms

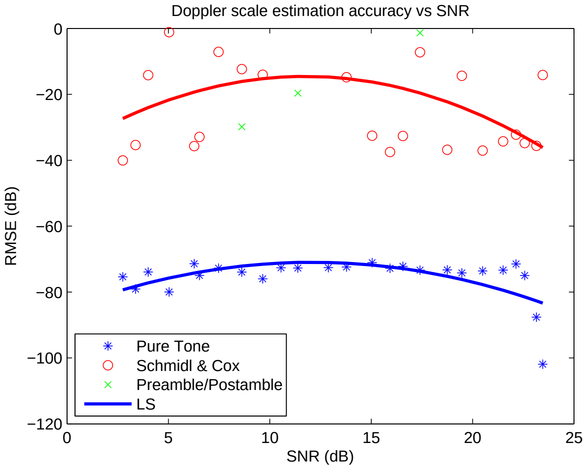

Fig. 16. Doppler scale estimation Laboratory test $f_d = 20$ Hz at $f_0 = 30$ kHz with SNR sweep

evaluated in this study for Doppler scale estimation in a real environment. In order to observe each algorithm accuracy at any scenario we perform two tests. First one is an SNR sweep which depicts RMSE between the real Doppler scale value and the estimation at different SNR values. And SNR variations were performed by signal power variations, then the SNR displayed in following figures is an estimated SNR defined as the ratio between the signal and the noise of each channel.

Then we repeat same test, but this time with a constant SNR of 15 dB and a variable Doppler scale value, from 0 to 70 Hz.

After the evaluation of each Doppler scale estimation algorithm we apply a frequency correction to the OFDM frame with each frequency shifting estimation obtained in previous stage, and we evaluate the MSE of the data frame after this correction and channel equalization, results are carried out in figures 18 and 19 respectively.

For this real test we repeat both SNR and Doppler frequency sweeps for observing the algorithms performance for any water channel status.

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 same depth at 1.5 m of separation of the transmitter. Both hydrophones where attached at 0.5 m of the seafloor. OFDM signals were transmitted while hydrophones were moving free in an horizontal axis by the sea currents.

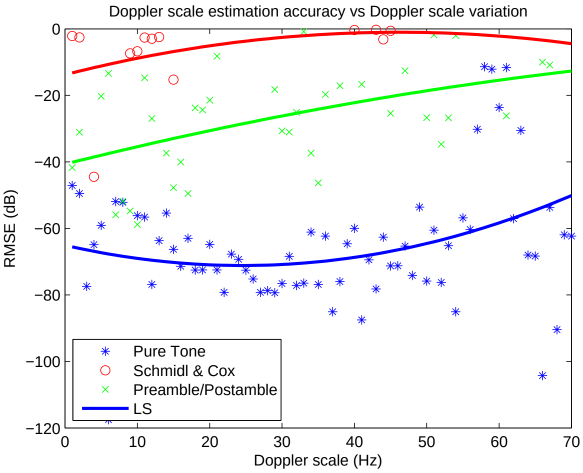

Fig. 17. Doppler scale estimation Laboratory test $\mathrm{SNR} = 15$ dB at $f_0 = 30$ kHz with f_d sweep

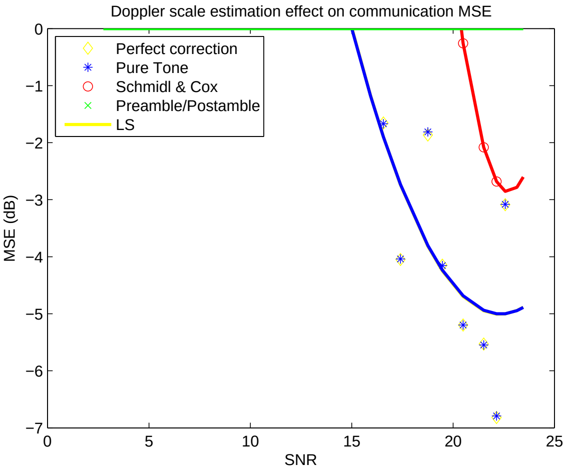

Fig. 18. Frame MSE Laboratory test after Doppler scale compensation on an SNR sweep with constant $f_d = 20$ Hz

Like it has been done in previous subsection V-A, we first evaluate Doppler scale algorithms RMSE in estimation, figure 20, and then this algorithm correction is applied to an OFDM frame as a Doppler shifting correction where we can evaluate frame MSE for each case, figure 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 to the water channel, so we are not able to increase the SNR up to the same levels shown in Laboratory tests.

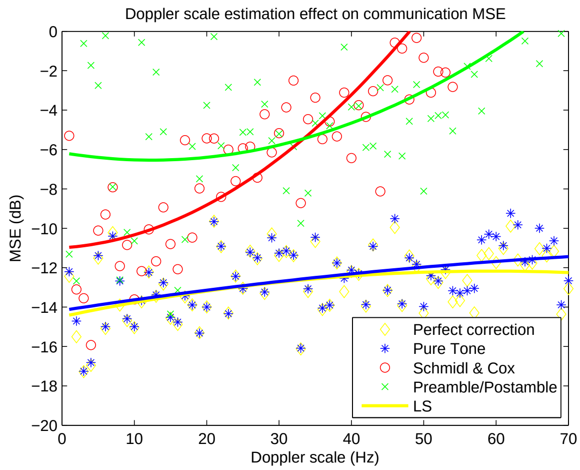

Fig. 19. Frame MSE Laboratory test after Doppler scale compensation on a Doppler scale frequency sweep with constant $\mathrm{SNR} = 15$ dB

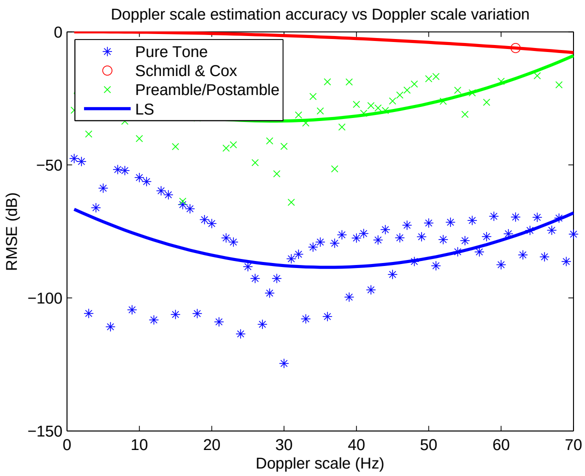

Fig. 20. Doppler scale estimation at OBSEA $SNR = 15$ dB at $f_0 = 30$ kHz with fd sweep

This test shows how Pure tone outperforms Schmidl & Cox algorithm and Preamble/Postamble LFM one, in all the possible scenarios. It is really close in terms of Doppler scale estimation and compensation to a perfect correction. So this algorithm will be the one used for the synchronization part of this study.

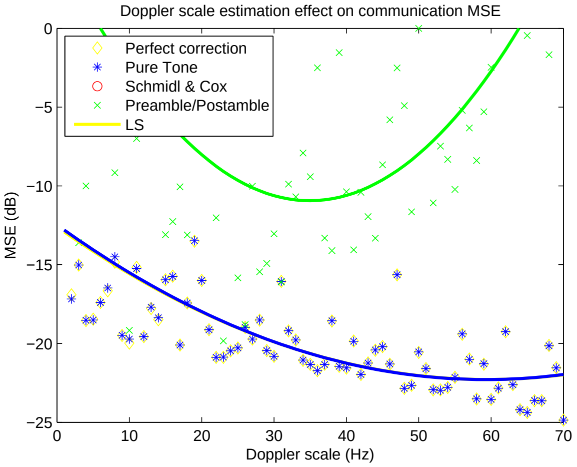

Fig. 21. Frame MSE at OBSEA after Doppler scale compensation on a Doppler scale frequency sweep with estimated $\mathrm{SNR} = 15$ dB

B. Time synchronization

In this section we repeat same simulation workbench, section IV-H, but this time we send the signal to the cRIO module [23] instead of simulating the physical layer. Then, the signal is sent through the test tank in subsection V-B1 and through a real water channel in front of OBSEA in subsection V-B2. Only the water channel is modified from simulation to these real tests, then we expect similar performance than in simulation with some shades 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 currents by adding Doppler scaling in our signal. Since we have simulated currents, they won't change during the experiment, so it results in similar time accuracy to simulation, as shown in figure 22.

2) Real test: In real tests, we should have a time synchronization evaluation close to the previous one. But in this case we do not have sea currents control, then they can vary along time, causing that the Kalman filtering and WLSE must be limited along time to avoid extrapolating Doppler scaling from different sea current sources. As can be seen in figure 22, time accuracy starts to decrease after the eighth message, what means after two minutes approximately since the beginning of the test. So for this case we can assume that sea currents vary significantly every two minutes. In a real application this time should be auto-adjusted in function of time

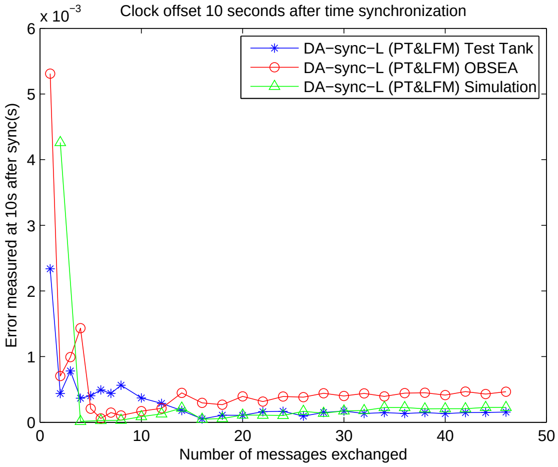

Fig. 22. Clock offset error after 10 s versus number of messages with clock skew set at 50 PPM and node movement simulation of 1 m/s.

synchronization error evolution, what is linked with the length of 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 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, a water tank and shallow-water experiments. Using a SNR of 15 dB and 20 Hz of acoustic signal frequency shifting, and this results in RMSE of Pure tone outperforming both Preamble and Postamble and Schmidl & Cox Doppler scale estimators.

Over clock skew of 50 PPM, using Pure tone algorithm for Doppler scale estimation and Linear Frequency Modulation for frame detection, since this scenario provided the best results in the presented study, time synchronization accuracy after 10 seconds was 179 μs in simulation, 170 μs in Laboratory, and 400 μs in 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 time synchronization algorithm decrease its performance with large message exchange procedures. Since water channel medium changes its current characteristics during synchronization procedure, then Kalman filtering for velocity estimation is worthless.

ACKNOWLEDGMENT

The authors would like to thank reviewers for their thoughtful comments.

This work was partially supported by the projects NeXOS and FixO3 from the European Unions Seventh Programme for research, technological development and demonstration under grants agreement No 614102 and No 312463 respectively.

NexOS (Innovative multi-functional sensors for in-situ monitoring of marine environment and related maritime activities) 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 6 European countries. Visit: www.nexosproject.eu

The FixO3 (Fixed point Open Ocean Observatory network) 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/

The authors would also like to thank National Instruments Spain for their support providing us the necessary hardware used in this study.

REFERENCES

[1] Liu, J., Wang, Z., Zuba, M., Peng, Z., Cui, J. H., & Zhou, S. (2014). DA-Sync: A Doppler-Assisted Time-Synchronization Scheme for Mobile Underwater Sensor Networks. IEEE Transactions on Mobile Computing, (3), 582-595.

[2] Ribas, J, & Stojanovic, M. (2009). Underwater wireless video transmission using acoustic ofdm (Doctoral dissertation, Masters thesis, Massachusetts Institute of Technology).

[3] Debbah, M. (2004). Short introduction to OFDM. White Paper, Mobile Communications Group, Institut Eurecom.

[4] Li, B., Zhou, S., Stojanovic, M., Freitag, L., & Willett, P. (2008). Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts. Oceanic Engineering, IEEE Journal of, 33(2), 198-209.

[5] Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. Ad hoc networks, 3(3), 257-279.

[6] Lee, K., Eidson, J. C., Weibel, H., & Mohl, D. (2005, October). Ieee 1588-standard for a precision clock synchronization protocol for networked measurement and control systems. In Conference on IEEE (Vol. 1588, p. 2).

[7] Jiang, H., Liu, X., Wang, X., Liu, W., & Wang,Y. (2009). Tri-Message: A Lightweight Time Synchronization Protocol for High Latency and Resource-Constrained Networks, IEEE International Conference on Communication

[8] Syed, A. A., & Heidemann, J. S. (2006, April). Time Synchronization for High Latency Acoustic Networks. In INFOCOM.

[9] Chirdchoo, N., Soh, W. S., & Chua, K. C. (2008, September). MU-Sync: a time synchronization protocol for underwater mobile networks. In Proceedings of the third ACM international workshop on Underwater Networks (pp. 35-42). ACM.

[10] Lu, F., Mirza, D., & Schurgers, C. (2010, September). D-sync: Doppler-based time synchronization for mobile underwater sensor networks. In Proceedings of the Fifth ACM International Workshop on UnderWater Networks (p. 3). ACM.

[11] Liu, J., Wang, Z., Peng, Z., Zuba, M., Cui, J. H., & Zhou, S. (2011, December). TSMU: A time synchronization scheme for mobile underwater sensor networks. In Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE (pp. 1-6). IEEE.

[12] Lee, K., Eidson, J. C., Weibel, H., and Mohl, D. (2005, October). IEEE 1588-Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In Conference on IEEE (Vol. 1588).

[13] Weibel, H., & Bchaz, D. (2004, September). Implementation and performance of time stamping techniques. In Proceedings of the 2004 Conference on IEEE (Vol. 1588).

[14] Expandable Seafloor Observatory OBSEA, url: 'http://www.obsea.es'. Accessed: October 2015

[15] Liu, Z., & Yang, T. C. (2014). On Overhead Reduction in Time-Reversed OFDM Underwater Acoustic Communications. Oceanic Engineering, IEEE Journal of, 39(4), 788-800.

[16] Le Nir, V., van Waterschoot, T., Duplicy, J., & Moonen, M. (2009). Blind coarse timing offset estimation for CP-OFDM and ZP-OFDM transmission over frequency selective channels. EURASIP Journal on Wireless Communications and Networking, 2009, 44.

[17] Bouvet, P. J., & Loussert, A. (2011, September). An analysis of MIMO-OFDM for shallow water acoustic communications. In OCEANS 2011 (pp. 1-5). IEEE.

[18] Stovanovic, M., & Freitag, L. Integrated doppler tracking and efficient resampling for phase coherent acoustic communication. IEEE of Oceanic Engineering.

[19] Kathiroli, P., Beaujean, P. P., & Xiros, N. (2011). Source speed estimation using a pilot tone in a high frequency acoustic modem (pp. 1-8). IEEE.

[20] Schmidl, T. M., & Cox, D. C. (1997). Robust frequency and timing synchronization for OFDM. Communications, IEEE Transactions on, 45(12), 1613-1621.

[21] Diamant, R., Feuer, A., & Lampe, L. (2012, November). Choosing the right signal: Doppler shift estimation for underwater acoustic signals. In Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems (p. 27). ACM.

[22] Sharif, B. S., Neasham, J., Hinton, O. R., & Adams, A. E. (2000). A computationally efficient Doppler compensation system for underwater acoustic communications. Oceanic Engineering, IEEE Journal of, 25(1), 52-61.

[23] Pallares, O., Bouvet, PJ., & del Rio, J. (2014). Underwater acoustic communication messaging time stamp applied to global time synchronization. Sensor Systems for a Changing Ocean (SSCO), IEEE DOI: 10.1109/SSCO.2014.7000380

[24] Hayashi, Fumio (2000). Econometrics. Princeton University Press. ISBN 0-691-01018-8.

[25] Moulton, B. R. (1986). Random group effects and the precision of regression estimates. Journal of econometrics, 32(3), 385-397.

Oriol Pallares 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. He is currently working toward the PhD degree and 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. He is a member of Open Geospacial Consortium (OGC).

Pierre-Jean Bouvet (M10) 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) in Caen, France, as baseband decoding architect for digital TV demodulator products (DVB-T/T2). Since 2009, he has been an associate professor in the department of embedded systems, acoustic and communications in the engineering college of ISEN, Brest, France. His current research interests include underwater acoustic communications, MIMO transmission, iterative reception and synchronization algorithms.

Joaquin del Rio was born in Catalonia, Spain, in 1976. He received his Bs, Ms and PhD degrees in Telecommunication Engineering and Electronic Engineering, in 1999, 2002 and 2011 respectively from Technical University of Catalonia (UPC), Spain.

Since 2001 is Professor in the Electronic Engineering Department of the Universitat Politècnica de Catalunya, Barcelona, Spain.

Member of the research group Remote acquisition systems and data processing (SARTI), focus his research about Electronic Instrumentation, Interoperability in Marine Sensor Networks and Wireless Sensor Networks.

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

He is involved in projects with the industry and funded public research projects and is a National Instruments Certified Instructor for teaching official LabVIEW courses.