Signal (Stream) synchronization with *White noise* sources, in biomedical applications

P. Vaz *, V. Almeida, L.R. Ferreira, C. Correia, J. Cardoso

Department of Physics, University of Coimbra, Rua Larga, 3004-516 Coimbra, Portugal

A B S T R A C T

When multiple acquisition systems are used to simultaneously acquire signals, synchronization issues may arise potentially causing errors in the determination of acquisition starting points and continuous clock offsets and shifts on each device. This paper introduces a processing method to efficiently synchronize these signals in the presence of white noise sources without the requirement of clock sharing or any other digital line exchange. The use of a signal source, such as white noise with a very wide frequency band, is of great interest for synchronization purposes, due to its aperiodic nature. This high bandwidth signal is simultaneously acquired by all the acquisition channels, on distinct systems, and, synchronized afterwards using cross-correlation methods. Two different correlation methods were tested; a global method, used when clock system frequencies are exactly known, and a local method, used when independent clocks evidence shifts over time that cumulatively account for long term acquisition errors in the synchronization process. In a computational simulation with known clock frequencies the results show a synchronization error of $\approx 1/10$ of the time resolution, for both methods. For unknown clock frequencies, the global method achieved an error of 24/10 the time resolution, indicating a much poorer performance. In the experimental set-up, only the local method was tested. The best result shows a synchronization error of 4/10 of the time resolution. All the signal conditioning and acquisition parameters were chosen taking into account potential biomedical applications.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The human body is a highly complex system which is composed by many subsystems with specific functions. The assessment of human health condition is made by visual observation or biomedical signals acquisition. These signals can be composed by multiple variables (different physical quantities) which are measured at different body locations [1]. Some examples of biomedical signals are: body temperature; bio-electric potentials; arterial blood pressure (ABP); or respiratory rate [2].

The assessment of multiple biomedical signals is a common medical situation that produces valuable information to improve clinical diagnosis. Usually, biomedical signals are acquired by different devices, each one with its own acquisition set-up. Different equipments have different sampling frequencies, resolutions, ranges, signal-to-noise-ratios (SNR) and temporal references [3].

The absence of a common temporal reference between signals occurs when multiple signals are acquired from the same patient. This situation makes the combined analysis of these signals extremely difficult. A common situation in biomedical applications is the use of different acquisition modules, from different manufacturers, each one with a specific internal clock source. The inclusion of an external clock is often very difficult or impossible. Even if the clocks are equal, an offset in the starting points of the acquisitions appears.

To overtake these issues, it is mandatory to have a time reference, of some sort, in order to guarantee the synchronization of independent data streams. This work proposes a scheme where an electronic circuitry (*white noise* generator) is used to automatically synchronize data signals acquired from different devices and setups.

This technique is based on the acquisition of a random signal (*white noise*) by all the data acquisition systems (DAS) during their normal operation. The biomedical signal must be internally synchronized with the random signal by each device. Thereafter, these random signals are processed using cross-correlation methods to determine the delay between each physiological signal and to build a common temporal reference. Cross-correlation is a widely used method for delay determination in random shifted signals [4–6], including for biomedical purposes [7].

* Corresponding author. Tel.: +351 239410109.
 E-mail addresses: pvaz@lei.fis.uc.pt (P. Vaz), jmrcardoso@uc.pt (J. Cardoso).
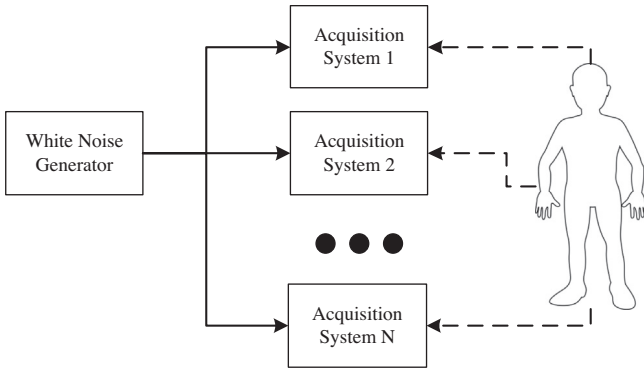
Fig. 1. Block diagram for hardware connections.

2. Methods

2.1. General approach

The proposed methodology requires an external hardware implementation that generates the synchronization signal (*white noise* generator). This signal is digitized by all the DAS (Fig. 1). At this point, one should say that this requirement is a drawback because the majority of the biomedical devices, used in clinical practice, do not allow either hardware or firmware changes. Nevertheless, in research areas that use laboratory-phase equipment, these changes are easily applied.

White noise is a random nature process with constant spectral density (Eq. (1)), *i.e.*, all the frequencies presented in the spectrum have the same power. Moreover, it presents a Gaussian probability density function (PDF) given by Eq. (2) [8].

$$|H(jw)|^2 = \text{constant} \tag{1}$$

$$p(x) = \frac{e^{-((x-\mu)^2/2\sigma^2)}}{\sigma\sqrt{2\pi}} \tag{2}$$

where $p(x)$ is the PDF, μ is the mean of the distribution and σ is the standard deviation. In the frequency domain, the *white noise* is characterized by a constant distribution, yielding a theoretical infinite bandwidth (Eq. (1)), which means that two adjacent samples are completely independent. The randomness of *white noise* makes it aperiodic, in other words, a signal with infinite period.

The autocorrelation of a periodic function ($P \in \Re$) (Eq. (3)) is itself periodic with the same period.

$$\nexists P \in \Re : f(x+P) = f(x) \tag{3}$$

A correlation is performed between two synchronization signals acquired by different DAS and delayed by a certain number of samples. The autocorrelation becomes a cross-correlation but the considerations about periodic and aperiodic signals are still valid [9,10].

To our application, the correlation can be written as Eq. (4) where the correlated signals are the same ($x[n]$) but with one of them delayed by a random number of samples (δ).

$$c_{xx}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x[n+\delta+k] \tag{4}$$

where x is the synchronization signal and N represents the number of samples of the largest signal. The result from the cross-correlation is a measure of the similarity of both signals as function of the delay k.

2.2. Computational simulation

A simulation was conducted to prove the concept of *white noise* synchronization using cross-correlation. This simulation started with the generation of a random signal with a Gaussian PDF (S_{noise}), with a large sampling frequency ($f_s^0 = 1$ MHz) and 20 s length. Fig. 2 (left) shows the histogram of the S_{noise}, which is similar to a Gaussian function with a zero mean and a standard deviation equals to one. The spectrum is presented in Fig. 2 (right). Although with some fluctuations, the spectrum is constant for all the frequencies. These fluctuations occur due to the existence of a finite number of samples.

To prevent the cross-correlation to be periodic, a signal with infinite bandwidth is the perfect solution. However, the complete independence of two consecutive samples in S_{noise} cause problems after the digitalization. Fig. 3 shows two signals (S_1 and S_2), that result from the downsampling of the S_{noise}, with the same frequency and delayed by only one sample. Signals are completely different from each other.

Moreover, by the Nyquist theorem, when the signal is digitized at a specific sampling frequency (f_s), the resulting bandwidth is restricted in the range [0, $f_s/2$]. Aliasing occurs when the original data has frequencies above this limit distorting the digitized data. These limitations can be bypassed with the use of a low-pass filter.

A low-pass filter limits the bandwidth of the synchronization signal and causes samples to be dependent on past values. The cut-off frequency selection is the key point in the design of the method. To choose a correct frequency it is necessary to take into account the sampling frequencies used by all the DAS. A high cut-off frequency will lead to aliasing but a low cut-off frequency can compromise the correlation and cause a periodic output where it is difficult to compute the real delay between signals.

The selection of the digitalization sampling rate depends on the acquired physiological signals. For example, ECG sampling rate vary from 125 Hz to 1 kHz [11,12], pulse oximetry from 125 Hz to 1 kHz [13–15] and pulse waveform from 1 kHz to 20 kHz [16,17]. Only in special cases higher frequency rates are utilized [18].

According to that information, a minimum sampling rate of 2000 samples/s was considered, leading to an anti-aliasing second order (Butterworth) low-pass filter with a cut-off frequency of 1 kHz. This frequency ensures enough precision for biomedical applications and prevent the occurrence of a periodic signal. The filter transfer function is defined as:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (\omega/\omega_c)^4}} \tag{5}$$

where ω_c is the cut-off angular frequency in rad/s.

Fig. 4 shows the frequency response of the applied filter (red). This response is characteristic of a second order low-pass filter with a roll-off of 40 dB/decade. The spectrum signal (blue) shows the filtered *white noise* signal spectrum. This new signal would be denominated S_{rand}. The amplitude of the spectrum decreases above 1 kHz but high frequencies are not completely extinguished.

Low order filters (first and second) are suitable for this application since the existence of some high frequencies is not a critical issue and the selection of a cut-off frequency of 1 kHz ensure sufficient precision for biomedical applications.

2.2.1. Global method

The S_{noise} was intentionally created with a large number of samples, to allowing for a simulation of the acquisition process. This simulation requires a downsampling of the signal with a fixed sampling rate and delay. Two different DAS have been simulated with sampling frequencies of 20 kHz and 2000 Hz and with a random delay between the starting points. No external noise (*e.g.*
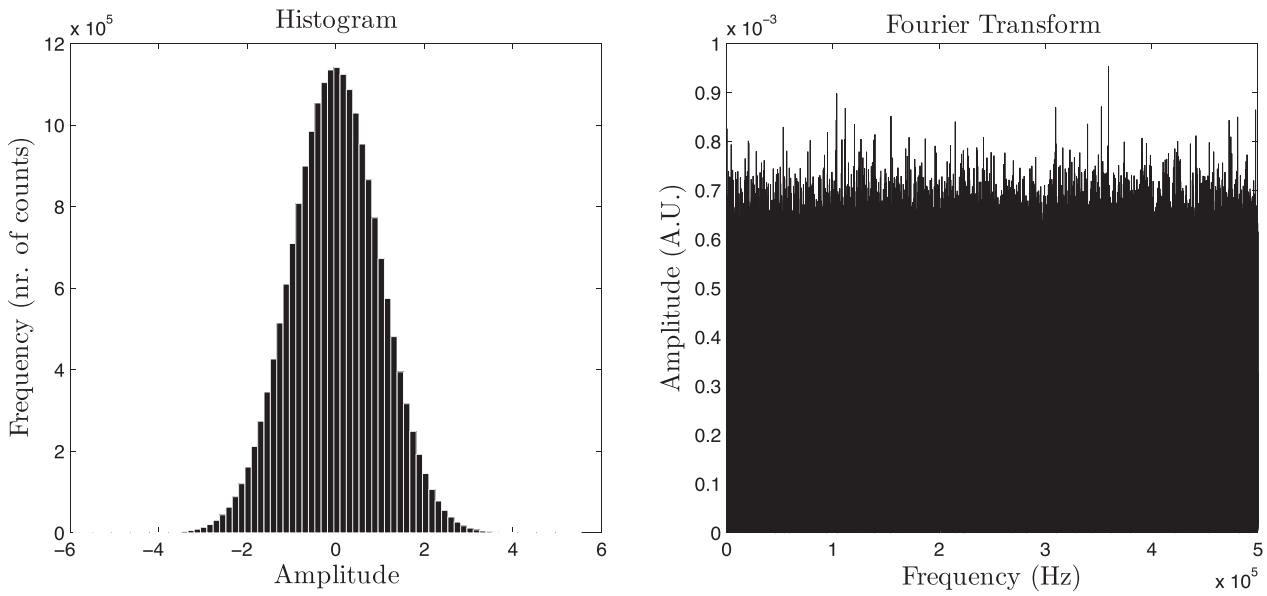
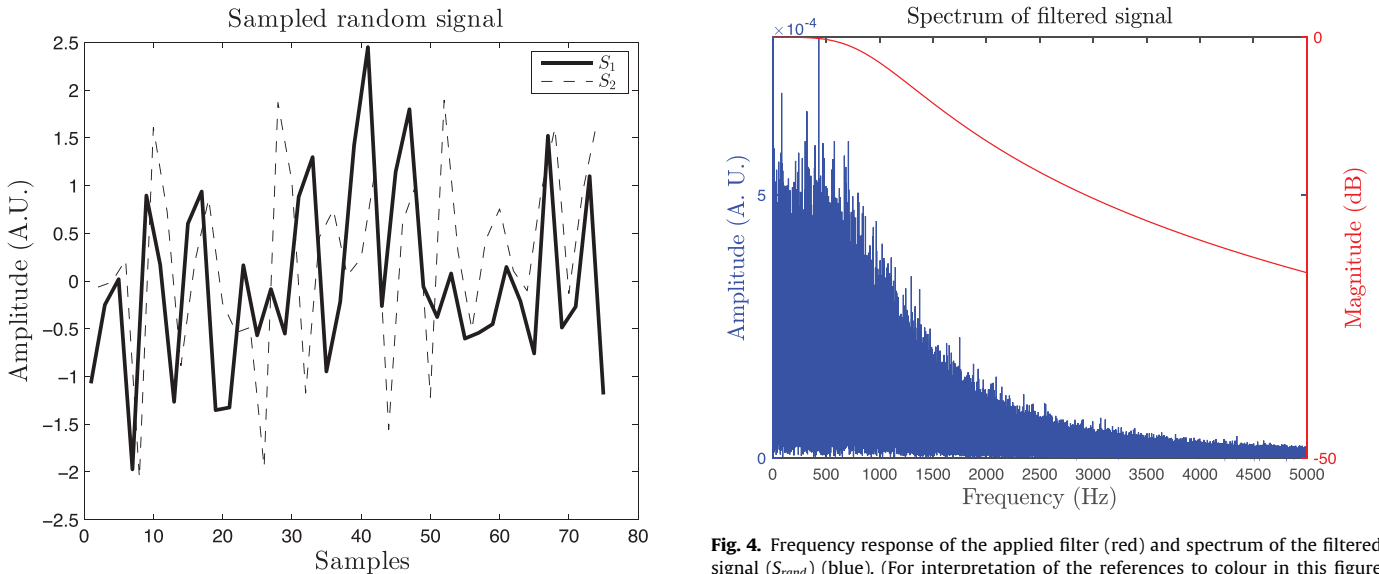**Fig. 2.** Analysis of the *white noise* signal (S_{noise}): histogram (left); frequency spectrum (right).

Fig. 3. Two samplings of *white noise* signal with f_s = 500 kHz and with 1 sample delay.

Fig. 4. Frequency response of the applied filter (red) and spectrum of the filtered signal (S_{rand}) (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

artificial thermal noise, shot noise or ADC quantization noise) was added to simulate a situation without interferences.

Cross-correlation methods are only applicable for signals with the same sampling frequency but, in our study, signals present different sampling rates. To solve this issue there are two options: (1) using interpolation methods to increase the number of samples in the signal with lower time resolution; (2) perform a downsampling in the signal with the highest time resolution. In this work, the (2) option was used because the random signal has an unpredictable behaviour and using interpolation methods can lead to errors in the reconstruction of the signal.

Fig. 5 shows two examples of cross-correlations between signals with different sampling frequencies. In the first case Fig. (5-Left), one signal is "acquired" at $f_s^1 = 20$ kHz and the other one is "acquired" at $f_s^2 = 2000.0$ Hz. The signals were delayed by 25,432 samples in the original data (it is equivalent to 25.432 ms @ $f_s^0 = 1$ MHz). The maximum correlation is founded at a delay equal to 51 samples (25.5 ms @ $f_s^2 = 2$ kHz) and it is a unique peak protruding

from the noise area (value equal to 2.0×10^{-3})). The exact error in the delay determination is 0.068 ms which is less than the temporal resolution of the signal with the lower sampling rate (0.5 ms @ $f_s^2 = 2000.0$ Hz). The cross-correlation was computed using the function *xcorr* of MATLAB®.

This method shows good results in the simulation case because the sampling rates of the two signals are high enough to obtain a good temporal resolution in the cross-correlation. Delay identification problems may appear with lower sampling rates.

When the acquisition systems does not have a temporal reference for the acquired data, it is necessary to estimate the temporal separation of consecutive samples based on the device sampling rate. In some devices, it is possible that the sampling rate is slightly different than the expected one. This fact causes a mismatch between the estimated time and the real acquisition time. Since the error is cumulative, i.e., the error in the last sample is the sum of the errors in each sample ($N \times error$). This effect is critical in long term acquisitions because some problems occur both in
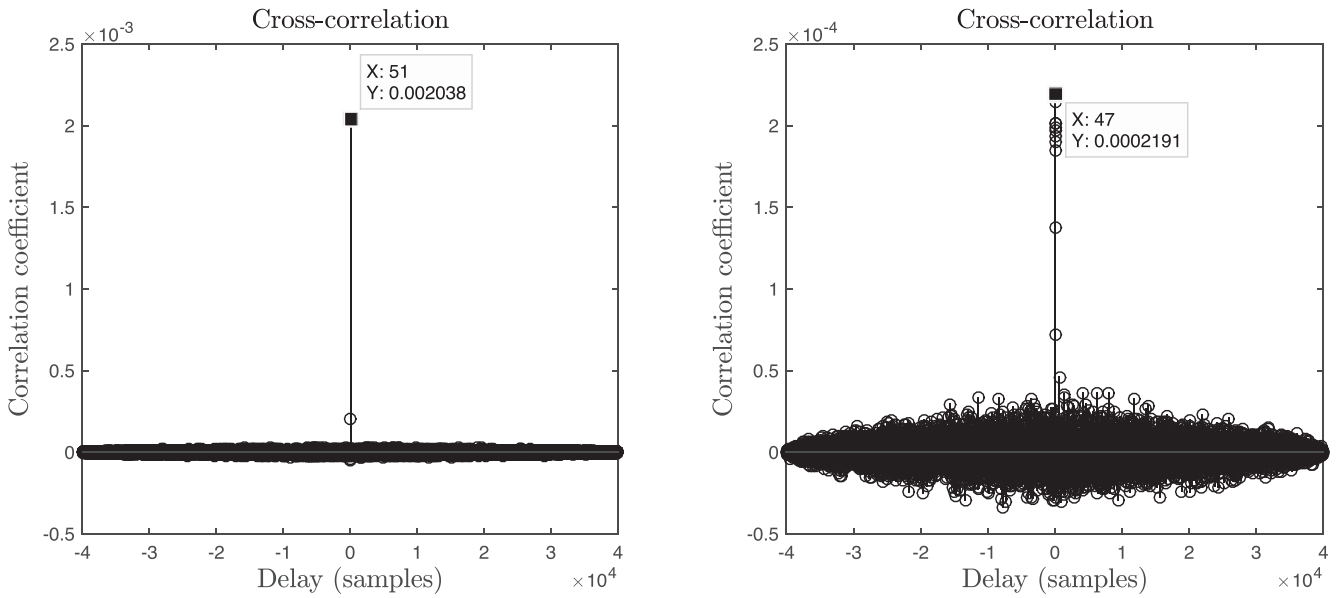
Fig. 5. Cross-correlation between two similar signals delayed by 25.432 ms, one acquired at $f_s^1 = 20$ kHz and downsampled to 2 kHz and the other acquired at: Left – $f_s^2 = 2000.0$ Hz; Right – $f_s^2 = 2000.5$ Hz.

the identification of the initial delay and in the synchronization of further samples.

The cross-correlation in Fig. 5 – right shows a simulation for the above situation. In this case, the cross-correlation was computed with one signal sampled with $f_s^1 = 20$ kHz and the second sampled signal with $f_s^2 = 2000.5$ Hz. This small variation in the sampling frequency corresponds to only 0.025% of the initial value (2000 Hz). The delay in the original data was maintained (25.432 ms). The decline in the correlation is clear. Both the peak enlargement and the lower value of the correlation (2.2×10^{-4}) are the proof of the worse result. Furthermore, the delay is identified at 47 samples, which correspond to 23.5 ms if the 2 kHz sampling rate is considered.

Fig. 5 – right situation happens because the correlation method minimize the error between signals in a global way. Although the initial samples contain only small errors, larger errors are detected in the final samples of the signal. These errors contribute to push the maximum of the correlation coefficient to a wrong value. With this delay, if we superimpose the two data, large errors are encountered in the initial and final part of the signal. The trigger method also does not solve this problem. Even if the trigger can synchronize correctly the initial point of the acquisition, samples far away from the trigger accumulate large errors.

To overtake this issue a new approach was necessary. A local method, based on the same cross-correlation, was implemented. This methodology uses small segment of one signal to perform the cross-correlation with the other one (complete signal). The main drawback of this methods lies in the need of a reliable temporal base for one of the signals, i.e., knowing the correct sampling rate.

The simulation that follows tries to define the final algorithm that will be applied in real data.

2.2.2. Local method

The local method was applied to two signals, described earlier, with $f_s^1 = 20$ kHz, $f_s^2 = 2000.5$ Hz and the same delay (25.432 ms) in the acquisition starting point. The first step of the local method consists in the selection of a short segment (0.5 s) of the second signal (f_s^2). After that, a cross-correlation between the short segment and the complete first signal was performed. The maximum of the cross-correlation is identified and its position (d_1) is saved, together

with the initial short segment point (s_1). Moreover, the full width at half maximum (FWHM), for the correlation peak, is computed as a measure of the quality of the correlation.

By selecting a new segment, displaced by 0.5 s, and performing the same procedure, a new pair (s_2, d_2) is recorded. This procedure is performed successively up to the last segment (s_N, d_N), allowing the construction of the matrix represented in (6). A linear regression is applied to this data in order to define a mapping function where the points of the signal without temporal reference (s) correspond to points in the reliable signal (d). The slope of the linear regression measures the inaccuracy of the sampling rate estimation. If the sampling rate is equal to the estimated one, the slope of the line will be equal to 1.

$$V_{s,d} = \begin{bmatrix} s_1 & d_1 \\ \vdots & \vdots \\ s_i & d_i \\ \vdots & \vdots \\ s_N & d_N \end{bmatrix} \qquad (6)$$

Fig. 6 shows a scatter plot of the pairs (s_i, d_i). The points are marked with their respective FWHM (error bars). The error bars are not visible because of their small size when compared to the plot scale. The black line corresponds to the linear regression of the data. The very good fitting is confirmed by the value of the determination coefficient ($R^2 = 1.0000$).

The linear regression (mapping function) corresponds to $s(d) = 0.99975 \times d + 49.6438$ where d is the sample in the segmented signal that corresponds to the sample ($d = 1$) in the time vector. For example, the first sample of the delayed signal ($d = 1$) corresponds to the sample number 51 of the time vector signal ($s = 51$). This delay corresponds to the correct result achieved in the simple cross-correlation method (25.5 ms). Since decimal values have no meaning, because we are dealing with discrete points, all the results need to be rounded.

The performed simulation intends to prove the ability of a *white noise* signal to synchronize different DAS in a biomedical context, even when the sampling frequencies are not accurate. The next
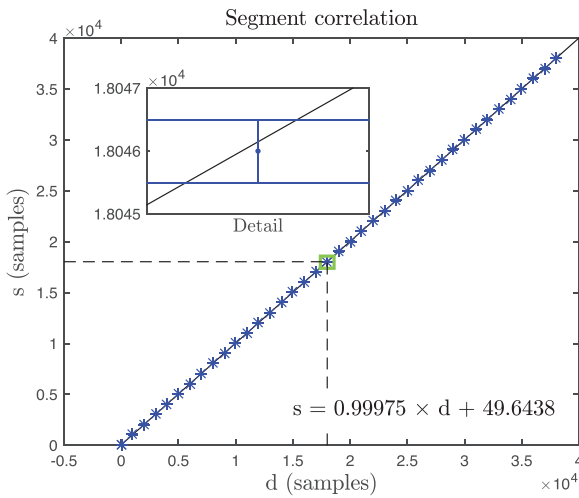
Fig. 6. Scatter plot of the pairs (s_i, d_i) jointly with theirs errorbars (blue, see detail). The errorbars are small compared with the y-axis scale. The black line corresponds to the linear regression with the equation expressed in the bottom right hand corner. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

section shows a simple experimental assembly to record synchronization signals in real conditions.

3. Experimental results

The experimental set-up was based on the architecture shown in Fig. 1. A simple experiment was conducted by using two DAS and a *white noise* generator. A two pole active low-pass filter was implemented with a cut-off frequency of $\approx$1 kHz [19].

Moreover, an instrumentation amplifier was added to control the gain and the DC level of the synchronization signal.

The synchronization signal was digitized by a National Instruments^{TM}DAQ (NI-USB-6210) (DAS 1) with a resolution of 16 bits and an input range of $\pm$10 V. An Arduino^{TM} Duemilanove (DAS 2) with a resolution of 10 bits and an input range of 0–5 V was used as a second independent acquisition platform. The preferential approach would be to split the signal using two channels and to adjust the gain and the offset to fill the input range of the two DAS. However, this method adds a noise level to the synchronization signal (from the analogue electronic components) leading to different noise levels for each one of the channels. To prevent this effect, an adjustment of the signal range to the shortest input range was performed. In our experiment, the Arduino^{TM}has the smaller input range, so the signal was conditioned to a range of 0–5 V.

The NI-DAQ has acquired the signal with a sampling rate of 20 kSamples/s and the Arduino^{TM}with an approximate sampling rate of 2 kSamples/s. One must stress that data acquired by NI-DAQ has a reliable temporal vector when compared to the Arduino^{TM}. These features (reliable temporal vector and highest sampling rate) makes the NI-DAQ as the reference DAS for the construction of Arduino^{TM}temporal reference. Three data sets (DS) were acquired on both DAS. The acquisition was started in the NI-DAQ and, a random time later, in the Arduino^{TM}. The starting time of Arduino^{TM}acquisition was recorded in the NI-DAQ by using an external trigger.

Thereafter, the algorithm explained in Section 2.2.2 is applied to the data. A segment with 0.5 s was selected with a displacement of 0.5 s. The resultant data with (pairs (s_i, d_i)) was converted to time, which is a more intuitive physical quantity, and it is presented in Fig. 7.

In DS2 (green line) and DS3 (red line), 19 segments were correlated. In DS1 (blue line) only 17 segments were correlated due to NI
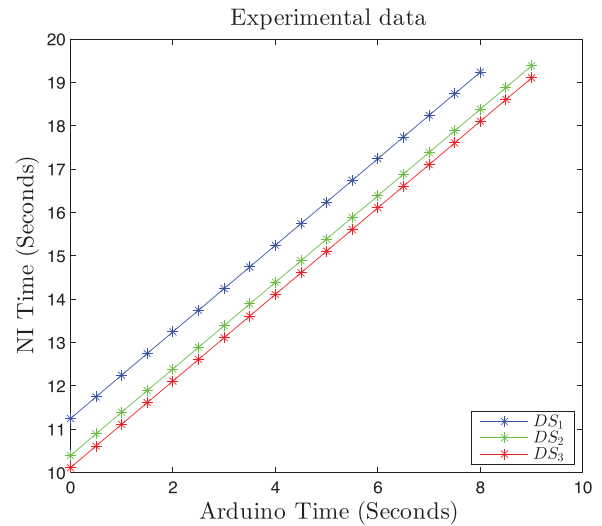

Fig. 7. Correspondence between the NI time samples and Arduino^{TM}time samples. Each line represents one different data set (DS).

Table 1
Computed and real delays for the initial point of 3 distinct data-sets.

Delay (s)	Data-set 1	Data-set 2	Data-set 3
Computed	11.249 ± 0.0015	10.395 ± 0.0015	10.116 ± 0.001
Real delay	11.2495 ± 0.00005	10.3955 ± 0.00005	10.1162 ± 0.00005

signal length constrains. By using this data, it is possible to conduct a linear regression that associates the reliable NI time reference to the Arduino^{TM}estimated time vector. The parameters estimated from data are presented, for the three DS, in Eqs. (7)–(9), respectively.

$$t_{NI} = 0.9984 \times t_{Arduino} + 11.249 \qquad (7)$$

$$t_{NI} = 0.9984 \times t_{Arduino} + 10.395 \qquad (8)$$

$$t_{NI} = 0.9984 \times t_{Arduino} + 10.116 \qquad (9)$$

The Arduino^{TM}time reference was constructed, for each acquisition, by using these equations. The slope of the three lines is identical which is expected since the acquisition systems are the same. This means that the change between the real sampling rate and the estimated one exists and it is equal for all the cases. If the estimated sampling rate and the real sampling rate were the same, the slope of the line would be exactly 1. In this case, their slope is 0.9984 which means that exists a small difference. For all the cases the R^2 was 1.0000.

The computed and real delays, for the first sample of Arduino^{TM}acquisitions are presented in Table 1.

The resolution of the real delays is better than the computed ones. The real time delays are determined by using a trigger signal acquired from the NI-DAQ data. The identification of this trigger is unambiguous which leads to a minimum identification error corresponding to the NI-DAQ time resolution, i.e., 0.05 ms.

On the other hand, the temporal resolution of the computed delay is poorer due to the Arduino^{TM}temporal resolution (0.5 ms). This error is also combined with an estimated error calculated using the correlation $\pm$ FWHM/2 for a major estimation. The results show a minimum delay identification error of 0.2 ± 1 ms (DS 3) and a maximum of 0.5 ± 1.5 ms (DS 2). These values are close to a zero since the correct samples were identified in all 3 data sets. The minimal resolution of 0.5 ms causes the uncertainty in error identification to be larger than its absolute value. A large uncertainty is obtained because it was enlarged by the consideration of an error equals to the FWHM. The highest obtained FWHM was 11 samples
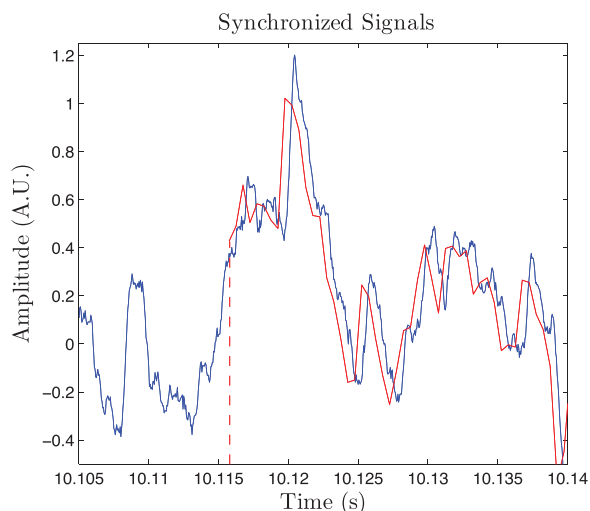
Fig. 8. Final synchronization. The blue line shows the NI-DAQ signal with its original sampling rate. The red line shows the Arduino signal with the new temporal vector. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

which correspond to 5.5 ms but it is clearly an outlier. All the other cross-correlations the FWHMs never surpass 5 samples.

The final result is the creation of a temporal vector for each data set. Fig. 8 shows the superposition of the synchronized signal from the DS3. The blue signal shows the NI-DAQ data which starts at time equals 0 s. On the other hand, the acquisition of the Arduino^TM only starts at 10.116 s (red signal). By the observation of the figure it is possible to identify a slight delay in the Arduino^TM data. This error is comprised in the correlation FWHM.

4. Applicability

The proposed system was implemented in a multi-channel acquisition system developed by the Electronics and Instrumentation Group (GEI) of the University of Coimbra [16]. This system is composed by 5 acquisition modules: Multifunctional DAS – National Instruments (NI) 6210; Pulse wave velocity (PWV) identification; Electrocardiography (ECG); Photoplethysmography (PPG); and Pulse wave analysis (PWA).

The PWA modules incorporates its own acquisition and control system (dsPIC33 from Microchip ^®) while the other modules signals (PPG, ECG and PWV) are digitized by the NI-DAQ. The PWA module introduces a delay of 112 ms between the pulse waveform signal and the synchronization signal acquisitions. In addition, the *dsPIC33* executes real-time signal processing tasks like signal integration and baseline removal. This firmware complex activity of the dsPIC33 adding to the external clock ageing and temperature dependences could lead to impairments on the acquisition times.

The local cross-correlation method was developed to account for NI-DAQ and dsPIC clock offsets and mismatches when long term acquisition are performed. In that specific case, we believe that the local method is a better synchronization algorithm that would lead to more reliable results.

NI-DAQ signals are internally synchronized. The *white noise* generator circuit was included in the EEG module due to size constrains and its signal transmitted, using a bus, to the NI-DAQ and PWA modules.

5. Conclusions

In this paper a synchronization method, based in cross-correlation, for biomedical signals which are acquired with

different devices is proposed. This method exploits the random nature of *white noise* and its large frequency band. These features guarantee that it is unlikely to find the same segment of signal in a short time period making it a good candidate for time synchronization applications.

A computation simulation was performed in absence of quantization noise and using an infinite frequency band *white noise*. In these perfect conditions, an error of 0.068 ms (13% of the time resolution) was achieved. This error is less than the time resolution of the cross-correlation (0.5 ms) which means that the correct sample is identified for the delay calculi.

In the case of minimal variations in the sampling frequency (0.025%) the simple cross-correlation achieved a minimal error of 1.93 ms (259% relative error). This result shows a poor performance in the identification of a delay between two biological signals. Moreover, the quality of the correlations decreases with the increase in the sampling rate difference. To overpass this problem a more complex approach was used.

The local approach uses a short segment of the synchronization signal to perform the cross-correlation and identify, in the signal with a reliable sampling rate, the correspondent points. By selecting many segments it is possible to obtain a set of points on which a linear regression is performed. With this linear regression it is possible to create a new time vector for the data with unknown sampling rate. This method identifies the initial delay with an error of 0.068 ms, in accordance with the previous result.

Section 3 shows the results achieved with a real experimental set-up. A National Instruments^TM NI-DAQ and an Arduino^TM were used as the DAS. The first system uses a sampling frequency of 20 kHz and the second one an undefined frequency of approximately 2 kHz. Three different data sets were recorded and processed using the algorithm presented in Section 2.2.2. The best result was achieved with the DS3 with a acquisition starting point identification error of 0.2 ms (40% relative error).

Concluding, the presented methods are capable of synchronize acquired biomedical signals with different devices with an accuracy of a millisecond and, most significantly, under the signal time resolution. In the physiological processes, this scale is small enough to be considered a good result.

References

[1] R.M. Rangayyan, Introduction to biomedical signals, in: Biomedical Signal Analysis, John Wiley & Sons, Inc., 2001, pp. 1–59, http://dx.doi.org/10.1002/9780470544204.ch1.

[2] K. Najarian, R. Splinter, Biomedical Signal and Image Processing, Taylor & Francis Group, 2012 http://www.google.pt/books?id=Gi-GOtnHAtAC

[3] T. Penzel, B. Kemp, G. Klösch, a. Schlögl, J. Hasan, a. Värri, I. Korhonen, Acquisition of biomedical signals databases, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 25–32 http://www.ncbi.nlm.nih.gov/pubmed/11446206

[4] P. Désesquelles, T. Ha, A. Korichi, F. Blanc, A. Olariu, C. Petrache, Fast analytical methods for the correction of signal random time-shifts and application to segmented HPGe detectors, Nucl. Inst. Methods Phys. Res. B 267 (3) (2009) 542–547, http://dx.doi.org/10.1016/j.nimb.2008.11.042.

[5] E. Oczeretko, J. Swiatecka, A. Kitlas, T. Laudanski, P. Pierzynski, Visualization of synchronization of the uterine contraction signals: running cross-correlation and wavelet running cross-correlation methods, Med. Eng. Phys. 28 (1) (2006) 75–81, http://dx.doi.org/10.1016/j.medengphy.2005.03.011 http://www.ncbi.nlm.nih.gov/pubmed/15919226

[6] B. Jovic, C. Unsworth, G. Sandhu, S. Berber, A robust sequence synchronization unit for multi-user DS-CDMA chaos-based communication systems, Signal Process. 87 (2007) 1692–1708, http://dx.doi.org/10.1016/j.sigpro.2007.01.014.

[7] A. Camacho, A.M. Hernandez, Z. Londono, L.Y. Serna, M. Mananas, A synchronization system for the analysis of biomedical signals recorded with different devices from mechanically ventilated patients, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol.

2012, 2012, pp. 1944–1947, http://dx.doi.org/10.1109/EMBC.2012.6346335 http://www.ncbi.nlm.nih.gov/pubmed/23366296

[8] W. Davenport, W.L. Root, An introduction of random signals and noise, in: I. Press (Ed.), An Introduction of Random Signals and Noise, No. I, Wiley-interscience, 1987, Chapter 8.

[9] R. Quiroga, A. Kraskov, T. Kreuz, P. Grassberger, On the performance of different synchronization measures in real data: a case study on EEG signals, Phys. Rev. E 65 (4) (2002) 041903, http://dx.doi.org/10.1103/PhysRevE.65.041903.

[10] M. Weeks, Digital Signal Processing: Using MATLAB and Wavelets, Electrical Engineering Series, Infinity Science Press, LLC, 2007 http://books.google.pt/books?id=rLqQYPqa82YC

[11] M. González, M. Chimeno, J. Castro, Bias and uncertainty in heart rate variability spectral indices due to the finite ECG sampling frequency, Physiol. Meas. 25 (2) (2004) 489–504, http://dx.doi.org/10.1088/0967-3334/25/2/008 http://stacks.iop.org/0967-3334/25/i=2/a=008?key=crossref.48c77aad8912e1c3367885a5946ee8de

[12] D. Lucani, G. Cataldo, J. Cruz, G. Villegas, S. Wong, A portable ECG monitoring device with Bluetooth and Holter capabilities for telemedicine applications, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, 2006, pp. 5244–5247, http://dx.doi.org/10.1109/IEMBS.2006.260798 http://www.ncbi.nlm.nih.gov/pubmed/17946295

[13] P. Baheti, H. Garudadri, An ultra low power pulse oximeter sensor based on compressed sensing, in: Sixth International Workshop on Wearable and Implantable Body Sensor Networks, IEEE, 2009, pp. 144–148, http://dx.doi.org/10.1109/BSN.2009.32 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5226900

[14] B. Kim, S. Yoo, Motion artifact reduction in photoplethysmography using independent component analysis, IEEE Trans. Biomed. Eng. 53 (3) (2006) 566–568, http://dx.doi.org/10.1109/TBME.2005.869784 http://www.ncbi.nlm.nih.gov/pubmed/17271609

[15] C. Poon, Y.-t. Zhang, A novel biometrics method to secure wireless body area sensor networks for telemedicine and M-health, IEEE Commun. Mag. (2006, April) 73–81.

[16] V. Almeida, New Methods for Hemodynamic Evaluation: A Multi-Parametric Approach, Ph.D. thesis, University of Coimbra, 2013 http://dspacetestes.bg.uc.pt/handle/10316/23933

[17] T. Pereira, I. Santos, T. Oliveira, P. Vaz, T. Correia, T. Pereira, H. Santos, H. Pereira, V. Almeida, J.a. Cardoso, C. Correia, Characterization of optical system for hemo-dynamic multi-parameter assessment, Cardiovasc. Eng. Technol. 4 (1) (2013) 87–97, http://dx.doi.org/10.1007/s13239-013-0125-y.

[18] T. Pereira, P. Vaz, T. Oliveira, I. Santos, H. Pereira, V. Almeida, C. Correia, J. Cardoso, Empirical mode decomposition for self-mixing Doppler signals of hemodynamic optical probes, Physiol. Meas. 34 (3) (2013) 377–390, http://dx.doi.org/10.1088/0967-3334/34/3/377.

[19] B. Carter, Application Report: Filter Design in Thirty Seconds, Tech. Rep. December, Texas Instruments, 2001.