Distributed Full Synchronized System for Global Health Monitoring based on FLSA

Giuseppe Coviello, Senior Member, IEEE, Antonello Florio, Graduate Student Member, IEEE, Gianfranco Avitabile, Senior Member, IEEE, Claudio Talarico, Senior Member, IEEE, and Janet M. Wang-Roveda.

Abstract—In modern medicine, smart wireless connected devices are gaining an increasingly important role in aiding doctors’ job of monitoring patients. More and more complex systems, with a high density of sensors capable of monitoring many biological signals, are arising. Merging the data offers a great opportunity for increasing the reliability of diagnosis. However, a huge problem is constituted by synchronization. Multi-board wireless-connected monitoring systems are a typical example of distributed systems and synchronization has always been a challenging issue. In this paper, we present a distributed full synchronized system for monitoring patients’ health capable of heartbeat rate, oxygen saturation, gait and posture analysis, and muscle activity measurements. The time synchronization is guaranteed thanks to the Fractional Low-power Synchronization Algorithm (FLSA).


I. INTRODUCTION

Time synchronization (usually simply referred to as “synchronization”) in computer networks and distributed systems theory has always gathered attention from the scientific community. As summarized by Freris et al. [1], there are typically three levels of specifications for time synchronization. In the first level, it is only important the event order. The goal is to create the right chronology for the events in the network, while the exact knowledge of each time instant is not important. In the second level, it is important to achieve a relative synchronization in the network, that is, estimating and compensating the relative clock drift among different clocks; the nodes do not need to agree on a common time reference, but they have enough information to transpose each clock value to their own time reference. In the third and strongest level the goal is to achieve absolute synchronization. Each node owns a local clock value and the synchronization aims to have each clock of the network agree on a common time. It is evident that the a higher level of synchronization implies the characteristics of lower ones. Also, as we consider upper levels, improved accuracy is reached for the entire synchronization process, determining an increase of the computational load on the network facilities. The choice of the level is strongly influenced by the type of application, as this latter introduces some constraints on the allowable complexity of the synchronization protocol, and requires trading-off complexity and accuracy.

Over the years, due to the introduction of new technology paradigms like the Internet of Things (IoT), the meaning of the term ‘network’ has significantly changed. In the IoT, a huge density of smart and, usually, wireless-connected devices join networks with different extensions. In this paper, we will focus on Wireless Body Area Networks (WBANs), i.e. networks with a small geographical extension, based on a mesh or star topology [2]. Each node of a WBAN is usually equipped with sensors and powered by a battery, to enhance its portability. In this context, IoT is often employed for supporting medicine giving rise to the Internet of Medical Things (IoMT).

In IoMT synchronization is crucial, since data coming from different sensors placed or implanted along the body are merged to obtain the final diagnosis. However, different biological parameters vary with different rates. Hence, we can introduce two possible levels of synchronization, depending on the level of required accuracy. In weak network time synchronization (WTS), we allow for a lower degree of accuracy, thus relaxing constraints on the synchronization process complexity. WTS is suitable for slowly varying phenomena such as body and environment temperature variations. In strong network time synchronization (STS), the degree of accuracy must be kept as high as possible since even small mismatches on time-stamping can lead to huge errors in the data merging phase. This is the case of surface electromyography (sEMG) correlated to Inertial Measurement Units (IMUs) for gait analysis. Nevertheless, a higher degree of synchronization calls for higher complexity for the potential increased number of exchanged messages.

Another difference when comparing IoMT devices is between online and offline data acquisition for post-processing.
In particular, we refer to online processing when a device forwards the sensor measurement to the gateway and/or the outer network as soon as the data is consolidated. In the offline processing, the data is stored in a persistent memory with which the device is equipped, and only after a predefined time interval, the acquisitions are sent to the gateway. Online data processing calls for a stable network infrastructure that is usually feasible to be realized in controlled environments such as medical laboratories. For most medical protocols, it is important for the patients to behave naturally, as when no measurement is in progress. In [3] Renggli et al. provide data highlighting a huge difference between offline and online data processing for what concerns the reliability of the measurements for diagnosis purposes. In particular, a long off-laboratory off-line processing is more reliable than an in-lab online measurement.

In this paper, we introduce an absolute low-cost synchronized platform for off-line acquisition of heartbeat rate (HR), peripheral oxygen saturation (SpO₂), gait analysis, posture analysis, and muscle activity related to gait analysis. The system is arranged in a star topology with a master/slave (M/S) architecture. The network time synchronization is achieved thanks to the algorithm that the authors presented in [4] and [5].

A. Related Works

In literature, it is possible to find many examples of synchronized IoMT systems. Mo et al. in [6] propose a synchronized system for monitoring the spent energy and breath volume. The system comprises three different units, arranged in an M/S architecture. Two units are equipped with an accelerometer (placed, respectively, on the wrist and on the hip) and another unit uses a ventilator sensor (placed on the abdomen). The synchronization problem arises because, along with the clock drift problems, depending on the configuration, packets can be long enough to determine synchronization artifacts. The proposed approach is a slot-data synchronization method, performed at the master level. The master is in charge of the network synchronization, and it reorders the packets in the correct time order taking into account the reception time, the length of the packet, and the sampling time of the sensors.

In [7], Vedaei et al. propose a WTS system based on Raspberry Pi Zero for monitoring the HR, the SpO₂, and temperature for COrorna Viirus Disease 2019 (COVID-19) diagnosis. The system transmitted the data acquired from the sensors to a smartphone and, then, to a cloud infrastructure for further processing. In this case, STS is not required in the system since heart rate, SPO₂ and temperature are slowly varying parameters. Thus, in this case, a large degree of error on the synchronization is accepted.

The solution proposed in [8] allows for both WTS and STS. In this work, Pathinarupothi et al. propose an IoT system for global health monitoring based on edge computing. The system uses an M/S hierarchy when STS is required, for example in the case when it is used for symptomatic patients. The master is chosen by the gateway, connecting the BAN with the outer networks, and it is in charge of guaranteeing the network synchronization. In the case of WTS mode, a peer-to-peer (P2P) structure is adopted, with the gateway being the network coordinator, loosely constraining the time synchronization.

The work of Wang et al. [9] introduces a planter pressure acquisition system for diabetic foot prevention and diagnosis. The system is composed of two special shoes with pressure sensors connected to a wireless Data Acquisition (DAQ) board, one for each leg. Each DAQ is synchronized to the other and it is in charge of querying the connected sensors, collecting data, and exposing them to the outer network through the Bluetooth connection of a smartphone. In this domain, STS is mandatory, since it is a constraint of the gait analysis [10]. However, no information on the employed synchronization protocol is available.

B. Main contributions and paper organization

We summarize here the most relevant contributions of our work.

- We introduce a low-cost synchronized platform for offline patients’ long-term global health monitoring. The system is low-power and relies on custom-designed boards based on the nRF52832 SoC [11], specific sensors for measuring the HR, SpO₂, gait behavior, posture, and muscle activity related to the gait through sEMG. The employed network topology is star-like with one predefined master and several slaves.
- We discuss FLSA, the employed synchronization algorithm implementation, briefly recalling the theoretical background introduced in [4].
- We highlight and propose a solution to the mutual synchronization problem between master and slaves. This problem arises when all the system units expose sensors for capturing data that are involved in the data merging process.
- We implement the FLSA in the monitoring system. Thus, we present the preliminary results in terms of achieved synchronization accuracy after one-month-long experiment in the laboratory and the ones obtained by testing the entire monitoring system on a healthy subject in a one-day test in everyday life.

The remaining part of the paper is organized as follows. In Section II we recall the theory behind the synchronization algorithm proposed by the authors, highlighting its limits as it does not allow for a mutual synchronization between master and slaves, in the way it was deployed. Hence, we introduce a supplementary procedure to handle also this situation. In Section III we describe in detail the low-cost system for health monitoring. Section IV describes an experimental campaign validating the system’s operations. Conclusions close this work.

II. FRACTIONAL LOW-POWER SYNCHRONIZATION ALGORITHM

The authors proposed FLSA [4], [5] as a possible solution for the low-power multi-board time synchronization problem in medical applications. The timer correction is performed...
Initialization

When a slave is said to be synchronous it can switch-off its radio section. The slot-skipping routine operates in terms of synchronization slots. First of all, the number of slots in which the node \( j \) can be considered to be synchronous to the master, \( S_{\text{sync}}(j) \), is updated, incrementing by one unit the previously stored value. We assume the slots in which the radio section is turned off to be synchronous, by hypothesis. The radio is switched off when the number of synchronous slots exceeds a predefined threshold \( S_{\text{sync}}^{\text{min}} \). The ST is adjusted at the end of each synchronization slot, taking into account the weighted mean value of the delays. This value represents the average error for each synchronization slot and it is subtracted in each skipped synchronization interval. After that, the skipped slot count is updated. The routine ends with the radio being switched on for receiving a new synchronization packet, at the reception of which, the RMQM routine is executed. The skipped slot number linearly increases when the systems are synchronous.

The Slave Timer correction Routine is governed by the set of equations:

\[
\begin{align*}
\sum_{i=A}^{B} \Delta R_C^i \Delta I_i &= N_{M}^{\text{new}} \\
\sum_{i=A}^{B} \Delta I_i &= \left\lfloor \frac{T_M}{t_{\text{RTC}}} \right\rfloor \\
\{I_i, R_C^i\} &> 0 \forall i
\end{align*}
\]

being \( R_C^i \) the updated value of comparator \( i \), \( \Delta I_i \) the updated value of its weight and \( N_{M}^{\text{new}} \) the compensated time slot duration. The system of equations in (4) does not have a single solution. Therefore, a solution can be found imposing the constraint \( \delta R_A^i = \delta R_B^i = 0 \) and start an iterative procedure. The simplified equation system is:

\[
\begin{align*}
\delta B &= \Delta N_M \\
\delta A &= -\Delta N_M
\end{align*}
\]

The new parameters to be set are available only when the system has a solution. If that is not the case, the sign of \( \Delta N_M \) indicates the advance or delay of the slave with respect to the master, and so new values of \( I_A \) and \( I_B \) are chosen with the initialization step being executed. It is very important to highlight that the method detects the \( t_{\text{RTC}} \) minimum deviation while the correction precision is equal to \( t_{\text{RTC}} \), that is, the minimum allowed value determined by the specific hardware implementation.

The preliminary measurement campaign was 7 days long and it employed a time periodicity for the synchronization \( T_M = 20 \text{ s} \). The experiment showed that in the best case for about 22 hours the time mismatch was lower than 500 \( \mu \text{s} \). From a power consumption perspective, the power saving obtained was in the 85-96\% range when compared to the continuous transmission of synchronization packets.

B. The problem of M/S reciprocal synchronization

In the first implementation of FLSA [5], we focused only on guaranteeing the maximum time accuracy on which slaves agree given the lowest power consumption. This is sufficient totally at hardware level, thus making the approach flexible and transparent to the upper layers of the system. In this section, we briefly recall its formulation and results, along with some new, recently introduced improvements.

A. A fractional approach to timer correction

FLSA performs synchronization thanks to four sub-algorithms, which are summarized in Figure 1 along with their interactions. FLSA makes use of two \( N/N + 1 \) type comparators/counters \( \{A,B\} \) to perform the dual modulus division. The number of ticks, \( N_M \), needed to form a synchronization slot of length \( T_M \) is:

\[
N_M = \left\lfloor \frac{T_M}{t_{\text{RTC}}} \right\rfloor
\]

(1)

where \( \lfloor \cdot \rfloor \) represents the rounding at the closest integer number operator, and \( t_{\text{RTC}} \) is the granularity of the Real-Time Clock (RTC). Directly from the RTC it is possible to extract the System Timer (ST) granularity \( t_{ST} \) with the formula [4]

\[
\begin{align*}
R_C^A &= \left\lfloor \frac{t_{ST}}{t_{\text{RTC}}} \right\rfloor \\
R_C^B &= R_C^A + 1
\end{align*}
\]

(2)

with \( R_C^i \) being the number of ticks needed to form a \( t_{ST} \) for comparator \( i \). The synchronization slot is determined by the variation of the comparator values, according to

\[
N_M = R_C^A \cdot I_A + R_C^B \cdot I_B
\]

(3)

with \( I_i \) being the number of times comparator \( i \) must be used.

Every \( T_M \) seconds the master broadcasts to the slaves a synchronization packet containing its ST. The synchronization packet is processed with the highest priority with respect to every other possible type of message thanks to the Received Messages Queue Management (RMQM) Routine, and this is true for master and slaves. After the synchronization message has been received, each slave first evaluates the equivalent accumulated time difference in a single slot, \( T_M \). Then, the slave uses this value in the evaluation of the number of the time interval in which keep the radio off according to the slot-skipping routine.

Fig. 1: FLSA routines interaction diagram [4].
During the TXIDLE phase, in which the radio is transmitting a packet, the END point, the READY phase, the radio is ramping up and preparing for transmission. After this phase, the packet is generated on the transmitter side. After will refer to what is represented in Figure 2a. In the first

For the description of the general transmission scheme, we consider the stages during the transmission and reception of a data packet on the nRF52832 SoC by Nordic [11], which will be also the SoC employed in our experiments. The SoC is built around an Arm® Cortex™-M4 CPU with floating-point unit running at 64 MHz. We will refer to the microcontroller unit as μC. The radio section (to which we will refer to as radio) is employed for the radio to switch between the transmission (Tx) and the reception (Rx) of packets and vice-versa, maintaining the same frequency. The meaning of the TxDelay value will be explained later on.

In order to better appreciate the impact of the duration of each phase, we summarize in Table I each timing value from the datasheet [11]. From those data, it is possible to say that the minimum delays introduced by the radio are in the order of hundreds of microseconds in both transmission and reception. In addition to these time values, we also should consider the preparation time and parsing time for the packet, as well as the traveling time of the packet in the radio medium.

Therefore, one key factor is to reduce the number of boards employed, and so the master itself should expose some sensors in order to capture biosignals.

As a consequence, a further improvement on FLSA is necessary, since the slave agreement on a common time is a necessary but not sufficient condition to the master agreement. This is due to several reasons. To better understand, let us consider the stages during the transmission and reception of a data packet on the nRF52832 SoC by Nordic [11], which will be also the SoC employed in our experiments. The SoC is built around an Arm® Cortex™-M4 CPU with floating-point unit running at 64 MHz. We will refer to the microcontroller unit as μC. The radio section (to which we will refer to as radio) is able to handle multiple protocols and it is also being capable of managing full protocol concurrency. In particular, it supports Bluetooth LE, NFC, ANT, and 2.4 GHz proprietary protocols.

TABLE I: Nominal time values for the phase duration between the listed event/signal endpoints declared on the datasheet for the nRF52832 SoC [11].

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value [μs]</th>
<th>Time interval endpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{set,tx} )</td>
<td>140</td>
<td>( { \text{TXEN,READY} } )</td>
</tr>
<tr>
<td>( t_{dis,tx} )</td>
<td>6@1MSps/4@2MSps</td>
<td>( { \text{TXDISABLE,DISABLED} } )</td>
</tr>
<tr>
<td>( t_{set,rx} )</td>
<td>140</td>
<td>( { \text{RXREADY,READY} } )</td>
</tr>
<tr>
<td>( t_{dis,rx} )</td>
<td>0</td>
<td>( { \text{RXDISABLE,DISABLED} } )</td>
</tr>
<tr>
<td>( \min { t_{switch} } )</td>
<td>20</td>
<td>Tx ≡ Rx</td>
</tr>
</tbody>
</table>

Finally, the case of transmission and immediate reception of a packet and vice-versa are shown in Figures 2a and 2b respectively. In these cases the radio is not disabled, so we considered the time \( t_{switch} \) employed for the radio to switch from the transmission (Tx) to the reception (Rx) of packets and vice-versa, maintaining the same frequency. The meaning of the TxDelay value will be explained later on.

In order to better appreciate the impact of the duration of each phase, we summarize in Table I each timing value from the datasheet [11]. From those data, it is possible to say that the minimum delays introduced by the radio are in the order of hundreds of microseconds in both transmission and reception. In addition to these time values, we also should consider the preparation time and parsing time for the packet, as well as the traveling time of the packet in the radio medium.
The delay can be automatically evaluated by the master, packet. preparation phase since it simply sends back the same received set ready state \((t_{\text{switch}})\) and, then, from Tx to Rx \((t_{s})\) and vice-versa. Furthermore, the slave does not have a packet preparation phase since it simply sends back the same received packet. The delay can be automatically evaluated by the master, comparing the timestamps at the beginning of packet transmission and reception, obtaining the total time from which, in turn, it is possible to extract the value to be summed to the timestamp sent by the master to the slave to obtain a perfect synchronization. In the practical implementation, the estimated values will be rounded to the implemented \(t_{ST}\), which, in our case, is equal to 100 \(\mu s\). As mentioned before, the measured transmission time from master to slave is slightly greater than the reciprocal time. Based on this experimental evidence, the delay can be evaluated using the formula:

\[
\Delta T_{M}^{\text{sync}} = \frac{T_{rx} - T_{tx}}{2} \cdot t_{ST} = \frac{\Delta_t}{2} \cdot t_{ST}
\]

where the operator \(\lceil \cdot \rceil\) returns the rounding to the greater integer and \(T_{tx}\) and \(T_{rx}\) are the sent timestamp and the current timestamp expressed as multiples of \(t_{ST}\).

As reported before, a 2.12 ms total time was measured and, according to (6):

\[
\Delta T_{M}^{\text{sync}} = 1.11\text{ms}
\]

From the master perspective, the whole M/S mutual calibration procedure is summarized in Algorithm 1. For what concerns the slaves, the calibration steps are summarized in Algorithm 2. This sequence should be repeated several times and the obtained values averaged in order to obtain the optimal \(\Delta T_{M}^{\text{sync}}\). In our experiments the value obtained is the same as in (7).

### III. System description

As described in Section I, in this paper we also present a low-cost synchronized platform for the long-term off-line global health monitoring of patients. The system has an M/S architecture with a star network topology. In particular, the system is made up of one master unit and a group of three slaves.

The master unit is the HRSpO2board-HW-1-0-0 board shown in Figure 4a. It is based on an nRF52832 SoC, allocating on-board a high-sensitivity SpO2 sensor, an HR sensor for
wearable health on the backside of the board, and a light and environment temperature sensor. The board supports microSD cards for data storage and it is power supplied by a CR2032 coin lithium battery. It samples the onboard IMU sensor with a 100 Hz rate.

The sEMG-HW-1-0-0 is shown in Figure 4c. It is an extension of the IMUboard-HW-1-0-0 previously described. It allocates a nine DoF MPU-9250 nine-axis MEMS MotionTracking™ device from TDK-InveSense (San Jose, CA, USA), a microSD for data storage, the circuit for the sEMG, and it is powered by a lithium battery connected to the JST connector. It samples the onboard IMU sensor at 100Hz and the sEMG output circuit at 50Hz. The sEMG output circuit is a square wave that indicates if there is muscle activity or not as described in [12].

### IV. Measurement Campaign

In this section, we present the results achieved during two measurement campaigns. In particular, after implementing the FLSA in the monitoring system described in the previous section, we first validated the results in terms of achieved synchronization accuracy after a one-month-long experiment in the laboratory. Second, having demonstrated the feasibility of the approach, we tested the entire monitoring system on a healthy subject for one day.

#### A. Validation of the time synchronization accuracy

In a preliminary phase, we verified the M/S synchronicity. The free-running timer is visualized by measuring the GPIO toggling rate on an oscilloscope. Setting the ST $T_{ST} = 100 \mu s$ and toggling the GPIO every 50 ticks results in a $T_f = 10 \text{ ms}$ nominal square-wave output period ($f_t = 100 \text{ Hz}$). The reported results show that at the beginning of the measurement campaign, each unit has a different frequency, quite far from $f_t$, as reported in Table II and Figure 6a. This measure of the RTC mismatch is due to many reasons, including the tolerance of the low-cost crystal oscillator used, the aging of the component, the PCB parasitics, etc [13]. After the synchronization phase, the master imposes its ST to the slaves, as shown in Table II and Figure 6b. It is worth noting that the STs are synchronous, despite the clocks being not synchronized and having, in any case, a $\pm 100 \mu s$ uncertainty. The $\pm 100 \mu s$ uncertainty is now related to the master clock time.

In this measurement campaign, we also evaluated what happened during the transient of the synchronization phase. Figure 7b sketches the experimental setup. A PC was connected to the master using Bluetooth Low-Energy (BLE) to issue the

| TABLE II: Comparison for the values of the Master and Slaves $f_t$ |
|-------------------|-----------------|-----------------|-----------------|-----------------|
|                   | Before the synchronization phase |                   |                   |                   |
| Master [Hz]       | Slave#1 [Hz]    | Slave#2 [Hz]    | Slave#3 [Hz]    |
| 72.84             | 75.86           | 93.62           | 83.95           |
| After the synchronization phase |                   |                   |                   |
| Master [Hz]       | Slave#1 [Hz]    | Slave#2 [Hz]    | Slave#3 [Hz]    |
| 72.84             | 72.84           | 72.82           | 72.82           |

Fig. 4: The custom designed boards. In detail: (a) The HRSpO2board-HW-1-0-0 board that has role of master in the proposed system (b) the sEMG-HW-1-0-0 and (c) IMUboard-HW-1-0-0 boards having slave roles.
ST start command. The master, then, sent a packet, using a proprietary protocol, to the set of slaves, starting their own STs (see 7a). After that, the master waited for the previously calculated interval, $\Delta T_{\text{SYNC}}^M$, to start its own ST, thus ensuring the simultaneous start for all the units in the system. Starting from this phase, each device wrote on the microSD card every 10 ms its own timestamp. The field dedicated to the sensor was kept empty to prevent recording any possible delay associated with the availability of the data from the sensors.
TABLE III: Synchronization experimental results.

<table>
<thead>
<tr>
<th>Slave #1</th>
<th>Slave #2</th>
<th>Slave #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>max $S_s$</td>
<td>253</td>
<td>229</td>
</tr>
<tr>
<td>min $S_s$</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>avg $S_s$</td>
<td>82</td>
<td>87</td>
</tr>
</tbody>
</table>

We used a $T_M = 10$ s and imported the data in MATLAB for analyzing the ST variation of each unit. In the forthcoming figures, for sake of clarity, we reported only the STs of the master and only a single slave. Figure 5a reports the overall ST variation while Figure 5b zooms on the initial phase, highlighting how eventually the slave synchronizes its timestamp to the master even if initially it was faster. To achieve this result, it changes its fractional ST parameters to become slower than the master. It is worth highlighting that in Figures 5a and 5b the slave samples have been normalized to the master ones and, then, interpolated. Fig. 5c shows the timestamp difference between master and slave, while Figure 5d zooms on an apparent noise generated by the fact that, as discussed in Section II, the STs have been synchronized but not the clocks.

In the second group of experiments, the units operated continuously for four weeks in the laboratory, to verify the effectiveness of the already tested algorithm [5], with the addition of the new mutual M/S synchronization phase. Table III summarizes the measured results, where we recall that $S_s(j)$ denotes the skipped slots for node $j$. The obtained values emphasize the perfect synchrony of the slaves with the master. Each slave stored in a microSD the time in which the synchrony is lost and, it assumes not to be in synchrony with the master if its ST is 500 $\mu$s greater or lower than that received by the master. The apparently low value of the means is due to the fact that the FLSA, every time it loses a single packet, it resets itself. The results are very good, showing that the slaves work for almost 9.5 hours before losing the synchrony, with an evident power consumption reduction.

B. System validation

In this last group of measurements, the units have been used in a real scenario, with a healthy subject wearing them for a 1-day long monitoring campaign. The units have been positioned as shown in Fig. 8a, with the goal of measuring the gait, the calf muscle activity, the subject posture, the HR and the SpO$_2$. The master is positioned in a hat on the subject head. This position gives the minimum interference with everyday activities, allowing a correct measurement of HR and SpO$_2$ [14].

Fig. 8: Overview of the real-world scenario experiments. In detail: (a) Positioning of master and slaves boards (b) Analysis of HR, SpO$_2$ and rolls data collected through the experiments (c) Close-up on the gait analysis
Figure 8b reports an excerpt of the data processing during the campaign recording. In the upper part there are shown the collected HR and SpO2 values, recorded each second, while in the lower part there are shown the roll angles calculated in MATLAB R2021b starting from the shin-bones data recorded by the slaves on the left and right sides. The left and right sEMG are reported, too. The recorded data can be easily processed through a PC, allowing for finer and more complex analyses correlating the data recorded by many units.

In Fig. 8c the observation time has been reduced, giving better insight into the elaborated data. The shin-bones angles during the walk, along with the muscles activity, are clearly shown. These values are used for gait analysis, showing the perfect symmetry between right and left legs in a healthy subject. The result indirectly validates that the data collected by different units placed in different body regions still maintain perfect synchrony due to the proposed algorithm.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a distributed full synchronized system for monitoring patients’ global health, along with an evolution of the FLSA. In particular, this latter provided for the synchronization not only among the slaves but also with the master. The system was first tested in the laboratory and then in a one-day-long test on a healthy subject. The achieved results in terms of synchronization accuracy demonstrated how the improved version of FLSA allows reaching low synchronization errors and it is stable with time. Also, the preliminary results on a healthy patient demonstrated how the algorithm performs well in a real-world scenario.

It is beyond the scope of this paper to make an in-depth analysis of the measurement campaign on biological signals itself. However, in future developments, the system will be integrated with drug therapy and posture or gait analysis to evaluate the efficacy of the treatment [15].

REFERENCES


Giuseppe Coviello was born in Bari, Italy in 1981. He received the Laurea degree in electronic engineering in 2006 and the Ph.D. degree in Electric and Information Engineering in 2015 from Polytechnic University of Bari, Italy. He is currently assistant professor at the Department of Electrical and Information Engineering of the Polytechnic University of Bari. His research interests include RF and mixed-signal circuits and embedded systems for healthcare.

Antonello Florio was born in Bari (Italy) in 1985. He received the BSc in Computer and Automation Engineering and the MSc in Telecommunications Engineering both with honors from Polytechnic University of Bari (Italy). He took part to a Double Degree program with University of Nice-Sophia Antipolis (France) where he achieved the Master de Recherche in Ubiquitous networking and computing. He is currently a Ph.D. student in Electronics at Polytechnic University of Bari. His research interests involve phased arrays, theories and techniques for Localization and Green Wireless Sensory Networks.
Gianfranco Avitabile (M’08–SM’19) was born in Livorno, Italy in 1958. He received in 1982 the B.S. and M.S. degrees in electronics from the Florence University, Italy. He joined soon after the Department of Electronics of University of Florence as Assistant Professor. From the 1998 he joined the Department of Electronics of the Polytechnic University of Bari as Associate Professor, where currently works. His fields of interest are mainly the high frequency circuits and systems for both civil and military applications and high-performance analog integrated circuits for telecommunication applications.

Claudio Talarico received the B.S. and M.S. degrees in electrical engineering from the University of Genoa, Italy, and the Ph.D. degree in electrical engineering from the University of Hawaii. He is currently a Professor of electrical and computer engineering with Gonzaga University. Before joining Gonzaga University, he was with Eastern Washington University, the University of Arizona, and with industry, where he held both engineering and management positions at Siemens Semiconductors, IKOS Systems, and Marconi Communications. His research interests include digital and mixed analog/digital integrated circuits and systems, computer-aided design methodologies, and design and analysis of systems-on-chip.

Janet M. Wang-Roveda is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Arizona in Tucson. She received her M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1998 and 2000, respectively. She was a recipient of the NSF career award and the Presidential Early Achievement Award for Science and Engineering at White House in 2005 and 2006, respectively. She was the recipient of the 2008 R. Newton Graduate Research Award from the EDA community, and the 2007 USS University of Arizona Outstanding Achievement Award. Her primary research interests focus on robust VLSI circuit design, biomedical instrument design, Smart grid, VLSI circuit modelling/design and analysis, and low power multi-core system design.