

Ultra-Low Power Estimation of Heart Rate Under Physical Activity Using a Wearable Photoplethysmographic System

Elisabetta De Giovanni, David Atienza

Embedded Systems Laboratory (ESL)

École Polytechnique Fédérale de Lausanne (EPFL)

Lausanne, Switzerland

Email: {elisabetta.degiovanni, david.atienza}@epfl.ch

Srinivasan Murali, Francisco Rincon

SmartCardia GmbH

Lausanne, Switzerland

Email: {srinivasan.murali, francisco.rincon}@smartcardia.com

Abstract—In the last years, the need for enhancing health and preventing problems with remote monitoring is increasing. A non-invasive low-cost technique for processing bio-signals and monitoring vital parameters, at rest and during physical activity, is the use of wearable PhotoPlethysmoGraphic (PPG) systems. However, in order to detect a relevant vital parameter, such as the heart rate during demanding exercises, motion artifacts must be removed from the signals retrieved.

In this paper, we present a fast and easy to implement algorithm to estimate the heart rate value which does not need to reconstruct the noise-free signal nor does it apply adaptive filtering as existing algorithms, thus gaining computational time and stored memory space. The method consists of applying the Fast Fourier Transform on short windows of data and removing motion artifacts relying on single-sided amplitude spectrum analysis of PPG and 3-axis accelerometer signals. The results show that our algorithm manages to remove a wide range of motion artifacts achieving an average absolute error of only 1.27 BPM between the heart rate estimated by the algorithm every second and the ground-truth value. The method was successfully implemented on a wearable PPG device achieving an execution time of 226 ms per second, hence obtaining a battery lifetime of 9.37 days.

Keywords—PhotoPlethysmoGraphic (PPG); wearable device; heart rate estimation; physical activity;

I. INTRODUCTION AND RELATED WORK

New approaches to improve and prevent health issues are being developed and start to enter the healthcare and wellness fields. A well spread example is the domain of Wireless Body Sensor Network (WBSN) systems which can detect vital parameters as heart rate, blood pressure, respiration and activity level. Moreover, monitoring the mentioned parameters gives information about activity and fitness level for healthy people.

Different companies are producing WBSN, such as Polar [1], which provides a heart rate sensor positioned on a chest belt connected via Bluetooth to a fitness mobile application that shows the variation of the heart rate. Other chest belts are provided by MyZone [2], Wahoo Fitness [3], Suntoo [4]. These devices use electrodes for heart rate monitoring whose cost is higher than other techniques which can reach the same accuracy, such as PhotoPlethysmoGraphy (PPG).

Wearable PPG systems apply a non-invasive, low-cost, optical technique which detects the blood volume changes in vessels. They contain a sensor which consists of a Light Emitting Diode (LED) illuminating the skin and a photodetector receiving the light reflected from the tissues. The alternating current component of the PPG waveform gives information about the cardiac rhythm, therefore it is possible to estimate the heart rate within the standard range between 0.67 Hz and 3.67 Hz, i.e. 40 BPM (Beats Per Minute) and 220 BPM, considered in this work.

During exercises and physical activities, the PPG signal can be affected by strong motion artifacts (MA) in the same range of frequencies, which must be removed to make accurate heart rate estimation. There are commercial wrist-based devices for fitness applying algorithms for MA removal, such as Mio Alpha 2 [5], which presents a high accuracy but a large variation from subject to subject [6]. Apart from the commercial world, different research studies present various methods for estimating heart rate in corrupted PPG signals. One of the common methods used is the periodic moving average filter, based on the quasi-periodicity of the PPG signal [7]. This filter segments the signal into periods and resamples each period. However, in-band noise occurs when the spectra of the MA and the PPG one overlap. Another technique is using acceleration-based adaptive filters. They require a reference signal to minimize the mean squared error between the filter output and the reference [8]. One relevant framework for motion removal from PPG signal is TROIKA [9], which claims that the three steps of the method are necessary for this purpose. The method applies a signal decomposition, which partially removes MA frequency components and reconstructs the noise-free signal. In addition, it applies a sparse signal reconstruction for high-resolution spectrum estimation, which requires solving an optimization problem to improve performance. The last step is a spectral peak tracking with verification, which analyzes the PPG spectrum to detect the heart rate as a peak and verifying it looking at previous windows. The method shows an average absolute error (see Section IV) for heart rate estimation of 2.34 BPM, with a standard deviation of 0.82

BPM. Another work was proposed by the same authors [10], but it is not designed for embedded devices as we target in this work.

In this paper, we propose a method for monitoring heart rate in real-time which analyzes the spectrum retrieved from the Fast Fourier Transform (FFT). In order to gain computational time, speed and memory space and decrease power consumption in the embedded device, we avoid noise-free signal reconstruction, focusing on the detection of MA as peaks in the range of frequencies previously mentioned. Integer arithmetic is employed as well to reduce execution time. The method works on short windows of data, which makes it applicable to real-time processing on WBSN devices. It does not require a reference signal, as the history of estimated values is used to update the current one. The method detects a wide range of MA and it manages to estimate the heart rate when PPG and MA spectra overlap. It shows an average absolute error of only 1.27 BPM with a standard deviation of 0.91 BPM. We compute the execution time of the algorithm on-board, i.e. 226 ms per second, which grants a battery autonomy of 9.37 days for the fully working device.

The paper is divided as follows. In Section II and III, we present respectively the structure of the proposed method and optimizations for the on-board implementation. In Section IV, we describe the set-up of the experiments provided in the IEEE Signal Processing Cup (SPC) 2015 [11] and the features of the final WBSN implementation of the algorithm. In Section V, we validate the method applied in post-processing on the dataset used and we show the results of the implementation on device in terms of accuracy, execution time and power consumption related to it. Finally, we present the conclusions of the work in Section VI.

II. PROPOSED ALGORITHM FOR HEART RATE ESTIMATION

During physical activity PPG signal is affected by MA and the frequency spectrum shows different peaks corresponding to it aside from the pulse. Fig. 1 shows an example of PPG signals of a subject at rest (a) and during running (c), and the corresponding single-sided spectra (b, d). In the spectrum (d), the peak due to the MA is highlighted and shows a high amplitude compared to the peak corresponding to the heart rate.

A simple method to detect and remove MA is adopting a 3-axis accelerometer. This gives information about proper acceleration in a 3-axis reference system due to the movement of the body part where the system is worn. Moreover, the selection of the wavelength of the LED employed is relevant to decrease MA in the PPG signal. The green light (530 nm) was shown to be more suitable than the red light for monitoring the heart rate in daily life and the PPG has relative freedom from MA compared to the red light [12], [13]. Therefore, we use a PPG system with a green LED

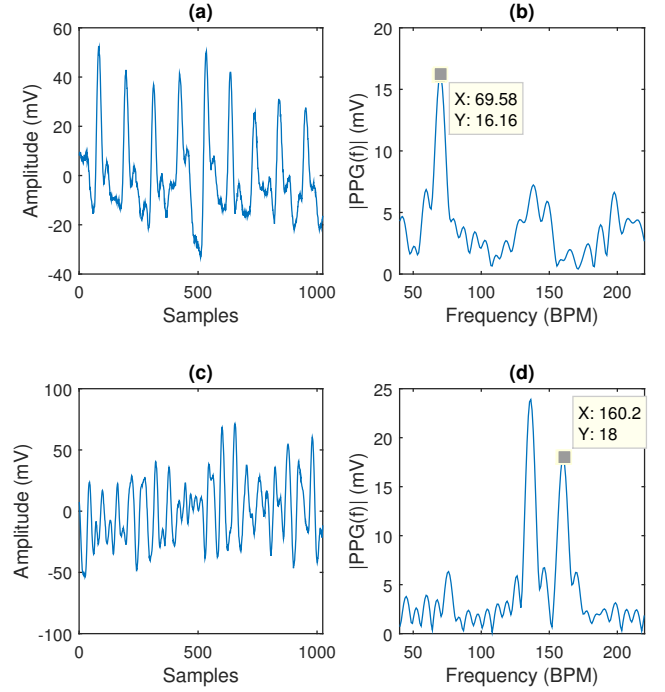


Figure 1. PPG signals and corresponding spectra at a sampling frequency of 125 Hz. (a) represents a PPG signal of a subject at rest and (b) its single-sided amplitude spectrum. (c) represents a PPG signal while the subject is running with strong MA and (d) the corresponding spectrum.



Figure 2. Block diagram of the main steps of the proposed algorithm

in reflectance mode, since our measurement position is the upper arm or the wrist.

The developed algorithm is divided in three main steps, as the block diagram in Fig. 2 shows. These steps are explained in details in Sections II-A, II-B, II-C.

A. Frequency Analysis and Peak Detection

As mentioned in Section I, the FFT analysis enables a reduction of computational time compared to methods that involve adaptive filtering or complex signal decomposition and reconstruction. Since the PPG signal is non-stationary and quasi-periodic, a Fourier series analysis is not directly applicable. It can only be applied in a cycle-by-cycle basis [14]. Therefore, we have chosen the use of the FFT applied on short windows of data (8 s) sliding by 1 s per iteration, assuming that the main frequency is stable.

Fig. 3 shows this first step of the algorithm. The method initially rescales the PPG signal to maximize its dynamic

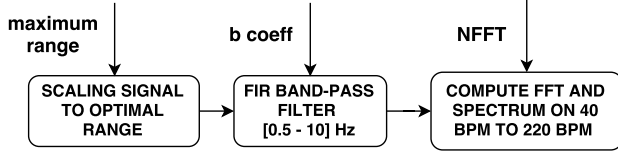


Figure 3. Block Diagram of Step A: Frequency Analysis

range, considering its amplitude and the maximum value that can be represented by a 32-bit integer, which is the data type used to store PPG samples.

A band-pass Finite Impulse Response (FIR) filter is then applied to remove low and high frequencies with cut-off values set at 0.5 Hz and 10 Hz, in order not to lose the range of frequencies needed because of attenuation. FIR filters are more suitable to be implemented with integer arithmetic, as we use, instead of Infinite Impulse Response (IIR) filters, which have problems in scaling the coefficients, losing precision for the filtering step.

The algorithm computes the FFT and the relative spectrum of PPG and accelerometer signals, limiting the range of frequencies to 40 - 220 BPM, as discussed in Section I. The input length of the FFT, shown in Fig. 3, is set to minimize its resolution applying a 7:1 ratio of zero-padding to the window of data. Given a sampling frequency of 125 Hz, a window of 8s (1024 samples) and NFFT of 8192 samples, the resolution will be:

$$res_{fft} = \frac{F_s}{NFFT} = 0.0153 \text{ Hz} \quad (1)$$

where F_s is the sampling frequency of the signals, $NFFT$ is the FFT length. The resolution in BPM is computed as

$$res_{ftbpm} = res_{fft} \times 60 = 0.9180 \text{ BPM} \quad (2)$$

where res_{fft} is the resolution of the FFT in Hz and res_{ftbpm} is the resolution of the FFT in BPM. We use FFT length as power of two because it is faster [15].

The peak detection in the PPG and accelerometer spectra is shown in Fig. 4. We consider the amplitude of the maximum peak in the spectrum and we choose to take the peaks with amplitude greater than a specific percentage of

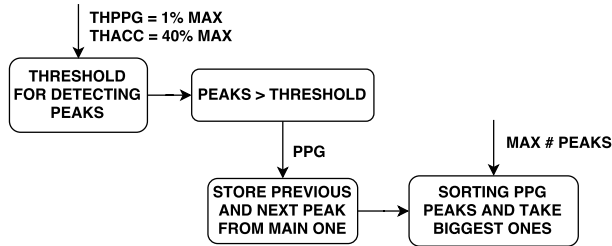


Figure 4. Block Diagram of Step A: Peak Detection

the maximum peak. The thresholds are different for PPG and accelerometer spectra. Since in the PPG the pulse rate can correspond to a low peak compared to the MA, a threshold of 1% of the maximum peak is chosen for this signal. Meanwhile, when the movement occurs, the spectrum of the accelerometer data contains few peaks with high amplitude, therefore the threshold is set at 40% of the maximum peak, value empirically chosen.

The PPG peaks are sorted and the first 7 biggest peaks are chosen from the PPG, though this parameter can be adjusted as needed. The algorithm stores the main and second peaks from the sorted array and, considering the main one, the previous and next peaks from the non-sorted array.

B. Motion Artifacts Removal

The second step of the algorithm consists of the analysis of the PPG and accelerometer peaks found in the previous step. During movement, in the array of PPG peaks detected, MA appear in the range of frequencies as they appear in the three arrays of accelerometer peaks. As the PPG spectrum is a superimposition of different frequencies due to the pulse rate and MA, the frequency at which the movement occurs is not exactly the same as the one appearing in the accelerometer spectrum. Therefore, in order to detect MA in the PPG spectrum, the algorithm sets a range of fluctuation of the frequency of the movement occurring in the PPG spectrum. This interval depends on the frequency in the accelerometer spectrum.

Algorithm 1 MA removal

```

1: for  $i = 0$  to  $lengthPeaks_{PPG}$  do
2:   for  $j = 0$  to  $lengthPeaks_{ACC}$  do
3:     if  $p_{ACC}(j) \times 0.98 < p_{PPG}(i) < p_{ACC}(j) \times 1.02$ 
       then
4:       if  $p_{PPG}(i) \neq mainPeak_{PPG}$  then
5:          $p_{PPG}(i) = 0$ 
6:          $value_{PPG} = 0$ 
7:       end if
8:     else
9:       if  $p_{ACC}(j) < 95$  &  $flagPeak = 1$  then
10:         $p_{PPG}(i) = 0$ 
11:         $value_{PPG} = 0$ 
12:      end if
13:    end if
14:  end for
15: end for
{This code is executed for X, Y, Z of the accelerometer}
  
```

In Algorithm 1, the condition in line 3 shows the interval in which MA are detected in the PPG spectrum. Hence, every peak of the PPG should have a fluctuation of $\pm 2\%$ from one of the peaks in the accelerometer. Fig. 5 shows the spectra of both PPG and X-axis accelerometer. The peak highlighted in the figure, with value 160.2, is removed

because is lower than $159.3 \times 1.02 \approx 162.49$, as the condition requires. Lines 5 and 6, 10 and 11 show how the peak is removed.

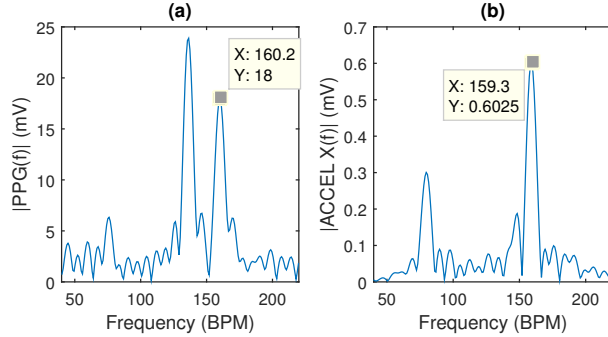


Figure 5. Peak recognition in PPG spectrum when it is in the range of one of the accelerometer peaks. (a) PPG spectrum. (b) Accelerometer spectrum.

As shown in line 4, if the movement occurs, the algorithm checks if the PPG peak analyzed is the main one. The conditions in line 9 correspond to two events: the main peak of the PPG corresponds to the horizontal movement of the arm or the wrist ($p_{ACC}(j) < 95$) and the pulse rate is in the neighbourhood of the main peak ($flagPeak = 1$). We analyze these events sequentially.

If we take a 3-axis reference system on the upper arm or the wrist and consider the Z axis as the vertical movement, the frequency along Z corresponds to the step frequency, unlike the horizontal one which corresponds to half of the step frequency, i.e. to complete an arm swing two footsteps are necessary [16]. While running, the heart rate can reach values from 70% to 90% of the maximum heart rate (from 105 BPM to 140 BPM as minimum value) [17], far from the maximum frequency of horizontal movement considered. The step frequencies range reached during this activity starts from 2.2 Hz (fast walking, corresponding to 132 BPM) to 3.2 Hz (190 BPM) or even more, as extrapolation of previous studies [18]. Therefore, if there is a peak in the accelerometer at half of the frequency of the maximum considered (190 BPM, that is 95 BPM), it is removed from the PPG peaks.

On the other hand, while running, if the pulse rate gets closer to the step frequency, the MA peak should be removed under precise conditions. In this case, the peak is close to the MA, though with a low amplitude, or merged with the accelerometer peak. As shown in Algorithm 2, the method checks the neighbourhood of the PPG main peak. Since the peaks greater than 95 BPM are already removed if they represent the MA, the algorithm checks if the conditions at line 5 are verified. In the first one, $dist(p_{PPG}, mainPeak_{PPG})$ represents the distance between every peak and the main one and 10 BPM is the maximum distance considered. In the second condition, $height_{PPG}$ represents the amplitude

of a peak in the spectrum and the threshold for detecting a neighbour peak is set at 20% of the maximum amplitude $height_{mainPeak}$. If the conditions are verified, the algorithm assigns 1 to $flagPeak$ (line 6). This flag shows that a peak exists in the neighbourhood, which is likely to be the real heart rate, therefore the PPG main peak, representing the step frequency, is removed.

Algorithm 2 Check neighbourhood for close peaks

```

1: if  $lengthPeaks_{PPG} > 1$  &  $mainPeak_{PPG} > 95$  then
2:   for  $i = 1$  to  $lengthPeaks_{PPG}$  do
3:     if  $p_{PPG}(i) > 95$  then
4:        $dist = dist(p_{PPG}(i), mainPeak_{PPG})$ 
5:       if  $dist \leq 10$  &  $height_{PPG} > 0.2 \times$ 
          $height_{mainPeak}$  then
6:          $flagPeak = 1$ 
7:       end if
8:     end if
9:   end for
10: end if
    {This code is executed before the MA removal}

```

It can occur that, in the neighbourhood, close peaks are not detected. In this case the heart rate value is likely merged with the PPG main peak. To detect it, the algorithm computes the 2nd order derivative of the spectrum and then finds the peaks. Algorithm 3 shows the details of this case.

Algorithm 3 Check neighbourhood for hidden peaks

```

1: if  $flagPeak = 0$  then
2:   compute 2nd order derivative
3:   find peaks 2nd order derivative
4:   if  $dist(peak_{prev}, mainPeak_{PPG}) > 15$  then
5:      $peak = peak_{prev}$ 
6:   end if
7:   if  $dist(peak_{next}, mainPeak_{PPG}) > 15$  then
8:      $peak = peak_{next}$ 
9:   end if
10:   $dist = dist(peak, mainPeak_{PPG})$ 
11:  if  $dist \leq 12$  &  $height_{2nd_{peak}} > 0.2 \times$ 
     $height_{2nd_{mainPeak}}$  then
12:     $flagPeak = 1$ 
13:  end if
14: end if
    {This code is executed before the MA removal and after
    checking close peaks}

```

The method detects the direction of the hidden peak by checking if the distance between the previous or next peak and the main peak in the original spectrum is greater than 15 BPM, as shown in the conditions in lines 4 and 7. If the conditions in line 11 are verified, the new peak is added in the PPG peaks and the main peak is removed

($flagPeak = 1$). The conditions are similar to the ones in Algorithm 2, except that the neighbourhood size checked is larger in order to detect the hidden peak.

The parameters of distance and threshold on the amplitude are set depending on the resolution of the FFT and the corresponding spectrum. In this case they are fixed, though they can be adjusted changing the resolution, statically or dynamically.

After removing MA, the algorithm checks the main and second peak in case errors in the signal occur. Hence, an error is detected as shown in Algorithm. 4.

Algorithm 4 Check previous 5 heart rate values

```

1: while  $m > 0$  do
2:   if  $dist(mainPeak_{PPG}, hr_{prev}(m)) \geq 5 \times sec$  then
3:      $flagHROK = 0$ 
4:   end if
5:    $m = m - 1$ 
6:    $sec = sec + 1$ 
7: end while
   {This code is executed after the MA removal}

```

The algorithm sets a maximum variation of the current heart rate from the one in the previous window to 5 BPM. We set this value because it is unlikely that in one second the heart rate varies of more than 5 BPM. Line 2 shows the distance from the 5 previous windows. Therefore, the maximum variation from the 5th previous windows is 25 BPM, from the 4th 20BPM, etc. If these conditions are not verified, the algorithm sets $flagHROK$ to 0 (line 3). This flag is used in the next step of the algorithm, where the current heart rate is updated considering the previous window, to contain the variation of heart rate in case of errors.

C. Adjustment and Updating of the Heart Rate Value

In Section II-B, the algorithm sets $flagHROK$ to 0 if the main peak and second peak of the PPG are not in the maximum range of variation of heart rate considered, after removing MA. This can happen when the acquired PPG signal is noisy compared to the accelerometer one, therefore the peaks found in the PPG spectrum are mainly MA related. Algorithm 5 shows the third step of the method.

The algorithm manages the accumulated error by decreasing the maximum variation of the current heart rate from the previous window to 2 BPM. The direction of the variation is chosen as the position of the current PPG main peak from the previous heart rate value ($sign(mainPeak_{ppg} - HR_{old})$). The heart rate value in the current window is updated as shown in the formula in line 2, where HR_{old} is the heart rate value in the previous window.

If no error occurs, the algorithm checks the distance between every remaining peak and the previous heart rate. The closest one is chosen as the suitable heart rate value.

Algorithm 5 Adjusting considering previous window and updating

```

1: if  $flagHROK = 0$  then
2:    $HR_{new} = HR_{old} + sign(mainPeak_{ppg} - HR_{old}) \times 2$ 
3: else
4:   if  $lengthPeaks_{PPG} > 0$  then
5:      $dist_{toLastPeak} = 1000$ 
6:     for  $i = 0$  to  $lengthPeaks_{PPG}$  do
7:        $dist = dist(ppg(i), HR_{old})$ 
8:       if  $ppg(i) \neq 0$  &  $dist \leq dist_{toLastPeak}$  then
9:          $HR_{new} = HR_{old} + sign(ppg(i) - HR_{old}) \times$ 
            $min(dist, 5)$ 
10:         $dist_{toLastPeak} = dist$ 
11:      end if
12:    end for
13:  end if
14: end if
15: if  $lengthPeaks_{PPG} > 0$  then
16:    $HR = HR_{new}$ 
17:    $HR_{old} = HR_{new}$ 
18: else
19:    $HR = HR_{old}$ 
20: end if
21:  $hr_{prev}(pr) = HR$ 
   {This code is executed after removing MA and checking
   5 previous windows}

```

In this case, the current window is updated as shown in the formula at line 9, where $dist$ is the distance of every PPG peak from the old heart rate. If the distance is lower than 5 BPM, the algorithm updates the value using $dist$, otherwise it updates the heart rate by 5 BPM.

In the end, if the algorithm does not detect any peaks, it assigns the value of the previous window as the current heart rate, as shown in line 19.

In Section III we present the optimizations applied for the implementation.

III. OPTIMIZATIONS FOR WBSN EXECUTION

Two main optimizations of the algorithm presented in Section II for WBSN execution are achieved and presented in this section. The first one is lowering the execution time of a routine. In this work, the main portion of computation time gained is related to the FFT routine. The algorithm uses a fixed-point short integer FFT [19] faster than the one using floating-points to achieve this goal. Applying the integer arithmetic can also affect the memory space allocated for the computation, which is the second optimization.

The device contains a PPG sensor and a 3-axis accelerometer sampling data at 125 Hz and 250 Hz and a ultra-low power 32-bit microcontroller unit (MCU) with a 48 KB RAM. The memory portion stored at 125 Hz for both PPG

and accelerometer, downsampled in order to have the same sampling frequency, is calculated as addition of equations 3 and 4. The memory space related to the window of data where we apply the FFT is:

$$mem_{wind} = 2 * 1024 * 4 + 2 * 3 * 1024 * 2 \simeq 20 \text{ KB} \quad (3)$$

where 1024 represents the length of the window buffer of 8,192 s of data (power of two used for the speed of the FFT) at the specified sampling frequency, in samples. The value is multiplied by 2 because we use a circular buffer for the changing window and a sorted buffer as actual input to the algorithm. The first adding represents the PPG data, which consist of 22-bits per sample, therefore we store integer values each occupying 4 bytes of memory. The second adding represents the 3-axis accelerometer data, short integer values each occupying 2 bytes of memory. The memory space related to the sliding window of data used for updating the heart rate value is:

$$mem_{slide} = 2 * 128 * 4 + 2 * 3 * 128 * 2 \simeq 2.5 \text{ KB} \quad (4)$$

where 128 represents the length of the sliding buffer of 1,024 s of data at the specified sampling frequency, in samples. The value is multiplied by 2 because we use two buffers in the interrupt routine in order to execute the algorithm routine while the signal is sampled.

As the memory portion stored at 125 Hz is almost half of the memory available, we reduce the sampling frequency to 31.25 Hz which, according to the Nyquist-Shannon sampling theorem, can represent a bandwidth of $\frac{F_s}{2}$, that is 16 Hz, greater than the maximum frequency of 3.67 Hz considered in the algorithm. We manage to reduce of $\frac{1}{4}$ the memory stored compared to 125 Hz of sampling rate, that is 5.8 KB.

In Section V, we report the results of the implementation on device compared to the signals provided by the IEEE SPC 2015 resampled at 31.25 Hz. The results of the gold standard in post-processing to validate the algorithm are shown at both 125 Hz and 31.25 Hz.

IV. EXPERIMENTAL SET-UP

The first validation of the algorithm was conducted in Matlab R2014b and involved twelve datasets provided by the 2015 IEEE SPC. They were recorded when subjects performed various physical exercises. Two-channel PPG signal and 3-axis accelerometer were recorded from the subject's wrist and on-channel ElectroCardioGram (ECG) from the subject's chest as ground-truth of the heart rate, each sampled at 125 Hz. As the heart rate, output of the ECG recorded, is updated every 2 s, the output of the algorithm is also plotted every 2 s, even if computed every second. We used only one of the two PPG channels and the 3-axis accelerometer data. The algorithm is applied on signals sampled at 125 Hz and downsampled at 31.25 Hz, on a data

window of 8 s. For the validation, the output of the algorithm is plotted compared to the ECG ground-truth value. Two type of analysis are conducted to show the behaviour of the algorithm: Average Absolute Error (AAE) and median value.

$$AAE = \frac{1}{N} \times \sum | HR_{alg}(i) - HR_{true}(i) | \quad (5)$$

where N is the number of window steps considered, $HR_{alg}(i)$ is the output of the algorithm at each step and $HR_{true}(i)$ is the ground-truth value from the ECG. The AAE is used in order to compare it with the existing algorithms, while the median shows the difference of the two values not biased by small or big values. It is computed considering the values estimated and the ECG after 15 steps, that is 30s, giving the algorithm time to reach stability.

The on-board processing was implemented in C, using the device mentioned in section III. The sampling of PPG and accelerometer signals were simulated storing static arrays of 30 s of data for the twelve subjects considered and filling the buffers in the interrupt routine.

The execution time of the algorithm routine is computed on one of the signals on 30s window of data. The power consumption of the device is computed considering the duty cycle of the PPG sensor for both receiving and transmission modes, the accelerometer, the microcontroller active and sleep mode and the execution of the algorithm for heart rate estimation.

The Section V presents the results of post-processing and embedded execution of the algorithm and reports the indicators to validate it.

V. RESULTS AND VALIDATION

Table I shows the AAE of the twelve subjects and the median value for both sampling frequency mentioned in Section IV. It shows also the average and standard deviation of the AAE and median value for the twelve subjects.

Table I
ANALYSIS ON TWELVE SUBJECTS OF SPC 2015 - AAE AND MEDIAN IN BPM FOR SIGNALS SAMPLED AT 125 HZ AND 31.25 HZ

	Analysis in BPM			
	125 Hz		31.25 Hz	
	AAE	Median	AAE	Median
S1	1.62	0.91	1.87	1.40
S2	1.42	0.71	2.97	2.19
S3	1.26	0.65	2.08	1.82
S4	1.40	0.49	2.53	1.73
S5	0.61	0.39	1.56	1.48
S6	1.55	0.60	2.00	1.30
S7	0.47	0.45	1.24	1.18
S8	0.43	0.39	2.10	1.7
S9	0.36	0.31	1.43	1.23
S10	3.78	2.03	4.99	3.04
S11	1.14	0.80	1.49	1.31
S12	1.16	0.77	2.60	2.20
MEAN	1.27	0.7	2.24	1.71
STD	0.91	0.46	1.01	0.54

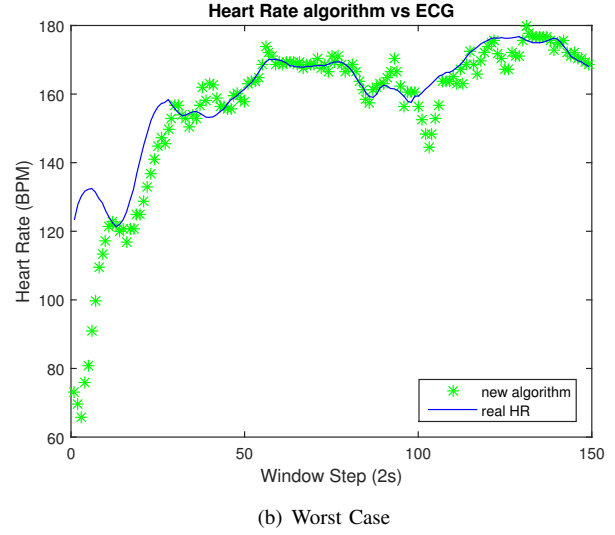
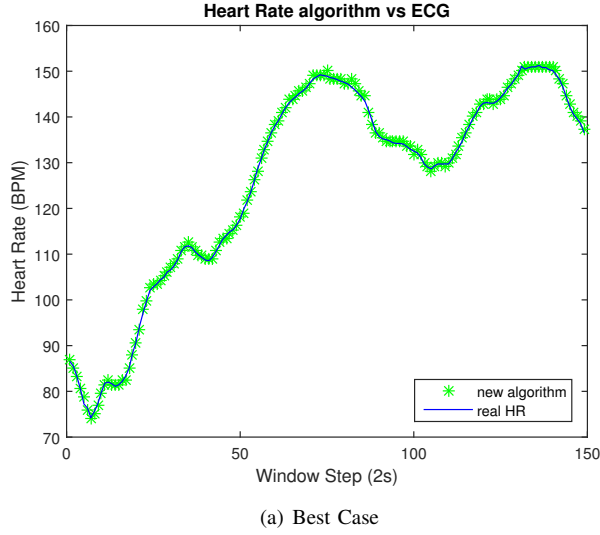


Figure 6. Heart rate estimated compared to ECG true heart rate for best case subject (a) and worst case subject (b), for a sampling frequency of 125 Hz

Fig. 6(a) and Fig. 6(b) show the results for two of the twelve subjects, the best and worst case, for a sampling frequency of 125 Hz. As shown in Table I, at 125 Hz, the AAE for the twelve subjects is **1.27 BPM \pm 0.91 BPM**, which compared to the TROIKA firmware (2.34BPM, see section I) is 1BPM lower and the maximum value ($1.27+0.91 = 2.18\text{BPM}$) of AAE is 0.16BPM still lower than the TROIKA result. Eleven subjects out of twelve show an AAE lower than 1.7BPM, while an outlier shows an AAE of 3.78BPM, which compared to the TROIKA is only 1BPM more, acceptable considering the advantages of the method. The outlier is shown in Fig. 6(b): after reaching stability the algorithm follows the ground-truth pretty well, even in the worst case subject. The best subject is shown in Fig. 6(a), which has an AAE of 0.36BPM. The median value in the worst case is 2.03BPM and in the best case is 0.31BPM, showing that the unbiased difference between the two values is very low. At 31.25 Hz, the AAE for the twelve subjects is $2.24\text{BPM} \pm 1.01\text{BPM}$, 1BPM more than the one at 125 Hz, because of the resampling precision. The value compared to the TROIKA framework is still lower. The median value is lower than the AAE, because not biased from small or big values. The outlier has an AAE higher than the mean value but it is still acceptable considering the advantages for real-time processing. The results show clearly that it is possible to avoid signal reconstruction and focus directly on the spectra of the PPG and accelerometer and relative peaks, obtaining high performance in terms of accuracy.

Therefore, the method is suitable to be implemented on embedded system and Table II shows the absolute error between the post-processing results and the same data streamed on a real wearable device. The average error has a

mean value of **0.42 BPM** which is due to the approximation precision for using integer arithmetic implementation on device, while in Matlab the heart rate is computed as floating-point value.

Table II
ABSOLUTE ERROR BETWEEN HEART RATE VALUE COMPUTED IN POST-PROCESSING AND THE ONE COMPUTED IN THE EMBEDDED DEVICE

	S1	S2	S3	S4	S5	S6
Absolute Error (BPM)	0.41	0.37	0.3	0.2	0.26	0.27
	S7	S8	S9	S10	S11	S12
Absolute Error (BPM)	0.36	0.24	0.31	0.4	0.24	1.72

Table III
AVERAGE CURRENT CONSUMED FOR HARDWARE COMPONENTS AND ALGORITHM PROCESSING

	Current (mA)	Time (s)	Average current (mA)
PPG Tx On-Rx On	1.225	10%	0.123
PPG Tx Off-Rx On	0.6	10%	0.06
Accelerometer	0.5	100%	0.5
Baseline current	0.045	100%	0.045
HR processing	10.5	23%	2.415
Sleep mode	0.018	77%	0.014
Total			3.16

We computed the execution time of the algorithm routine, obtaining an average of **226 ms** per second, which is the time between two outputs of the algorithm. Table III shows

the total average current computed considered the features mentioned in section IV in order to retrieve the total amount of power consumed while the device is fully working.

Considering the battery rating of 710mAh we successfully obtain a battery lifetime of **9.37 days**.

VI. CONCLUSIONS

The demand for improvement and prevention of acute or chronic problems and enhancing health through remote vital parameter monitoring has increased the manufacturing of wearabel device. In particular, heart rate monitoring using wearable PhotoPletysmoGraphic (PPG) systems have the advantage of being non-invasive, due to the optical sensor, and low-cost.

One challenge is to detect the heart rate during physical activities from a PPG signal affected by MA. We proposed a method which applies the Fast Fourier Transform on short windows of data and removes MA depending on the spectrum of PPG and 3-axis accelerometer signals, avoiding complex reconstruction of the signal or adaptive filtering. The results showed that the algorithm removes a wide range of MA achieving a high degree of accuracy. The method was suitable to be implemented in a wearable embedded device running 22.6% of the time between two heart rate values. The device can fully work for 9.37 days, including algorithm processing.

VII. ACKNOWLEDGMENTS

This work has been partially supported by the Hasler Foundation (project no. 15048), and the BodyPoweredSenSE RTD project (no. 20NA21 143069), evaluated by the Swiss NSF and funded by Nano-Tera.ch with Swiss Confederation financing.

REFERENCES

- [1] "Heart Rate Monitors, activity trackers and bike computers." [Online]. Available: <http://www.polar.com/us-en>
- [2] "Myzone." [Online]. Available: <http://www.myzone.org/>
- [3] "TICKR X Heart Rate Monitor with Motion Analytics | Wahoo." [Online]. Available: <http://uk.wahoofitness.com/devices/wahoo-tickr-x-heart-rate-strap.html>
- [4] "Wearable tech for athletes, outdoor adventurers & divers." [Online]. Available: <http://www.suunto.com/>
- [5] "Mio ALPHA 2 Heart Rate Activity Tracker Watch." [Online]. Available: <http://www.mioglobal.com/EN-US/Mio-ALPHA-2-Heart-Rate-Sport-Watch/Product.aspx>
- [6] G. Valenti, K. R. Westerterp, and H. Biology, "Optical Heart Rate Monitoring Module Validation Study," pp. 2–3. [Online]. Available: <http://www.usa.philips.com/>
- [7] H.-w. Lee, J.-w. Lee, W.-g. Jung, and G.-k. Lee, "The Periodic Moving Average Filter for Removing Motion Artifacts from PPG Signals," *International Journal Of Control Automation And Systems*, vol. 5, no. 6, pp. 701–706, 2007.
- [8] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, "Wearable Photoplethysmographic SensorsPast and Present," *Electronics*, vol. 3, no. 2, pp. 282–302, 2014. [Online]. Available: <http://www.mdpi.com/2079-9292/3/2/282/>
- [9] Z. Zhang, Z. Pi, S. Member, and B. Liu, "TROIKA : A General Framework for Heart Rate Monitoring Using Wrist-Type Photoplethysmographic (PPG) Signals During Intensive Physical Exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522–531, 2015.
- [10] Z. Zhang and S. Member, "Photoplethysmography-Based Heart Rate Monitoring in Physical Activities via Joint Sparse Spectrum Reconstruction," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 1902–1910, 2015.
- [11] "2015 IEEE Signal Processing Cup." [Online]. Available: <http://www.zhilinzhang.com/spcup2015/>
- [12] J. Lee, K. Matsumura, K. I. Yamakoshi, P. Rolfe, S. Tanaka, and T. Yamakoshi, "Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 1724–1727, 2013.
- [13] Y. Maeda, M. Sekine, and T. Tamura, "Relationship between measurement site and motion artifacts in wearable reflected photoplethysmography," *Journal of Medical Systems*, vol. 35, no. 5, pp. 969–976, 2011.
- [14] K. Reddy, B. George, and V. Kumar, "Use of Fourier Series Analysis for Motion Artifact Reduction and Data Compression of Photoplethysmographic Signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1706–1711, 2009.
- [15] "Fast Fourier transform - MATLAB fft - MathWorks Schweiz." [Online]. Available: <http://ch.mathworks.com/help/matlab/ref/fft.html>
- [16] M. P. Ford, R. C. Wagenaar, and K. M. Newell, "Arm constraint and walking in healthy adults," *Gait and Posture*, vol. 26, no. 1, pp. 135–141, 2007.
- [17] "Target Heart Rates." [Online]. Available: http://www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/FitnessBasics/Target-Heart-Rates_UCM_434341_Article.jsp
- [18] E. Hirasaki, S. T. Moore, T. Raphan, and B. Cohen, "Effects of walking velocity on vertical head and body movements during locomotion," *Experimental Brain Research*, vol. 127, no. 2, pp. 117–130, 1999.
- [19] "integer_fft - www.jjj.de." [Online]. Available: http://www.jjj.de/crs4/integer_fft.c