The implementation of Fast Hilbert Transform with low pass filter for reconstructing breathing sound signals in LabView

I B I Purnama[1], I M Sumertayasa[1]

[1] Electrical Engineering Department, Politeknik Negeri Bali, Kampus Bukit Jimbaran, Bali, Indonesia

E-mail: ida.purnama@pnb.ac.id

Abstract. Many studies have stated that the respiration rate is one of the vital signs in individual health. It can be used to identify serious illness. Detecting either the chest movement or the exhausted gases properties using various sensors are the common methods to monitor respiration rate. Apart from both, breathing sound seems also effective for such purpose without requiring any complicated sensing devices. However, attaining respiration rate from the burst-type of breathing sound signals is challenging. Their high variations in frequency, amplitude, and the interval between inhalation and exhalation make the reconstruction of a smoother signal needs extra signals processing effort. This paper proposes a technique to reconstruct breathing sound which is acquired using a built-in headset microphone and implemented using Fast Hilbert Transform with Low Pass Filter in LabView. Through adjusting a proper filter order and cut-off frequency, the result shows that the proposed method is able to derive the representation of breathing sound signals in a smooth curve and convenient form. Compare to the raw ones, this reconstructed signals will make the peak detection easier. It is expected that this sound-based approach will be a cheap and effective solution for further respiration rate monitoring systems.

1. Introduction

Breathing is one of the fundamental functions to sustain life in living beings [1]. It is a cyclic process which consists of inhalation and exhalation. Both of these cycle numbers per minute is called respiration rate and often treated as a vital sign of individual health [2]. Many studies have shown that the alteration of the respiration rate can be used for early detection to predict potentially serious clinical events [3]. Furthermore, it can be used to monitor the progression of serious illnesses such as asthma, bronchitis, pneumonia, tuberculosis, lung cancer and chronic obstructive pulmonary diseases (COPD) [1-4]. Hence, continuous and proper respiration monitoring is critically useful. Using various sensors and approaches, recent studies in respiration rate monitoring have rapidly developed.

Generally, the respiration monitoring approach can be classified as invasive and non-invasive. For invasive approach, the sensing device is attached to the subject's body, while it is not for the non-invasive one [5]. The invasive approach mostly implemented the detection of either the chest movement by measuring impedance, a ratio of voltage vs. current phasor, across the chest or the exhausted gases properties by measuring temperature, humidity, particle concentration, and airflow. Studies [6-8] used an accelerometer and pressure sensors attached on the body, while [9] used Thermistor sensor, and [10] used modulating effect on Electrodiagram (ECG) to derive respiration rate. In the non-invasive approach, [11] used an optical technique in vision sensor to detect thoracic movement, while [12,13] used thermal sensor and imaging-based sensor that could follow facial features related to respiration.

Different approaches obviously need different sensing devices. Among those, a headset microphone could be the simplest and most easy-to-find device that can be used for sound-based acquisition.

Since the airflow has a close correlation to the amplitude of the respiratory sound, this makes the respiration rate monitoring based on the breathing sound is possible [14]. Several previous studies attempted to extract the features of breathing sound to perform the sound-based investigation as a new approach for diagnosis, especially for abnormal breathing sounds such as wheeze detection [15], asthma detection [16], and apnea detection [12]. Practically, acquiring breathing sound is relatively easy using microphone, but due to its nature which is highly varied in frequency, amplitude, and duration, as well as in the interval between inhalation and exhalation, make the processing task of this signal is more challenging. Therefore, a better signal representation for further analysis is critically needed.

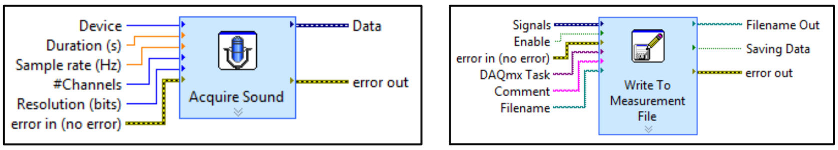
From the aforementioned context, this paper aims to propose a technique to reconstruct breathing sound signals and derive the smooth representation signals by mainly implementing Fast Hilbert Transform (FHT) with Low Pass Filter (LPF) in LabView. In this case, the FHT is applied to envelop the signals, while LPF is used to filter the unwanted signals. It is expected that the smoother signal can be gained to make the peak detection for respiration rate calculation easier and more convenience. Here, LabView is used since it provides understandable data flow in block diagram with many built-in functions and interactive visualization.

2. Materials and methods

This section describes the material and methods used in this paper. It begins with the acquisition of breathing sound, then it is followed by signals envelope using Fast Hilbert Transform (FHT), signals filtering using Low Pass Filter (LPF), and signal reconstruction using Build Waveform Function (BWF).

2.1. Breathing sound signals acquisition

The breathing sound used for this study was recorded using a built-in microphone of earphone and placed underneath the nose. The benefit of using this high definition audio device is no additional microprocessor devices are needed. The Acquire Sound Function in LabView, Figure 1.a, is used to automatically configure an input task, acquires the data, and clears the task after acquisition complete. By wiring data out to the Write to Measurement File Function, Figure 1.b, the sound is then digitized by 16 bits resolution, sampled in 22.05 kHz rate, recorded in one-minute duration, and saved in binary format (.tdms). The resolution, sampling rate, and duration lead to the enormous series of data.



(a) (b)

Figure 1. (a) Acquire sound function, (b) Write to measurement file function.

2.2. Signal envelope using Fast Hilbert Transform

A Hilbert Transform (HT) is a convolutional operator which is commonly referred to as a wideband 90° phase shifter. The filter embedded in the HT performs a -90° and +90° phase shift for positive and negative frequencies in a signal, respectively [17]. The HT of a function x(t) is defined as:

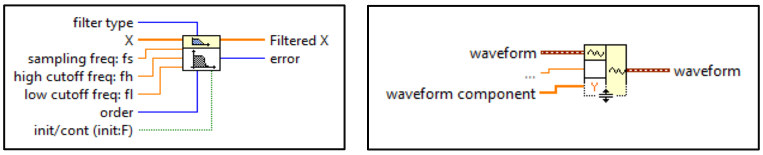
$$h(t) = H\{x(t)\} = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau \tag{1}$$

Using Fourier identities, it can be shown the Fourier Transform of the Hilbert Transform of $x(t)$ is $h(t) \Leftrightarrow H(f) = -j\,sgn(f)\,X(f)$, where $x(t) \Leftrightarrow X(f)$ is a Fourier Transform pair and

$$sgn(f) = \begin{cases} 1 & f > 0 \\ 0 & f = 0 \\ -1 & f < 0 \end{cases} \quad (2)$$

This Hilbert Transform is valuable to calculate instantaneous attributes of a time series signal especially frequency and amplitude before the signal is enveloped by tracking the successive peak values [18].

2.3. Signal filtering using Low Pass Filter (LPF)
Low Pass Filter is a filter that passes signals with a frequency lower than a selected cut-off frequency and attenuates signals with frequencies higher than the cut-off frequency. Here, a digital Butterworth Filter is generated by calling the Butterworth Coefficients VI, Figure 2.a. In the time domain, this digital filter is characterized by its pulse or impulse response. For LPF, the low cut-off frequency holds a critical role. Here, X is the input signal to filter and Filtered X is the output array of filtered samples.



(a) (b)

Figure 2. (a) Low Pass Filter function, (b) Build Waveform function.

2.4. Signal reconstruction using Build Waveform Function (BWF)
The output of LPF, Filtered X, is then used as input components for reconstructing the original signal. Using Build Waveform Function, Figure 2.b, an analog waveform will be built by modifying the original waveform based on the components that are supplied from the output of LPF.

3. Results and discussion
This section displays the results and discusses their significances. First, the block diagram in LabView is presented where all the subsystems in the methodology section are implemented. Second, the series visualization of the signals are shown which are the raw breathing sound signal both in sorter and larger period, the envelope signal, the filtered signal, and the final reconstruction signal respectively.

3.1. The LabView block diagram
Figure 3 shows the block diagram of this proposed monitoring system in LabView which consists of four subsystems: (1) Loading File, (2) Fast Hilbert Transform, (3) Low Pass Filter, and (3) Build Waveform Function. Signal data will flow from the first subsystem and experience several conversions data type based on the needs of the passed function. It is started from the Read From Measurement File Function which is used to load the recorded breathing sound file. Since the file type is in binary (.tdms), the Time Stamps and Generic Text File sections are not available. This is followed by displaying signal into two graphs: for a chunk of short period and for a longer period which is 1 minute, Figure 4. The short period graph aims to give a closer outlook of the variability of signal, while the one minute period gives the general picture of the breathing pattern as the breathing sound in this study is sampled in high frequency 22.05 kHz. In this subsystem, the data type of the signal is called dynamic data.

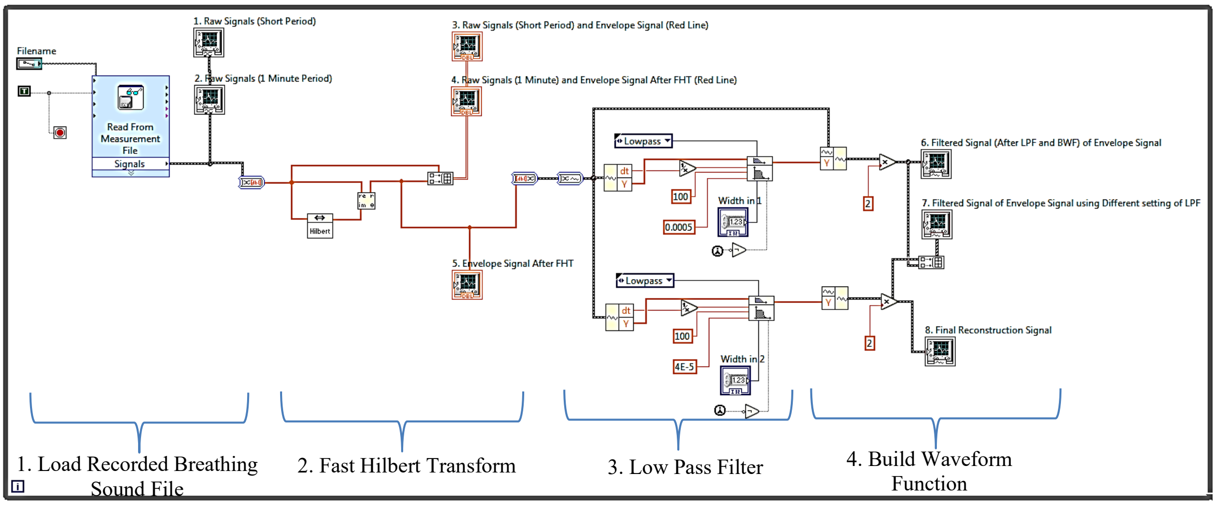


Figure 3. The LabView block diagram for reconstructing signals with four subsystems.

Entering the second subsystem, data is converted into array (1D array of double, ~15 digit precision) where Fast Hilbert Transform works to envelope signal on this 1D array. Then, the Re/Im Polar Function will convert the rectangular components of a complex number into its polar components. This is followed by concatenating both signals, raw signal, and envelope signal, using Build Array Function before they are displayed as in Figure 4, while the single envelope signal is displayed in Figure 5.a.

The next subsystem will first convert array into dynamic data and then into analog waveform. Before entering the Low Pass Filter, the waveform will pass through Get Waveform Components Function to get *dt* which will return the time interval in seconds between data points in the waveform and *Y* which return the data values of the waveform. These *dt* and *Y*, together with cut-off frequency and filter order, are the input components for LPF. Then, the output of LPF will become the components for modifying the envelope signal in the Build Waveform Function subsystem to produce the final reconstructed signal.

3.2. Raw signals of breathing sound

Figure 4 shows the raw breathing sound signals in time domain acquired at 27/06/2019 and displayed for (1) the short period which is around 150ms and (2) the long period which is 1 minute. It can be seen that the characteristic of the signals highly oscillates with intense amplitude variation due to inhale and exhale motion. In this very short period, extracting a piece of relevant information from this signal is difficult. On the other hand, longer signals can give cycle patterns. They look like immediate bursts signals which represent the instances of inhalation and exhalation signs separated by the idle period representing the low detected sound. Their forms are not similar to each other showing high variability. In addition, some background noises can also be seen. As the in-exhale signs are formed by a group of many signals, it will be useful if those signs can be represented by only one smooth curve which may be closely associated with a sinusoidal waveform. This is the main goal of this paper for which further signal processing is needed. The following subsections will present the processed signals step by step.

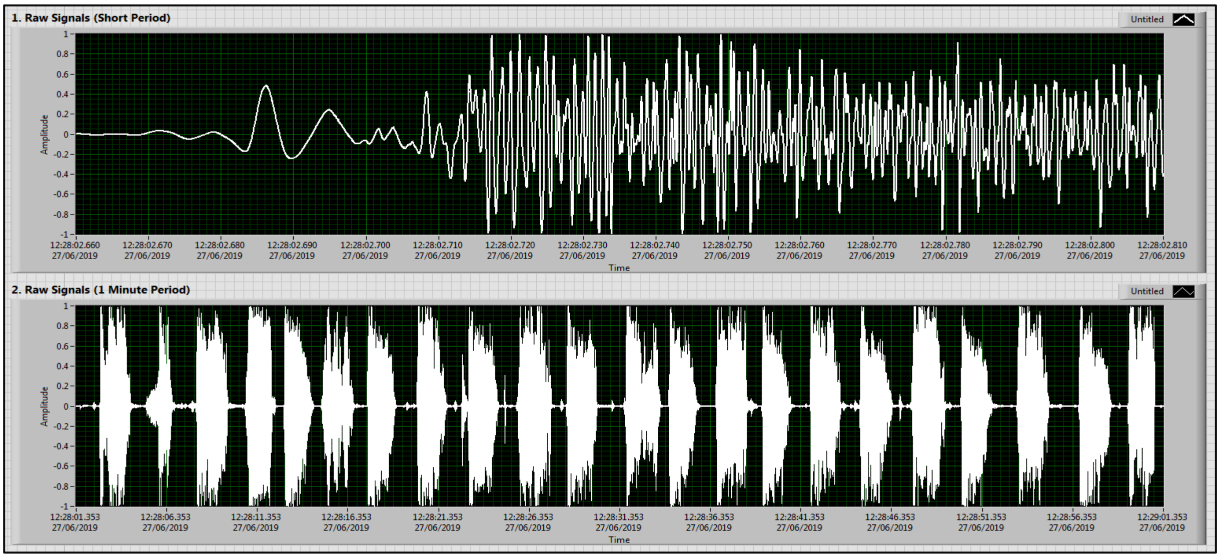


Figure 4. The breathing sound raw signals in short (150ms) and long (1 minute) duration.

3.3. Envelope signal

To reconstruct a smooth curve, it is started with enveloping the raw signals. Figure 5 presents the envelope signals, denoted in the red line, which are the result of the Fast Hilbert Transform. In this case, only the upper signal with positive values is enveloped because the lower signal with negative value basically is the copy of the upper ones. It can be seen that the envelope signal is able to track the consecutive peak values. It pretends that they are connected by outlining on each peak of the raw signals. Ideally, the envelope process aims to produce a smooth signal. However, as the raw signal highly oscillates with a fast change of amplitudes, this makes the envelope signal is still far from a smooth curve. Downsampling for reducing the sample rate could be possible to help, but there is a risk of data loss. Therefore, only implementing the Low Pass Filter is conducted in the next processing stage.

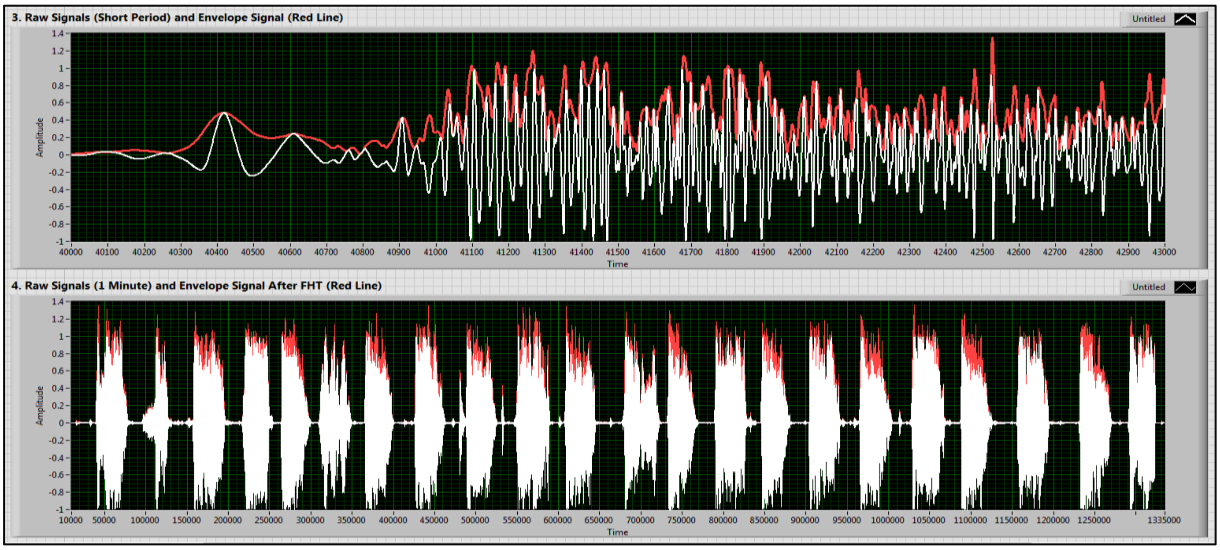


Figure 5. The raw (white) and envelope (red) signals.

3.4. Upper envelope and filtered signal

Figure 6 shows the upper envelope signal, followed by the filtered signal. The upper envelope signals are almost similar to the upper part of the raw signal. Using 0.5 mHz as the lower cut-off frequency and 2 as the filter order, the filtered signals start to give a likely single form of signal in their bottom part, while their peak parts still oscillate. The smoother graph is critically needed to facilitate the peak detection based on threshold more easily in order to calculate the respiration rate. In this case, the proper adjustment of the lower cut-off frequency is required to get rid of the remaining high-frequency parts.

Figure 6. The upper envelope signals and their filtered ones.

3.5. Reconstructed signal

By decreasing the cut-off frequency of LPF to the 0.04 mHz, the smoother signal can be attained as depicted in Figure 7 (the green line). This tells that the very low cut-off frequency produces a smoother signal compared to the previous signals (the red line). This shows the transformation of breathing sound signals from the highly populated and oscillated signals to the smoother individual curve where one signal now can represent the inhalation and exhalation cycle. In addition, it can be seen that the shape of each generated signal is not exactly similar showing the characteristics of their raw breathing sound signals are still inherited to this new signals which vary in amplitude, interval, and width. However, using this reconstructed signals, it is expected that the calculation of the respiration rate will be easier.

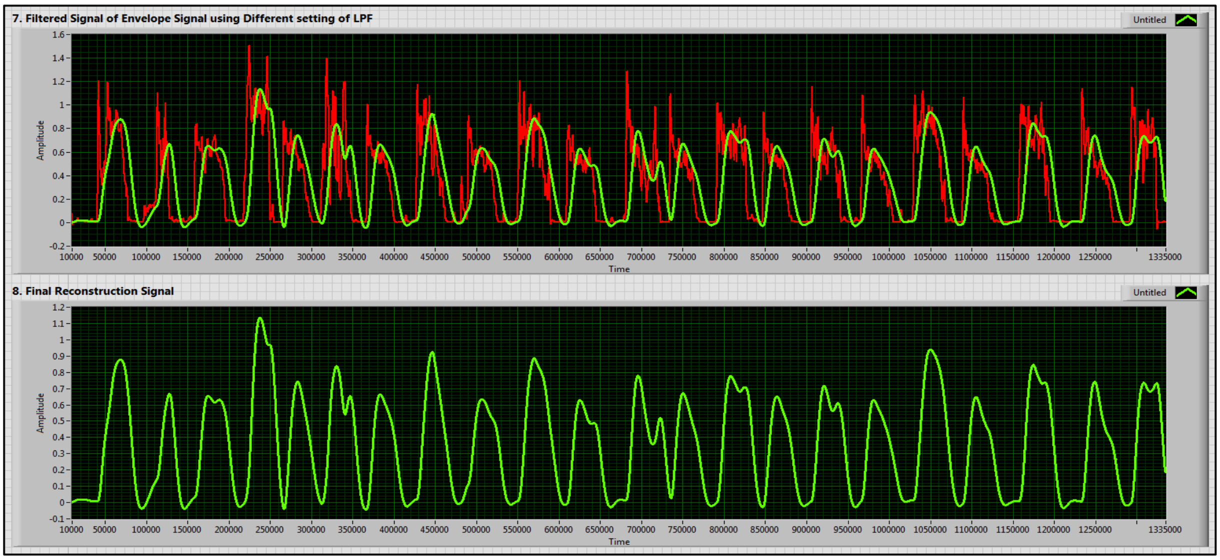
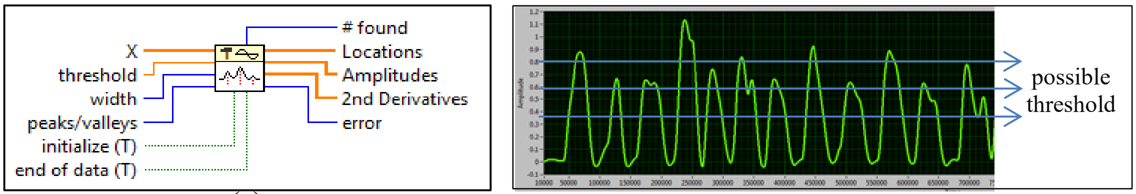


Figure 7. The reconstructed signals.

3.6. The usability of the reconstructed signal
As the aforementioned discussion, one of the possible usability of this reconstructed signal is for calculating the respiration rate which is usually counted per minute. Here, the peak threshold and signal width can be implemented which aims to avoid the certain level and width of signal that maybe not the signal of inhalation and exhalation. The peak signals that higher than both the threshold and width are only counted. This means their values will instruct the function to ignore peaks that are smaller. Towards addressing this need, in LabView, it can be implemented using the Peak Detector Function as depicted in Figure 8. However, to determine the proper peak threshold and width is quite challenging because the shape of each signal is not exactly similar. For example, if the blue lines on the right figure are considered as the threshold, they will give a different peaks number. Therefore, a proper mechanism of the peak threshold and width determination is critically needed which will be the future work.



(a) (b)

Figure 8. (a) The Peak Detector Function, (b) The possible options of threshold.

4. Conclusion
The primary objective to reconstruct and derive the breathing sound signals, from the highly populated and oscillated raw signals to the smooth individual curve for each breathing cycle, has been achieved successfully. In the data acquisition part, using only a built-in microphone in headphone makes the sensing process relatively easy, straightforward, and cost-effective. In the signal processing part, the use of Fast Hilbert Transform with Low Pass Filter in LabView is able to envelop and filter the signals reasonably. Here, the 2nd filter order and the very low cut-off frequency, 0.04 mHz, can produce the representation of the prospective signals for further analysis. It is expected that this method can also be implemented not only for breathing sound signals but also for other signals that have similar bursts-cycled characteristics. In the near future, this research will be continued to test this proposed method using various types of breathing sound signals such as fast, slow, wheeze and other abnormal breathes. This aims to investigate the accuracy and reliability of the method whether it is adaptive enough for different types of breathing sound signals. Furthermore, the calculation of respiration rate using the Peak Detector Function in LabView will be conducted with the investigation of a proper mechanism for threshold determination.

5. References
[1] Arlotto P, Grimaldi M, Naeck R and Ginoux J-M 2014 *Sensors* **14** 15371-15386
[2] Pramono RXA, Bowyer Sand Rodriguez-Villegas E 2017 *PLoS ONE* **12** 1-43
[3] Al-Khalidi F Q, Saatchi R, Burke D, Elphick H E and Tan S 2011 *Pediatric Pulmonology Wiley* **46** 523
[4] Mhetre R and Bagal U R *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)* **9** 42-46
[5] Murthy R, Pavlidis I, Tsiamyrtzis P 2014 *Conf. Proc. Engineering in Medicine and Biology Society IEEE* **2** 1196–1199
[6] Reinvuo T, Hannula M, Sorvoja H, Alasaarela E and Myllyla R 2006 *Proceedings IEEE International Symposium on Biomedical Imaging: from nano to macro IEEE International Symposium on Biomedical Imaging* 1493-1496
[7] Dehkordi P K, Marzencki M, Tavakolian K, Kaminska M and Kaminska B 2011 *Conf Proc IEEE Eng Med Biol Soc.*

[8] Morillo D Z, Ojeda J L R, Foix L F C and Jimenez A L 2010 *IEEE Transactions Information Technology in Biomedicine* **14** 491–499
[9] Gupta M and Qudsi H *Annual Bioengineering Conference (NEBEC)*
[10] Mazzanti B, Lamberti C and Bie J D 2003 *Computers in Cardiology* **30** 613-616
[11] Nakajima K, Matsumoto Y and Tamura T 2001 *Physiological Measurement* **22** 21-28
[12] Hsu C H and Chow J 2005 *Biomedical Engineering* **17** 126-133
[13] Zhue Z, Fei J and Pavlidis I *In the proceeding of the 5th IEEE Symposium on Bioinformatics and Bioengineering (BIBE'05)* **19** 227-231
[14] Yadollahi A and Moussavi Z 2008 *Proceedings of IEEE Engineering in Medical and Biology Society*
[15] Li S H, Lin B S, Tsai C H, Yang C T and Lin B S 2017 *Sensors-MDPI* **17** 1-15
[16] Badjatiya S, Mhetre M, Kodgule R and Abhyankar H 2017 *International Journal of Computer Techniques* **4** 1-8
[17] Yang X, Ji S and Song L 2014 *Sensors and Transducers* **176** 165-171
[18] Olkkonen H, Pesola P and Olkkonen J T 2014 *Journal of Signal and Information Processing* **1** 18-23

Acknowledgments
Authors wish to acknowledge that this research is fully funded by the Centre of Research and Community Services, Politeknik Negeri Bali, Indonesia, via Grant of DIPA PNB No: SP.DIPA-042.01.2.401006.2019, December 5th, 2019.