Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Multidisciplinary : Rapid Review : Open Access Journal Vol. 4, No. 2, May 2022, pp.68-77 eISSN: 2656-8624

RESEARCH ARTICLE OPEN ACCESS

Manuscript received October 30, 2021; revised March 3, 2022; accepted April 21, 2022; date of publication May 30, 2022
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v4i2.4

Improving Heart Rate Measurement Accuracy by Reducing Artifact Noise from Finger Sensors Using Digital Filters

Anita Miftahul Maghfiroh[1], Liliek Soetjiatie[1], Bambang Guruh Irianto[1], Triwiyanto T[1], Nuril Hidayanti[1] and, Achmad Rizal[2]
[1] Department of Electromedical Engineering, Poltekkes Kemenkes Surabaya, Jl. Pucang Jajar Timur No. 10, Surabaya, 60245, INDONESIA
[2] School of Electrical Engineering, Telkom University, Bandung, INDONESIA

Corresponding author: Liliek Soetjiatie (e-mail: lilik06@poltekkesdepkes-sby.ac.id)

ABSTRACT Heart rate is an important indicator in the health sector that can be used as an effective and rapid evaluation to determine the health status of the body. Motion or noise artifacts, power line interference, low amplitude PPG, and signal noise are all issues that might arise when measuring heart rate. This study aims to develop a digital filter that reduces noise artifacts on the finger sensor to improve heart rate measurement accuracy. Adaptive LMS and Butterworth are the two types of digital filters used in this research. In this study, data were collected from the patient while he or she was calm and moving around. In this research, the Nellcor finger sensor was employed to assess the blood flow in the fingers. The heart rate sensor will detect any changes in heart rate, and the measurement results will be presented on a personal computer (PC) as signals and heart rate values. The results of this investigation showed that utilizing an adaptive LMS filter and a Butterworth low pass filter with a cut-off frequency of 6Hz, order 4, and a sampling frequency of 1000Hz, with the Butterworth filter producing the least error value of 7.57 and adaptive LMS maximum error value of 27.65 as predicted by the researcher to eliminate noise artifacts. This research could be applied to other healthcare equipment systems that are being monitored to increase patient measurement accuracy.

INDEX TERMS Heart Rate, Finger Sensor, Artifact, Butterworth Digital Filter, Adaptive LMS Digital Filter

I. INTRODUCTION

Heart rate is an essential indicator in the health sector that may be used as an effective and quick evaluation tool to determine the body's health status. In this case, knowing the accuracy and effectiveness of heart rate measurement equipment is important since it makes it easier for health care institutions to measure and gather patient data [1]. Checking the heart rate is also related to blood flow because the number of heartbeats is the same as the pulse rate. The heart rate measurement is positioned on the fingertips, toes, and ears. It has been established that the heart rate measurement on the fingertips and ears is more apparent and dependable than the toe measurement [2][3][4]. The finger sensor is one of the sensors used to measure heart rate. Photodiode and infrared are used in the finger sensor. Infrared will reflect light that enters the bloodstream, which will be picked up by the photodiode [5][6][7][8][9]. The sensor was placed on the fingers of the participants in this study to assess heart rate. This method of measuring heart rate is currently employed in all health establishments. Motion or noise artifacts, power line interference, low amplitude PPG, and signal noise are all issues that might arise when measuring heart rate [10][11][12]. A noise signal is caused by incorrect sensor positioning. The presence of noise must be reduced to a specific degree to maintain the signal's validity and produce an accurate BPM value and heart rate signal. To reduce noise caused by movement, a bandpass filter with a cutoff frequency of 0.1 Hz to 5 Hz is used [13][14][15]. To apply the LMS adaptive filter principle to noise cancellation, changes in signal properties might happen extremely quickly, necessitating the employment of adaptive and Butterworth algorithms that integrate quickly [16][17][18][19]. The LMS method is a popular and easy signal processing algorithm that may be used to solve various problems such as noise, echo, and interference reduction. The target means square error and the processing speed that must be reached limits the filter order. The smaller the mean square error, but the slower the processing time, the higher the rank of the filter. Coefficient regulators in adaptive filters can respond to changing environmental circumstances and system modifications [20][21][22]. Butterworth filters have a passband amplitude response that is almost flat (maximally flat) with no ripple, making them superior to Chebyshev and Elliptic filters, which are also frequent filter designs. As a result, this study will compare the LMS adaptive filter and the Butterworth filter to see which one is best for reducing noise artifacts when measuring heart rate.

Based on a literature review, Mohammad Abu Raihan Miah et al conducted a study in 2013 on heart rate monitoring utilizing a fingertip and microphone port with an optical sensor based on infrared technology to detect changes in blood volume. This study benefits from a low-cost, computer-based heart monitoring device that can be used to study and diagnose heart problems including sleep arrhythmias. However, it has the drawback of requiring careful placement of the fingertips during installation to avoid inaccurate readings caused by hand movements [23]. Yang Liu et al. conducted a study in 2013 on a method for decreasing noise in audio signals utilizing an adaptive LMS filter and an adaptive NLMS algorithm for noise reduction in speech applications and evaluated the suggested noise reduction method's performance. The proposed approach was tested using audio data sampled at 48 kHz with a recording period of 4.5 seconds. This study has an additional benefit of being able to improve the quality of noisy audio signals. It does indeed, however, it also have the drawback of just employing one sample audio signal [16]. Furthermore, Rekha Chandra R et al studied the design of a tiny pulse oximeter device for continuous SPO2 and heart rate monitoring using the PPG method in 2015. This study has the advantage of a wireless pulse oximeter system that can be used to assess non-invasively for a lengthy period. The results were examined using Matlab when the values have been obtained in digital form. However, due to motion artifacts, skin pigmentation, venous blood, and other factors, it produced noise and interference under certain conditions, necessitating a high level of filtering to recognize PPG waveforms obtained from lesser reflectance sensors [14]. I Putu Anna Andika further carried out research on a portable pulse oximeter employing the MAX30102 sensor in 2019. This study benefits from the portability of a mobile device. According to the results obtained, this tool is suitable for use because the highest tolerance level for error is Pulse Oximeter 1%, according to the Guidelines for Testing and Calibrating Medical Devices published by the Ministry of Health of the Republic of Indonesia in 2001. However, it has a flaw that it will produce a big erroneous value if there is finger movement during the measurement [24]. In 2020, Ahmad Zaky used the MAX30100 sensor to study on monitoring blood volume and heart rate signals with data storage and two parameters, namely SPO2 and BPM. There is data storage in this study, which is a plus. However, if there is a movement of artifacts created by the respondent's finger movements, this will increase the error value, and the intensity of light around the MAX30100 sensor can alter the readings of the SpO2, and BPM measurement parameters, this further will increase the error value [25].

Various things need to be developed based on the description of the literature study that has been provided, including the presence of noise when there is patient movement or noise artifacts. The author's goal in this study is to design a digital filter to eliminate noise artifacts on the finger sensor to improve heart rate measurement accuracy. The pulse technique, which measures blood flow in the fingers, was used to determine heart rate in this study. Then, to separate the sensor performance data, an appropriate filter that may decrease or eliminate noise caused by patient movement was created. The adaptive LMS and Butterworth digital filters were also used in this experiment. The heart rate sensor will detect any changes in heart rate, and the measurement data will be displayed as a signal on a personal computer (PC). The goal of this study's data analysis is to evaluate the measurement findings acquired from a heart rate sensor that has been through two distinct types of digital filters and then draw accurate conclusions. Health facilities should be able to readily measure and obtain patient heart rate data as a result of this research.

Chapter 1 explains the background knowledge of this research, Chapter 2 explains the research methods, Chapter 3 explains the results and analysis of the research, Chapter 4 explains the results of the study, compares with previous researchers and the implications of the research, while Chapter 5 explains the conclusions and future research.

II. MATERIALS AND METHODS

A. EXPERIMENTAL SETUP

The respondent's heart rate/pulse was detected using a finger sensor put on the finger in this investigation. Data were collected at random from ten adult respondents. Relaxed conditions before the Butterworth filter process, conditions for movement after the Butterworth filter process, relaxed conditions before the adaptive LMS filter process, and conditions for movement after the adaptive LMS filter process were used to retrieve data four times.

1) MATERIALS AND TOOLS

This study utilized a finger sensor (Nellcor, 130010075/SA11Y-01X, China) that detected heart rate and pulse. To amplify and create photoplethysmograph signals, a finger sensor circuit, AC RED amplifier and filter, and AC IR amplifier and filter circuit were also used. The data generated by the Arduino Uno module and Arduino software (Version

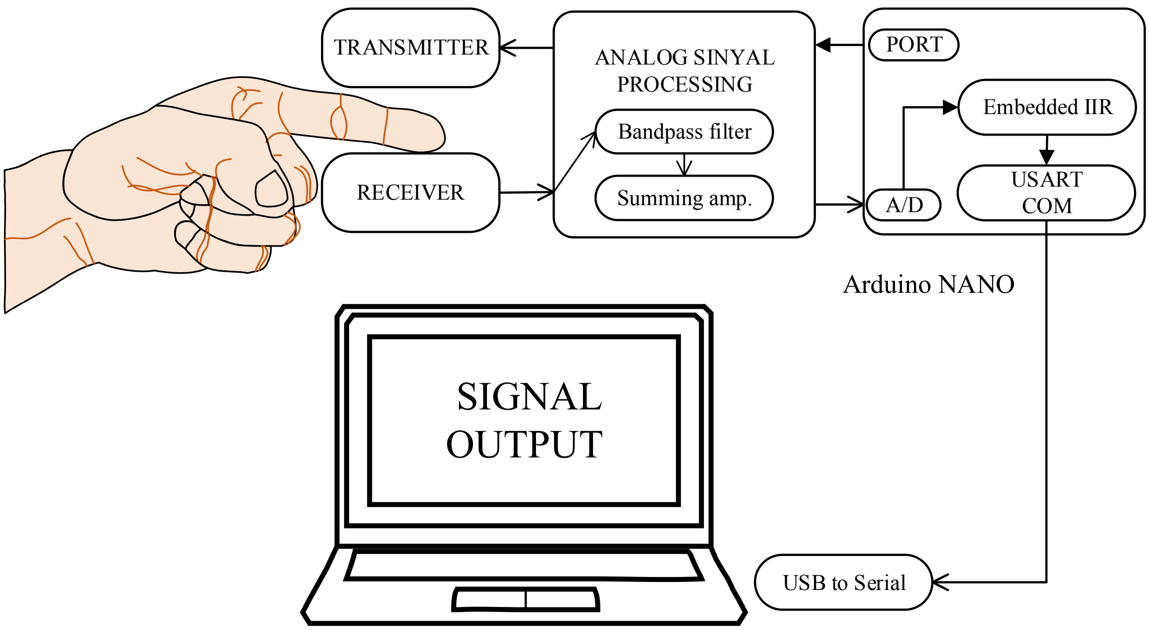

FIGURE 2. The Diagram Block system. The output of the photodiode sensor is a signal as a result of blood flow then signal processing was carried out in an analog filter circuit. The output of the ASP was then filtered using digital signal processing to remove noise artifacts.

1.8.14) was then transferred from the module to the computer display via USB.

2) EXPERIMENT

In this study, the respondent involved were 6 men with an average age of 18 years ± 2 years who did not have heart defects. The participants were given informed consent so that they could study and understand the experimental protocol. In this case, the experimental protocol has passed the Surabaya Health Polytechnic Ethics Committee, Ministry of Health, Indonesia (No: 035/S/KEPK/V/2017).

The data acquisition was done from the finger sensor to get the heart rate value (BPM) which was then filtered using an analog filter circuit. The analog filter output was then filtered digitally to reduce motion artifacts.

Data collection was carried out in two ways, namely the subject in a relaxed state and the subject in a movement state. From the data collection, the signal-to-noise ratio (SNR) was analyzed by comparing two digital filter methods, namely the Butterworth filter and the Adaptive LMS filter.

To determine the level of accuracy or error from this research, statistical analysis was carried out by calculating the value, error, and standard deviation according to the calibration uncertainty formula [26].

3) METHODS

The digital filter methods used in this research are the adaptive least mean square (LMS) filter and the IIR filter. Where the adaptive filter least mean square is the real programmable filter, with x(n) being the adaptive filter's input, y(n) being the adaptive filter's output, d(n) being the desired result, and e(n) being the adaptive algorithm's error based on FIGURE 1 The target means square error and the processing speed that must be reached limits the filter order. The smaller the mean square error, but the slower the processing time, the higher the rank of the filter Based on Equation (1) [27].

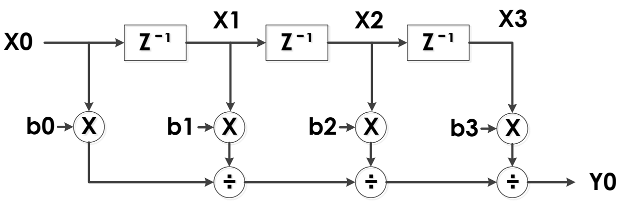

FIGURE 1. Transfer function digital filter

$$y=b[0]*x0 + b[1]*x1 + b[2]*x2 + b[3]*x3 + b[4]*x \quad (1)$$

One of the most extensively used traditional filter design methods is the Butterworth filter. The Butterworth filter is preferable because it has a passband amplitude response that is practically flat (maximally flat) and has no ripple. This is a low-frequency IIR filter. The IIR filter is very good and has a lot of processing capacity Based on Equation (2) [27].

$$y=b[0]*x0 + b[1]*x1 + b[2]*x2 + b[3]*x3 + b[4]*x4 - a[1]*y1 - a[2]*y2 - a[3]*y3 - a[4]*y4 \quad (2)$$

B. THE DIAGRAM BLOCK

FIGURE 2 shows when the infrared light from the transmitter penetrated the finger and was collected by the photodiode sensor at the receiver. The photodiode output generated a signal as a result of the blood flow. An analog signal conditioning circuit will handle the photodiode sensor's output signal. However, there was still noise in the signal, which necessitates the use of a filter circuit to remove it. An adaptable LMS digital filter and a Butterworth digital filter were utilized. The signal will be converted into digital data using the Arduino microcontroller analog to digital converter before being treated on the filter (ADC). Before the filter procedure is carried out, the results of the signal can be seen in Arduino programming. The outcomes of the digital signal conversion will also be managed using Arduino's digital filter programming. The measurement findings after the signal filter are then shown as a signal on a personal computer (PC). The final stage will involve determining which filter circuit is best for improving heart rate measurement accuracy.

C. THE FLOWCHART

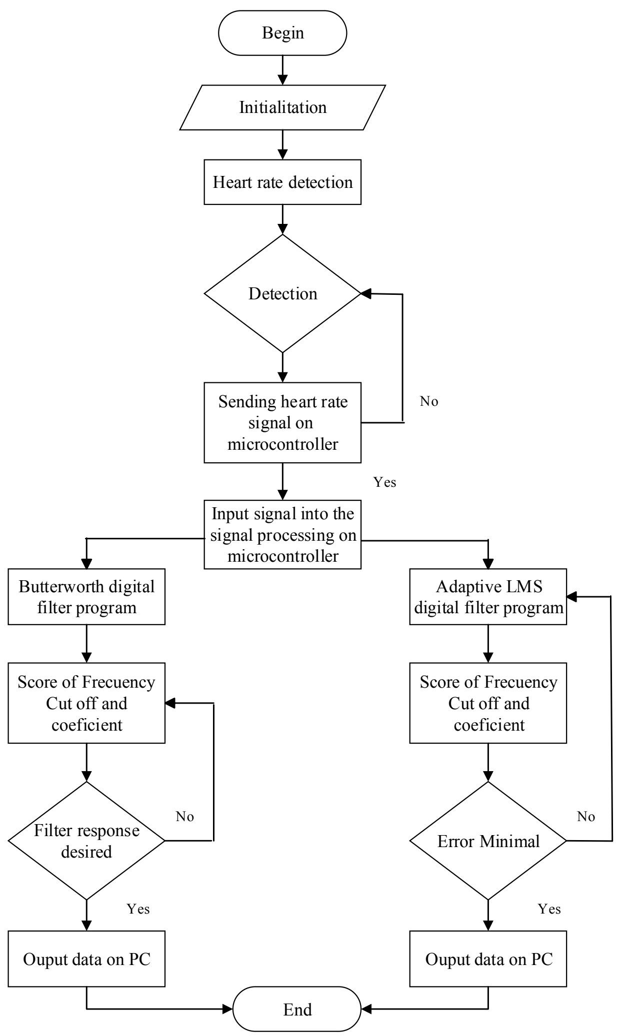

FIGURE 3. In the Flowchart of Arduino Process, a heart rate was detected, a digital filter process was carried out, and there was a selection process, namely the Butterworth digital filter and the adaptive LMS filter, each of which is a cut-off frequency score which then outputs data to a personal computer.

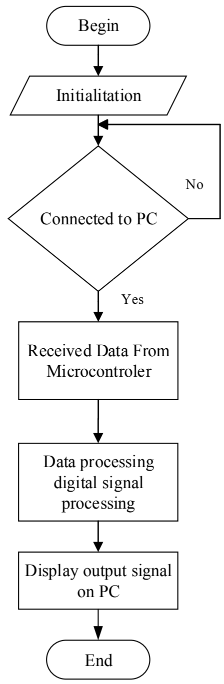

FIGURE 4. The Flowchart of the Personal Computer is the process of initializing the program to the microcontroller so that it is connected to the PC. In this case, the data is received by the PC to carry out the digital filter process which will then be displayed on the PC screen.

FIGURE 3 shows the heart rate was detected by the finger sensor that was affixed to the patient's finger. The heart rate detection will be repeated if the heart rate is not detected. If the heart rate is recognized, the heart rate signal measurement results are communicated to the Arduino microcontroller. The data that would be filtered was then selected. It further began by configuring the LMS Adaptive Filter and the frequency cutoff and coefficient value were calculated. The data will be shown on the PC if the minimal error value has been reached; if the minimum error value has not been reached, the reprogramming procedure will be initiated automatically. Furthermore, the coding Butterworth filters are started. The frequency cutoff and coefficient value were calculated. The desired filter result data were then obtained and presented on the computer. The reprogramming process will be carried out if the desired filter data cannot be obtained. The measurement findings were then be shown on a computer's data display (PC).

After the digital filter process using the Butturworth filter and the LMS adaptive filter is complete, the next step is programming the microcontroller to display the output signal that is connected to a personal computer (PC) as described in FIGURE 4. It was shown that a flow diagram of the Personal Computer. Initialization will come before starting the tool. The microcontroller will then be connected to a computer (Personal Computer). The microcontroller will be re-initialized if it has not been connected to a PC (Personal Computer). If a PC (Personal Computer) is linked to the microcontroller, the PC will receive data from the microcontroller. The Arduino microcontroller was then used to perform the input signal filtering stage. On a computer, the filtering process' findings will be shown and analyzed (Personal Computer). Then it is complete.

III. RESULT

A finger sensor put on the respondent's finger was used to detect the respondent's heart rate/pulse in this investigation. The data were collected from ten people, with a comparison between the module and the pulse oximeter.

A. DESIGN MODULE BUILD

FIGURE 5 presents the output from the finger sensor that further passed to the analog signal conditioning circuit, where it passed through a 2.34 Hz passive high pass filter and an active 2.34 Hz low pass filter. The high pass filter passes frequencies above the cut-off frequency, while the low pass filter passes frequencies below the cut-off frequency and amplifies the output, resulting in an AC infrared signal (AC IR) and a red led at the filter's output (AC RED). The finger sensor circuit generated a large and distinct PPG signal.

The output of this analog signal conditioning circuit will be fed into an Arduino Uno pin and shown in the Delphi program. According to the results of the PPG signal plotting comparison, the signal after the Butterworth digital filter differs from the signal before the Butterworth filter (in a state of movement). The signal after the adaptive LMS digital filter differs from the signal before the LMS adaptive filter by a little amount (in a state of movement).

FIGURE 5. Module Design consists of an analog signal processing circuit, power supply, and a microcontroller circuit.

B. RESULTS OF COLLECTING BPM VALUE DATA FROM RESPONDENTS

The results of the data collection on the BPM value from the respondents involved were 6 men with an average age of 18 years ± 2 years who did not have heart defects. Statistical calculations were then performed to determine the error value and accuracy of this study as shown in TABLE 1 and FIGURE 6.

TABLE 1
Average BPM and Standard Deviation Measurement of Each LMS Adaptive Filter and Butterworth Filter With Relaxed Conditions and Motion Condition

Mean and SD BPM Measurement			
Butterworth Filter (Relax)	Adaptive LMS Filter (Relax)	Butterworth Filter (Motion)	Adaptive LMS Filter (Motion)
82.27 ± 11.59	102.63 ± 15.93	87.33 ± 13.62	112.78 ± 18.77
Error BPM Measurement			
Butterworth Filter (Relax)	Adaptive LMS Filter (Relax)	Butterworth Filter (Motion)	Adaptive LMS Filter (Motion)
± 7.57	± 27.65	± 10.73	± 19.81

The results of testing the respondents' BPM values before the Butterworth filter was relaxed resulted in an error value of ±7.57, with an average standard deviation of 82.27±11.59 based on the results of testing the respondents' BPM values. Before the adaptive LMS filter was relaxed, the BPM value created an inaccuracy of ±27.65, with an average standard deviation of 102.63±15.93. When the Butterworth filter was in motion, the BPM value provided an erroneous value of ±10.73, with an average standard deviation of 87.33±13.62. When the adaptive LMS filter was in motion, the BPM value provided an error value of ±19.81, with an average standard deviation of 112.78±18.77.

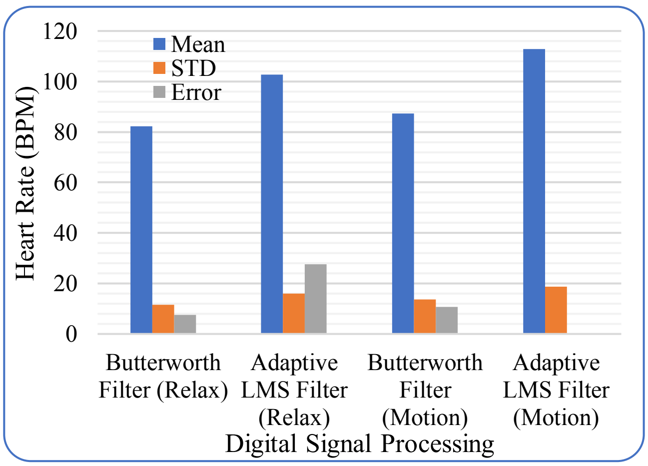

FIGURE 6. Comparison between Butterworth filter and Adaptive LMS filter with two conditions, namely relaxed condition and motion condition.

C. SNR MEASUREMENT

To determine the reliability of the digital filter system, data were collected under two conditions, namely relaxed conditions and motion conditions, as shown in TABLE 2 and FIGURE 7. The SNR is based on Equation (3) [28].

$$SNR: \frac{\bar{S}}{\sigma_N} \quad (3)$$

where $\bar{S}$ average is signal power and σ_N is the baseline and activation, associated with noise. Statistical analysis was carried out to obtain the average value and standard deviation of the signal-to-noise ratio (SNR).) of Adaptive LMS filters and Butterworth filters.

TABLE 2
SNR Measurement Average Value And Standard Deviation Of Each LMS Adaptive Filter And Butterworth Filter With Relaxed Conditions And Motion Conditions

Mean and SD SNR Measurement			
Butterworth Filter (Relax)	Adaptive LMS Filter (Relax)	Butterworth Filter (Motion)	Adaptive LMS Filter (Motion)
2.01±4.70	1.34±4.37	1.34±4.87	1.005±4.7

The average value and standard deviation of the Butterworth filter SNR in a relaxed condition are 2.01±4.70 the results of signal testing utilizing the signal-to-noise ratio in the Matlab application. In a relaxed condition, the average value and standard deviation of the SNR adaptive LMS filter are 2.01±4.70. In a state of motion, the average value and standard deviation of the SNR filter Butterworth are 1.34±4.87. In a state of movement, the average value and standard deviation of the SNR adaptive LMS filter are 1.005±4.7.

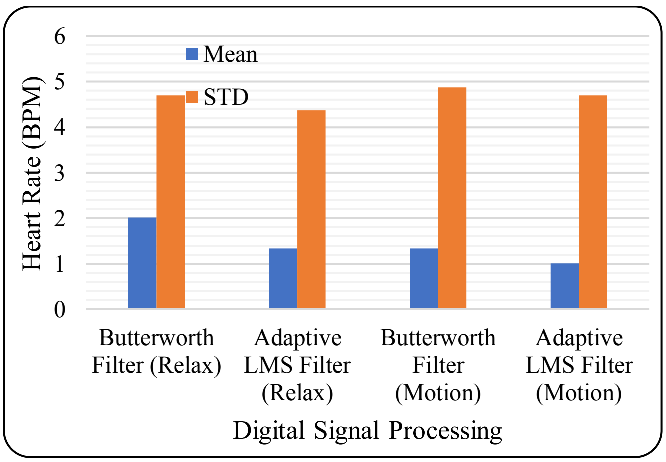

FIGURE 7. **Comparison between mean and standart deviation SNR measurement Butterworth filter and Adaptive LMS filter with two conditions, namely relaxed condition and motion condition.**

Planning fundamentals, noise, and DC harmonics are data collected using SNR Test using Butterworth Filter in Relaxed Conditions as described in FIGURE 8.

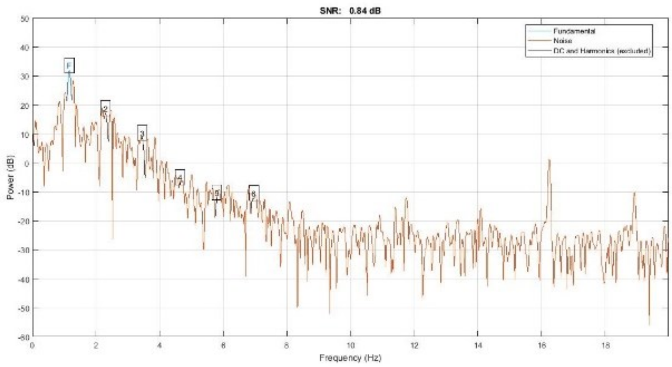

FIGURE 8. **SNR Test using Butterworth Filter in Relaxed Conditions. Blue chart is fundamental, red is noise, and DC harmony is displayed in black**

Based on FIGURE 8, it is explained that the dominant signal from the data collected is depicted in blue charting, which is the fundamentals. The noise at the time of data collection is indicated in red plotting. The DC harmonics are displayed in black. Meanwhile, FIGURE 9 is SNR Test using Adaptive LMS Filter in Relaxed Conditions

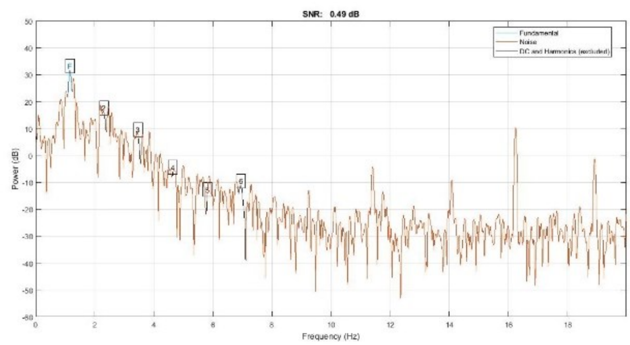

FIGURE 9. **SNR Test using Adaptive LMS Filter in Relaxed Conditions. the blue chart is fundamental, red is noise and DC harmony is displayed in black**

Planning fundamentals, noise, and DC harmonics is data collection with SNR Test using Butterworth Filter in Motion Conditions as described in FIGURE 10. Meanwhile, FIGURE 11 is SNR Test using Adaptive LMS Filter in Motion Conditions. Based on FIGURE 10 and FIGURE 11, it is explained that the dominant signal from the data collected is depicted in blue charting, which is the fundamentals. The noise at the time of data collection is indicated in red plotting. The DC harmonics are displayed in black.

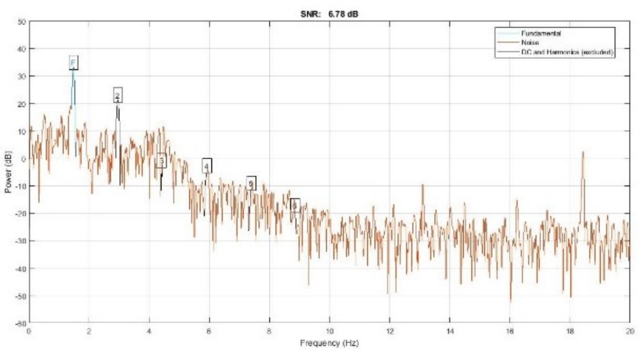

FIGURE 10. SNR Test using Butterworth Filter in Motion Conditions. blue chart is fundamental, red is noise and DC harmony is displayed in black

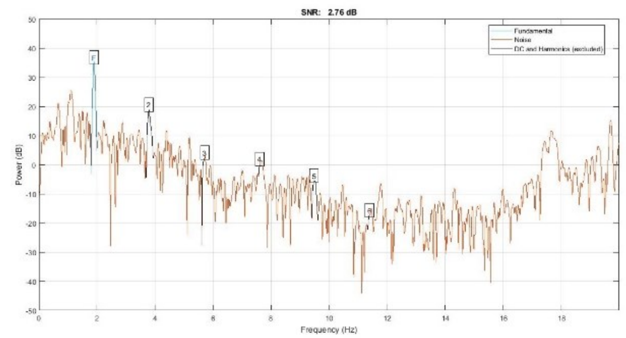

FIGURE 11. SNR Test using Adaptive LMS Filter in Motion Conditions. blue chart is fundamental, red is noise and DC harmony is displayed in black

IV. DISCUSSION

In this study, the finger sensor's output was transmitted to the demultiplexer circuit, which was further divided the output into infrared signals and red led. The demultiplexer's output was then fed into the passive high pass filter of 2.34 Hz and the active low pass filter of 2.34 Hz. The high pass filter passed frequencies above the cut-off frequency, while the low pass filter passed frequencies below the cut-off frequency and amplifies the output, resulting in an AC infrared signal (AC IR) and a red led at the filter's output (AC RED). The finger sensor circuit generated a large and distinct PPG signal. The output of this analog signal conditioning circuit was then fed into an Arduino Uno pin and shown in the Delphi program. According to the results of the PPG signal plotting comparison, the signal after the Butterworth digital filter differs from the signal before the Butterworth filter (in a state of movement). The signal after the adaptive LMS digital filter differs from the signal before the LMS adaptive filter by a little amount (in a state of movement).

According to the findings of testing the BPM value on 10 respondents, the results of testing the respondents' BPM values before the Butterworth filter was relaxed resulted in an error value of ±7.57, with an average standard deviation of 82.27±11.59 based on the results of testing the respondents' BPM values. Before the adaptive LMS filter was relaxed, the BPM value created an inaccuracy of ±27.65, with an average standard deviation of 102.63±15.93. When the Butterworth filter is in motion, the BPM value provided an erroneous value of ±10.73, with an average standard deviation of 87.33±13.62. When the adaptive LMS filter was in motion, the BPM value provides an error value of ±19.81, with an average standard deviation of 112.78±18.77. With these data, the Butterworth filter generated the smallest error value ±7.57 and the adaptive LMS filter generated the biggest error value ±27.65.

It can be used to gather data or monitor PPG signals and BPM values based on the program used in this module, which includes detection, display, and signal processing programs, as well as BPM values before and after the Butterworth filter and the adaptive LMS filter. The Delphi application displays the results of the Arduino program processing. The weakness in this study is that the threshold process for calculating BPM is still manual and does not use an adaptive threshold.

David Pollreisz et al compared several filter methods to remove motion artifacts from the PPG output signal [4] including the filters used were Discrete Wavelete Transform (DWT), Adaptive LMS filters, Variable frequency complex demodulation and EMD techniques, but it would be difficult to use the data set used.

Based on the findings of signal testing in the Matlab program utilizing the signal-to-noise ratio, the average value and standard deviation of the Butterworth filter SNR in a relaxed condition are 2.01±4.70 the results of signal testing utilizing the signal to noise ratio in the Matlab application. In a relaxed condition, the average value and standard deviation of the SNR adaptive LMS filter are 2.01±4.70. In a state of motion, the average value and standard deviation of the SNR filter Butterworth are 1.34±4.87. In a state of movement, the average value and standard deviation of the SNR adaptive LMS filter are 1.005±4.7. Implications of this study serves as information for other researchers in order to be able to choose the right filter to reduce interference from motion conditions that the better the noise reduction results, the higher the SNR value obtained. So that it can be used as a reference if you want to develop research in this field.

V. CONCLUSION

The purpose of this study is to design a digital filter to eliminate noise artifacts on the finger sensor to improve heart rate measurement accuracy using adaptive LMS and Butterworth digital filters. This study was able to use a finger sensor, the pulse technique, which measures blood flow in the fingers, was used to determine heart rate. The Butterworth filter generated the smallest error value of ±7.57 and the adaptive LMS filter generated the biggest error value of ±27.65. The better the noise reduction results, the higher the SNR value obtained in Butterworth filters. In the future, it could be applied to other healthcare equipment systems that are being monitored to increase patient measurement accuracy.

REFERENCES

[1] T. P. Tunggal and A. Latif, "Low-cost portable heart rate monitoring based on photoplethysmography and decision tree Low-Cost Portable Heart Rate Monitoring Based on Photoplethysmography and Decision Tree," *AIP Conf. Proc. 1755*, vol. 090004, no. July 2016, p. 6, 2018, doi: 10.1063/1.4958522.

[2] A. Sarkar and A. L. Abbott, "Biometric Authentication Using Photoplethysmography Signals," *Dep. Electr. Comput. Eng.*

[3] H. Mansor, S. S. Meskam, and N. Sakinah, "Portable Heart Rate Measurement for Remote Health Monitoring System," in *International Islamic University Malaysia (IIUM) Ministry of Higher Education*, 2015, no. June 2013, pp. 0–4.

[4] J. N. Lygouras and P. G. Tsalides, "Optical-Fiber Finger Photo-Plethysmograph Using Digital Techniques," *IEEE Sens. J.*, vol. 2, no. 1, pp. 20–25, 2002.

[5] N. H. Wijaya, A. G. Alvian, A. Z. Arfianto, J. E. Poetro, and F. Waseel, "Data Storage Based Heart and Body Temperature Measurement Device," *J. Robot. Control*, vol. 1, no. 1, pp. 11–14, 2020.

[6] A. B. Tadesse, A. A. Mohammed, and N. S. Ramaiah, "Design and Implementation of Heart Monitoring Using PIC Microcontroller," in *National Conference on Communication and Image Processing (NCCIP- 2017) 3rd National Conference by TJIT, Bangalore*, 2017, no. November 2019, p. 4, doi: 10.1109/SmartTechCon.2017.8358479.

[7] H. Shirzadfar, M. S. Ghaziasgar, Z. Piri, and M. Khanahmadi, "Heart beat rate monitoring using optical sensors," *Int. J. Biosens. Bioelectron.*, vol. 4, no. 2, pp. 45–51, 2018, doi: 10.15406/ijbsbe.2018.04.00097.

[8] T. P. Tunggal, S. A. Juliani, H. A. Widodo, R. A. Atmoko, and P. Thanh, "The Design of Digital Heart Rate Meter Using Microcontroller," *J. Robot. Control*, vol. 1, no. 5, pp. 141–144, 2020, doi: 10.18196/jrc.1529.

[9] S. Namani, M. Ramasamy, and S. Paripelli, "Arduino Microcontroller Based Heart Rate Monitor Using Fingertip Sensors," *Saudi J. Eng. Technol. (SJEAT) Arduino Microcontroller Based Hear. Rate Monit. Using Fingertip Sensors*, vol. 6272, no. 6264, pp. 478–483, 2017, doi: 10.21276/sjeat.2017.2.12.4.

[10] D. Pollreisz and N. Taherinejad, "Detection and Removal of Motion Artifacts in PPG Signals," *Mob. Networks Appl.*, vol. 10.1007, p. 11, 2019.

[11] J. C. Sci, S. Biol, A. Chatterjee, and R. Uk, "Algorithm to Calculate Heart Rate and Comparison of Butterworth IIR and Savitzky-Golay FIR Filter," *J. Comput. Sci. Syst. Biol.*, vol. 11, no. 2, pp. 171–177, 2018, doi: 10.4172/jcsb.1000268.

[12] T. Tamura, "Current progress of photoplethysmography and - SPO 2 for health monitoring," *Biomed. Eng. Lett.*, vol. 9, no. 1, pp. 21–36, 2019, doi: 10.1007/s13534-019-00097-w.

[13] T. Chang, P. Chan, C. Chen, K. Chen, and C. Yang, "Fingertip Pulse Signals Enhanced by Using Intermodulation Multiplication of Active High-Sensitivity Split-Ring Resonator," *2018 IEEE/MTT-S Int. Microw. Symp. - IMS*, pp. 1416–1418, 2018.

[14] R. C. R, K. P. Safeer, and P. Srividya, "Design and Development of Miniaturized Pulse Oximeter for Continuous Spo2 and HR Monitoring with Wireless Technology," *Int. J. New Technol. Res.*, vol. 1, no. 1, pp. 11–15, 2015.

[15] Y. Aiboud, J. El Mhamdi, A. Jilbab, and H. Sbaa, "Review of ECG signal de-noising techniques," *Proc. 2015 IEEE World Conf. Complex Syst. WCCS 2015*, 2016, doi: 10.1109/ICoCS.2015.7483313.

[16] Y. Liu, M. Xiao, and Y. Tie, "A Noise Reduction Method Based on LMS Adaptive Filter of Audio Signals," in *3rd International Conference on Multimedia Technology*, 2013, pp. 1001–1008.

[17] Z. Zhu, X. Gao, L. Cao, D. Pan, Y. Cai, and Y. Zhu, "Analysis on the adaptive filter based on LMS algorithm Optik Analysis on the adaptive filter based on LMS algorithm," *Opt. - Int. J. Light Electron Opt.*, vol. 127, no. 11, pp. 4698–4704, 2018, doi: 10.1016/j.ijleo.2016.02.005.

[18] U. Pradesh, "LMS Adaptive Filters for Noise Cancellation : A Review," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 5, pp. 2520–2529, 2017, doi: 10.11591/ijece.v7i5.pp2520-2529.

[19] R. A. Rachman, I. D. G. H. Wisana, and P. C. Nugraha, "Development of a Low-Cost and Effisient ECG devices with IIR Digital Filter Design," *Indones. J. Electron. Electromed. Eng. Med. informatics*, vol. 3, no. 1, pp. 21–28, 2021, doi: 10.35882/ijeeemi.v3i1.4.

[20] K. F. Akingbade and I. A. Alimi, "Separation of Digital Audio Signals using Least-Mean-Square (LMS) Adaptive Algorithm," *Int. J. Electr. Comput. Eng.*, vol. 4, no. 4, pp. 557–560, 2014.

[21] Z. Shen and R. Wang, "Design and Application of an Improved Least Mean Square Algorithm for Adaptive Filtering," *Eur. J. Electr. Eng.*, vol. 21, no. 3, pp. 303–307, 2019.

[22] S. Chaudhary and R. Mehra, "ADAPTIVE FILTERS DESIGN AND ANALYSIS USING LEAST," *Int. J. Adv. Eng. Technol.*, vol. 6, no. 2, pp. 836–841, 2013.

[23] A. R. Miah, S. Basak, and A. Roy, "Low Cost Computer Based Heart Rate Monitoring System Using Fingertip and Microphone Port," *Electr. Electron. Eng.*, vol. 13, no. 978-1-4799-0400–6, p. 4, 2013.

[24] I. P. A. Andika and T. Rahmawati, "Portable Pulse Oximeter," *jeemi*, vol. 1, no. 1, pp. 28–32, 2019, doi: 10.35882/jeeemi.v1i1.6.

[25] A. Zaky, P. C. Nugraha, and A. Pudji, "Bed Measuring Estimate Blood Volume and Cardiac Output With TFT Display Equipped With Data Storage (SpO 2 and BPM)," *ijeemi*, vol. 2, no. 1, pp. 6–12, 2020, doi: 10.35882/ijeeemi.v2i1.2.

[26] J. Duvernoy, "Guidance on the Computation of Calibration Uncertainties," *World Meteorol. Organ.*, no. 119, 2015, [Online]. Available: http://www.wmo.int/pages/prog/www/IMOP/publications-IOM-series.html%0D.

[27] L. Tan, *Digital Signal Processing Fundamentals and Applications*, no. c. Georgia: DeVry University Decatur, 2008.

[28] M. Welvaert and Y. Rosseel, "On the definition of signal-to-noise ratio and contrast-to-noise ratio for fMRI data," *PLoS One*, vol. 8, no. 11, 2013, doi: 10.1371/journal.pone.0077089.

APPENDIX

1) BEFORE THE DIGITAL FILTER, THERE IS A PPG SIGNAL PROGRAM.

```
SignalNF=analogRead(A2);
sampleCounter += 2;
if (SignalNF < thresh && N > (IBI / 5) * 3) {
    if (SignalNF < T) {
      T = SignalNF;}}
  if (SignalNF > thresh && SignalNF > P) {
    P = SignalNF; }
```

2) BEFORE AND AFTER THE DIGITAL FILTER, THE PROGRAM BPM VALUE.

```
if (N > 250) {
if ( (SignalNF > thresh) && (Pulse == false) && (N >
(IBI / 5) * 3) ) {
Pulse = true;
IBI = sampleCounter - lastBeatTime;
lastBeatTime = sampleCounter;
if (secondBeat) {
secondBeat = false;
for (int i = 0; i <= 9; i++) {
rate[i] = IBI;
} }
if (firstBeat) {
firstBeat = false;
secondBeat = true;
}
word runningTotal = 0
for (int i = 0; i <= 8; i++) {
rate[i] = rate[i + 1];
runningTotal += rate[i];
}
rate[9] = IBI;
runningTotal += rate[9];
runningTotal /= 10;
BPM = 60000 / runningTotal;
QS = true;                    }}
if (SignalNF < thresh && Pulse == true) {
Pulse = false;
amp = P - T;
thresh = amp / 2 + T
P = thresh;
T = thresh;      }
if (N > 2500) {
thresh = 500;
P = 500;
T = 500;
lastBeatTime = sampleCounter;
firstBeat = true;
secondBeat = false;
BPM = 0;        }
total = total - readings[readIndex];   readings[readIndex]
= BPM ;
total = total + readings[readIndex];
readIndex = readIndex + 1;
if (readIndex >= numReadings) {
readIndex = 0;
  }
average = total / numReadings;
  BPMNF = average;
  return true;
}
```

3) AFTER THE DIGITAL FILTER, PPG SIGNAL PROGRAM

```
//DF Butterworth orde 6 LPF fc 3Hz
double c[7] =
{0.000076214549009309991,0.00045728729405585997
,0.00114321823513965,0.0015242909801861998,0.001
14321823513965,0.00045728729405585997,0.0000762
14549009309991};    //NUMERATOR
double d[7] = {1,-
4.182389579168496,7.4916110845876407,-
7.3135959668907287,4.0893499318331035,-
1.2385253717767073,0.15842763255178344};
//DENUMERATOR
double
p,p10,p9,p8,p7,p6,p5,p4,p3,p2,p1,q,q10,q9,q8,q7,q6,q5,
q4,q3,q2,q1,q0;

//DF LMS orde 6 LPF fc 3Hz
double a[7] = {0.12292899210550082,
0.14158376958352209,0.1537839892525007,0.1580266
0571912633,0.1537839892525007,0.141583769583522
09,0.12292899210550082};
double y,y10,y9,y8,y7,y6,y5,y4,y3,y2,y1,y0;

p=c[0]*q0 + c[1]*q1 + c[2]*q2 + c[3]*q3 + c[4]*q4 +
c[5]*q5 + c[6]*q6 - d[1]*p1 - d[2]*p2 - d[3]*p3 -
d[4]*p4 - d[5]*p5 - d[6]*p6 ;
SignalBW = p;

y=a[0]*y0 + a[1]*y1 + a[2]*y2 + a[3]*y3 + a[4]*y4 +
a[5]*y5 + a[6]*y6 ;
SignalLMS = y;
sampleCounter1 += 2;
int N1 = sampleCounter1 - lastBeatTime1;
if (SignalBW < thresh1 && N1 > (IBI1 / 5) * 3) {
if (SignalBW < T1) {
T1 = SignalBW;
    } }
if (SignalBW > thresh1 && SignalBW > P1) {
P1 = SignalBW;

sampleCounter2 += 2;
int N2 = sampleCounter2 - lastBeatTime2;
if (SignalLMS < thresh2 && N2 > (IBI2 / 5) * 3) {
 if (SignalLMS < T2) {
T2 = SignalLMS;
} }
```

```
if (SignalLMS > thresh2 && SignalLMS > P2) {
P2 = SignalLMS;
```

4) DATA IS BEING SENT TO A DELPHI DISPLAY BY A PROGRAM.

```
Serial.print("g");
Serial.print(BPMNF);
//Serial.print(" ");
Serial.print("h");
//Serial.print("BPM BW:");
Serial.print("i");
Serial.print(BPMBW);
//Serial.print(" ");
Serial.print("j");
//Serial.print("BPM LMS:");
Serial.print("k");
Serial.print(BPMLMS);
//Serial.print(" ");
Serial.print("l");
Serial.print("a");
Serial.print(SignalNF);
//Serial.print(" ");
Serial.print("b");
Serial.print("c");
Serial.print(SignalBW);
//Serial.print(" ");
Serial.print("d");
Serial.print("e");
Serial.print(SignalLMS);
Serial.print("f");
digitalWrite(3,LOW);
digitalWrite(3,HIGH);
//delayMicroseconds(7000);
//delay(20);}
```

ATTACHMENT

- Schematic and Board : https://drive.google.com/drive/folders/1jFgGGeOkrVcjlUyK52GBSEY1OG0iggmx?usp=sharing
- Listing Program : https://drive.google.com/drive/folders/1uhA7610ofLT65xglIoPzZ1eewUotvPIB?usp=sharing