Implementation of Real-Time Oscillometric Based Algorithm for Blood Pressure Measurement in Patient Monitor

Trio Adiono*, Clarence Amadeus*, Sindy Novaria Cicilya Sinaga†, and Teuku Rafifsyah Thomi†
*School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia
†PT. Xirka Dama Persada, Indonesia

***Abstract*—As blood pressure is considered one of the most important metrics of human health, it is critical to have a device to measure blood pressure accurately and practically. Non Invasive Blood Pressure(NIBP) device is one of the solutions. To be able to perform well, NIBP devices need to be equipped with the right algorithm to measure blood pressure. One of the most popular blood pressure measurement algorithms is the oscillometric-based algorithm. This algorithm involves Oscillometric Waveform(OMW) extraction, envelope detection, and blood pressure estimation. In this paper, the authors want to develop and implement an oscillometric based algorithm for real-time NIBP measurement in a prototype patient monitor device. The algorithm will be tested in case-based and (Beat Per Minute) BPM-based data, and will be evaluated according to Indonesian Ministry of Health's standard. Our test yields promising result, with errors below 5mmHg**

***Keywords*—** NIBP, Oscillometric, Mean Arterial Pressure, Systolic Blood Pressure, Diastolic Blood Pressure

1. Introduction

Blood pressure is one of the most important metrics for human health. Humans can be very prone to disease if their blood pressure is too high or too low. Some of the most dangerous disease caused by blood pressure is Cardiac Arrest, Stroke, and many more. [1] To avoid this disease, periodical monitoring of blood pressure is vital. Traditionally, blood pressure is measured by medical staff using traditional medical equipment. This method requires good skill from the medical staff to be able to measure the patient's blood pressure. Another problem with this method is that this method requires a meeting between the medical staff and the patient. This is often impractical, especially in emergencies where the patient is unable to go anywhere.

To solve this problem, a simple and portable machine is developed to measure blood pressure automatically. This device is often called NIBP(Non Invasive Blood Pressure) Monitor. Such devices can be purchased easily in health equipment stores. Even some of the hospitals/clinics also use this kind of device because of its practicality [2]. Most NIBP monitor devices use an oscillometric-based algorithm to measure blood pressure. This algorithm uses a cuff that is inflated on the patient's arm to stop blood flow in the arm, then the pressure of the cuff is monitored using a high-precision pressure sensor. From the pressure data, the Oscillometric waveform can be separated and from this waveform, Mean Arterial Pressure(MAP), Systolic Blood Pressure (SBP), and Diastolic Blood Pressure (DBP) can be calculated. To be able to measure blood pressure accurately, the oscillometric algorithm needs to be tuned carefully. In this paper, the authors want to develop and implement an oscillometric based algorithm for real-time NIBP measurement in a prototype patient monitor device.

2. Previous Work

Basically, in NIBP devices, the Oscillometric based algorithm is widely used. It usually involves taking the maximum pressure in the Oscillometric Waveform as Mean Arterial Pressure (MAP), then deciding SBP and DBP based on the MAP. The algorithm to decide SBP and DBP is still an open problem, because different NIBP devices may have different approaches/algorithms. Some approaches use statistical-based method, some use DSP-based method, and some uses Neural Network Approach.

Some of the statistical-based methods involve statistical models to find the systolic and diastolic characteristic ratios. In [3] and [4], Bayesian Model is used to determine the SBP and DBP ratio, while in [5], Gaussian Mixture Model is used to estimate SBP and DBP ratio. These works try to find systolic and diastolic ratios by finding the posterior probability distribution function of these ratios. Although the ratios found using this approach is accurate, the process of calculating these statistical model can be hard and complex. This can make the blood pressure measurement time longer, which is not desired.

Another approach to estimate SBP and DBP is to use the derivative of the Oscillometric Waveform Envelope(OMWE). In [6] and [7], the maximum slope of OMW is used to estimate SBP and DBP ratios. This technique can also produce accurate SBP and DBP ratios compared to the empirically chosen ratios. The problem with this method is that the accuracy of the ratios is highly dependent on the quality of the OMWE. If the OMWE contains noise then the performance of this method can degrade significantly.

Besides using ratios, SBP and DBP can also be obtained from the oscillometric pulse itself. This can be done by evaluating the pulse shape in different cuff pressures [8]. In [9], Digital Signal Processing (DSP) based method is used to evaluate this pulse shape to obtain various pulse morphologies. This technique can lead to a better understanding of the relationship between blood pressure and OMW but requires a good understanding of the DSP concept, which is not practical. Another problem with this technique is that this technique is very noise prone because whenever the OMW is corrupted by noise, its pulse morphology can change drastically.

With the development of AI, various AI models have been proposed to estimate SBP and DBP. For example, Deep Neural Network is proposed to estimate SBP and DBP directly from OMWE in [10]. In [11], Adaptive Neuro-Fuzzy Inference System(ANFIS) is used to estimate SBP and DBP from OMWE. In [12], Principal Component Analysis (PCA) is combined with Neural Networks to estimate SBP and DBP from OMWE. These networks can estimate SBP and DBP accurately and are not sensitive to noise.

The downside of this approach is that some models may require large datasets and computational power.

Despite the success of implementing various concepts to estimate SBP and DBP, the most common and empirical way to determine SBP and DBP is to use the fixed ratio algorithm. In the fixed ratio algorithm, SBP and DBP are estimated using a predetermined fixed ratio to the MAP. This ratio is chosen usually by trial and error. This algorithm usually involves Oscillometric Waveform(OMW) extraction, envelope detection, and SBP-DBP estimation. Some papers discuss this algorithm, and their comparison can be seen in Table 1.

3. Proposed Method

In this paper, the authors used an Oscillometric based algorithm to measure blood pressure. The block diagram of this algorithm is shown in Figure 1. As seen in Figure 1, this algorithm takes the raw cuff pressure data read by the pressure sensor as an input. From this input, the following steps are performed:

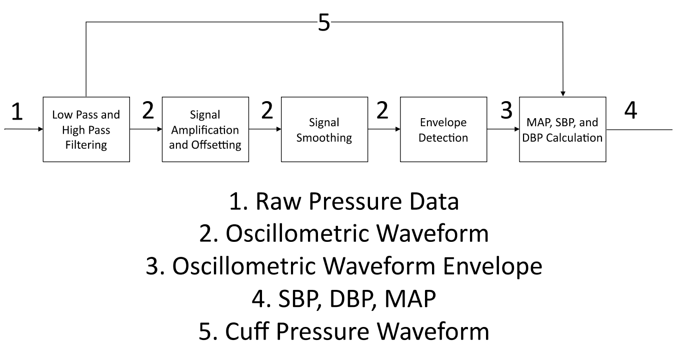

Figure 1. Block Diagram of The Algorithm Used in This Paper

1) Filtering
   The main objective of this step is to separate the Oscillometric Waveform from the Cuff Pressure Waveform. To separate the Oscillometric Waveform from the Cuff Pressure Waveform, low pass, and high pass filtering is performed on the input signal. The signal is first low-pass filtered to get the cuff pressure waveform. Then the cuff pressure waveform is further filtered using the high pass filter to obtain the oscillometric waveform. The low pass filter and high pass filter cutoff frequency are tuneable parameters in this step. An example of the raw data and filtered data can be seen in Figure 2 and Figure 3, respectively.
   As seen in Figure 3, the output of the low pass filter is the declining pressure of the deflating cuff. The MAP, SBP, and DBP will be determined from this cuff pressure waveform. The position of the MAP, SBP, and DBP will be obtained from the oscillometric waveform, which is the output of the high-pass filter.
2) Signal Amplification and offsetting
   The oscillometric waveform obtained from the cuff pressure waveform is still very weak and it contains negative values. To make the oscillation clearer, the signal will be amplificated. After the signal is amplificated, a dc offset will be added to get rid of negative values.
3) Signal Smoothing

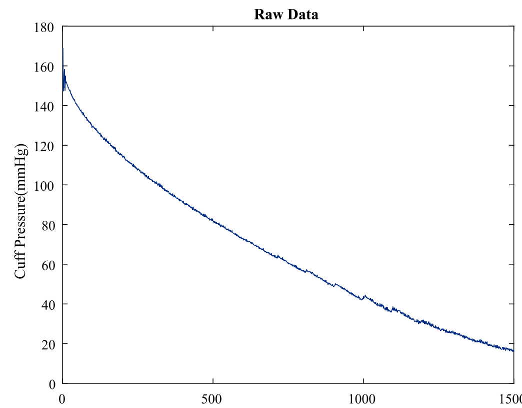

Figure 2. Raw Data of the Cuff Pressure Read by the Sensor

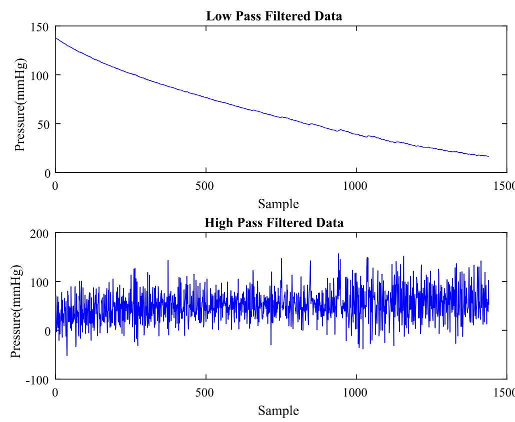

Figure 3. The Result of the Low Pass and High Pass Filter

   The obtained oscillometric waveform is often still noisy, or it has too much oscillation. This will make detecting the envelope of the waveform hard. To avoid this, the signal will be smoothed using a moving average filter. The frame size and repetition of the filter are tuneable parameters in this step. The formula used in the moving average filter is shown in Equation 1.

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \tag{1}$$

   where y[n] is the smoothed signal, N is the length of the signal, and k is the sumation iteration index
   The comparison of the Oscillometric Waveform before and after signal smoothing can be seen in Figure 4. As seen in Figure 4, the oscillometric waveform obtained from the high pass filter is very noisy. This will cause the signal envelope hard to be obtained. Using a moving average filter, the oscillometric waveform is successfully smoothed and the signal is less noisy.
4) Envelope Detection
   After the Oscillometric Waveform is obtained, the envelope of this waveform will be taken. This signal envelope is often called Oscillometric Waveform Envelope(OMWE). To obtain OMWE, an envelope detector

TABLE 1. COMPARISON OF FIXED-RATIO ALGORITHM IN DIFFERENT PAPERS

Paper	OMW extraction	Envelope Detector	SBP and DBP estimation
[13]	Filter 0.3 Hz-6 Hz	not mentioned	ratio of 0.55 and 0.85
[14]	Polinomial fitting	Cubic spline interpolation	ratio of 0.487 and 0.658
[15]	Filter 0.5Hz-1.5 Hz	Cubic spline interpolation	least square method

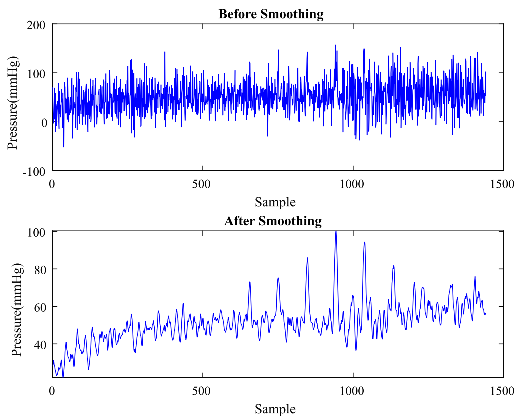

Figure 4. Oscillometric Waveform Before and After

algorithm is used. The flowchart of the algorithm is shown in Figure 5.
As seen in the flowchart, the algorithm takes the maximum point between i^{th} sample and $(i+frame)^{th}$ sample of the oscillometric waveform. After the maximum point is taken, the signal envelope is incremented until it reaches the maximum point. If the signal envelope is larger than the maximum point, the envelope signal will be decremented. After the signal envelope is obtained, it can be smoothed by the moving average filter. The frame length is the only tuneable parameter in this step.

5) MAP, Systolic, and Diastolic calculation
The last step of this algorithm is to determine MAP, SBP, and DBP, which are the value of Mean Arterial Pressure, Systolic Pressure, and Diastolic Pressure output of the measurement. The steps of the calculation are shown in Figure 6. As seen in Figure 6, the program accepts cuff pressure waveform(CPW) and Signal Envelope as input. The first step is to calculate yMAP, which is the maximum amplitude of the envelope signal. Then, the program will calculate ysys and ydia, which is the value of yMAP $\times$ rs and yMap $\times$ rd, respectively. The parameter rs and rd is a pre determined fixed ratio and will be the tuneable parameter of this step. After yMAP, ysys, and ydia is obtained, xMAP, xsys, an xdia is determined. The value of xMAP is taken as the index of yMAP. The value of xsys is taken as the index of min(yMAP-ysys), and the value of xdia is taken as the index of min(yMAP-ydia). The last step is to obtain MAP, SBP, and DBP. MAP is taken as CPW[xMAP], SBP is taken as CPW[xsys], and DBP is taken as CPW[xdia].
An example of MAP, SBP, and DBP calculation can be seen in Figure 7.
As seen in Figure 7, MAP is taken as the pressure in Cuff Pressure Waveform whose index corresponds to the

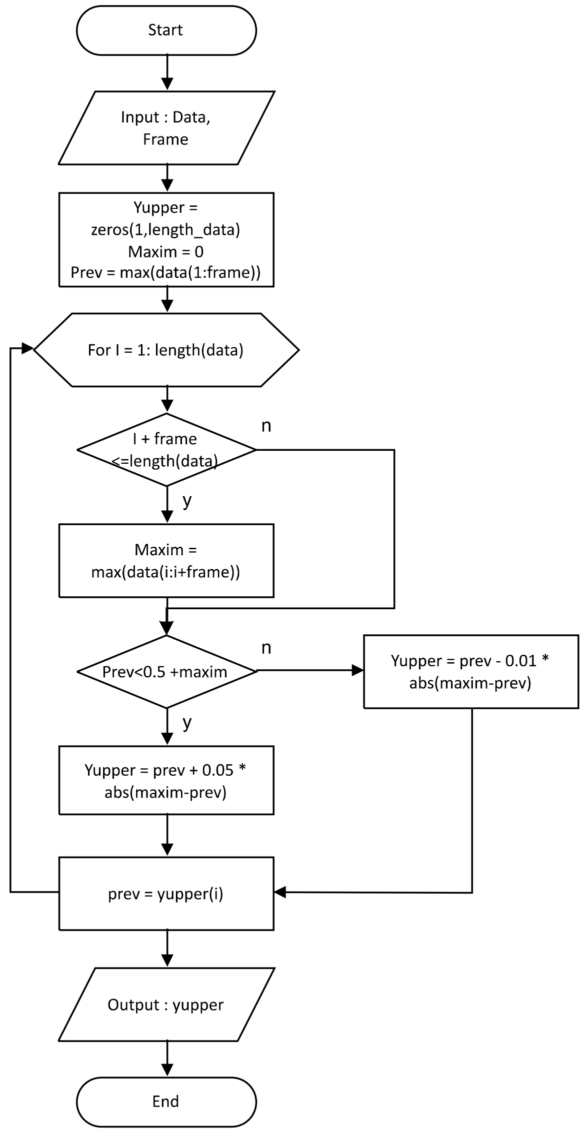

Figure 5. Envelope Detector Flowchart

maxpoint. Maxpoint is defined as the maximum pressure in the Oscillometric Waveform. SBP is taken as the pressure in Cuff Pressure Waveform which index corresponds to the pressure in the Oscillometric Waveform which is closest to sysratio*maxpoint. DBP is taken as the pressure in Cuff Pressure Waveform which index corresponds to the pressure in the Oscillometric Waveform which is

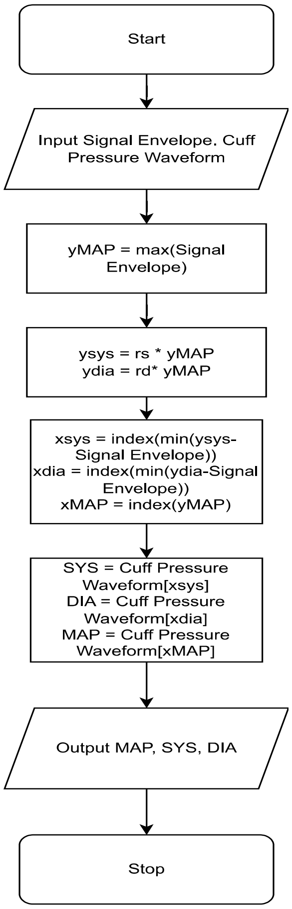

Figure 6. MAP,SYS,DIA Calculation Flowchart

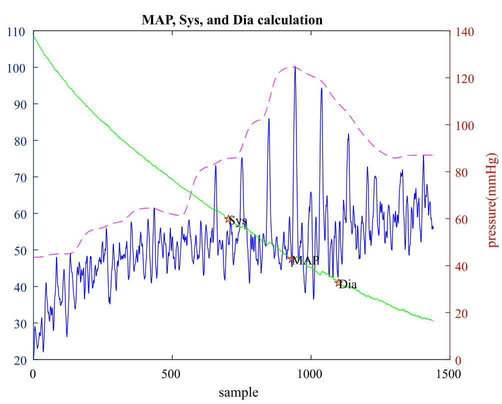

Figure 7. An Example of MAP, SBP, and DBP calculation

closest to diaratio*maxpoint. In this case, MAP is 42 mmHg, SBP is 59 mmHg, and DBP is 33 mmHg.
Although this algorithm is enough for most of the cases, sometimes there are cases such as shown in Figure 8. MAP is 211, SBP is 255, and DBP is 79. It can be seen that DBP is very wrongly estimated because MAP is 215, SBP is 255, and DBP should be 186. To minimize this kind of error, the determination of SBP and DBP can be limited to a certain part of the signal, centered at the maxpoint of the envelope signal. This introduces a new parameter p to determine how much portion to the left and right of the signal envelope is to be considered in SBP and DBP estimation. The example of SBP and DBP estimation of the case in Figure after using the p parameter is shown in Figure 9. Therefore, the tuneable parameters in this step are SysRatio, DiaRatio, and p.

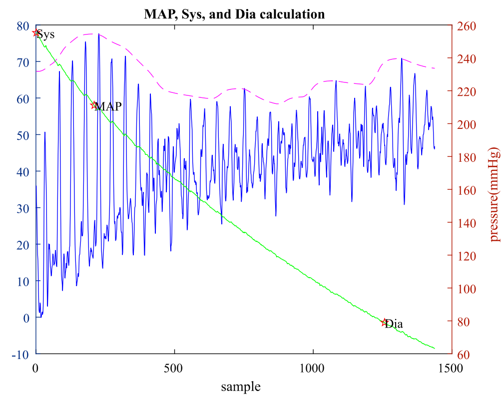

Figure 8. An Example of DBP Estimation Error

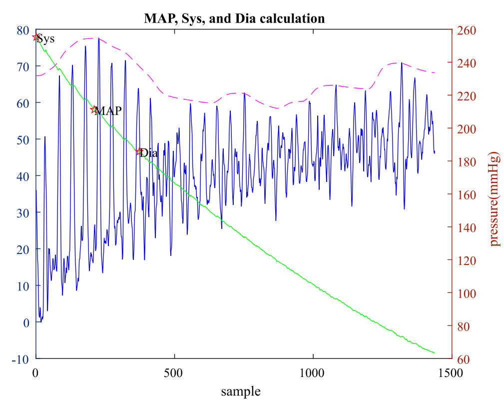

Figure 9. An Example of DBP Estimation After Using p Parameter

3.1. Parameter Selection

In the algorithm that is used in this paper, several tuneable algorithms are available. The value of each parameter that is used in this paper is shown in Table 2

3.2. Dataset Creation

The dataset used in this paper to evaluate the accuracy of the signal is made using CONTEC MS200 NIBP Simulator. The machine can be set to produce a cuff pressure waveform of a certain BPM, Systolic, and Diastolic blood pressure. The pressure produced by this calibrator is read using a pressure sensor, controlled by an ESP32 microcontroller. Finally, the data is printed into the serial monitor and saved to a txt file. The Systolic and Diastolic variation in this dataset can be seen in Table3. Each variation is

TABLE 2. PARAMETER OF THE ALGORITHM

parameters	value
low pass filter cutoff	20Hz
high pass filter cutoff	0.3Hz
moving average filter frame	10 , 5
envelope frame length	100
moving average filter frame	40
p	0.2*data length
r_s	0.75
r_d	0.9

taken for 30,60,120, and 180 BPM, and is repeated 6 times, so in total the dataset has 168 data.

TABLE 3. DATASET DISTRIBUTION

Case ID	Systolic(mmHg)	Diastolic(mmHg)	MAP(mmHg)
1	60	30	40
2	80	50	60
3	100	65	76
4	120	80	93
5	150	100	116
6	200	150	166
7	255	195	215

3.3. Algorithm Implementation

The program implementation is divided into two phases, which are algorithm development and real time implementation of the algorithm. The details of those phases will be explained in the next subsubsections.

3.3.1. Algorithm Development. The algorithm is developed using MATLAB R2023a. MATLAB is chosen because of its versatility in signal processing applications and to make it easier to plot and analyze the signal in each step of the algorithm. Although MATLAB has its library and toolbox to implement various DSP algorithms, in this paper the authors try to minimize the usage of such functions. This is done to make it easier to port the algorithm into other languages, for example, C language.

3.3.2. Real-Time Implementation. The algorithm is implemented in a prototype patient monitor device developed using ESP32. Block Diagram of the system can be seen in Figure 10. As seen in Figure 10, the system consists of a ESP32 microcontroller, a mprls0300yg00001b pressure sensor, 2 valves and valve drivers, a motor driver and a motor pump. The ESP32 controls the system by reading the pressure sensor, turning the pump on/off, and opening/closing the valve. This system is implemented for a prototype patient monitor that can be seen in Figure 11, while an example of the patient monitor display is shown in Figure 12.

4. Result and Discussion

In this section, the testing result will be displayed and discussed

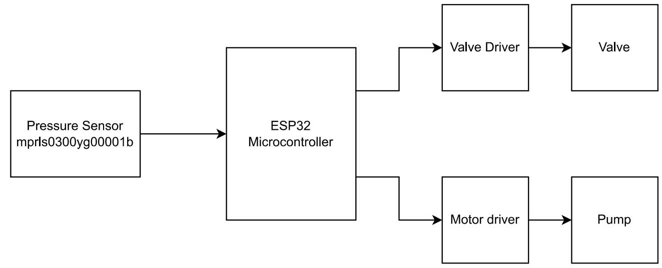

Figure 10. Block Diagram of the Patient Monitor

Figure 11. The Developed Patient Monitor

4.1. Algorithm Performance

The algorithm is tested in two ways. One way is to test the algorithm BPM based, and the other way is to test the algorithm Case-Based. These two methods are done to see if the algorithm is robust in terms of case and BPM. If the algorithm's performance is different in each case, then different parameters need to be adjusted in each case/BPM. The BPM and case-based performance of the algorithm can be seen in Table 4 and 5, respectively.

TABLE 4. BPM-BASED PERFORMANCE OF THE ALGORITHM

BPM	MAP Error(mmHg)	Sys Error	Dia Error
30	3.1667	4.0476	2.8333
60	3.0476	3.1905	2.3810
120	3.1429	4.0952	3.2143
180	3.4762	2.8571	2.5238

4.2. Discussion

According to the document issued by Indonesian Health Ministry on Bed Side Monitor Device testing, the maximum difference/error of the blood pressure reading is +-5 mmHg. This standard is taken to evaluate the performance of the algorithm.

Figure 12. The Developed Patient Monitor

TABLE 5. CASE-BASED PERFORMANCE OF THE ALGORITHM

Case	MAP Error(mmHg)	Sys Error	Dia Error
1	2.9583	3.8333	2.3750
2	3.1250	4.5	1.625
3	2.875	3.2917	2.7083
4	3.25	2.2083	2.3333
5	3.4167	4.8333	3.2083
6	3.5	4.375	3.5
7	3.4167	3.125	3.4167

Table 4 and 5 shows that in each BPM and case, no average error is more than 5 mmHg. This means that the algorithm works well in each BPM and each case. Hence, no further parameter tuning is needed.

5. Conclusion

In this paper, a algorithm for calculating blood pressure using NIBP method is proposed. The algorithm is suitable for real-time NIBP measurement. To evaluate the algorithm, it is then tested using a database created using the NIBP machine calibrator according to the Indonesian Ministry of Health standard. The algorithm manages to read Systolic and Diastolic blood pressure with mean error below +-5 mmHg, which satisfies the condition required by the Indonesian Health Ministry. This performance is valid for 7 different blood pressure cases and 4 different BPM, which shows the robustness of this algorithm. To further improve the work done in this paper, this algorithm can be implemented to measure a real patient's blood pressure. This will further test the algorithm's robustness, as well as the complexity of the algorithm.

Acknowledgment

This work was supported in part by Indonesia Endowment Fund for Education Agency (LPDP) under Grant PRJ-50/LPDP/2020, S-248/LPDP.4/2021, and S-446/LPDP.4/2023.

References

[1] *Clinical methods: The history, physical, and laboratory examinations*, 3rd ed. LexisNexis UK, 1990.

[2] P. S. Lewis, "Oscillometric measurement of blood pressure: A simplified explanation. a technical note on behalf of the british and irish hypertension society," *Journal of Human Hypertension*, vol. 33, no. 5, pp. 349–351, 2019.

[3] S. Lee, G. Jeon, and G. Lee, "On using maximum a posteriori probability based on a bayesian model for oscillometric blood pressure estimation," *Sensors (Basel, Switzerland)*, vol. 13, pp. 13 609–23, 10 2013.

[4] S. Lee, S. Rajan, H. R. Dajani, V. Z. Groza, and M. Bolic, "Determination of blood pressure using bayesian approach," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, 2011, pp. 1–5.

[5] S. Lee, J.-H. Chang, S. W. Nam, C. Lim, S. Rajan, H. R. Dajani, and V. Z. Groza, "Oscillometric blood pressure estimation based on maximum amplitude algorithm employing gaussian mixture regression," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 12, pp. 3387–3389, 2013.

[6] M. Mafi, S. Rajan, M. Bolic, V. Z. Groza, and H. R. Dajani, "Blood pressure estimation using maximum slope of oscillometric pulses," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 3239–3242.

[7] A. H. Drzewiecki G, Hood R, "Theory of the oscillometric maximum and the systolic and diastolic detection ratios," *Annals of Biomedical Engineering*, vol. 22, 01 1994.

[8] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, S. Rajan, and I. Batkin, "Oscillometric blood pressure estimation: Past, present, and future," *IEEE Reviews in Biomedical Engineering*, vol. 8, pp. 44–63, 2015.

[9] M. Mafi, S. Rajan, M. Bolic, V. Groza, and H. Dajani, "Blood pressure estimation using oscillometric pulse morphology," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2011, pp. 2492–6, 08 2011.

[10] Å. Ãolak and C. Isik, "Blood pressure estimation using neural networks," 08 2004, pp. 21 – 25.

[11] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, and S. Rajan, "Adaptive neuro-fuzzy inference system for oscillometric blood pressure estimation," in *2010 IEEE International Workshop on Medical Measurements and Applications*, 2010, pp. 125–129.

[12] ——, "Oscillometric blood pressure estimation using principal component analysis and neural networks," in *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, 2009, pp. 981–986.

[13] R. Jaafar, H. M. Desa, Z. Mahmoodin, M. R. Abdullah, and Z. Zaharudin, "Noninvasive blood pressure (nibp) measurement by oscillometric principle," in *2011 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, 2011, pp. 265–269.

[14] S. Kumar, S. Yadav, and A. Kumar, "Oscillometric waveform evaluation for blood pressure devices," *Biomedical Engineering Advances*, vol. 4, p. 100046, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667099222000226

[15] P. Huang, J. Fu, and Y. Zhao, "A matlab demo software for nibp measurement based on hybrid oscillometric method," in *2011 4th International Congress on Image and Signal Processing*, vol. 5, 2011, pp. 2536–2539.