Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 21 No. 1 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v21i01.51717

PAPER

Real-Time Optimization of VMD in Healthcare Embedded Systems Using Parallel Processing with OpenMP

Zakaria El Khadiri[1](✉), Rachid Latif[1], Amine Saddik[1,2]

[1]Laboratory of Systems Engineering and Information Technology (LISTI), National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco

[2]Faculty of Applied Sciences, Ibn Zohr University, Ait Melloul, Morocco

zakaria.elkhadiri@edu.uiz.ac.ma

ABSTRACT

Variational mode decomposition (VMD) is an advanced signal processing technique used to analyze photoplethysmogram (PPG) signals to extract vital physiological indicators such as heart and respiratory rates. However, implementing VMD on embedded systems presents challenges due to limited computational resources and the need for real-time performance. This paper investigates the optimization of VMD using OpenMP to improve the performance of healthcare-focused embedded systems. We first describe the naive VMD implementation and identify computational bottlenecks. We then optimize the algorithm using OpenMP by parallelizing critical sections, including iterative updates and loop structures, within a homogeneous CPU architecture. We evaluate performance based on metrics such as processing time, efficiency, and the accuracy of heart rate extraction. Experimental results demonstrate that the optimized VMD algorithm achieves a near-linear speedup, with processing times reduced by up to 9.45 times compared to the naive single-threaded version while maintaining efficient resource utilization. This optimization enables real-time signal processing in healthcare applications, improving the performance and reliability of medical devices for patient monitoring and care.

KEYWORDS

healthcare, embedded systems, OpenMP, variational mode decomposition (VMD), photoplethysmogram (PPG)

1 INTRODUCTION

Embedded systems are specialized computing platforms tailored to perform dedicated functions within broader systems. In the healthcare domain, they are integral to devices such as wearable health monitors, portable diagnostic tools, and real-time patient monitoring equipment. Wearable health monitors, for instance, are designed to continuously collect and analyze physiological data from sensors attached to the body, including accelerometers, gyroscopes, and photoplethysmogram (PPG) sensors [1–6]. PPG sensors measure blood volume changes in the microvascular

El Khadiri, Z., Latif, R., Saddik, A. (2025). Real-Time Optimization of VMD in Healthcare Embedded Systems Using Parallel Processing with OpenMP. *International Journal of Online and Biomedical Engineering (iJOE)*, 21(1), pp. 97–117. https://doi.org/10.3991/ijoe.v21i01.51717

Article submitted 2024-08-14. Revision uploaded 2024-11-02. Final acceptance 2024-11-04.

bed of tissue, crucial for determining heart rate and other cardiovascular metrics. Portable diagnostic tools are handheld or easily transportable devices that perform diagnostic tests at the point of care, enhancing accessibility and efficiency in healthcare delivery. Real-time patient monitoring systems continuously track patients' vital signs and other health metrics, proving essential in intensive care units (ICUs) and remote patient monitoring setups.

These healthcare applications demand efficient, low-power processing to handle the continuous data streams generated by various sensors [7–12]. Embedded systems in this context must be capable of executing complex algorithms for signal processing and analysis, often operating under stringent power and resource constraints. For instance, a wearable health monitor must operate for extended periods on battery power, necessitating the use of energy-efficient processors and optimized software to minimize power consumption. Additionally, the miniaturization of hardware components is critical to maintaining the comfort and usability of wearable devices, further emphasizing the need for low-power solutions. Real-time analysis of physiological signals, such as heart rate and respiratory rate, is a critical function in these systems. Accurate and timely processing of these signals can significantly enhance patient care by providing immediate insights into the patient's health status [13–15]. Because PPG signals are typically noisy and non-stationary, advanced signal processing techniques are essential to extract relevant physiological information accurately [16–22].

Variational mode decomposition (VMD) is one such technique that decomposes the PPG signal into its constituent modes, facilitating the extraction of heart rate and respiratory rate with high accuracy [23–26]. VMD's ability to adaptively separate signal components based on their intrinsic characteristics makes it particularly suitable for analyzing complex physiological signals. However, the computational complexity of VMD presents significant challenges, particularly in embedded platforms that have limited processing power and memory. This necessitates the use of optimization techniques to ensure real-time performance without compromising accuracy. Implementing computationally intensive algorithms such as VMD on embedded systems requires optimization strategies such as parallel processing with OpenMP. OpenMP allows for the distribution of computational tasks across multiple processor cores, effectively reducing the execution time and enhancing processing efficiency. By leveraging OpenMP, it is possible to achieve significant performance improvements, enabling the real-time processing capabilities necessary for healthcare applications. Efficient implementation of these algorithms not only improves the responsiveness of the monitoring system but also maintains the accuracy of the extracted physiological parameters.

A state-of-the-art method for signal processing, VMD decomposes a signal into its modes, each of which represents a different frequency component. This technique is especially useful for extracting important metrics such as heart and breathing rates from non-stationary physiological data such as PPGs. However, VMD's computational complexity presents many difficulties, particularly in embedded systems and real-time applications where memory and processing capacity are scarce. Effective use of VMD is essential to guaranteeing accurate and rapid analysis, which has an immediate effect on patient care and monitoring. To address these challenges, we leverage OpenMP, a widely used framework for parallel programming. By dividing workloads among several processor cores, OpenMP makes it possible to parallelize computationally demanding processes, significantly enhancing processing efficiency. Our goal is to enhance VMD's functionality and enable real-time healthcare applications in embedded systems by integrating OpenMP with it. This approach not only meets the stringent requirements of low-power, high-efficiency embedded systems but also ensures that the computational demands of VMD are met, thereby facilitating its widespread adoption in healthcare technologies.

This paper describes a comprehensive study that uses OpenMP for parallel processing to optimize the VMD implementation in healthcare-embedded systems. The main objective is to drastically reduce the processing time of VMD, making it suitable for real-time analysis of physiological signals. To illustrate the usefulness of parallel processing in embedded healthcare systems, the study also aims to assess the computational speedup efficiency of the optimized VMD algorithm. Ultimately, our purpose is to contribute to the development of more efficient and reliable healthcare monitoring solutions that can function effectively within the constraints of embedded systems.

The rest of the paper is structured as follows: Section 2 explores the principles and applications of VMD, also illustrates parallel processing, and provides a summary of OpenMP's capabilities and benefits for parallel programming, with its use in the medical field. The optimization process and methodology are described in Section 3, along with performance evaluation criteria and details on algorithm design and OpenMP parallel implementation. In Section 4, experimental results are presented, comparing the processing times of optimized and naive VMD implementations, highlighting the improvements achieved. Finally, Section 5 concludes the paper by summarizing key findings.

2 VARIATIONAL MODE DECOMPOSITION AND PARALLEL PROCESSING

2.1 Variational mode decomposition

K. Dragomiretskiy et al. [27] developed a non-recursive signal decomposition method that aims to decompose a given signal into a set of band-limited intrinsic mode functions (IMFs); the algorithm is fully established on a mathematical framework. The main idea behind VMD is to extract the IMFs all at once via an iterative process; hence the number of modes K should already be established. By increasing the value of K, this condition causes the VMD's complexity and calculation time to increase proportionately. In addition, the method might produce a noisy or mixed mode if K had an ill-set value. As a result, many researchers offer some methods for figuring out K appropriately. The most notable is known as detrended fluctuation analysis (DFA), and several writers [28, 29] employ it. In addition, the VMD technique is associated with three primary ideas: frequency shifting to baseband by complex harmonic mixing, the Hilbert transform for constructing single-sideband analytic signals, and the Wiener filter for denoising signals. Variational mode functions (VMFs), which are the major decomposed modes, are capable of reproducing the original signal with various sparsity characteristics. Additionally, the following is a concise summary of the VMD theory:

1. Initialize:

$$\left\{U_k^1\right\}, \left\{\omega_k^1\right\}, \lambda^1, n \leftarrow 0$$

Where:

$\{U_k\}$: = $\{U_1, \ldots, U_k\}$ are shorthand notations for the set of all modes.

$\{\omega_k\}$: = $\{\omega_1, \ldots, \omega_k\}$ are shorthand notations for the center frequencies of the modes.

The iteration is from 1 to k = number of modes.

$\{\lambda^n\}$: are the Lagrangian multiplier.

$\hat{f}(\omega)$: is the Fourier transform of the original signal.

α: is a parameter for balancing.

2. Compute the following values until the number of modes (K) is reached:

$$\hat{U}_k^{n+1}(\omega) \leftarrow \frac{\hat{f}(\omega) - \sum_{i<k} \hat{U}_i^{n+1}(\omega) - \sum_{i>k} \hat{U}_i^{n}(\omega) + \frac{\hat{\lambda}^{n}(\omega)}{2}}{1 + 2\alpha\left(\omega - \omega_k^n\right)^2} \tag{1}$$

$$\omega_k^{n+1} \leftarrow \frac{\int_0^\infty \omega \left|\hat{U}_k^{n+1}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{U}_k^{n+1}(\omega)\right|^2 d\omega} \tag{2}$$

3. Update the Lagrangian multiplier through the dual ascent method as follows:

$$\lambda^{n+1}(\omega) \leftarrow \hat{\lambda}^{n}(\omega) + \tau\left(\hat{f}(w) - \sum_k \hat{U}_k^{n+1}(\omega)\right) \tag{3}$$

4. Repeat steps 2 and 3 until the function is converged based on convergence criteria satisfied by the condition:

$$\sum_k \left\|\hat{U}_k^{n+1} - \hat{U}_k^{n}\right\|_2^2 \Big/ \left\|\hat{U}_k^{n}\right\|_2^2 < \epsilon, \text{ where } \epsilon \text{ is a given accuracy requirement.}$$

Furthermore, the following Pseudo-code shows the VMD approach (see Algorithm 1):

Algorithm 1: VMD

```
Start algorithm
// Variational Mode Decomposition – VMD
// The VMD function has the following parameters
function VMD (f, alpha, tau, K, DC, init, tol)
  // Initialize variables
  N ← length(f); // Length of input signal f
  f_hat ← fftshift(fft(f)); // Fourier transform of f, shifted
  u_hat ← zeros (K, N); // Initialize mode functions
  omega ← if init == 1 then 0.5 * pi * (0: K – 1) / K else init
  lambda_hat = zeros (1, N); // Initialize Lagrange multipliers
  for n = 1: MaxIter do // Main iteration loop
   u_hat_old ← u_hat;
   for k = 1: K do // Update each mode
    u_hat (k, :) ← (res. * exp (–1j * t * omega(k))). / (1 + 2 * alpha * (t – omega(k)). ^ 2);
    end for
    for k = 1: K do //Update center frequencies
      omega(k) ← sum (t. * abs (u_hat (k, :)). ^ 2) / sum (abs (u_hat (k, :)). ^ 2);
    end for
    lambda_hat = lambda_hat + tau * (f_hat – sum (u_hat, 1));
    if norm (u_hat – u_hat_old, 'fro') / norm (u_hat_old, 'fro') < tol then break
  end for
  for k = 1: K do
    u (k, :) = ifft (ifftshift (u_hat (k, :))); // Compute final modes in time domain
  end for
  return u, omega
end function
END algorithm
```

The length of the input signal, its Fourier transform, mode functions, and Lagrange multipliers are the first variables that the VMD algorithm sets up. The user can specify the initialization of center frequencies, or they can be set equally. After that, the algorithm iterates, updating the mode functions and modifying them in the frequency domain for each iteration based on residual calculations. In order to make sure that the sum of all modes closely resembles the original signal, it also updates the Lagrange multipliers and the center frequencies using a weighted average. By measuring the relative change in modes, convergence is verified. The result is the collection of decomposed modes and their center frequencies. The modes are translated back to the time domain once the algorithm converges. The capacity of VMD to decompose complex signals into their constituent parts, such as PPG data, makes it very useful in the medical field. Because PPG signals record changes in blood volume, they are essential for monitoring cardiovascular health. We can separate several physiological components, especially those associated with cardiac and respiratory activities, by applying VMD to PPG signals. For example, by identifying peaks that correspond to heartbeats, the cardiac component of the PPG signal, once separated via VMD, can be evaluated to extract the heart rate. In the same way, the breathing pattern is revealed by the respiratory component, which is similarly isolated using VMD. The frequency of these respiratory cycles can then be used to calculate the respiratory rate.

For more convenience, consider an original PPG signal with overlapping cardiac and respiratory data. We can decompose this signal into other modes using VMD, and we can then determine which of these modes correspond to cardiac and respiratory activity by assessing their frequency content. This procedure demonstrates how well VMD extracts physiological indications from PPG signals, enhancing the quality of health monitoring. On the other hand, there are various technical obstacles when implementing VMD in embedded systems, notably for real-time healthcare applications. Processing time is one of the biggest challenges since it directly affects the feasibility of adopting VMD in real-time applications such as continuous physiological signal monitoring.

2.2 Parallel processing and OpenMP

Through the use of several processors or cores, a problem can be divided into smaller subproblems and solved concurrently through the use of parallel processing [30–35]. This method is especially helpful for real-time applications in the healthcare industry, where timely data analysis is crucial, as it greatly increases and enhances computational efficiency and decreases processing time. On embedded systems, complicated algorithms such as VMD can be executed more efficiently. For more convenience, the following Figures 1 and 2 show an overview of parallel processing and the fork and join model:

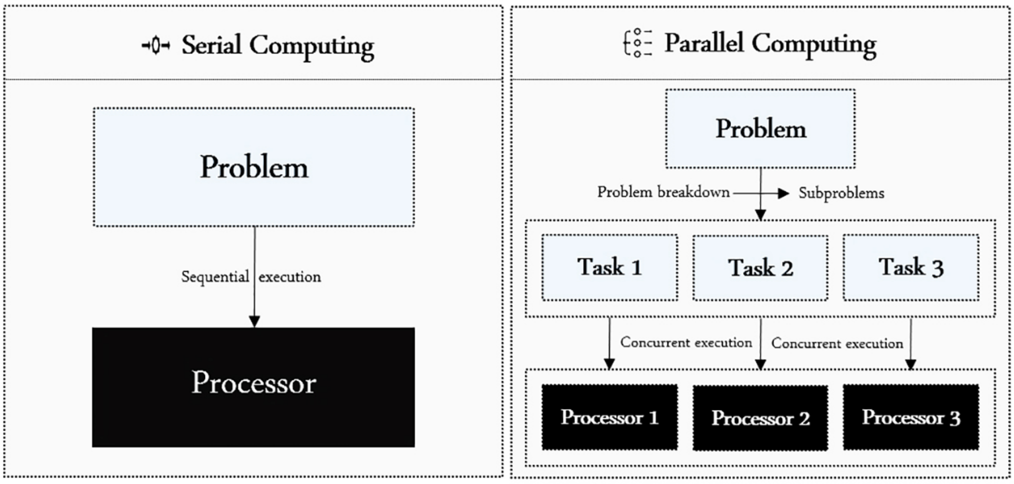

Fig. 1. Serial computing and parallel computing

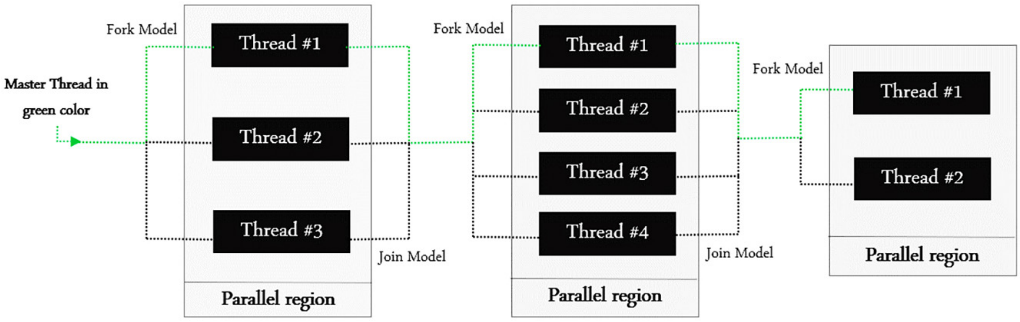

Fig. 2. Fork and join model

As shown in Figures 1 and 2 above, parallel processing and OpenMP offer significant advancements over traditional serial computing by enabling the simultaneous execution of multiple tasks, thus improving computational efficiency and speed. In serial computing, a problem is processed directly by a single processor, executing one instruction at a time. This sequential approach can be time-consuming, especially for complex or large-scale problems. Conversely, parallel computing divides a problem into multiple smaller tasks, each of which is executed concurrently across multiple processors. This division allows for a more efficient use of computational resources, significantly reducing the overall processing time. OpenMP is a widely used API that simplifies the implementation of parallel processing in programs. It provides a set of compiler directives, library routines, and environment variables that facilitate the parallelization of code. By leveraging OpenMP, developers can efficiently manage and distribute tasks across available processors, leading to enhanced performance and scalability in various applications, including real-time signal processing in healthcare-embedded systems.

2.3 Overview OpenMP in healthcare applications

From real-time data analysis in wearable devices to image processing in medical imaging, OpenMP has proven useful in a variety of healthcare applications [36–38].

For example, using OpenMP to construct parallel algorithms can speed up the processing of large datasets produced by ECG or PPG monitoring systems, facilitating quicker diagnosis and decision-making. Figure 3 shows the application of OpenMP in healthcare:

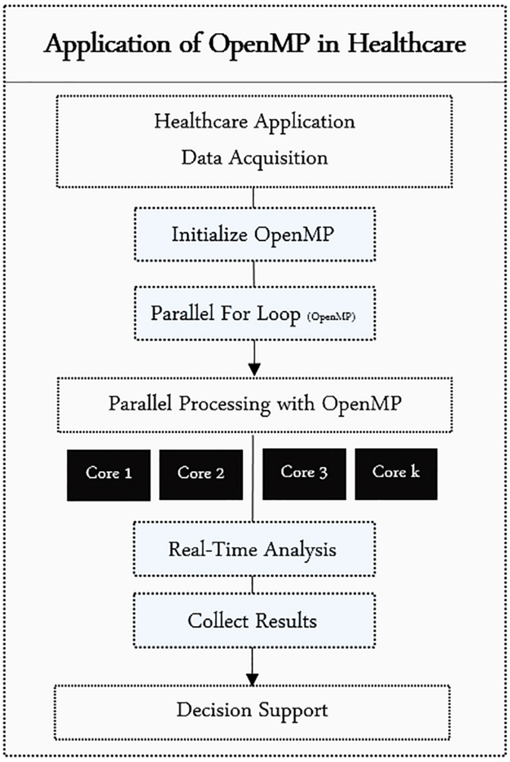

Fig. 3. Application of OpenMP in healthcare

3 OPTIMIZATION METHODOLOGY

To estimate the heart rate using the BUT PPG database, we propose an approach that includes signal normalization, signal decomposition, selection of relevant IMFs, application of Fourier Transform, and extraction of physiological parameters. Initially, the PPG signal is retrieved from the BUT PPG database [39–41], and it is normalized to maintain amplitude consistency and remove baseline drift. The normalized signal is then decomposed using VMD to separate it into several IMFs. For this study, we employ both a naive version and an optimized version of VMD using OpenMP. The naive version processes the signal sequentially, while the optimized version leverages parallel processing to enhance computational efficiency. After decomposition, the relevant IMFs, which contain significant frequency components related to heart rate, are selected. The selected IMFs are then subjected to Fourier Transform to convert the time-domain signal into the frequency domain. From the frequency spectrum, the dominant frequency corresponding to the heart rate is identified. This approach not only allows for accurate heart rate extraction but also demonstrates the performance benefits of the optimized VMD implementation in real-time healthcare applications. Our proposed approach can be depicted in Figure 4:

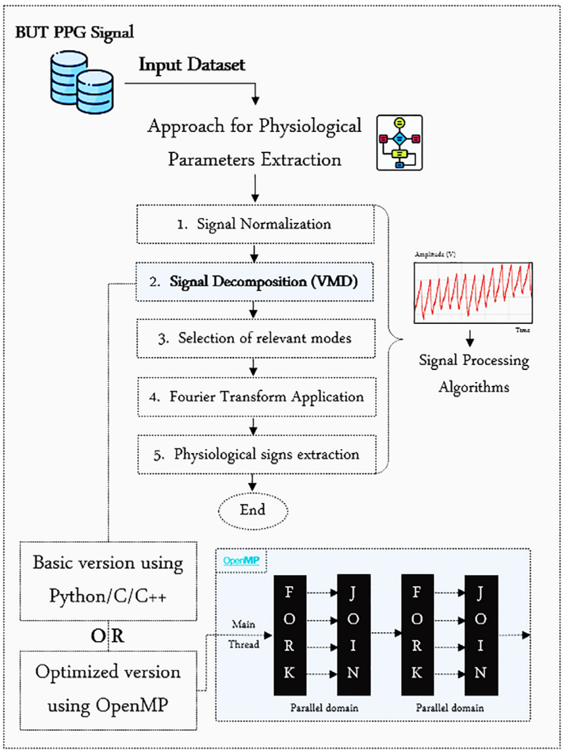

Fig. 4. Overview of the proposed methodology

The optimization of the VMD algorithm focuses on increasing computational efficiency and enabling real-time processing capabilities. As aforementioned above, the main phases involved in VMD include initialization, iterative mode updating, frequency updating, and convergence checking. Each of these steps presents opportunities for optimization. Hereunder, Figure 5 shows the most prominent steps involved in the VMD method.

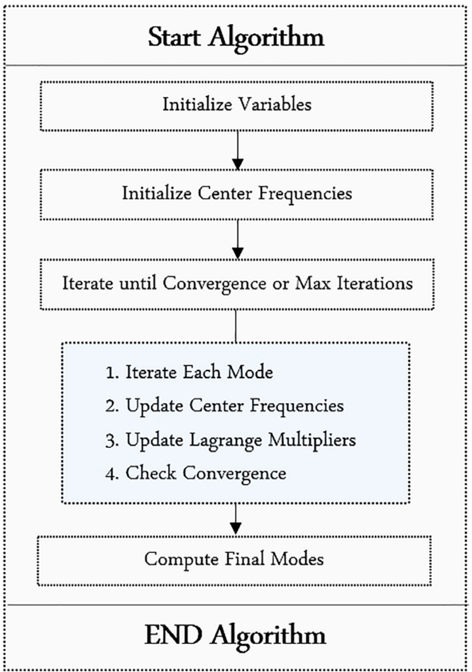

Fig. 5. Main steps of the VMD method

In the optimization process, we examine each step's computing complexity to find portions that can be parallelized. The objective is to distribute the workload across multiple processing units using OpenMP, thereby reducing the overall execution time. As stated before, OpenMP uses an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran. It was chosen for this study due to its simplicity and powerful capabilities for parallelizing computational tasks. The proposed parallel implementation of VMD using OpenMP involved initialization as the first step, which means setting up the signal and initial parameters for VMD. After that, we proposed a parallel mode update by distributing the computation of mode updates across multiple threads. Each thread handles a subset of the modes, updating them concurrently. Then, frequency updates are performed by computing the center frequencies for each mode in parallel, leveraging the shared memory model of OpenMP. In the end, we found the Lagrange multiplier update, which illustrates the adjustment of the Lagrange multipliers to ensure convergence, performed in parallel to reduce overall computation time. Furthermore, the following Pseudocode illustrates the parallelized sections of the VMD algorithm (see Algorithm 2) using OpenMP directives:

Algorithm 2: Parallelized VMD Algorithm

```
// Section of VMD's code
// Application of OpenMP directives
// Parallel mode update (OpenMP)
#pragma omp parallel for
for k = 0: K do
    // Update mode function u_hat(k)
    sum_others = calculate_sum_others (u_hat, k);
    res = f_hat – sum_others – lambda_hat/2;
    u_hat[k] = update_mode_function (res, omega[k], alpha, t);
end for
// Parallel frequency update (OpenMP)
#pragma omp parallel for
for k = 0: K do
    // Update center frequency omega(k)
    omega[k] = calculate_center_frequency(u_hat[k], t);
end for
// Parallel Lagrange multiplier update (OpenMP)
#pragma omp parallel for
lambda_hat = lambda_hat + tau * (f_hat – sum (u_hat, 1));
// Check for convergence
```

To assess the efficiency of our optimization, we established multiple performance measures. The main metric is processing time, which is the overall amount of time needed to decompose the signal into its constituent modes. We also took into account speedup, which is determined by dividing the processing time of the optimized implementation by the processing time of the naive implementation. However, the processing time is the way to compare the execution time of the naive and optimized versions of the VMD algorithm. Afterward, the computing of the speedup factor serves as the ratio of the naive execution time to the optimized execution time. After that, the efficiency, which is used to assess the efficiency of parallelization by analyzing the utilization of computational resources, then, the scalability test of the implementation by varying the number of processing cores and observing the performance changes. Here below are the formulas for the speedup and efficiency calculations:

$$S = \frac{T_{naive}}{T_{optimized}} \tag{4}$$

$$E = \frac{S}{Number\ of\ cores} \tag{5}$$

Where:

- S is the speedup.
- E is the efficiency.
- T_{naive} is the total time for the naïve version of the VMD method.
- $T_{optimized}$ is the total time for the optimized version of the VMD method.

4 RESULTS

The results of our heart rate estimation approach will be presented in this section, along with a comparison with the results produced by the Brno University of Technology's Department of Biomedical Engineering's cardiology team and their numerous annotators and a comparison between the naïve and the optimized version in terms of processing time, speedup, and efficiency. Let's start by looking at an illustration of a PPG signal that was utilized in our study, as seen in Figure 6.

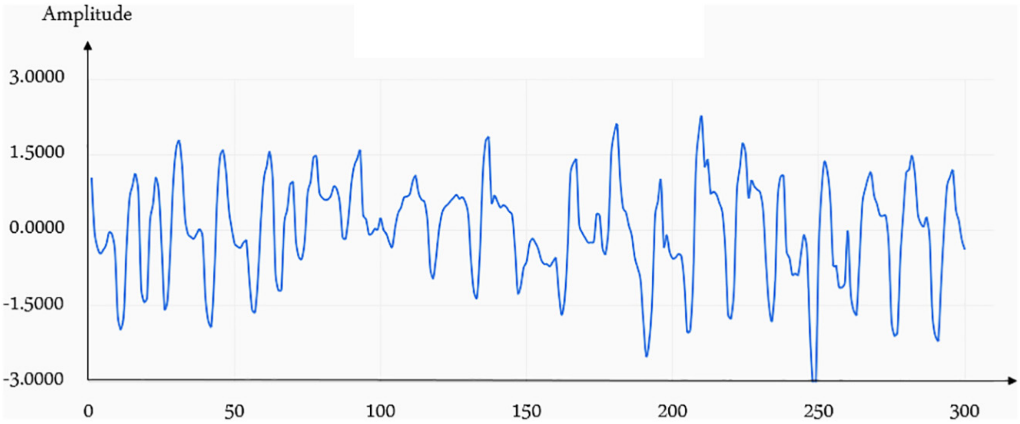

Fig. 6. Normalized PPG signal

To evaluate the efficacy of our proposed VMD-based heart rate extraction method, we compared heart rate values obtained from both the naive and optimized versions of the algorithm with those annotated by five different annotators from the BUT PPG database. This approach enables us to ensure robustness by validating against a diverse set of reference annotations. The heart rate values retrieved from the PPG signals using our method are shown in Table 1.

Table 1. Heart rate values extracted by annotators vs VMD approach

Gender	Age	Weight	Ant. 1	Ant. 2	Ant. 3	Ant. 4	Ant. 5	VMD appr.
F	51	58	82	84	84	82	84	84
F	51	58	85	84	84	84	81	84
F	51	58	83	84	84	83	87	84
F	54	63	66	67	67	66	67	69
F	54	63	70	72	69	69	70	72
F	54	63	71	70	71	70	71	72
F	61	70	68	68	68	68	67	66
F	61	70	66	67	66	65	67	66
F	61	70	67	67	66	66	68	66
M	23	71	82	90	0	78	80	78
M	23	71	90	92	90	90	92	90
M	23	71	102	102	95	100	98	102
M	24	84	110	108	110	110	110	108
M	24	84	120	122	120	120	124	120
M	24	84	113	116	116	115	117	114
F	21	69	83	82	84	80	82	84
F	21	69	93	93	93	95	91	90
M	59	80	64	63	63	64	61	66
M	59	80	68	66	68	68	68	66
M	23	73	76	75	75	76	74	78
M	23	73	105	105	106	100	117	102
M	24	70	72	76	72	72	75	72
M	24	70	76	73	78	74	80	78
M	24	70	75	75	75	75	75	72

As shown in Table 1, the heart rate values extracted by both versions of our VMD-based method closely match those recorded by the annotators. The version that has been optimized by utilizing OpenMP for parallel processing is extremely appropriate for real-time applications since it not only closely matches the annotators' output but also drastically reduces the processing time.

To evaluate the performance of our heart rate estimation algorithm, we use the Bland-Altman plot and scatter plot [42]. These visualizations and statistical metrics provide insights into the relationship between estimated heart rates and the reference values annotated by the cardiology team. Figures 7–16 display the comparison's findings. The heart rate estimation results from our technique are represented by each data point in the scatter plot, which is shown against the relevant reference value provided by the cardiology team and their annotators. The plot's trends and point distribution can provide information on the precision and coherence of our methodology. The degree and direction of the linear association between the estimated heart rates and the reference values are quantified by the correlation coefficient. A correlation strength that is close to one suggests that our methodology yields extremely accurate heart rate estimates. Conversely, a score near 0 denotes a poor association and raises the possibility that our estimations are not as accurate.

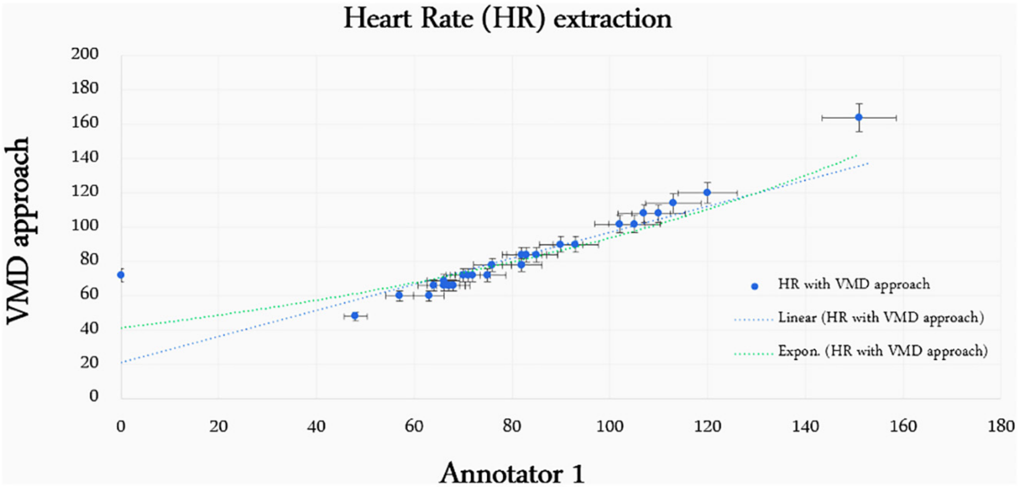

Fig. 7. Scatter plot between annotator 1 and VMD

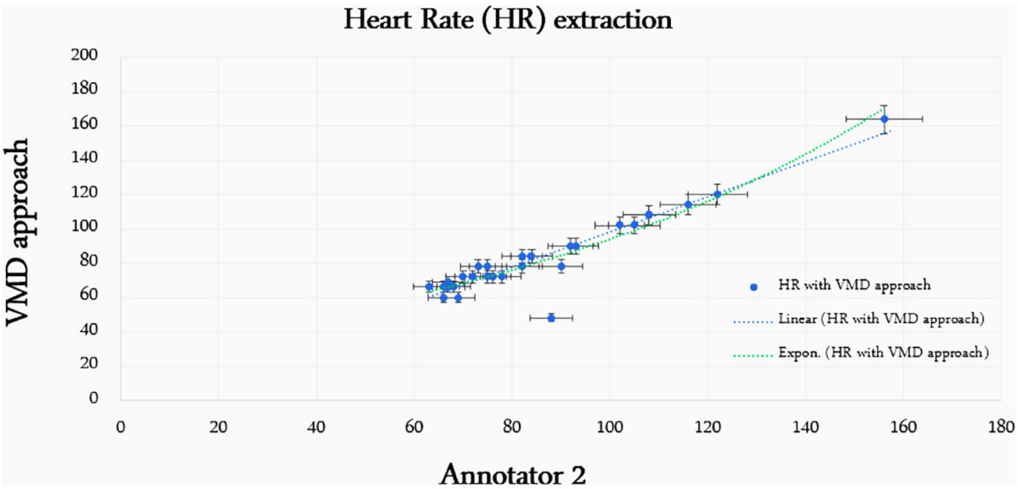

Fig. 8. Scatter plot between annotator 2 and VMD

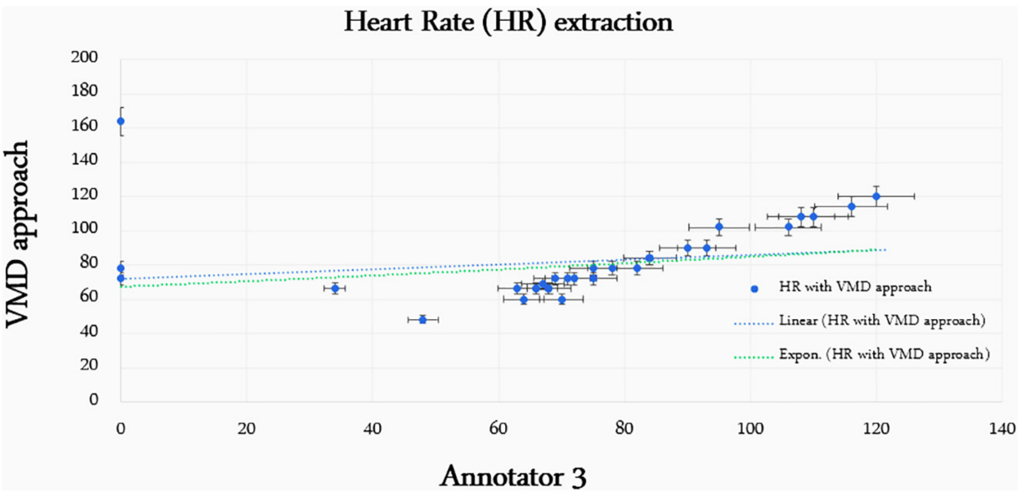

Fig. 9. Scatter plot between annotator 3 and VMD

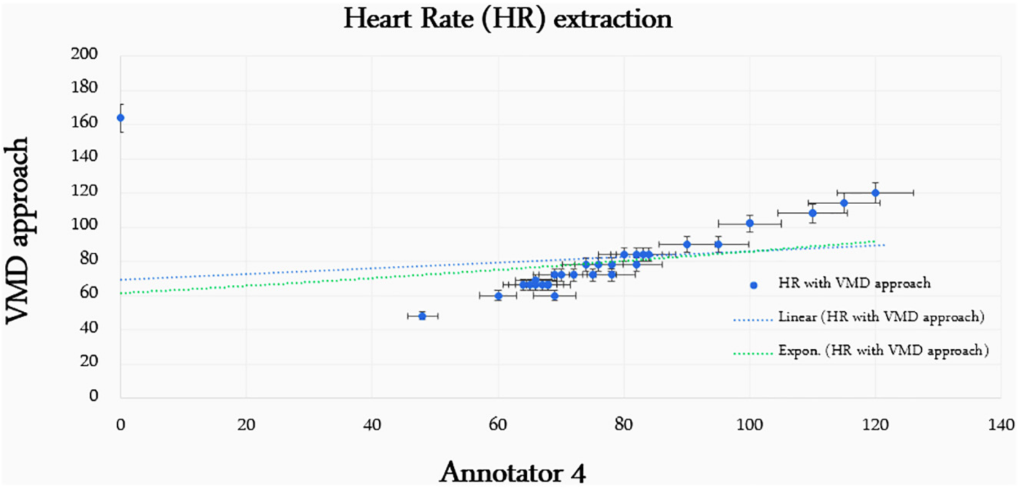

Fig. 10. Scatter plot between annotator 4 and VMD

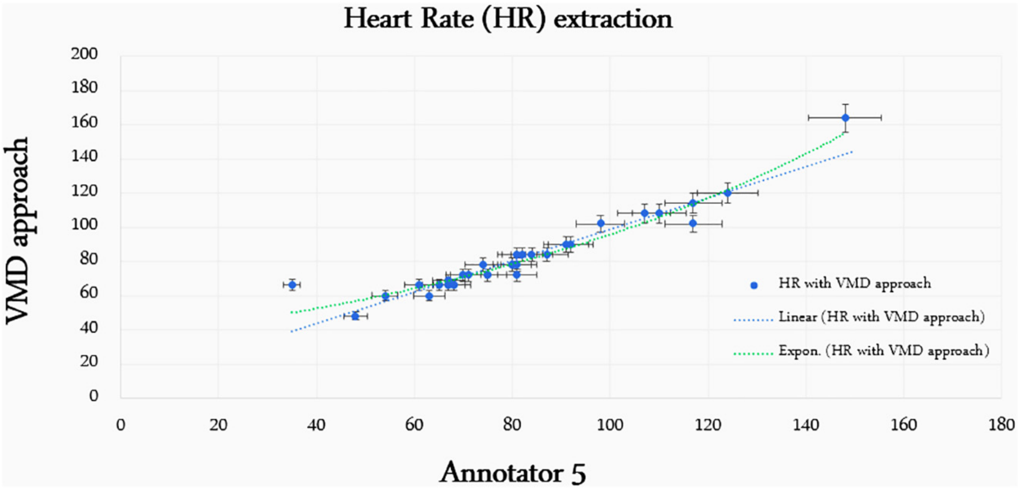

Fig. 11. Scatter plot between annotator 5 and VMD

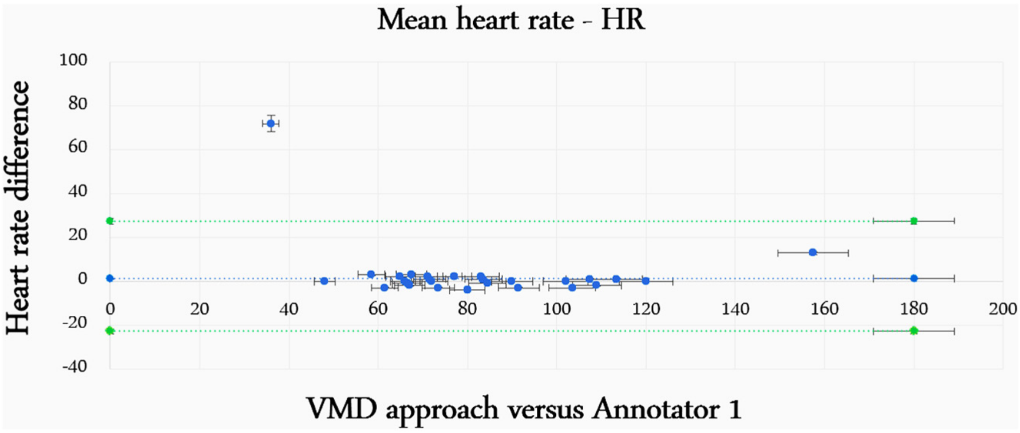

Fig. 12. Bland Altman between annotator 1 and VMD approach

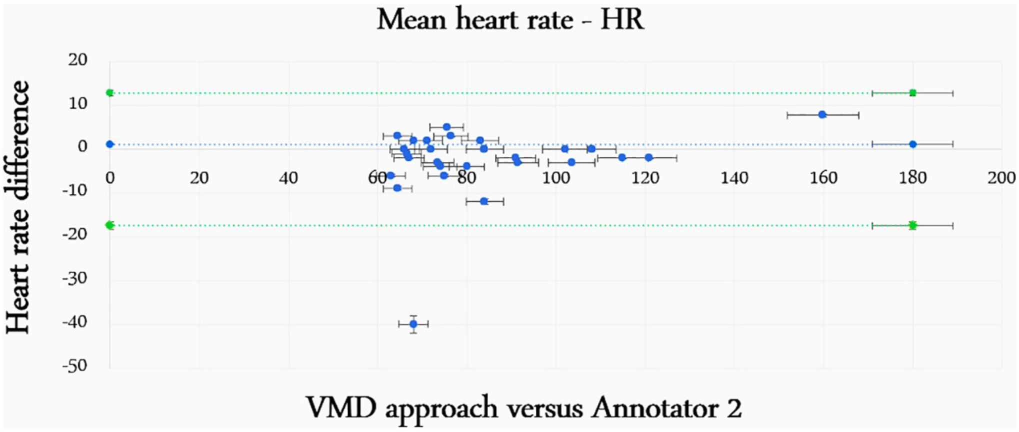

Fig. 13. Bland Altman between annotator 2 and VMD approach

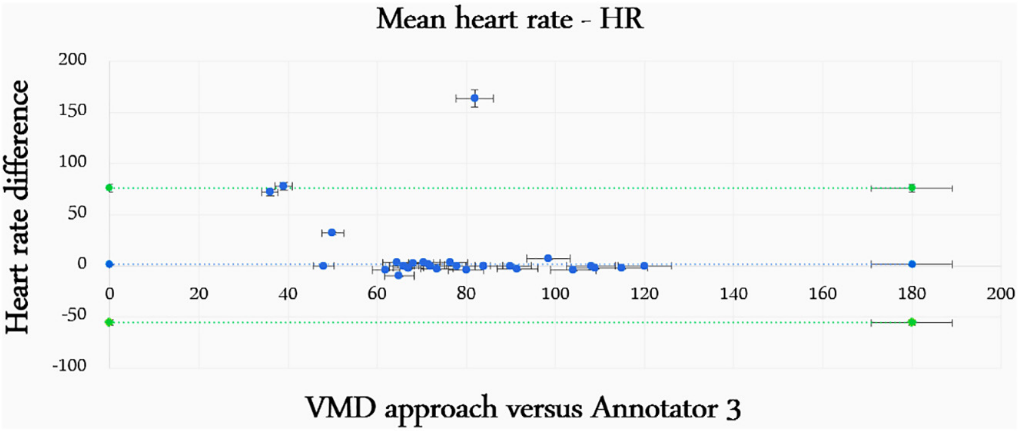

Fig. 14. Bland Altman between annotator 3 and VMD approach

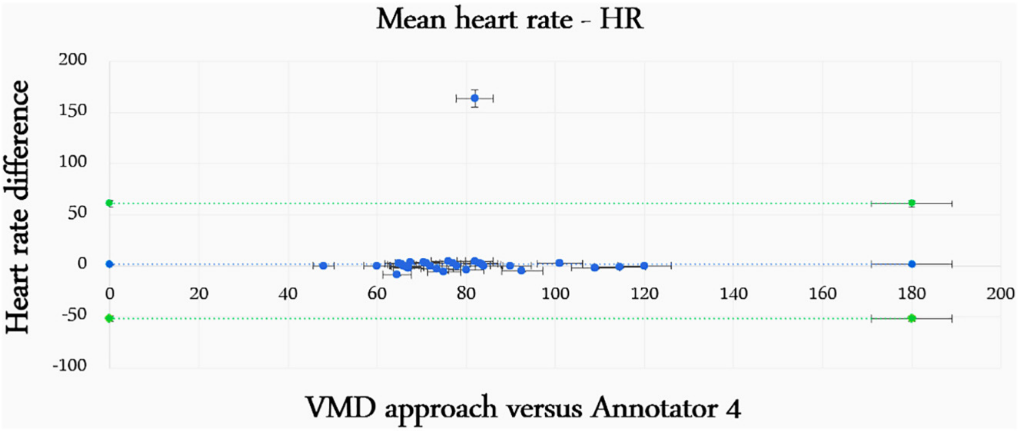

Fig. 15. Bland Altman between annotator 4 and VMD approach

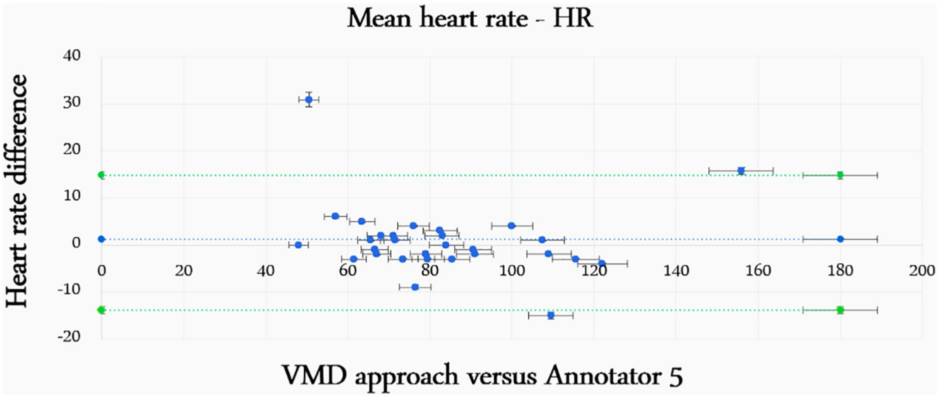

Fig. 16. Bland Altman between annotator 5 and VMD approach

Our experiments were conducted on an Intel(R) Core (TM) i5-6300U, 2.4 GHz, with 4 CPUs to assess the performance benefits of the enhanced VMD implementation using OpenMP. The experiments utilized Python for implementation, with the BUT PPG database providing the photoplethysmography (PPG) signals. These signals were used to extract heart rates, enabling a comparison between the naive and optimized VMD implementations. We evaluated the parallelized VMD implementation utilizing OpenMP by comparing the processing times of the optimized version (multi-threaded) and the naïve version (single-threaded). The reduction in processing time and the resultant computational speedup were the main performance indicators. Figures 17 and 18 illustrate the processing times at different PPG signal durations for both the optimized and naive VMD implementations:

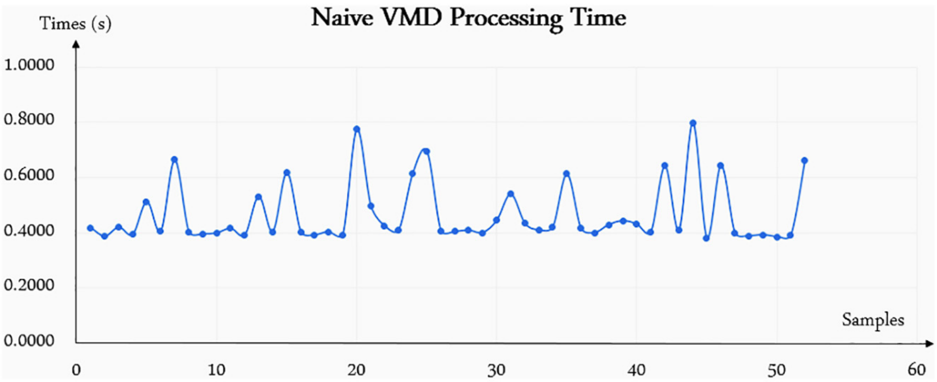

Fig. 17. Processing time using naïve VMD version

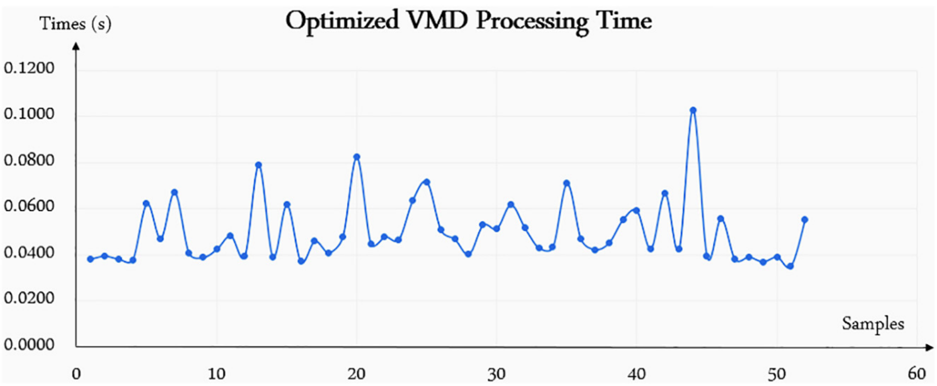

Fig. 18. Processing time using optimized VMD version

On the other side, the performance improvements were measured using efficiency and speedup calculations. The ratio of the naive implementation's execution time to the optimized implementation's execution time is known as speedup. The speed increase divided by the number of processors in use is efficiency. Figures 19 and 20 illustrate the speedup and efficiency between the naive and optimized versions.

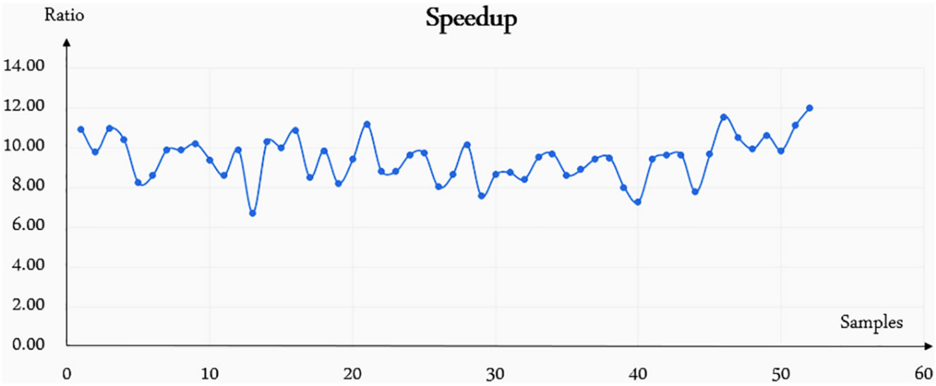

Fig. 19. Computational speedup

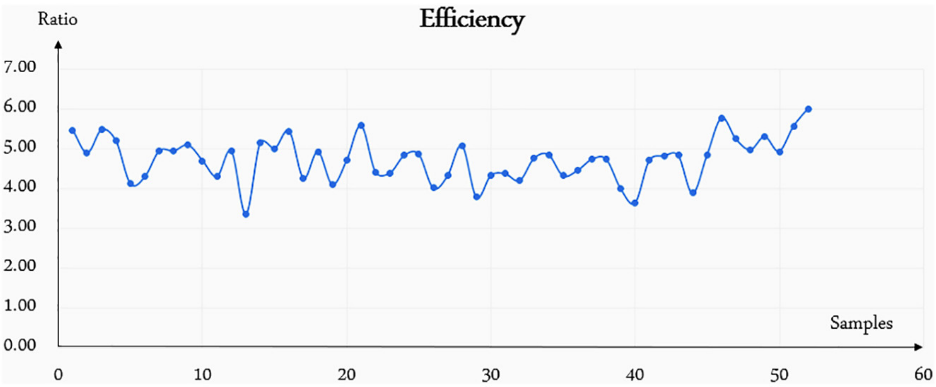

Fig. 20. Computational efficiency

The performance evaluation of the VMD algorithm, both in its naive and optimized forms, reveals significant improvements when using OpenMP for parallel processing. The naive VMD implementation has an average processing time of 0.466 seconds with a standard deviation of 0.109 seconds. In contrast, the optimized VMD implementation significantly reduces the average processing time to 0.050 seconds, with a standard deviation of 0.013 seconds. This substantial reduction in processing time results in an average speed up of 9.452 with a standard deviation of 1.097, demonstrating the efficiency of parallel processing. Furthermore, the efficiency of the optimized VMD implementation, measured as the speedup per processor, averages 4.727 with a standard deviation of 0.549. Table 2 presents our latter findings.

Table 2. Processing time using optimized VMD version

	Average	STD
Naive VMD Time (sec)	0.466	0.109
Optimized VMD Time (sec)	0.050	0.013
Speedup	9.452	1.097
Efficiency	4.727	0.549

The experimental results demonstrate that the parallelized VMD algorithm using OpenMP significantly improves the processing time for PPG signal decomposition in healthcare-embedded systems. The optimized implementation achieved a substantial reduction in processing time, making it suitable for real-time applications where quick response times are critical. The near-linear speedup with the number of cores utilized highlights the scalability of the OpenMP-based approach. This suggests that further performance gains can be achieved with higher core counts, making this optimization technique promising for future advancements in healthcare-embedded systems. In summary, the parallelized VMD algorithm using OpenMP offers a practical and effective solution for real-time signal processing in healthcare applications.

5 CONCLUSION

Our work based on the optimization of the VMD algorithm through parallel processing with OpenMP, significantly improves the real-time processing capabilities of healthcare-embedded systems. The optimized VMD approach produced a near-linear speedup by effectively utilizing multi-core processors, resulting in processing speeds that were × 9.45 times faster than the naive single-threaded version. This substantial improvement in computational efficiency ensures that complex signal processing tasks, such as decomposing PPG signals to extract vital physiological parameters such as heart and respiratory rates can be performed swiftly and accurately. Consequently, this optimization contributes to the development of more responsive and reliable medical devices for patient monitoring, ultimately enhancing the quality of healthcare delivery and patient outcomes through improved real-time analysis and timely interventions.

6 REFERENCES

[1] J. Park, H. S. Seok, S. S. Kim, and H. Shin, "Photoplethysmogram analysis and applications: An integrative review," *Frontiers in Physiology*, vol. 12, 2022. https://doi.org/10.3389/fphys.2021.808451

[2] Z. El Khadiri, R. Latif, and A. Saddik, "An efficient hybrid algorithm for non-contact physiological sign monitoring using plethysmography wave analysis," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 11, no. 6, pp. 2305–2321, 2023. https://doi.org/10.1080/21681163.2023.2227741

[3] H. W. Loh *et al.*, "Application of photoplethysmography signals for healthcare systems: An in-depth review," *Computer Methods and Programs in Biomedicine*, vol. 216, p. 106677, 2022. https://doi.org/10.1016/j.cmpb.2022.106677

[4] A. Pankaj *et al.*, "A review on computation methods used in photoplethysmography signal analysis for heart rate estimation," *Archives of Computational Methods in Engineering*, vol. 29, pp. 921–940, 2022. https://doi.org/10.1007/s11831-021-09597-4

[5] Q. K. Cabanilla *et al.*, "Technology adoption of computer-aided instruction in healthcare: A structured review," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, no. 23, pp. 160–181, 2023. https://doi.org/10.3991/ijet.v18i23.44083

[6] G. N. Georgieva-Tsaneva and I. Serbezova, "Research on the impact of innovative interactive technologies in the education of health care students," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, no. 20, pp. 283–291, 2022. https://doi.org/10.3991/ijet.v17i20.32903

[7] I. Mustapha, N. Khan, M. I. Qureshi, A. A. Harasis, and N. T. Van, "Impact of Industry 4.0 on healthcare: A Systematic Literature Review (SLR) from the last decade," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 15, no. 18, pp. 116–128, 2021. https://doi.org/10.3991/ijim.v15i18.25531

[8] A. Setyono, H. Haryanto, and E. Z. Astuti, "An adaptive web-based framework for mobile telemonitoring system," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 1, pp. 40–52, 2017. https://doi.org/10.3991/ijim.v11i1.6031

[9] B. G. Mohammed and D. S. Hasan, "Smart healthcare monitoring system using IoT," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 1, pp. 141–152, 2023. https://doi.org/10.3991/ijim.v17i01.34675

[10] R. Jeya, H. A. Karim, and S. B. Mansor, "Artificial intelligence and mobile apps support intelligent healthcare systems for mental health services," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 18, no. 20, pp. 157–168, 2024. https://doi.org/10.3991/ijim.v18i20.50743

[11] S. Jacques, A. Ouahabi, and T. Lequeu, "Remote knowledge acquisition and assessment during the COVID-19 pandemic," *International Journal of Engineering Pedagogy (iJEP)*, vol. 10, no. 6, pp. 120–138, 2020. https://doi.org/10.3991/ijep.v10i6.16205

[12] C. J. Macdonald, D. Archibald, H. Kellam, R. Sun, E. Stodel, and D. Puddester, "Evaluation of online health and wellness resources for healthcare professionals," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 4, no. 2, pp. 18–23, 2011. https://doi.org/10.3991/ijac.v4i2.1659

[13] D. Peterson, T. Clark, R. Sprod, T. Verrall, L. English, and A. Thomson, "Bloody good! The impact of elearning on medical and nursing practice," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 10, no. 2, pp. 75–87, 2017. https://doi.org/10.3991/ijac.v10i2.7349

[14] C. J. Macdonald, D. Archibald, H. Kellam, R. Sun, E. Stodel, and D. Puddester, "Evaluation of online health and wellness resources for healthcare professionals," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 4, no. 2, pp. 18–23, 2011. https://doi.org/10.3991/ijac.v4i2.1659

[15] S. King, L. Liu, E. Stroulia, and I. Nikolaidis, "Using simulations to integrate technology into health care aides' workflow," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 6, no. 2, pp. 28–31, 2013. https://doi.org/10.3991/ijac.v6i2.2982

[16] M. S. Miah, S. S. Bashar, A. Z. Karim, and Z. Hasan, "Heart rate extraction from photoplethysmography signal: A multi model machine learning approach," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2020, pp. 1–6. https://doi.org/10.1109/IICAIET49801.2020.9257869

[17] H. El Boussaki, R. Latif, and A. Saddik, "A review on video-based heart rate, respiratory rate and blood pressure estimation," in *Advances in Machine Intelligence and Computer Science Applications, ICMICSA '22*, in Lecture Notes in Networks and Systems, N. Aboutabit, M. Lazaar, and I. Hafidi, Eds., Springer, Cham, vol. 656, 2023, pp. 129–140. https://doi.org/10.1007/978-3-031-29313-9_12

[18] Z. El Khadiri, R. Latif, and A. Saddik, "Breathing pattern assessment through the empirical mode decomposition and the empirical wavelet transform algorithms," in *The 3rd International Conference on Artificial Intelligence and Computer Vision (AICV2023)*, in Lecture Notes on Data Engineering and Communications Technologies, A. E. Hassanien et al., Eds., Springer, Cham, vol. 164, 2023, pp. 262–271. https://doi.org/10.1007/978-3-031-27762-7_25

[19] R. Latif, B. Addaali, and A. Saddik, "Real-time SPO2 monitoring based on facial images sequences," in *Digital Technologies and Applications, ICDTA '23,* in Lecture Notes in Networks and Systems, S. Motahhir and B. Bossoufi, Eds., Springer, Cham, vol. 668, 2023, pp. 474–483. https://doi.org/10.1007/978-3-031-29857-8_48

[20] H. El Boussaki, R. Latif, A. Saddik, Z. El Khadiri, and H. El Boujaoui, "Non-contact respiratory rate monitoring based on the principal component analysis," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023. https://doi.org/10.14569/IJACSA.2023.01409106

[21] Z. El khadiri, R. Latif, and A. Saddik, "Remote heart rate measurement using plethysmographic wave analysis," in *Advances in Machine Intelligence and Computer Science Applications, ICMICSA 2022,* in Lecture Notes in Networks and Systems, N. Aboutabit, M. Lazaar, and I. Hafidi, Eds., Springer, Cham, vol. 656, 2022, pp. 254–267. https://doi.org/10.1007/978-3-031-29313-9_23

[22] Z. El khadiri, R. Latif, and A. Saddik, "Efficient remote health monitoring using deep learning and parallel systems," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 10, 2024. https://doi.org/10.14569/IJACSA.2024.0151020

[23] N. Salankar, D. Koundal, and S. Mian Qaisar, "Stress classification by multimodal physiological signals using variational mode decomposition and machine learning," *Journal of Healthcare Engineering*, vol. 2021, no. 1, pp. 1–12, 2021. https://doi.org/10.1155/2021/2146369

[24] C. Kaur, A. Bisht, P. Singh, and G. Joshi, "EEG signal denoising using hybrid approach of variational mode decomposition and wavelets for depression," *Biomedical Signal Processing and Control*, vol. 65, p. 102337, 2021. https://doi.org/10.1016/j.bspc.2020.102337

[25] S. Hadiyoso, E. M. Dewi, and I. Wijayanto, "Comparison of EMD, VMD and EEMD methods in respiration wave extraction based on PPG waves," *Journal of Physics: Conference Series*, vol. 1577, no. 1, p. 012040, 2020. https://doi.org/10.1088/1742-6596/1577/1/012040

[26] M. Das, T. Choudhary, M. K. Bhuyan, and L. N. Sharma, "Non-contact heart rate measurement from facial video data using a 2D-VMD scheme," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 11153–11161, 2022. https://doi.org/10.1109/JSEN.2022.3169963

[27] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no, 3, pp. 531–544, 2014. https://doi.org/10.1109/TSP.2013.2288675

[28] Y. Liu, G. Yanga, M. Li, and H. Yin, "Variational mode decomposition denoising combined the detrended fluctuation analysis," *Signal Process*, vol. 125, pp. 349–364, 2016. https://doi.org/10.1016/j.sigpro.2016.02.011

[29] F. Li, B. Zhang, S. Verma, and K. J. Marfurt, "Seismic signal denoising using thresholded variational mode decomposition," *Explor. Geophy.*, vol. 49, no. 4, pp. 450–461, 2018. https://doi.org/10.1071/EG17004

[30] R. B. Hoffmann *et al.*, "OpenMP as runtime for providing high-level stream parallelism on multi-cores," *The Journal of Supercomputing*, vol. 78, pp. 7655–7676, 2022. https://doi.org/10.1007/s11227-021-04182-9

[31] L. G. García, E. Montoya, S. Isaza, and R. A. Velasquez, "An open edX extension for parallel programming assignments with automatic configurable grading," *International Journal of Engineering Pedagogy (iJEP)*, vol. 11, no. 4, pp. 7–22, 2021. https://doi.org/10.3991/ijep.v11i4.20097

[32] S. Jacques, A. Ouahabi, and T. Lequeu, "Remote knowledge acquisition and assessment during the COVID-19 pandemic," *International Journal of Engineering Pedagogy (iJEP)*, vol. 10, no. 6, pp. 120–138, 2020. https://doi.org/10.3991/ijep.v10i6.16205

[33] A. Klein, J. Müller, N. Link, and R. Schaffer, "Assessment and level modelling in fundamentals of electrical engineering," *International Journal of Engineering Pedagogy (iJEP)*, vol. 14, no. 6, pp. 37–53, 2024. https://doi.org/10.3991/ijep.v14i6.47865

[34] M. Gusev, S. Ristov, G. Velkoski, and B. Ivanovska, "E-learning and benchmarking platform for parallel and distributed computing," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 9, no. 2, pp. 17–21, 2014. https://doi.org/10.3991/ijet.v9i2.3215

[35] M. Serik, N. Karelkhan, J. Kultan, and Z. Zulpykhar, "Setting up and implementation of the parallel computing cluster in higher education," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 6, pp. 4–17, 2019. https://doi.org/10.3991/ijet.v14i06.9736

[36] H. Yviquel *et al.*, "The OpenMP cluster programming model," in *Workshop Proceedings of the 51st International Conference on Parallel Processing*, 2023, pp. 1–11. https://doi.org/10.1145/3547276.3548444

[37] M. Rakhimov, J. Elov, U. Khamdamov, S. Aminov, and S. Javliev, "Parallel implementation of real-time object detection using OpenMP," in *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, 2021, pp. 1–4. https://doi.org/10.1109/ICISCT52966.2021.9670146

[38] M. Aldinucci *et al.*, "Practical parallelization of scientific applications with OpenMP, OpenACC and MPI," *Journal of Parallel and Distributed Computing*, vol. 157, pp. 13–29, 2021. https://doi.org/10.1016/j.jpdc.2021.05.017

[39] A. Nemcova, R. Smisek, E. Vargova, L. Marsanova, M. Vitek, and L. Smital, "Brno University of Technology Smartphone PPG Database (BUT PPG): Annotated database for PPG quality assessment and heart rate estimation," *BioMed Research International*, vol. 2021, no. 1, pp. 1–6, 2021. https://doi.org/10.1155/2021/3453007

[40] A. Nemcova, R. Smisek, E. Vargova, L. Marsanova, M. Vitek, and L. Smital. "Brno University of Technology Smartphone PPG Database (BUT PPG) (version 1.0.0)," *PhysioNet*, 2021. https://doi.org/10.13026/7vy8-av04

[41] A. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. https://doi.org/10.1161/01.CIR.101.23.e215

[42] J. M. Bland and D. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement," *The Lancet*, vol. 327, no. 8476, pp. 307–310, 1986. https://doi.org/10.1016/S0140-6736(86)90837-8

7 AUTHORS

Zakaria El Khadiri received his M.S. degree from the Ibn Zohr University, Morocco in 2020. Currently, he is a PhD student at the Laboratory of Systems Engineering and Information Technology (LISTI) laboratory ENSA, University Ibn Zohr, Agadir, Morocco. His current research focuses on computer science and embedded systems (E-mail: zakaria.elkhadiri@edu.uiz.ac.ma).

Rachid Latif received a PhD. degree in signal processing in 2000, a Habilitation degree in 2005 from the faculty of sciences, and a Full Professor in Information Technology in 2015, from Ibn Zohr University, Morocco. Currently, he is a Professor at the Department of Industrial & Electrical Engineering, National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco. His research interests include biomedical signal and image processing, biomedical instrumentation, robotics and embedded systems (CPU, DSP, FPGA, CPU-GPU, CPU-FPGA), and artificial intelligence and he is working on the modeling, filtering, and analysis of fetal cardiac signals and monitoring of premature babies. From 2006 to 2014, he was the head of the signals systems and computer science team (ESSI). He is the head of the LiSTi Laboratory. Besides, he is the head of the Moroccan Society of Embedded Systems (MSES) and the head of Embedded Systems and Biomedical Engineering Master (SEIB), and also, he is also an evaluator expert at CNRST Morocco.

Amine Saddik received his M.S. degree from the Ibn Zohr University, Morocco, in 2019, and his Ph.D. degree in Computer Science and Embedded Electronics Systems from the Ibn Zohr University in collaboration with the University of Paris-Saclay, France, in 2022. He is currently an Associate Professor at the Faculty of Applied Science, Ibn Zohr University, Morocco.