

# Energy-Efficient Spectral Analysis of ECGs on Resource Constrained IoT Devices

Charalampos Eleftheriadis , *Graduate Student Member, IEEE*, and  
Georgios Karakonstantis , *Senior Member, IEEE*

**Abstract**—Power spectral analysis (PSA) is one of the most popular and insightful methods, currently employed in several biomedical applications, aiming to identify and monitor various health related conditions. Among the most common applications of PSA is heart rate variability (HRV) analysis, which allows the extraction of further insights compared with conventional time-domain methods. Unfortunately, existing PSA approaches exhibit high computational complexity, hindering their execution on power-constrained embedded internet of things (IoT) devices. Such IoT devices are increasingly used for monitoring various conditions mainly by processing the input signals in the less complex time-domain. In this paper, a new low-complexity PSA system based on fast Gaussian gridding (FGG) is proposed, which can be used to calculate the Lomb-Scargle periodogram (LSP) of a non-uniformly spaced RR tachogram. The proposed approach is implemented on a popular ARM Cortex-M4 based embedded system, which is widely used in common wearables, and compared with conventional LSP-based approaches. Utilizing this experimental setup, a meticulous analysis is performed in terms of power, performance and quality under different operational settings, such as the total input/output samples, precision of computations, computer arithmetic (floating/fixed-point), and clock frequency. The experimental results show that the proposed FGG-based LSP approach, when specifically optimized for the targeted embedded device, outperforms existing approaches by up-to 92.99% and 91.70% in terms of energy consumption and total execution time respectively, with minimal accuracy loss.

**Index Terms**—Power spectral analysis (PSA), lomb-scargle periodogram (LSP), fast Gaussian gridding (FGG), embedded system evaluation.

## I. INTRODUCTION

IN recent years, the demand for smart decentralized health monitoring systems has steadily increased [1]. Numerous wearables now capture electrocardiogram (ECG) or photoplethysmography (PPG) signals, offering vital insights into heart-related disorders, which is a leading global cause of mortality [2], [3], [4]. A key physiological indicator extracted by

these wearables is heart rate variability (HRV), determined by analyzing the RR-tachogram from an ECG of a patient. This indicator can be evaluated using either time or frequency domain metrics and provides valuable information about the activities of the sympathetic nervous system (SNS) and parasympathetic nervous system (PNS) [4], [5]. Moreover, several conditions can be detected, which are associated with respiratory sinus arrhythmia, like stress and respiration rate [6], understanding dynamic changes in heart rate during anesthesia [7], revealing correlations between depressive symptoms and HRV in youth [8], and exploring age-related effects on long-term HRV [9].

Currently, there are several devices (e.g., Apple Watch, Fitbit Watch, Garmin Watch, Oura Ring, etc.) relying on lightweight battery-powered internet of things (IoT)/ microcontroller unit (MCU)-based systems at the edge. Such IoT devices collect and analyze ECG/PPG data in the time-domain, while leaving the processing of more complicated methods like the power spectral analysis (PSA) to more capable devices/servers, where collected data have to be regularly transmitted [10], a fact which imposes many limitations. In particular, current edge-cloud systems primarily conduct PSA using complex algorithms to gain further insights into HRV beyond those obtained from traditional time-domain metrics [4]. This approach hinders the real-time analysis and faces limitations related with high wireless transmission losses and power consumption, and security/privacy vulnerabilities. Therefore, there is a critical need for enabling the energy-efficient processing of complex PSA algorithms at the edge. This can enable the establishment of remote patient monitoring (RPM), aiming to alleviate health-care system strain and enhance patient treatment quality by minimizing unnecessary hospital appointments [3].

While insightful, the implementation of PSA for HRV has yet to achieve widespread adoption on miniature IoT MCUs due to the high complexity of the involved algorithms. Current PSA algorithms utilized for HRV assessment [11], [12], [13], [14], [15], [16], [17] necessitate a transformation from the time to the frequency domain, resulting in substantial computational overhead and prohibitively high power consumption. In general, there is a multitude of methods, typically involving a resampling/interpolation stage and power spectral density (PSD) estimation stage. Specifically, the most popular approaches involve the utilization of an interpolator along with a direct fast Fourier transform (FFT) [11] or an autoregressive (AR) [12], [13], [14] scheme. Meanwhile, the most recent approaches have adopted the Lomb-Scargle periodogram (LSP)

Manuscript received 26 February 2024; revised 12 April 2024 and 16 May 2024; accepted 20 May 2024. Date of publication 29 May 2024; date of current version 29 May 2025. This paper was recommended by Associate Editor P. Motto Ros. (Corresponding author: Georgios Karakonstantis.)

Charalampos Eleftheriadis is with the Queen's University Belfast, BT71NN Belfast, U.K.

Georgios Karakonstantis is with the University of Thessaly, 38221 Volos, Greece (e-mail: gkarakon@uth.gr).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TBCAS.2024.3406520>.

Digital Object Identifier 10.1109/TBCAS.2024.3406520

[16], [17] in order to avoid the use of interpolation and re-sampling needed in traditional PSA approaches that may alter the frequency content [18]. Although all these existing works have proposed various techniques to limit the computational complexity and consumed energy, they often involve trade-offs between accuracy and energy which may not be acceptable in biomedical applications. Furthermore, despite these efforts, a comprehensive evaluation of a PSA system on a modern MCU has not yet been performed in the literature.

This paper aims at introducing a low complexity LSP system and evaluate it on a commonly used edge device to reveal the potential benefits over the conventional PSA approaches. It introduces a new algorithm that significantly reduces the computational complexity of existing approaches and characterizes it on a real embedded device, thus opening avenues for real-time PSA and new health related monitoring applications. The contributions of this paper can be summarized as follows:

- 1) Develop a low-complexity PSA system that performs PSD estimation and HRV (low frequency (LF), high frequency (HF) etc.) feature extraction based on the LSP. The proposed approach is based on a novel fast Gaussian gridding (FGG) algorithm that constructs more accurate grids than the conventional Lagrange-based fast Lomb-Scargle periodogram (FLSP) resulting in smaller over-sampled FFTs. Furthermore, a novel quantization method is introduced that greatly reduces the number of required multiplications at the gridding stage.
- 2) Optimize and implement the existing and the proposed LSP algorithms, on an ARM-Cortex M4 MCU using the STM32F401RE, which is widely used in several IoT/embedded system applications [19], [20], [21]. Additionally, the common microcontroller software interface standard (CMSIS) ARM library [22] is utilized to apply several optimized functions and math operations for the tested PSA systems, revealing the capabilities of modern state of the art MCUs.
- 3) Develop an experimental setup to enable the on-board characterization in real-time of the developed PSA system, in terms of the current consumption of either the whole PSA system or a part of it (gridding stage, FFT, communication cost), while executing real patient datasets [23]. This setup enables the evaluation of the proposed FGG-based and the existing Lagrange-based LSP approaches for different number of samples (64, 128, 256 and 512), precision levels (low, mid and high) and computer arithmetic schemes (floating/fixed-point). Concerning the MCU architecture level configurations, several of them have been explored regarding the clock frequency (21 MHz, 42 MHz and 84 MHz), power modes (RUN and SLEEP mode) and the utilization or not of the floating-point unit (FPU).
- 4) Perform a rigorous analysis for all the test scenarios and metrics including: the execution time, number of clock cycles, energy consumption and current dissipation for all the PSD estimation stages of the proposed fast Gaussian gridding-Lomb-Scargle periodogram (FGG-LSP) and the conventional Lagrange-based FLSP PSA systems. The experimental results demonstrate that the achieved

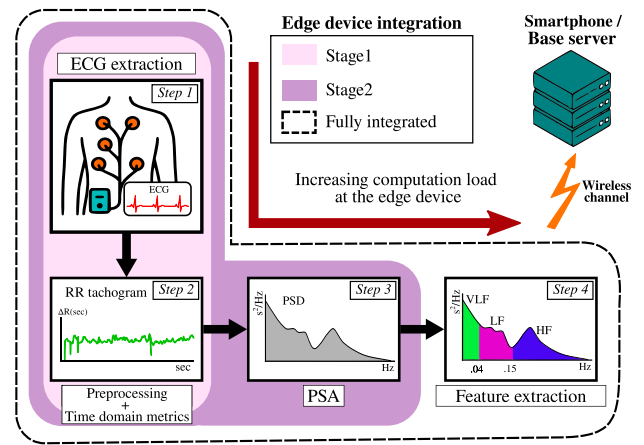


Fig. 1. PSA steps of HRV evaluation for several edge device integration stages.

savings of the proposed approach vary from 34.10% to 92.99% in terms of time and 33.39% to 91.70% in terms of energy consumption for a clock frequency of 84 MHz, utilizing the right amount of energy for a given quality level.

The rest of the paper is organized as follows. Section II discusses the background and challenges existing PSA systems. Section III presents the proposed FGG-LSP approach. Section IV introduces the experimental setup which is used in the work. Section V elaborates on the experimental results and the findings of this work. Section VI draws the final conclusions.

## II. BACKGROUND

This section presents some background and works relevant to PSA HRV evaluation systems and their implementation on portable devices along with some existing challenges that remain unaddressed. Initially, for completeness the existing time domain approaches are presented and then frequency-domain HRV is introduced, which have received less attention in past works and is the main focus of the paper.

### A. HRV System Overview

1) *Time-Domain*: A typical HRV processing system involves several steps, as depicted in Fig. 1 which are performed either on the edge or on cloud/server devices. Existing approaches involve time domain methods which process ECG signals that are recorded by attaching electrodes (single-channel or multi-channel) on a patient to capture the electrical signals generated by their heart (Step 1). Similarly, other wearable smartwatch devices (e.g. Apple Watch) use PPG signals to monitor the blood flow using LED sensors [24]. After acquiring the ECG/PPG signals, some preprocessing is conducted, including filtering, QRS detection and artifact rejection to extract RR (or NN) intervals/tachogram (Step 2). The resulting tachogram is a plot of the time intervals between successive R waves of the QRS wave, offering insights into the beat-to-beat variations in a patient's heart rate. The current wearable devices, predominantly perform lightweight HRV evaluation in time domain

using metrics like the SDNN and RMSDD, avoiding the more insightful and energy demanding frequency domain methods.

In several works the preprocessing steps in the time domain have been widely investigated with the aim of reducing the total energy dissipation. Specifically, in [25] the utilization of multicore systems has been examined for 12 lead ECGs on filtering, signal enhancement, feature extraction and inference in the time domain. Furthermore, in [26] an energy-efficient preprocessing method for RR peak detection has been proposed using an unsupervised learning technique called BayeSlope. Also in [27] a lightweight embedded algorithm for the detection of the QRS complex based on the shape, trend and slope of the ECG signal at a given instance. Finally, in [28], a Raspberry Pi has been utilized for the feature extraction of ECG signals using time domain techniques.

2) *Frequency-Domain*: Regarding the PSA step, which is the focus of this work, it consists of two main stages: interpolation/resampling and PSD estimation. The initial stage is crucial because the RR tachogram is known at non-uniformly spaced time intervals. There are several PSA algorithms to estimate the PSD of the RR tachogram, such as those based on a direct FFT [11] and on AR models [12], [13], [14]. However, such processes introduce approximations in the computed spectrum, prompting the exploration of alternative methods for direct PSD estimation. Recently, the LSP [15], [16], [17] has emerged as a promising approach for directly computing the PSD via least-squares fitting in the context of HRV. This paper specifically focuses on the latter method, hence a detailed explanation of its underlying principles is discussed in Section II-B.

Therefore, proceeding the PSD estimation algorithms, the HRV features of the estimated PSD are computed (*Step 4*), usually consisting of the total power (TP) (0.05 Hz to 0.4 Hz), the LF (0.05 Hz to 0.15 Hz), and HF (0.15 Hz to 0.4 Hz) frequency bands, as well as the LF/HF ratio. These HRV metrics, crucial for the analysis of specific cardiac anomalies (e.g., hypertension, congestive heart failure) [4], are instrumental in the diagnostic process.

3) *Data-Transmission*: Depending on the processing capabilities of the edge device, the wireless transmission of data occurs at different steps (Fig. 1). For the integration *Stage 1* only the preprocessing and time domain metrics are extracted at the edge, which is the case with most currently available popular wearable devices [10]. When the whole or part of the PSD estimation is also performed on the edge device, integration *Stage 1* is achieved as proposed in [12]. Finally, full integration [13] is achieved when only the essential HRV metrics (LF, HF, etc.) are transmitted at the smartphone or base server, thus significantly reducing communication energy costs. Note, when the latter approach is adopted the edge device bears almost all the computational load hence it is crucial to optimize the executed PSA algorithms which are the most energy demanding part of the whole system.

## B. Conventional Lomb-Scargle Periodogram

PSA in the frequency domain is more complex than it may seem. Initially, the RR tachogram, derived from the ECG

signal (depicted as *Step 2* in Fig. 1), constitutes an irregularly spaced time series. To evaluate the PSD of an  $N$  sized irregularly spaced RR tachogram, the LSP can be employed as shown in (1):

$$\text{PSD}(\omega) = \frac{1}{2} \left( \frac{\left[ \sum_{j=0}^{N-1} x_j \cos \omega(t_j - \tau) \right]^2}{\sum_{j=0}^{N-1} \cos^2 \omega(t_j - \tau)} + \frac{\left[ \sum_{j=0}^{N-1} x_j \sin \omega(t_j - \tau) \right]^2}{\sum_{j=0}^{N-1} \sin^2 \omega(t_j - \tau)} \right) \quad (1)$$

$\omega$ : angular frequency,  $t_j$ : observation time,  
 $x_j$ : Normalized RR interval,  $\tan(2\omega\tau) = \frac{\sum_{j=0}^{N-1} \sin(2\omega t_j)}{\sum_{j=0}^{N-1} \cos(2\omega t_j)}$

Also if  $C_0$ ,  $S_0$ ,  $C_1$ ,  $S_1$  are defined as:

$$\begin{aligned} S_0(\omega) &= \sum_{j=0}^{N-1} x_j \sin(\omega t_j), & C_0(\omega) &= \sum_{j=0}^{N-1} x_j \cos(\omega t_j) \\ S_1(\omega) &= \sum_{j=0}^{N-1} \sin(2\omega t_j), & C_1(\omega) &= \sum_{j=0}^{N-1} \cos(2\omega t_j) \end{aligned} \quad (2)$$

By using some trigonometric identities, the four summations of (1) can be computed as follows:

$$\sum_{j=0}^{N-1} x_j \cos \omega(t_j - \tau) = C_0(\omega) \cos(\omega\tau) + S_0(\omega) \sin(\omega\tau) \quad (3)$$

$$\sum_{j=0}^{N-1} x_j \sin \omega(t_j - \tau) = S_0(\omega) \cos(\omega\tau) - C_0(\omega) \sin(\omega\tau) \quad (4)$$

$$\begin{aligned} \sum_{j=0}^{N-1} \cos^2 \omega(t_j - \tau) &= \frac{N}{2} + \frac{1}{2} C_1(\omega) \cos(2\omega\tau) \\ &\quad + \frac{1}{2} S_1(\omega) \sin(2\omega\tau) \end{aligned} \quad (5)$$

$$\begin{aligned} \sum_{j=0}^{N-1} \sin^2 \omega(t_j - \tau) &= \frac{N}{2} - \frac{1}{2} C_1(\omega) \cos(2\omega\tau) \\ &\quad - \frac{1}{2} S_1(\omega) \sin(2\omega\tau) \end{aligned} \quad (6)$$

The computation of the PSD for  $N$  output frequencies  $\omega$  using the (1) requires  $O(N)$  operations are per frequency, thus the total complexity is  $O(N^2)$ . To reduce this quadratic computational complexity the FLSP method has been proposed [15], which efficiently computed the summations of (2) using Lagrange polynomials.

## C. Lagrange-Based Fast Lomb-Scargle Periodogram

The computational complexity of the LSP can be significantly reduced by utilizing Lagrange polynomials. In particular, (2) can be combined as complex summations:

$$X_0(\omega) = C_0(\omega) - iS_0(\omega) = \sum_{j=0}^{N-1} x_j e^{-i\omega t_j} \quad (7)$$

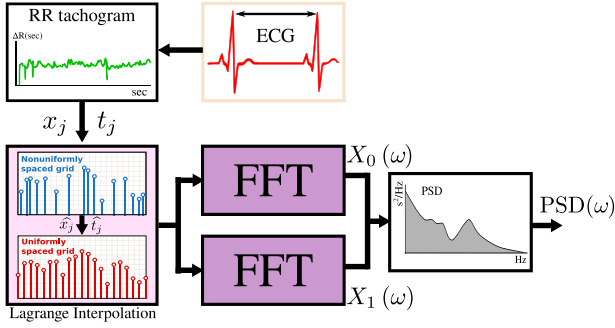


Fig. 2. The block diagram of the conventional FLSP; The ECG delineation is followed by a Lagrange based resampling algorithm succeeded by the  $X_0$  and  $X_1$  computation via two FFTs. These FFTs do not compute the PSD, as their only purpose is to lower the complexity the Lomb periodogram.

Note that the exponent terms in (7) are non-uniformly spaced due-to  $t_j$ . Thus they can be resampled in equally spaced points as a summation of  $M$  total factors:

$$g(t_j) = e^{-i\omega t_j} \approx \sum_{k=0}^{M-1} w_k(t_j) e^{-i\omega \hat{t}_k} \quad (8)$$

$w_k(t_j)$ :  $k^{th}$  order Lagrange polynomial  $\hat{t}_k$ : equally spaced  
Therefore, by integrating (8) in (7):

$$\begin{aligned} X_0(\omega) &= \sum_{j=0}^{N-1} x_j g(t_j) \approx \sum_{j=0}^{N-1} x_j \sum_{k=0}^{M-1} w_k(t_j) e^{-i\omega \hat{t}_k} \\ &= \sum_{j=0}^{N-1} \sum_{k=0}^{M-1} x_j w_k(t_j) e^{-i\omega \hat{t}_k} = \sum_{k=0}^{M-1} \hat{x}_k e^{-i\omega \hat{t}_k} \end{aligned}$$

$$\text{where: } \hat{x}_k = \sum_{j=0}^{M-1} x_j w_k(t_j) \quad (9)$$

It is evident that if  $\hat{x}_k$  is known, then (9) can be efficiently computed using the FFT. Additionally, to adequately approximate  $\hat{x}_k$ , an oversampling factor  $R_L$  has to be taken into account, which is also multiplied by a factor of 4 in order to avoid aliasing [15]. Consequently,  $M = 4R_L$ , where  $R_L$  takes on values of 2, 4, or 8, depending on the desired accuracy level, with a higher  $R_L$  corresponding to higher accuracy. A similar approximation for the summations  $C_1(\omega)$ ,  $S_1(\omega)$  can be derived, leading to an equation like (9). Consequently, the block diagram of the FLSP algorithms is depicted in Fig. 2 where there are two FFT blocks that are the most computationally expensive parts with computational complexities of  $O(4R_L N \log_2 4R_L N)$ .

#### D. Relevant Works

Both the LSP and the enhanced FLSP, discussed in the preceding sections, face significant computational challenges. The LSP comes with a quadratic computational complexity, while the FLSP, which has log-linear complexity, still demands a high oversampling factor  $R_L$  to achieve desirable accuracy levels.

This poses a notable challenge for portable, power-constrained devices, necessitating optimizations to balance computational efficiency and power consumption.

In pursuit of this balance, [16] demonstrated a method to reduce the FFT size by leveraging signal sparsity. This is achieved through a discrete wavelet transform (DWT) transform prior to the FFTs in FLSP, enabling truncation and approximation of operations. Their approach, tested on an RISC processor simulator, showcased energy-accuracy trade-offs with savings reaching up to 82% by significantly sacrificing signal quality. Another important contribution is from [29], where the authors developed an HRV system on chip (SoC) processor. This processor, based on the direct LSP approach, incorporated peripheral controller units and introduced a fixed-point LSP to further reduce computational demands but the main focus is on the design-level of the system rather than the optimization of the involved algorithms, the power/performance characterization and the achieved trade offs. Additionally, [30] employed the FFT-based Welch method to calculate the LF/HF ratio on an STM32F407ZET6 development board with ARM Cortex M4 MCU, alongside other time-domain methods. While, this work allows to extract various HRV characteristics the execution time of the proposed frequency-domain methods is high (4176 ms) as an interpolation/FFT approach is adopted, rather than the more accurate and lightweight LSP approach. Furthermore, [12], [13] proposed custom-designed hardware architectures for the fast Burg and Yule-Walker AR methods. [31] utilized an OLIMEX PIC33-PINGÜINO-OTG module for AR spectral analysis, which not only requires interpolation/resampling but also has a high computational complexity due to the required autoregressive coefficient estimation.

Beyond portable devices and the HRV evaluation, the LSP algorithm has found applications in various domains, prompting several optimization efforts. Notably, [32] introduced a software implementation that integrates the NFFT [33] library for expedited FLSP computations, particularly in astronomy. This approach focuses on accelerating trigonometric sum calculations inherent in the FLSP. Also, [34] harnessed the capabilities of modern GPUs for parallel frequency computations, effectively addressing the algorithm's intensive computational requirements.

#### E. Challenges

While the existing PSA estimation approaches for HRV evaluation in the frequency domain have tried to reduce the complexity and energy consumption for portable devices, there are significant challenges that remain unaddressed as listed below:

- In terms of the employed algorithms, many studies rely on direct interpolation of the RR tachogram followed by an FFT or AR methods for PSD estimation [8], [11], [12], [13], [30]. However, this underscores the need for algorithmic reformulation for the conventional Lagrange-based FLSP method [15], which requires a high oversampling factor  $R_L$  for maintaining adequate accuracy. This results in FFTs with prohibitively high sizes for IoT/embedded MCUs.



- The scope of approaches on embedded devices remains constrained, with a primary focus on achieving energy savings through custom chip design or FPGA-based solutions [12], [13], [29]. Furthermore, the existing embedded systems solutions do not use a proper experimental setup that allows for the accurate measurement of the energy consumption of various stages of the executed PSA algorithms [30], [31], focusing only on the feasibility of implementation and not the power optimization. This indicates a narrower exploration within the realm of modern IoT devices which offer greater flexibility and easier integration with existing technologies, especially concerning the latest ARM-Cortex M4 MCU that offers enhanced processing capabilities (e.g., FPU, CMSIS library) with low power consumption.
- There exists a critical need for in-depth exploration of various algorithmic (such as number of samples, precision level, and arithmetic) and architectural (clock frequency, power modes, and hardware utilization) parameters for contemporary IoT MCUs. Understanding their impact on energy consumption, execution time, and overall quality is essential, particularly in the context of real-time applications for different parts of the PSA system. It is also important to match the inherent accuracy of the PSA algorithm with the proper arithmetic scheme (floating, fixed-point), thus utilizing the right amount of energy to achieve a certain quality level.

### III. PROPOSED APPROACH

In this section, the proposed FGG method is presented which drastically reduces the computational complexity of the conventional Lagrange-based FLSP algorithm.

#### A. Gaussian Gridding

As mentioned in Section II the values  $x_j$  of the RR tachogram are known at unequally spaced time instances  $t_j$  the RR tachogram can be defined in the continuous domain for  $N$  consecutive samples as:

$$x(t) = \sum_{j=0}^{N-1} x_j \delta(t - t_j), \text{ where } \delta(t) : \text{Dirac function} \quad (10)$$

After examining (7) and (10) it is evident that  $X_0(\omega)$  is the continuous time Fourier transform (CTFT) of  $x(t)$  defined as:

$$X_0(\omega) = \int_{-\infty}^{\infty} \sum_{j=0}^{N-1} x_j \delta(t - t_j) e^{-i\omega t} \quad (11)$$

Therefore a Gaussian convolution in the continuous domain will be employed in order to efficiently compute  $X_0(\omega)$ .

Initially, the  $2\pi$  periodic continuous Gaussian kernel  $g_\beta$  is defined:

$$g(t) = \sum_{l=-\infty}^{\infty} e^{-\frac{(t-2\pi l)^2}{4\beta}} \quad (12)$$

The continuous convolution of  $x(t)$  and  $g(t)$  yield:

$$\begin{aligned} x_\beta(t) &= (x * g)(t) = \int_{-\infty}^{\infty} x(u)g(t-u)du \\ &= \int_{-\infty}^{\infty} \sum_{j=0}^{N-1} x_j \delta(u - t_j) \sum_{l=-\infty}^{\infty} e^{-\frac{(t-u-2\pi l)^2}{4\beta}} du \\ &= \sum_{j=0}^{N-1} x_j g_\beta(t - t_j) = \sum_{j=0}^{N-1} x_j \sum_{l=-\infty}^{\infty} e^{-\frac{(t-t_j-2\pi l)^2}{4\beta}} \quad (13) \end{aligned}$$

In the continuous frequency domain  $X_\beta(\omega) = X_0(\omega)G(\omega)$  where  $G(\omega) = \sqrt{\frac{\pi}{4\beta}} e^{-\omega^2 \beta}$  [35]. Thus the desired  $X_0(\omega)$  is computed by  $X_\beta(\omega)$  from (13) and dividing with  $G(\omega)$  (deconvolution operation).

Note that  $x_\beta(t)$  and the desired  $X_0(\omega)$  are at the continuous domain. Thus if  $N$  total output frequencies  $\omega_k = \frac{2\pi k}{N} f_{max} \mid k \in \{0, N-1\}$ ,  $f_{max} = 0.4$  Hz are needed, then according to the Nyquist theorem  $x_\beta(t)$  has to be sampled at least twice the maximum frequency to avoid aliasing. Therefore, for equally spaced time-intervals  $t_n = \frac{2\pi n}{R_G N f_{max}} \mid n \in \{0, R_G N - 1\} \cap \mathbb{N}$ :

$$x_\beta(t_n) = x_\beta[n] = \sum_{j=0}^{N-1} x_j \sum_{l=-\infty}^{\infty} e^{-\frac{(t_n-t_j-2\pi l)^2}{4\beta}}, \quad (14)$$

where  $R_G = 4$  is a constant oversampling factor. Also  $\beta = \frac{\pi M_{sp}}{3R_G^2 N^2}$ . So to reduce the computational complexity, the discrete signal  $x_\beta[n]$  can be approximated by computing the  $M_{sp}$  closer Gaussian pulses as:

$$\tilde{x}_\beta[n] \approx \sum_{j=0}^{N-1} x_j \sum_{l=-M_{sp}}^{M_{sp}-1} e^{-\frac{(t_n-t_j-2\pi l/R_G N)^2}{4\beta}}, \quad (15)$$

Finally, the desired output in the discrete frequency domain is computed as:

$$\tilde{X}[k] = \mathcal{F}_{R_G N} \{ \tilde{x}_\beta[n] \} \sqrt{\frac{\pi}{\beta}} \frac{e^{k^2 \beta}}{R_G N}, \quad (16)$$

where  $\mathcal{F}_{R_G N}$  is the scaled  $R_G N$ -sized FFT followed by the deconvolution operation (multiplication by  $\sqrt{\frac{\pi}{\beta}} e^{-k^2 \beta}$ ). Consequently, the total computational complexity is  $O(M_{sp} N)$  for the approximate convolution (15) and  $O(R_G N \log_2(R_G N))$  for the FFT.

#### B. Gaussian Smearing

The approximate convolution method, detailed in (15), computes  $\tilde{x}_\beta[n]$  for each sample  $n$  individually, leading to multiple evaluations for each  $x_j$ . To improve the efficiency, the Gaussian smearing technique is introduced, outlined in Alg. 1 and illustrated in Fig. 3, allowing to process each  $x_j$  separately. This technique is particularly effective for non-uniformly spaced points in the RR tachogram, as shown by the three red points in Fig. 3. These points influence neighboring grid points during convolution with a Gaussian pulse, with central points identified as per line 7 in Alg. 1. The algorithm then identifies  $M_{sp}$  left and right offset points (line. 12 in Alg. 1), less influenced by the RR tachogram points, where a larger  $M_{sp}$  enhances

**Algorithm 1** Approximate Gaussian Smearing

---

```

1:  $\tilde{x}_b \leftarrow \text{zeros}(R_G N, 1)$  ▷ Output vector
2: for  $l = 0$  to  $M_{sp}$  do
3:    $E_3[l] \leftarrow e^{\pi^2 l^2 / (R_G^2 N^2 \beta)}$  ▷ Precompute
4: end for
5: for  $j = 0$  to  $N - 1$  do
6:    $n_j \leftarrow R_G N f_{max} \cdot t_j$ 
7:    $n_{center} \leftarrow \lceil n_j \rceil$  ▷ Find center point
8:    $\Delta t \leftarrow \frac{2\pi}{R_G N} (n_j - n_{center})$  ▷ Time difference
9:    $E_1 \leftarrow e^{-\Delta t^2 / (4\beta)}$ 
10:   $E_2 \leftarrow e^{\pi \Delta t / (R_G N \beta)}$ 
11:  for  $l = -M_{sp}$  to  $M_{sp} - 1$  do ▷ Iter. Gaussian points
12:     $n_l \leftarrow n_{center} + l$  ▷ Offset center point
13:     $n \leftarrow n_l \bmod R_G N$  ▷ Periodicity
14:     $\tilde{x}_b[n] \leftarrow \tilde{x}_b[n] + x_j E_1 E_2^l E_3[|l|]$  ▷ Update  $\tilde{x}_b[n]$ 
15:  end for
16: end for

```

---

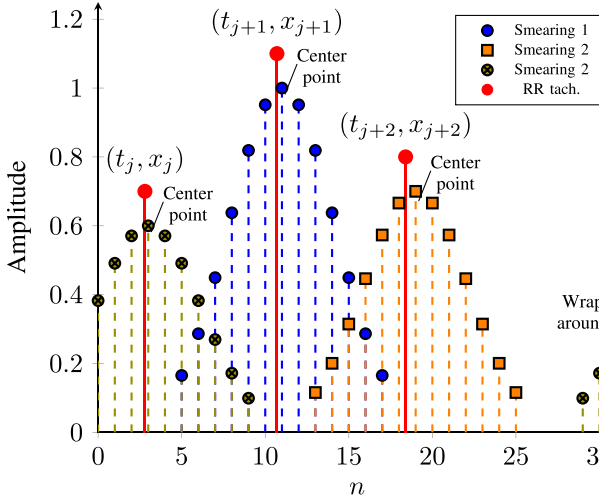


Fig. 3. Gaussian smearing of 3 non-uniformly spaced points  $(t_j, x_j)$  for  $M_{sp} = 6$ .

accuracy. Additionally, a modulo operation is applied in line 13 of Alg. 1 for the offset points in order to support wrap-around. The contributions from each RR point are aggregated into the uniformly spaced oversampled grid  $\tilde{x}_b[n]$  as shown in line 14. Notably, the exponential coefficient from (15) is divided into  $E_1$ ,  $E_2$ , and  $E_3$ , with the latter being precomputed as it does not depend on  $\Delta t$ , as explained in line 3 of Alg. 1.

**Quantized Step:** The time difference  $\Delta t$  (line 8) can take predetermined values as  $\Delta t \in (-2\pi/R_G N, 0]$ . Thus, the multiplications of the exponentials in line 14 can be avoided for a given index  $l$ , if the difference  $\Delta t$  is adequately approximated. To achieve this, a look-up table can be constructed which samples  $\Delta t$  in  $M$  total equally spaced intervals  $\tilde{\Delta t}_m = \frac{2\pi}{R_G N} (\frac{m}{M} - 1)$ , where  $m \in \{1, M\}$ , and computes the product  $LUT[m][l] \approx E_1 E_2^l E_3[|l|]$  for  $l \in \{-M_{sp}, M_{sp} - 1\}$ . An example of this optimization is illustrated in Fig. 4. The distance between the two center points  $\tilde{x}_b[n]$  and  $\tilde{x}_b[n+1]$  of the output grid is

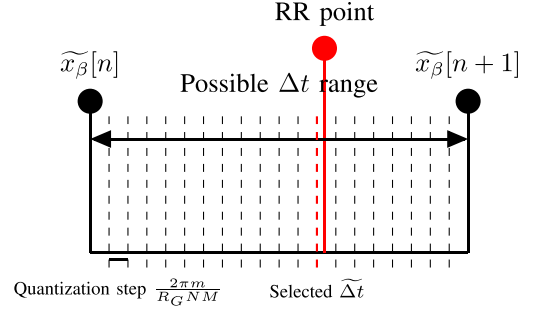


Fig. 4. Gaussian smearing of 3 non-uniformly spaced points  $(t_j, x_j)$  for  $M_{sp} = 6$ .

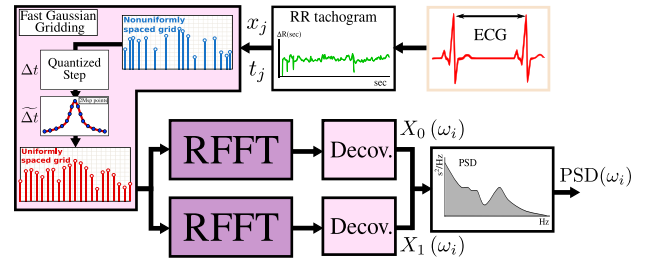


Fig. 5. Proposed FGG-based LSP PSA system.

quantized with a  $\frac{2\pi m}{R_G N M}$  step. The red RR point falls between the 12<sup>th</sup> and 13<sup>th</sup> quantization level ( $M = 20$ ) and thus the selected  $\tilde{\Delta t}$  is  $\frac{2\pi}{R_G N} (\frac{12}{20} - 1)$ . The maximum quantization error is computed as  $\epsilon_q = |\Delta t - \tilde{\Delta t}| \leq \frac{\pi}{R_G N M}$ , thus the final relative error is computed as:

$$\epsilon_R = \left| \frac{E - \tilde{E}}{E} \right| \approx \left| 1 - e^{-\frac{4\alpha^2 M_{sp} + 2\alpha}{4\beta M}} \right|, \quad (17)$$

where  $\tilde{E} = LUT[m][l]$ ,  $E = E_1 E_2^l E_3[|l|]$ ,  $\alpha = \frac{\pi}{R_G N}$

Therefore, given the parameters  $\alpha$  and  $M_{sp}$ , the total number of approximation points, denoted as  $M$ , is selected to be sufficiently large (in the range of approximately 500 to 1000). This choice ensures that the error introduced in the Gaussian approximation process (15) exceeds the quantization error. Thus, the use of the look-up table does not lead to any loss of accuracy in the computation. This quantization method incurs a modest memory overhead ranging from 1 to 4 MB, well within the capabilities of the tested MCU, eliminating the need for  $6M_{sp}$  multiplications (line 14). This leads to a more than 3x reduction in the gridding stage's time and energy consumption when employing the proposed quantization method on the FGG-LSP instead of directly computing the multiplications.

### C. Fast Gaussian Gridding-Based Lomb-Scargle Periodogram

In Fig. 5, the final proposed FGG-based LSP PSA system is illustrated. The process starts with the extraction of the RR tachogram from the ECG. This is followed by the application of the FGG step, which handles the non-uniformly spaced RR points  $(x_j, t_j)$ . Next, the  $\Delta t$  quantization process is executed

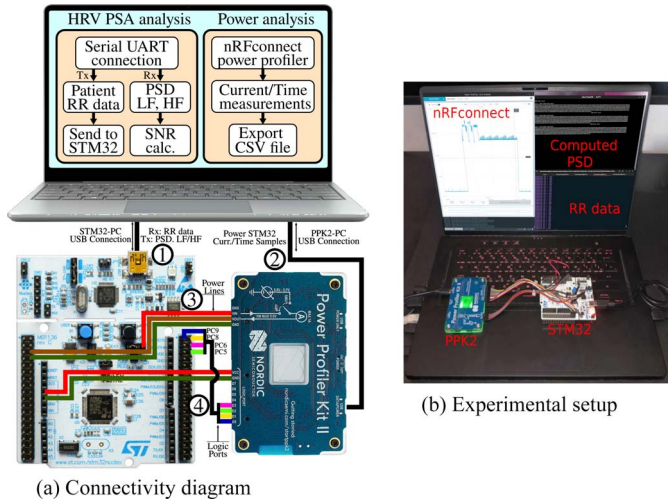


Fig. 6. Overview of the experimental setup: STM32 MCU, PPK2 and PC. ① Data Tx/Rx with PC. ② External power connection. ③ Monitored power lines. ④ Signaling logic ports.

for each input sample using the precomputed values from the look-up table. Gaussian smearing is then performed to establish an equally spaced grid. This procedure is replicated for all components in the summation of (2), requiring the creation of two equally spaced grids. Subsequently, real-valued fast Fourier transforms (RFFTs) are carried out, as the produced grids fall within the real number domain ( $\mathbb{R}$ ). This optimization notably reduces the computational workload by nearly half. For these operations, the efficient RFFTs algorithms from the CMSIS ARM library were utilized [22]. The process concludes with two deconvolution multiplications, as detailed in (16), to compute the final LSP summations  $X_0(\omega)$  and  $X_1(\omega)$ .

#### IV. EXPERIMENTAL SETUP

This section describes the experimental setup that is used for the implementation of the examined PSA systems and their characterization in terms of quality of results and energy consumption. Initially, an overview of the whole setup is discussed prior to the detailed presentation of each of the individual components; MCU, firmware and power profiler.

##### A. System-Level Overview

In Fig. 6 the connectivity of the STM32F401RE IoT board and the PPK2 with the PC is illustrated. Both components are connected via USB with the PC to facilitate data transmission over the serial universal asynchronous receiver-transmitter (UART), where preprocessed RR intervals are provided to the board to allow focusing on the evaluation of the LSP-based PSA systems. Specifically, RR tachogram data from several patients in the MIT-arrhythmia database [23] are transmitted from the PC to the STM32 board ①, where the PSD estimation is conducted. The computed PSD and/or HRV metrics LF, HF are then transmitted back to the PC from the STM32 board over the same UART line for further analysis and quality/precision evaluation. Similarly, the PPK2 board is connected to the PC via the serial

UART line ② to monitor in real-time the current consumption of the STM32 board. Note that ① captures the transmitted PSD or/and HRV data and does not provide power to the STM32 board, which is instead powered from the PPK2 via the power lines ③ in order to measure the current consumption. The current consumption monitored by the PPK2 can be visualized and exported to a CSV file using the nRFconnect power profiler app [36], while the logic ports ④ are used as flags to signal the time intervals during which the test MCU executes specific code sections/functions. (More details in Section IV-B).

##### B. MCU Configurations

**Hardware Configuration:** The STM32F401RE board is equipped with an ARM-Cortex M4 MCU, which is one of the most popular family of ARM processors [20], widely used in IoT products [21], [30]. The board offers 96 KB of SRAM and 512 KB of flash memory. It features a configurable AHB Prescaler, that allows to select different clock frequencies ( $f_{SYS} = 21, 42, 84$  MHz used in this work). The MCU also supports enhanced floating-point operations commonly encountered in signal processing systems by including a single-precision FPU. Additionally, the MCU supports various power modes. This study primarily utilizes the RUN mode for executing PSA algorithms and the SLEEP mode for reducing power consumption, by deactivating the MCU, while the peripherals remain active to maintain data reception over the UART line at 115 200 bauds, which is the fastest available setting. Furthermore, the MCU also offers 81 I/O ports, with ports PC5, PC6, PC8, and PC9 (as illustrated in Fig. 6) designated as general purpose input/output (GPIO) ports that allows to accurately profile each individual part of the test PSA systems.

**Firmware Configuration:** The STM32CubeMX [37] initialization code generator provided by ST, is employed for the MCU configuration, initializing the required libraries in C (e.g., HAL) and generating the Makefile containing compilation directives. The arm-none-eabi-gcc open-source compiler is utilized, with the Makefile facilitating the activation or deactivation of the FPU. For the test scenarios, the conventional LSP, Lagrange-based FLSP, and the proposed FGG-based LSP algorithms are evaluated. The optimized CMSIS library [22] is employed to enhance these algorithms, utilizing RFFT functions for hardware floating-point (HFP) or software floating-point (SFP), fixed-point 16 bits (Q16) and fixed-point 32 bits (Q32) and other optimized vector operations (e.g., multiplication, addition), which are specifically optimized by ARM for digital signal processing (DSP) operations.

##### C. Profiling and Characterization

In Fig. 7(a), a detailed comparison of current measurements is presented for both the conventional Lagrange-based FLSP and the proposed FGG-based LSP PSA systems, while in Fig. 7(b), a computed example of PSD for  $N = 256$  up to 0.4 Hz is illustrated. Initially, the MCU in each system is set to SLEEP mode, drawing 6.2 mA for fetching RR-data from the PC via the UART line. In RUN mode, the most computationally intensive tasks, namely the gridding processes (comprising Lagrange

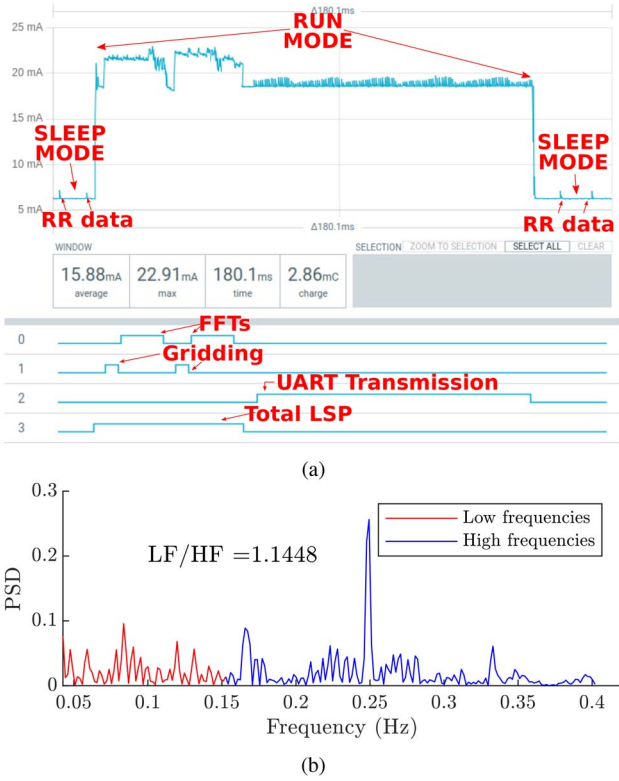


Fig. 7. (a) nRFconnect power profiler: current consumption measurements (b) Computed PSD using the conventional Lagrange-based FLSP for  $N = 256$ .

interpolation and Gaussian convolution) and the execution of two FFTs, are monitored using logic ports 0 and 1. Furthermore, the transmission of the PSD output data, identified as the most power-consuming process, is tracked by logic port 2, especially when HRV metrics such as LF and HF are not being processed by the MCU. For the evaluation of the PSD's quality and accuracy, data are transmitted byte by byte through the UART line. This involves transmitting 4 bytes per data point for the F32 and Q32 formats, represented as  $N_{bytes} = 4$ , and 2 bytes for the Q16 format, indicated as  $N_{bytes} = 2$ . Consequently, for  $N$  output frequencies, the total bytes transmitted amount to  $N_{bytes} \cdot N$ . The role of logic port 3 is crucial as it is used to capture the overall current consumption for the entire LSP computation, irrespectively of the inclusion of HRV metrics. It is important to note that in the example of Fig. 7(a) the PSD transmission cost dominates the total energy consumption as shown Fig. 8(a), while the FFT processing part accounts for approximately 1/4 of the total cost. To this end, by computing and transmitting the HRV metrics at the edge devices, the PSD transmission cost is eliminated, but the processing cost remains significantly high, especially for the FFT stage, as shown in Fig. 8(b). Furthermore, in Fig. 8(a) the Lagrange interpolation is the green part noted as "Grid", while the "Rest" part stands for any pre/post-processing prior/after the FLSP. The contribution of the "Rest" part slightly increases as shown in Fig. 8(b) as the HRV metric calculation (LF, HF) has to be taken into account. Therefore, it is important to investigate the achievable

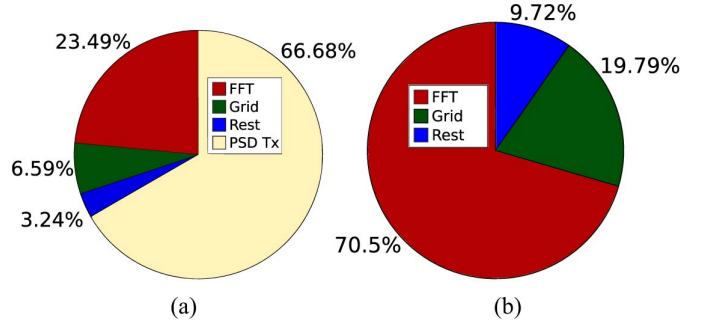


Fig. 8. Conventional Lagrange-based FLSP energy consumption for  $N = 256$ ,  $R_L = 4$ : (a) with PSD transmission. (b) HRV metrics transmission.

energy savings in terms of processing cost of the proposed FGG approach, especially for the gridding and FFT parts, compared with the conventional Lagrange-based counterpart.

## V. RESULTS

This section presents the experimental results of the PSA systems tested using the LSP. Initially, a complexity analysis is conducted, comparing different parts of the tested LSPs (direct LSP, Lagrange-based FLSP, and the proposed FGG-LSP). Subsequently, for different clock frequencies, the achieved savings in execution time, clock cycles and energy consumption are evaluated for the most energy critical components of the PSA systems averaged over 20 patient data for the MIT-arrhythmia database [23]. The selected samples are comprised of randomly selected subjects that are either healthy or have various clinically significant arrhythmias [23]. Following this, the LSP approaches are compared in terms of quality/accuracy (signal-to-noise ratio (SNR)) and energy efficiency. The influence of frequency scaling on power consumption is also examined, alongside the corresponding execution time.

### A. Complexity Analysis

In Fig. 9, a detailed comparison is made of the complexities of three distinct LSP PSA systems. The classic LSP method, serving as the baseline, shows quadratic complexity ( $O(N^2)$ ), as it requires  $N$  multiplications for each of the  $N$  output frequency points, leading to an excessive computational overhead (Fig. 9(a)). Therefore, for the real energy, time and current measurements the direct approach will not be taken into consideration. For the conventional Lagrange-based FLSP, the complexity is  $O(R_L N)$  in the gridding stage, with two  $4R_L N$  oversampled FFTs. The oversampling factor,  $R_L$ , is pivotal, especially for higher SNR levels. The proposed FGG-LSP approach presents an increased computational complexity in the gridding stage by about 1.8 times compared to the Lagrange-based method, however the FFTs size remains constant regardless of SNR requirements, allowing for up to 87.5% reduction. Consequently, the latter proposed strategy trades off the increased computational complexity in the gridding stage (linear complexity) with a reduced complexity in the FFT stage (loglinear complexity), significantly lowering the overall computational cost.



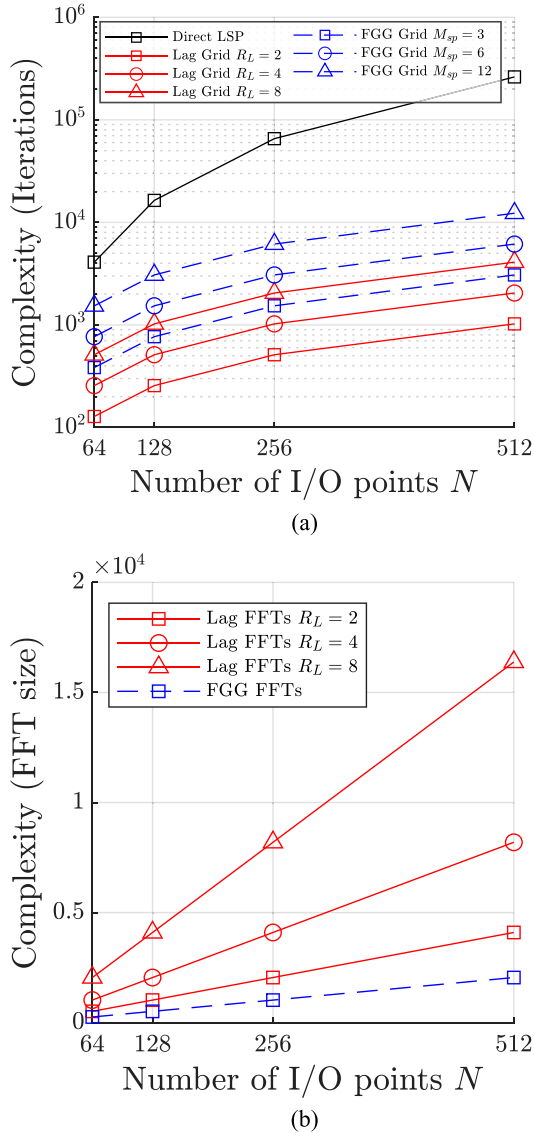


Fig. 9. Complexity analysis of (a) gridding stage (b) FFT stage for the Lagrange and FGG based LSP methods.

### B. Time & Clock Cycles & Energy Comparison

In Fig. 10, a comparison is shown between the conventional Lagrange-based FLSP and the proposed FGG-LSP approach in terms of clock cycles, energy consumption ( $\mu A \cdot ms$  at 1.9 V), and execution time (ms) at  $f_{SYS} = 84$  MHz (the class approach is omitted as it results in magnitudes of order higher clock cycles, time and energy). The bars in Fig. 10 are divided into two sections, representing the gridding and FFT stages, which are the main contributors to power consumption. The comparison includes four arithmetic schemes: HFP, Q16, Q32, and SFP. These methods are assessed across four different  $N$  sizes (64, 128, 256, 512) at three precision levels (low:  $R_L = 2$ ,  $M_{sp} = 3$ ; mid:  $R_L = 4$ ,  $M_{sp} = 6$ ; high:  $R_L = 8$ ,  $M_{sp} = 12$ ).

The FGG-based approach evidently surpasses the Lagrange-based method across all test parameters and scenarios. This

advantage primarily stems from the significantly smaller FFTs utilized in the proposed approach, coupled with a precomputed look-up table for quantized gridding. Specifically, as illustrated in Fig. 10(a)–10(c), for the Lagrange-based approach the FFT stage consumes the most clock cycles, time, and energy in all scenarios. Additionally, for higher precision levels, the FFT scales significantly, while the gridding part is less affected. Conversely, in the FGG-based approach (Fig. 10(d)–10(f)), the FFT component is considerably reduced, and higher precision impacts only the gridding stage, as more Gaussian points are considered. With respect to the different arithmetic schemes implemented, the HFP approach proves to be the most efficient. This efficiency is due to the MCU's dedicated FPU hardware accelerator for single precision floating point operations and the CMSIS library's optimized implementation of a RFFT (*arm\_rfft\_fast\_f32* function) specifically for this FPU [22]. Furthermore, the FGG approach demonstrates a lower scaling/doubling percentage in all three metrics (time, clock cycles, energy) as the total input/output points  $N$  increase, presenting a more efficient size scaling compared to the Lagrange-based method, as indicated in Table I. Nearly identical results are achievable for 21 MHz and 42 MHz thus they are omitted for simplicity.

In Fig. 11, the savings (%) in execution time and energy consumption of the proposed approach compared to the Lagrange-based method are presented for clock frequencies of 21 MHz (a-d), 42 MHz (b-e), and 84 MHz (c-f). The proposed FGG-LSP consistently outperforms the conventional method in all test scenarios. Under 21 MHz, the time and energy savings range from 39.75% to 90.02% and 42.81% to 91.11%, respectively. Under 42 MHz, the time and energy savings range from 40.98% to 90.15% and 45.90% to 90.08%, respectively. Under 84 MHz, these savings vary from 34.10% to 92.99% for time and 33.39% to 91.70% for energy. The precision level (low, mid, high) significantly influences the savings, as the FFT component of the proposed method is notably smaller. The size  $N$  has a minimal impact on the variation of savings. Notably, the SFP implementation yields slightly higher savings as it is significantly benefited from the CMSIS library, relative to the Q32, Q16, and HFP methods, with the latter showing marginally lesser benefits.

### C. SNR & Energy Comparison

In Fig. 12(a), the relationship between energy consumption and SNR is compared for a clock frequency of 84 MHz across different values of  $N$ , arithmetics (HFP, Q16, Q32, SFP), and precision levels for both the Lagrange-based and FGG-based approaches. The  $x$ -axis represents the achieved SNR,  $y$ -axis the size  $N$ , and  $z$ -axis the energy consumption. For clarity, Fig. 12(b) specifically illustrates the results for  $N = 64$ , identical to those in Fig. 12(a) for the same size  $N$ . The proposed solutions are marked in bold colors, while the Lagrange-based ones are in light colors. Precision levels are denoted as follows: ① for low precision ( $M_{sp} = 3$ ,  $R_L = 2$ ), □

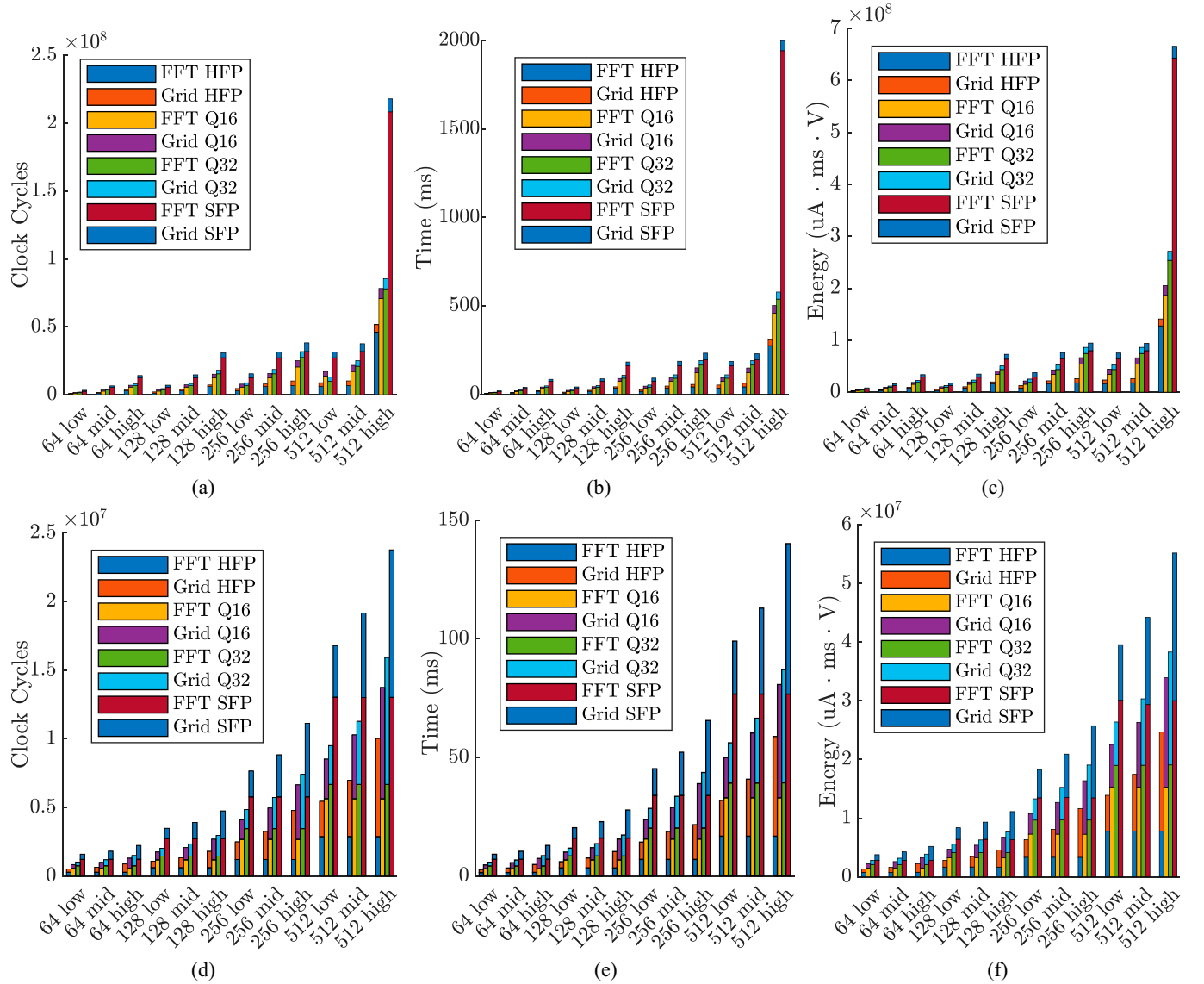


Fig. 10. Conventional Lagrange based (left figures) and proposed FGG based (right figures) LSP comparison at  $f_{SYS} = 84$  MHz: (a)–(b) clock cycles, (c)–(d) execution time, (e)–(f) energy consumption.

TABLE I  
SIZE SCALING/DOUBLING  $N$

Method	Percentage change (%)		
	Clock cycles	Time	Energy
Lan	144.61	128.59	139.02
FGG	119.11	115.12	116.23

for mid precision ( $M_{sp} = 6$ ,  $R_L = 4$ ), and  $\triangle$  for high precision ( $M_{sp} = 12$ ,  $R_L = 8$ ).

Analyzing all solutions in Fig. 12(a) and 12(b), the proposed approach consistently achieves lower energy consumption for a given SNR level in most test scenarios. The HFP arithmetic stands out for its superior energy-to-SNR performance, as floating-point operations have minimal quantization error compared to fixed-point methods. Additionally, the use of the

FPU and the optimized CMSIS library contributes to significant energy reductions. For a more meaningful comparison, the fixed-point approaches are contrasted with the SFP alternative, considering that many ultra-low power MCUs (ARM Cortex M0, M0+, M1) do not have an FPU. Excluding HFP solutions, the FGG-based Q16 approach (noted in bold-blue  $\textcircled{1}$ ) for low-precision ( $< 50$  dB) achieves the lowest energy consumption. For precision between 80 dB and 85 dB, the FGG-based Q32 (bold-green  $\square$ ) excels. In the highest precision range ( $> 100$  dB), the FGG-based SFP (bold-magenta  $\square$ ,  $\triangle$ ) shows the best performance. Regarding Lagrange-based methods, increasing the  $R_L$  factor does not enhance accuracy, rather it leads to more accumulative quantization errors due to the enlarged gridding part and FFTs in fixed-point arithmetic. This effect is less pronounced in the FGG-based solutions, as the FFTs size remains constant, affecting only the  $M_{sp} = 12$  solutions with many Gaussian points considered.

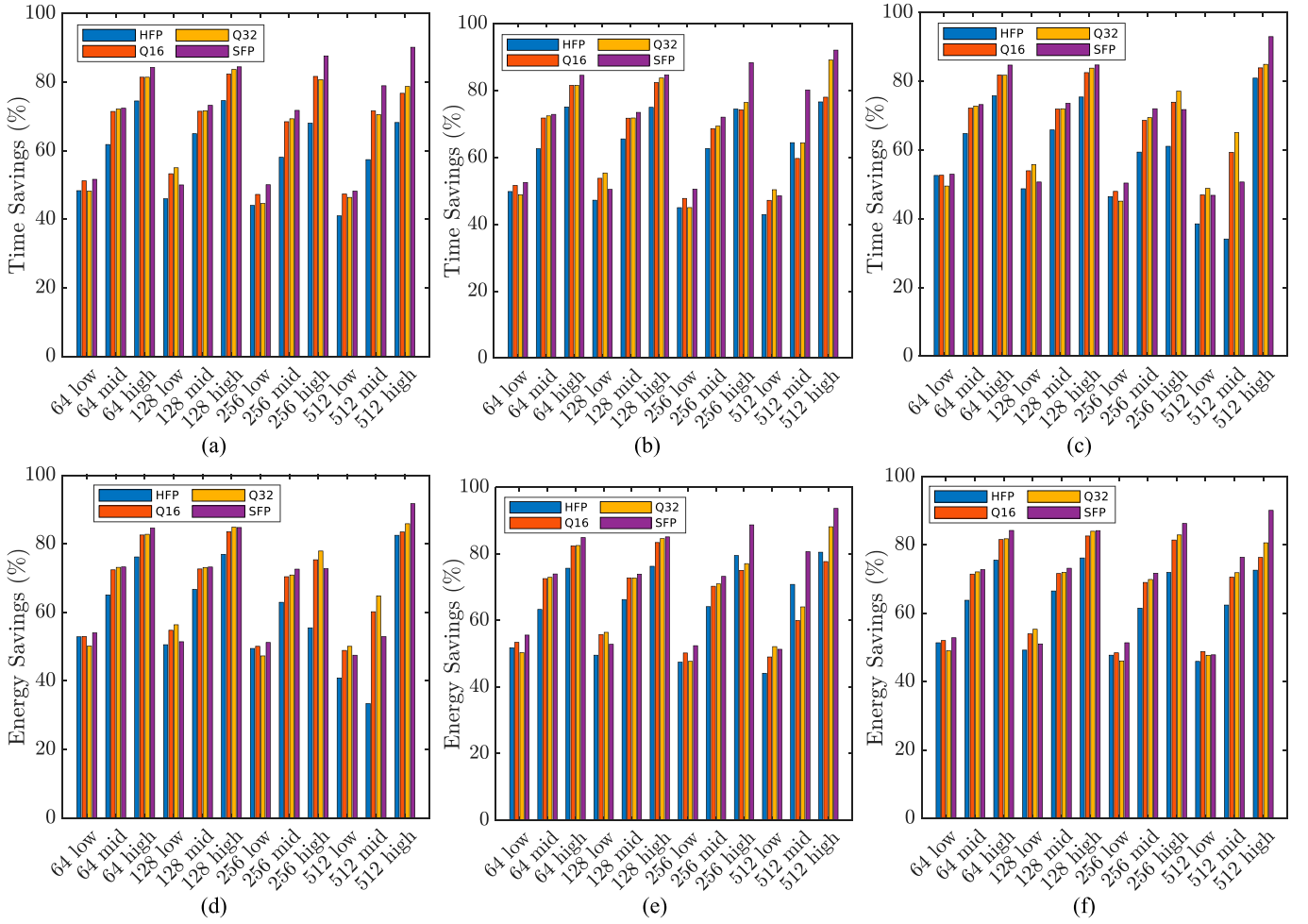


Fig. 11. Execution time and energy saving (%) of the proposed FGG approach compared the Lagrange based at:  $f_{SYS} = 21$  MHz: (a)–(b),  $f_{SYS} = 42$  MHz: (c)–(d),  $f_{SYS} = 84$  MHz: (e)–(f).

In Fig. 12(c), an analysis of energy savings (%) and SNR quality loss (%) is presented for all sizes  $N$  using non-FPU solutions. Here, the baseline is the SFP high precision ( $R_L = 8$ ) solutions, which, while being the most accurate, are also the most energy-intensive. It's observed that all FGG-based solutions demonstrate superior energy efficiency. The same solutions (bold-blue ①, bold-green □, and bold-magenta □, △) offer a distinct balance between energy savings and quality loss, effectively forming a *pareto front* for 84.75% and 3.87% up to 96.67% and 67.39% energy savings and quality loss respectively. The LF and HF bands using the proposed FGG-LSP and the conventional Lagrange-based LSP methods are have been calculated and compared against those derived from the classic direct LSP approach. In all test scenarios, the integral of the PSD from 0.05 Hz to 0.15 Hz (LF) and from 0.15 Hz to 0.4 Hz (HF) is calculated and its change (%) from the classic direct LSP approach is computed for all the test patient samples. As detailed in Table II, the Lagrange-based method's LF band presents an average change of 0.48%, with a standard deviation of 0.85%, and peaks at a change of 3.65%. Conversely, the HF band under this same method shows a notably higher average of 3.70%, a standard deviation of 5.29%, and reaches a maximum change of 28.29% for  $N = 512$ , Q16, and a high precision level

(rightmost light blue triangle in Fig. 12(a)). In comparison, the FGG-LSP method demonstrates a less significant error on both LF and HF ranges. Here, the LF presents an average change of 0.38%, a standard deviation of 0.51%, and a maximum change of 1.14%. The HF continues this trend with an average of 0.71%, a standard deviation of 0.71%, and a highest change of 2.37%. This data confirms that the FGG-LSP method accurately computes the LF and HF ranges within tolerable error margins for most arrhythmia detections, as shown in [16].

#### D. Power & Frequency Scaling Comparison

In Fig. 13(a), the effects of frequency scaling on average current consumption and execution time are investigated. The presented solutions are for  $N = 64$  to 512, for different arithmetics (HFP, Q16, Q32, SFP) and precision levels for both the Lagrange-based and FGG-based approaches. In general, the FGG solutions consume less current, while also having shorter execution time compared to their Lagrange-based counterparts, due to the smaller FFTs for each of the three clock frequencies. Furthermore, for the same solution, reducing the clock frequency decreases the average current but also increases the execution time. Thus, it is important to find the clock frequency

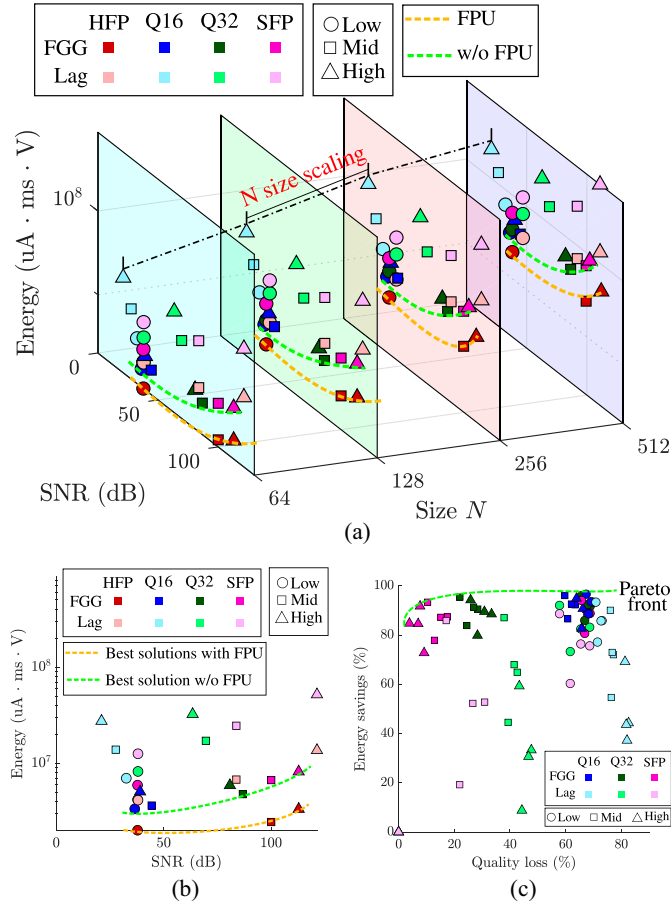


Fig. 12. Energy saving and SNR at 84 MHz for: (a) different sizes, (b) for  $N = 64$ . (c) Energy saving (%) and Quality loss (%) for different sizes at 84 MHz.

TABLE II  
LF, HF BANDS OF THE PROPOSED FGG-LSP AND THE  
CONVENTIONAL LAGRANGE-BASED APPROACHES

Method	Metric	Average Change (%)	Std Change (%)	Max Change (%)
Lan	LF	0.48	0.85	3.65
	HF	3.70	5.29	28.29
FGG	LF	0.38	0.51	1.14
	HF	0.71	0.71	2.37

that strikes the perfect balance between these two metrics, leading to energy efficient solutions.

In Fig. 13(b) and 13(c), a comparison is performed between average power (current  $\times$  voltage) and the percentage increase in execution time. The reference for each comparison is the solutions at  $f_{SYS} = 42$  MHz, with each solution (i.e.,  $N$  size, precision level, arithmetic) at the other two clock frequencies compared against the 42 MHz solutions. Additionally, iso-energy lines are highlighted, indicating points of equal energy consumption. It is evident that for both FGG and Lagrange-based solutions, the most favorable frequency is the highest, at 84 MHz. For the FGG-based solution, the mean power increase is 66.93%, with a standard deviation of 1.83%, and the mean time decrease is 50.59%, with a standard deviation of 1.76%. For the Lagrange-based approach, the mean power

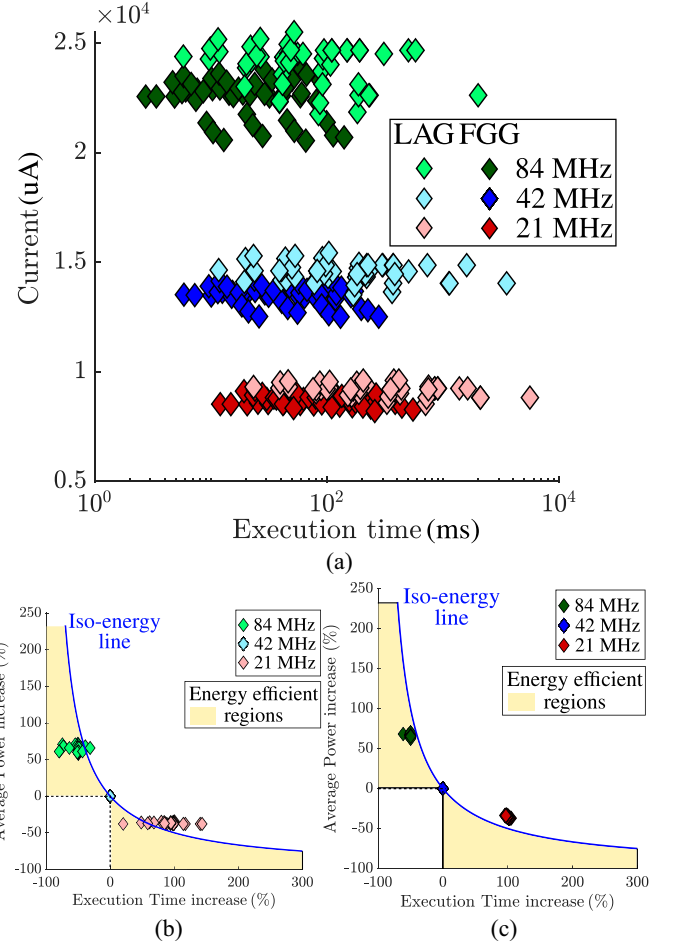


Fig. 13. (a) Current consumption and execution time at 1.9 V for  $f_{SYS} = 21$  MHz,  $f_{SYS} = 42$  MHz and  $f_{SYS} = 84$  MHz, (b)–(c) Power increase (%) and execution time decrease (%) with points at  $f_{SYS} = 84$  MHz as a reference for the conventional Lagrange-based FLSP (b), the proposed FGG-FLSP (c).

increase is 64.94%, with a standard deviation of 3.47%, and the mean time decrease is 51.90%, with a standard deviation of 8.57%. In contrast, at a clock frequency of 21 MHz, the FGG-based solution, the mean power decrease is 35.18%, with a standard deviation of 1.3%, and the mean time increase is 99.81%, with a standard deviation of 2.23%. For the Lagrange-based approach, the mean power decrease is 36.83%, with a standard deviation of 0.91%, and the mean time increase is 96.39%, with a standard deviation of 27.00%. Evidently, the FGG approach demonstrates a lower time standard deviation compared to the Lagrange-based solutions, making 84 MHz the most energy-efficient choice consistently. It is important to note that these findings consider only the power consumption during RUN mode. The current consumption during SLEEP mode, which varies based on the input data rate, is not included in this analysis.

### E. Impact

The proposed approach demonstrates the feasibility of executing complex PSA on state-of-the-art MCUs by effectively reducing the computational complexity of the LSP and efficiently utilizing the resources provided by the MCU. By fully



integrating the PSA system at the edge device, this approach limits and potentially eliminates the need for data transmission, thereby indirectly saving transmission power and avoiding associated losses, which account for a significant portion of total energy consumption. In contrast, a conventional system that performs time-domain analysis on the MCU or transmits all calculated PSD points and  $N$  RR interval samples to a smartphone/server would necessitate the transmission of  $2N$  or  $4N$  total bytes. Similarly, transmitting only the autocorrelation coefficients for AR PSD estimation requires  $2p$  or  $4p$  total bytes. However, by processing all data on-site and transmitting only essential HRV metrics (TP, LF, and HF), transmission costs are significantly reduced by 97.66%, 98.82%, and 99.41% for  $N = 128, 256, 512$  respectively, and by 81.25%, 90.63%, and 95.31% for  $p = 16, 32, 64$  respectively in the AR approach. Although on-site processing increases the energy overhead, the proposed FGG-LSP method addresses this issue by substantially reducing the energy requirements of the LSP, as elaborated in the preceding sections. Finally, the proposed methodology can be expanded and performed on other platforms, thus enabling the investigation of different MCUs (e.g., ARM Cortex M0, M0+, M1). This expansion would allow for exploring an additional dimension, encompassing various algorithmic and arithmetic configurations, clock frequencies, and MCU optimizations, which have already been investigated in this work.

## VI. CONCLUSION

This paper presents a new PSA system for HRV evaluation by replacing the existing Lagrange-based FLSP method with an optimized FGG-LSP-based one and implementing it on a real embedded system. Diverging from previous studies that focused solely on system implementation, this research conducts an extensive evaluation, measuring energy, the execution time, clock cycles, and current consumption of various parts of the PSD estimation process in real-time. Furthermore, it explores several architecture-level configurations of the MCU, including clock frequency, power modes and the FPU, aiming to balance energy consumption and computation quality, effectively paving the way for broader applications and real-time analysis of ECGs and other bio-signals. The experimental results show that the proposed FGG-based approach outperforms the conventional Lagrange-based method, achieving substantial execution time and energy savings ranging from 34.10% to 92.99% and from 33.39% to 91.70%, respectively, with minimal accuracy loss.

## REFERENCES

- [1] N. Arandia, J. Garate, and J. Mabe, "Embedded sensor systems in medical devices: Requisites and challenges ahead," *Sensors*, vol. 22, 2022, Art. no. 9917.
- [2] M. Amini et al., "Trend analysis of cardiovascular disease mortality, incidence, and mortality-to-incidence ratio: Results from global burden of disease study 2017," *BMC Public Health*, vol. 21, no. 25, pp. 1471–2458, 2021.
- [3] S. S. Mishra and A. Rasool, "IoT Health care Monitoring and Tracking: A Survey," in *Proc. 3rd Int. Conf. Trends Electron. Inform. (ICOEI)*, 2019, pp. 1052–1057.
- [4] Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology, "Heart rate variability: Standards of measurement, physiological interpretation and clinical use," *Circulation*, vol. 93, no. 5, pp. 1043–1065, 1996.
- [5] A. A. Khan, G. Y. H. Lip, and A. Shantsila, "Heart rate variability in atrial fibrillation: The balance between sympathetic and parasympathetic nervous system," *Eur. J. Clin. Investigation*, vol. 49, no. 11, pp. 1–8, 2019.
- [6] H.-G. Kim, E.-J. Cheon, D.-S. Bai, Y.-H. Lee, and B.-H. Koo, "Stress and heart rate variability: A meta-analysis and review of the literature," *Psychiatry Investigation*, vol. 15, no. 3, pp. 235–245, 2018.
- [7] H. Huang, H. Chan, P. Lin, C. Wu, and C. Huang, "Time-frequency spectral analysis of heart rate variability during induction of general anaesthesia," *Brit. J. Anaesthesia*, vol. 79, no. 6, pp. 754–8, 1997.
- [8] J. D. Blood et al., "The variable heart: High frequency and very low frequency correlates of depressive symptoms in children and adolescents," *J. Affect. Disorders*, vol. 186, pp. 119–126, Jul. 2015.
- [9] J. Zhang, "Effect of age and sex on heart rate variability in healthy subjects," *J. Manipulative Physiol. Therapeutics*, vol. 30, no. 5, pp. 374–379, 2007.
- [10] K. Li, C. Cardoso, A. Moctezuma-Ramirez, A. Elgalad, and E. Perin, "Heart rate variability measurement through a smart wearable device: Another breakthrough for personal health monitoring?" *Int. J. Environ. Res. Public Health*, vol. 20, 2023, Art. no. 7146.
- [11] F. Shaffer and J.-P. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers Public Health*, vol. 5, 2017, Art. no. 258.
- [12] S. Yoshida et al., "Energy-efficient spectral analysis method using autoregressive model-based approach for Internet of Things," *IEEE Trans. Circuits Syst. I*, vol. 66, no. 10, pp. 3896–3905, Oct. 2019.
- [13] K. Kajihara et al., "Hardware implementation of autoregressive model estimation using Burg's method for low-energy spectral analysis," in *IProc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, 2018, pp. 199–204.
- [14] C. Eleftheriadis and G. Karakostas, "ACOR: On the design of energy-efficient autocorrelation for emerging edge applications," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, 2023, pp. 1–9.
- [15] W. H. Press and G. B. Rybicki, "Fast algorithm for spectral analysis of unevenly sampled data," *Astrophys. J.*, vol. 338, pp. 277–280, Mar. 1989.
- [16] G. Karakostas, A. Sankaranarayanan, M. M. Sabry, D. Atienza, and A. Burg, "A quality-scalable and energy-efficient approach for spectral analysis of heart rate variability," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibition (DATE)*, 2014, pp. 1–6.
- [17] H. Guo and C. S. Burrus, "Wavelet transform based fast approximate Fourier transform," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, vol. 3, 1997, pp. 1973–1976.
- [18] A. I. Maistrou, "Implicit comparison of accuracy of heart rate variability spectral measures estimated via heart rate and heart period signals," in *Proc. IEEE Comput. Cardiol.*, 2008, pp. 553–556.
- [19] A. Metwally, J. P. N. Queralt, V. K. Sarker, T. N. Gia, O. Nasir, and T. Westerlund, "Edge computing with embedded AI: Thermal image analysis for occupancy estimation in intelligent buildings," in *Proc. Intell. Embedded Syst. Archit. Appl. Workshop*, 2020, pp. 1–6.
- [20] "Cortex M4." ARM. Accessed: Jan. 2024. [Online]. Available: <https://developer.arm.com/Processors/Cortex-M4>
- [21] D. Roggen, "ARM cortex M4-based extensible multimodal wearable platform for sensor research and context sensing from motion & sound," in *Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. Proceed. ACM Int. Symp. Wearable Comput.*, 2020, pp. 284–289.
- [22] "CMSIS-DSP library." ARM-software. Accessed: Jan. 2024. [Online]. Available: <https://github.com/ARM-software/CMSIS-DSP>
- [23] A. L. Goldberger, L. A. N. Amaral, L. Glass, and J. M. Hausdorff, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *J. Circ.*, vol. 101, no. 23, pp. 215–220, Jun. 2000.
- [24] J. Lee, K. Matsumura, K.-I. Yamakoshi, P. Rolfe, S. Tanaka, and T. Yamakoshi, "Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Society (EMBC)*, 2013, pp. 1724–1727.
- [25] E. De Giovanni, T. Teijeiro, G. P. Millet, and D. Atienza, "Adaptive R-peak detection on wearable ECG sensors for high-intensity exercise," *IEEE Trans. Biomed. Eng.*, vol. 70, no. 3, pp. 941–953, Mar. 2023.
- [26] E. De Giovanni et al., "Modular design and optimization of biomedical applications for ultralow power heterogeneous platforms," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3821–3832, Nov. 2020.

- [27] F. Tueche, Y. Mohamadou, A. Djeukam, L. C. N. Koueque, R. Seujip, and M. Tonka, "Embedded algorithm for QRS detection based on signal shape," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
- [28] E. S. Pramukantoro, K. Amron, V. Wardhani, and P. A. Kamila, "Comparative analysis of classification features for ECG signal classification on raspberry Pi using wearable sensor data," in *Proc. 8th Int. Conf. Sustain. Inf. Eng. Technol.*, 2023, pp. 612–618.
- [29] W.-C. Fang, H.-C. Huang, and S.-Y. Tseng, "Design of heart rate variability processor for portable 3-lead ECG monitoring system-on-chip," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1491–1504, 2013.
- [30] V. H. Rodriguez, C. Medrano, and I. Plaza, "Embedded system based on an ARM microcontroller to analyze heart rate variability in real time using wavelets," *Wireless Commun. Mobile Comput.*, vol. 2018, Art. no. 14, 2018, Art. no. 14.
- [31] D. Naranjo-Hernández, L. M. Roa, J. Reina-Tosina, G. Barbarov-Rostan, and O. Galdámez-Cruz, "Smart device for the determination of heart rate variability in real time," *J. Med. Eng.*, vol. 2017, Nov. 2017.
- [32] B. Leroy, "Fast calculation of the lomb-scargle periodogram using nonequispaced fast Fourier transforms," *Astron. & Astrophys.*, vol. 545, no. A50, 2012, Art. no. 9.
- [33] J. Keiner, S. Kunis, and D. Potts, "Using NFFT3 - A software library for various nonequispaced fast Fourier transforms," *ACM Trans. Math. Softw.*, vol. 36, pp. 1–30, 2009, Art. no. 19.
- [34] R. H. D. Townsend, "Fast calculation of the Lomb-Scargle periodogram using graphics processing units," *Astrophys. J. Suppl. Series*, 2010.
- [35] L. Greengard and J.-Y. Lee, "Accelerating the nonuniform fast fourier transform," *SIAM Rev.*, vol. 46, no. 3, pp. 443–454, 2004.
- [36] "nRF connect for desktop." Nordic. Accessed: Jan. 2024. [Online]. Available: <https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-desktop/download>
- [37] "STM32CubeMX." STMicroelectronics. Accessed: Jan. 2024. [Online]. Available: <https://www.st.com/en/development-tools/stm32cubemx.html>



**Charalampos Eleftheriadis** (Graduate Student Member, IEEE) received the M.E. degree in electrical and computer engineering from Democritus university of Thrace (DUTH), Xanthi, Greece, in 2020. Currently, he is a Marie Curie funded early stage Researcher and working toward the Ph.D. degree with Queen's University Belfast. His research interests include algorithm and hardware co-design of power spectral analysis (PSA) systems, energy efficient fast Fourier transform architectures and applied digital signal processing algorithms on hardware.



**Georgios Karakonstantis** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees from Purdue University, USA. He is an Associate Professor in the School of Electrical and Computer Engineering in University of Thessaly, Greece, and a honorary Visiting Academic in Queen's University Belfast, U.K. He has published more than 95 papers in peer reviewed journals and conferences, and he is an inventor of a US patent and author of three book chapters. His research interests include energy-efficient and error-resilient computing and storage

architectures for embedded and high-performance applications. He is the recipient of three best paper awards and nominations (in IEEE DATE and MICRO conferences), of three HiPEAC paper awards, and of a prize at the Altera Innovate Design Contest.