IEEE Open Journal of
Circuits and Systems

Received 29 March 2020; revised 29 June 2020; accepted 3 July 2020. Date of current version 26 August 2020.
This article was recommended by Guest Editor A. Bermak.

Digital Object Identifier 10.1109/OJCAS.2020.3009822

Low Power Optimisations for IoT Wearable Sensors Based on Evaluation of Nine QRS Detection Algorithms

JIAMIN LI[1] (Student Member, IEEE), ADNAN ASHRAF[2], BARRY CARDIFF[2] (Senior Member, IEEE), RAJESH C. PANICKER[1] (Member, IEEE), YONG LIAN[3] (Fellow, IEEE), AND DEEPU JOHN[2] (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583

[2]School of Electrical and Electronic Engineering, University College Dublin, Dublin 4, D04 V1W8 Ireland

[3]Department of Electrical Engineering and Computer Science, Lassonde School of Engineering, York University, Toronto, ON M3J 1P3, Canada

CORRESPONDING AUTHOR: D. JOHN (e-mail: deepu.john@ucd.ie)

This work was supported in part by the NSERC Canada Discovery Grant; and in part by the Microelectronic Circuits Centre Ireland under Grant MCCI-2018-03.

ABSTRACT This paper aims to reduce the power consumption of electrocardiography based wearable healthcare devices, by introducing power reduction approaches and considerations at system level design, where we have the highest potential to influence power. It focuses, in particular, on algorithm design and implementation, data acquisition, and transmission under constrained resources. A thorough investigation of the suitability of nine existing algorithms for on-sensor QRS feature detection is conducted, with respect to metrics such as sensitivity, positive predictivity, power consumption, parameter choice and time delay. Optimisation of data acquisition on CPU-based IoT systems is performed, and the current consumption is reduced by a factor of 3 using a combination of direct memory access (DMA) list approach and low-level register manipulations for task delegation. The acquisition data rate, sampling rate, buffer and batch size are also optimised. To reduce the power consumption by data transmission, the effect of on-sensor versus off-sensor processing is investigated. While focusing on CPU-based systems with experiments performed on a generic low-power wearable platform, the design optimisation and considerations proposed in this work could be extended to custom designs and allow further investigation into QRS detection algorithm optimisation for wearable devices.

INDEX TERMS Bluetooth low energy, direct memory access, Internet of Things, on-chip processing, QRS detection, wearable sensors.

I. INTRODUCTION

IN RECENT years, an increasing amount of effort has been devoted to developing real-time wearable cardiac monitoring devices, as they offer greater mobility and enable early detection of cardiovascular diseases as compared to traditional ambulatory Holter monitors. Through constant monitoring, these wearable devices can detect cardiac rhythm disorders before the disease deteriorates, allowing treatment at the pre-clinical stage which not only increases the chance of complete recovery but also curbs mounting healthcare expenditure. Currently, the detection of cardiac diseases relies on identifying, extracting and analysing the features of each heartbeat event from the person's electrocardiogram (ECG) signal.

There are three main events in typical ECG signals: the P wave, the QRS complex, and the T events, the duration and frequency of which suggest cardiac conditions. Compared to P and T events, the automatic detection of the QRS complex is found to be comparatively easier and more reliable, and therefore carries greater clinical significance [1]. As a result, extensive algorithms on QRS detection have been developed and enhanced, with the focus being improving the detection

accuracy and the robustness against commonly experienced noises such as power line interference [2], baseline drift [3], electrode noise and motion artefacts [3], [4]. Multimodal fusion techniques were developed to optimise the accuracy in case discontinuous or noisy data [5]. In [6], convolutional neural network based machine learning techniques were proposed to optimise the QRS detection accuracy. A dynamic plosion index based algorithm which doesn't require a manual threshold and can automatically adapt to input signal characteristics is presented in [7]. All the above works focus on improving accuracy of QRS detection in a noisy setting. However, numerical efficiency and power consumption are of secondary concern, as these algorithms are mostly designed for hospital settings, with high-speed processors and power supply readily available.

However, as the idea of remote and proactive healthcare gains popularity, emerging wearable technology is being actively employed in the health industry for self-monitoring of medical conditions. Unlike clinical equipment, wearable platforms are constrained by computational capability, and ultimately by power and battery capacity. Keeping the power consumption as low as possible would extend the battery life of a wearable device and thus enable longer continuous monitoring, thereby avoiding interruptions in ECG data collection and patient care. Although there are several attempts at developing low complexity QRS detection algorithms, there is very little attention to reducing system level power for embedded wearable applications. In [8], a lightweight min-max technique is proposed for complexity reduction. An algorithm combining matched filtering and Hilbert transform is proposed in [9] for QRS detection. In [10], a lightweight fast fourier transform (FFT) based algorithm is proposed for wearable smartvest application. Reference [11] presents a modular low complexity algorithm for real-time embedded systems. Except [11], none of the other techniques presents an embedded implementation and there is little focus on embedded power reduction techniques.

A typical wearable ECG device consists of three main functional blocks – sensing front-end for the acquisition of digitised ECG signals, a processing unit for feature extraction, and wireless transmitter for uploading the data [12]. The solutions presented in this paper applies to a typical Internet of Things (IoT) embedded wearable sensor with a highly integrated 2-chip solution. Here, a chipset such as TI ADS1292R which integrates a variable gain amplifier (VGA), analog to digital converter (ADC) and a serial interface could be used for biomedical data acquisition. Also, a System-on-Chip (SoC), such as Nordic Semiconductor nRF 52 series which integrates a Bluetooth transceiver and a micro-controller could be used for biomedical data processing and wireless transmission. This paper investigates power consumption at data acquisition, processing, and transmission stages on the IoT wearable sensor, and proposes techniques and design recommendations to reduce the system power consumption. The paper approaches this issue on system design level for potentially higher influence on power consumption [13]. The data acquisition and transmission design choices allow for better power-delay balancing. Adopting power-efficient data processing (i.e., QRS detection) algorithm design and implementation allows for generic power reduction while largely meeting the performance requirements for wearable applications. The techniques introduced in this paper has broader relevance for wearable ECG devices, whether CPU-based or custom designed [14], [15]. In the case of a fully custom hardware implementation of biomedical processing in the data acquisition chip itself, the power consumption of various algorithms reported in Table 4 gives an indication of the comparative computational complexity and power consumed in a fully hardware mode.

At the data acquisition stage, traditional direct memory access (DMA) and DMA list based solutions are implemented and compared in terms of power consumption. The influences of Serial Peripheral Interface (SPI) clock frequency, ECG signal sampling rate and DMA buffer list size are taken into consideration. Based on the results and analysis, an optimal way to acquire data over SPI is proposed. At the data processing (i.e., QRS detection) stage, the performance-power consumption trade-off analysis is conducted for nine popular QRS peak detection algorithms, which is critical in evaluating their performance and suitability for battery-operated devices. Parameter choices and time delay (i.e., suitability for real-time applications) are also considered. Such trade-off analysis is crucial, as the majority of existing QRS detection algorithms were designed for hospital settings where power consumption is not an important factor, whereas reasonable compromises in QRS detection performance with respect to power need to be made prior to their deployment on wearable devices. This analysis would help in optimising a QRS detection algorithm for wearable devices. The essentiality of on-board QRS detection for power reduction is also experimentally illustrated.

The paper is organised as elaborated below. Followed by the introduction, literature related to general QRS detection methodology, types of noises affecting ECG recordings, and selected QRS detection algorithms are presented. Section II discusses the methodology used for this work. Section III discusses the optimisations used for data acquisition. The next two sections present the optimisation for data processing, transmission, and experimental results of this paper. In particular, Section IV discusses comparison and optimisation for data processing. Section V details the data transmission stage while comparing the results of on-board processing and raw data transmission over Bluetooth Low Energy (BLE). This paper concludes with Section VI which discusses potential future work proposed to encourage further contributions in this field.

II. METHODOLOGY

This section presents the evaluation metrics for QRS detection algorithms and elaborates on how these metrics are measured in this study, both quantitatively and qualitatively.

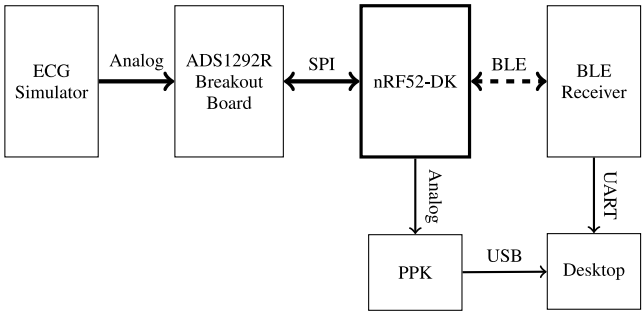

FIGURE 1. Block Diagram of the Hardware Platform.

A. EVALUATION METRICS

As this paper examines the power consumption of data acquisition, processing (i.e., QRS detection) on chip with respect to the performance of feature extraction and transmission, it is essential to define several common metrics and benchmarks so that a fair and reasonable comparison study/trade-off analysis could take place.

In this work, metrics for algorithm evaluation are chosen to be 1) power consumption, 2) sensitivity of QRS peaks, 3) positive predictivity of QRS peaks, 4) parameter choice (i.e., versatility), 5) time delay. While metrics 2-4 are good qualitative and quantitative representations of algorithm performance, metrics 1 and 5 examines the suitability of an algorithm on wearable platforms which calls for low power consumption and real-time response. Using them together would allow a thorough trade-off analysis among the nine algorithms selected.

B. POWER CONSUMPTION MEASUREMENT

To measure the power consumption, a hardware platform is set up to simulate the typical wearable wireless ECG device architecture, as shown in Fig. 1. An ECG simulator works as an ECG signals source, and Texas Instruments' ADS1292R breakout board acts as an analog front-end which filters and digitises the ECG signals. We opted for ADS1292R as it supports two channel ECG acquisition along with bio-impedance for respiratory rate estimation. Nordic Semiconductor's nRF52 Development Kit nRF52-DK acquires digitised ECG data from the analog front-end via SPI, extracts the QRS complexes and transmits useful data over BLE. The only serial interface supported by ADS1292R is SPI and therefore this interface was chosen for data. SPI DMA and SPI DMA List are mechanisms whereby the SPI transfer is controlled by a DMA controller instead of the host CPU in nRF52832. The nRF52- DK houses an nRF52832 SoC which is built around 64 MHz ARM Cortex-M4F CPU with 512 KB Flash and 64 KB RAM and has and integrated BLE 5.0 transceiver. The reason for choosing nRF52832 was its high level of integration with BLE and MCU into a single chipset along with the availability of a floating-point unit which helps in the implementing complex signal processing techniques. The BLE receiver in Fig. 1, is an additional board containing the same Nordic Semiconductor's nRF512832 SoC.

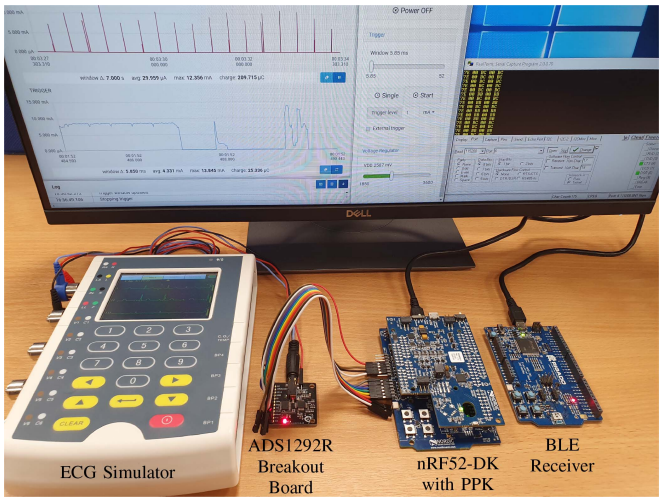

FIGURE 2. Experimental Setup for Current Measurement.

The power consumed by the nRF52832 SoC is measured using Nordic Semiconductor's Power Profiler Kit (PPK) nRF6707 which plugs directly onto the nRF52-DK. The PPK measures current from 1 μA to 70 mA and gives a detailed picture of the current profile. The PPK provides the real-time current measurement results on a desktop application with a resolution down to 13 μs. In this paper, all current measurements refer to the average current consumption of the nRF52832 SoC only when supplied with 2.85 V. The experimental measurement setup is illustrated in Fig. 2.

C. SENSITIVITY & POSITIVE PREDICTIVITY MEASUREMENT

The performance of a QRS detection algorithm is measured quantitatively by two metrics: sensitivity and positive predictivity. Sensitivity measures the percentage of correct QRS detections among all QRS complex in the database, calculated as $Sensitivity = TP/(TP + FN)$, where TP represents the number of true positives and FN the number of false negatives. Positive predictivity, on the other hand, measures the percentage of correct QRS detections among all QRS complex reported by the algorithm. It is calculated as $Positive\ Predictivity = TP/(TP + FP)$, where FP denotes the number of false positives (i.e., normal ECG data points incorrectly classified as QRS complex by the algorithm).

This paper uses MIT-BIT Arrhythmia Database [16] from PhysioNet [17], which is a collection of 48 two-channel Holter ECG recordings, each 30 minutes long, with a sampling frequency of 360 SPS and a resolution of 11 bits. Artefacts that commonly exist in clinical settings, as well as arrhythmia with significant variation in ECG features are represented in the excerpts, so as to best emulate the practical scenarios. Manual annotations are done for all recordings.

To obtain numerical values for these two metrics, each QRS detection algorithm is implemented in MATLAB in

real-time, and tested against MIT-BIH database. The average values of sensitivity and positive predictivity of all recordings are calculated and taken as the quantitative performance indicator of the algorithm.

D. VERSATILITY & TIME DELAY APPRAISAL

A closer look at QRS detection algorithms reveals that some parameters such as threshold ratio are almost always determined empirically, based on the algorithm performance in a particular database. This poses questions on whether the algorithm itself is indeed able to perform well in another database or a real-life scenario of notably different characteristics. By analysing how the parameters are determined and how much influence they have on the performance of a particular algorithm, the versatility of that algorithm could be inferred.

Time delay determines whether an algorithm is suitable for real-time application on an SoC, and is calculated by studying each individual QRS detection algorithm. For example, an algorithm with a higher-order FIR filter implies a larger time delay, and a decision rule involving more knowledge of future values results in longer lag time.

III. DATA ACQUISITION OPTIMISATION

The data acquisition stage continuously senses, collects and digitises ECG waves on the human body, and then passes the data to the processing stage via communication protocols such as Serial Peripheral Interface (SPI). This section explores the optimal dynamic memory access (DMA) SPI implementation, acquisition frequency, and buffer size, as well as their influence on power consumption.

The nRF52832 SoC is utilised as a CPU-based wearable device that acquires digitised ECG data from ADS1292R via SPI, with the former being SPI master and latter the slave. Upon initialising both devices and configuring ADS1292R to data streaming mode, the slave device will pull down its *Data Ready* pin every time a new conversion data is available, i.e., every sample interval of ADS1292R. The corresponding pin on the master device (i.e., nRF52832 SoC) acts as an interrupt source. Every time the *Data Ready* pin transitions to low, the SoC has to acquire 9 bytes of data (24 status bits + 24 bits × 2 channels) from ADS1292R via SPI.

In the DMA approach, once the SoC detects a falling edge on the interrupt pin, the interrupt handler will be invoked and the SoC will initiate reading of 9 bytes from ADS1292R via SPI. The SPI data transfer task is delegated to a DMA module on the SoC, which receives and stores SPI data into a DMA buffer in its RAM. The CPU itself is put into sleep mode during SPI data transfer. However, CPU has to increment the DMA buffer pointer and then initiate data transfer every time interrupt occurs to avoid overwriting previous data. Note that in this approach, the master CPU wakes up every time a new sample is available.

In the DMA list approach, every time the SPI data transfer is initiated, DMA automatically increments its DMA buffer pointer by 9 bytes, and the CPU doesn't have to do it manually. Besides, the SoC provides programmable peripheral interconnect (PPI) which enables peripherals to interact autonomously with each other using tasks and events independent of the CPU. Using the capability, PPI is employed to connect the interrupt event to the SPI data transfer task. Since SPI data transfer is now automatically initiated every time the interrupt occurs due to PPI, and the SPI DMA automatically increments its DMA buffer pointer every time due to the DMA list feature, and there is no involvement of the CPU. The CPU itself remains in sleep mode during this, and wakes up for DMA buffer pointer reset only upon a counter interrupt when a pre-defined amount of data has been collected. This approach is efficiently implemented using GPIOTE, SPIM, PPI and TIMER modules of the nRF52832 SoC.

TABLE 1. Current consumption by DMA and DMA list approach at different sampling rates.

Approach	Sampling Rate			
	125 SPS	250 SPS	500 SPS	1000 SPS
DMA	20.6 µA	38.5 µA	71.6 µA	86.1 µA
DMA List	8.3 µA	12.9 µA	21.7 µA	38.0 µA

TABLE 2. Current consumption at different SPI data rates.

1 Mbps	2 Mbps	4 Mbps	8 Mbps
43.6 µA	26.6 µA	17.3 µA	12.9 µA

Table 1 demonstrates the current measurements of data acquisition by the two approaches at different sampling rates with a fixed DMA buffer list size of 1000 samples and SPI data rate of 8 Mbps.

Based on the observations from Table 1, data acquisition by the DMA list approach consumes less than half the current consumed by DMA approach, due to task delegation and more efficient CPU utilisation. Therefore, it is intuitive to conclude that the DMA list approach is a power-efficient means for data acquisition via SPI.

The influence of SPI data rate on power consumption with fixed DMA buffer list size of 1000 samples is investigated as demonstrated in Table 2.

From Table 2, it is evident that higher data rates result in efficient data acquisition, primarily due to the reduced time spent in data acquisition. Thus, the maximum possible data rate of 8 Mbps is chosen for the rest of the experiments.

To examine the influence of sampling rate and DMA buffer list size on power consumption, current measurements with respect to different DMA buffer list sizes and sampling rates are summarised in Table 3 with 8 Mbps SPI data rate employing DMA list approach.

According to Table 3, there is no significant influence on power consumption due to changes in DMA buffer list size, possibly because there is no data processing involved at this stage after acquisition of these samples. No signal processing

TABLE 3. Current consumption at different DMA buffer list sizes and sampling rates.

Sampling Rate	DMA Buffer List Size			
	500	1000	1500	2000
125 SPS	8.5 µA	8.3 µA	8.0 µA	8.2 µA
250 SPS	13.3 µA	12.9 µA	12.7 µA	13.9 µA
500 SPS	21.9 µA	21.7 µA	21.9 µA	22.6 µA
1000 SPS	37.2 µA	38.0 µA	38.2 µA	38.1 µA

algorithms were running on the CPU, for the power numbers reported in Tables 1, 2, and 3. It is intuitive to say that processing the whole DMA buffer after data acquisition will result in higher current consumption at lower DMA buffer list sizes. On the other hand, a higher buffer list size will result in considerable RAM usage besides an increased delay in the processing outcome. Therefore, with the above considerations, a buffer size of 1000 samples $\times 9$ bytes/sample is chosen.

It could also be observed that there is a non-linear increase in current consumption due to data acquisition when the sampling rate increases, due to the frequent engagement of the peripherals and CPU. But lowering the sampling rate will result in loss of data which may influence the accuracy of QRS detection algorithms.

IV. COMPARISON AND OPTIMISATION OF DATA PROCESSING

Nine QRS detection algorithms based on amplitude thresholding & first derivative, first & second derivative, Okada, Pan-Tompkins, Hamilton-Tompkins, Hilbert transform, mathematical morphology, linear prediction, and digital filters have been reproduced using MATLAB to the best of our understanding and implemented on nRF52832 SoC, which helps in obtaining the relative performance and power consumption at different sampling rates for each of the nine algorithms. Together with parameter choice and time delay, they are summarised in Table 4, which will be scrutinised in detail in later sections.

A. DISCUSSIONS ON CHOICE OF SAMPLING RATE

To visualise how the sampling rate of ECG signals could possibly influence the sensitivity of QRS detection algorithms in general, the sensitivity measurements of each algorithm presented in Table 4 are summarised into the two-line charts below in Fig. 3.

From observation, most of the algorithm line graphs peak at 250 SPS, indicating that the number of QRS peaks correctly identified by them reaches its maximum when the ECG data is being sampled at 250 SPS. There are exceptions when algorithms based on amplitude thresholding & first derivative, first & second derivative, and linear prediction are employed. While the algorithm based on the first & second derivative has the best sensitivity at 500 SPS, the rest two works best at 1000 SPS.

TABLE 4. Measurements summary of different QRS detection algorithms.

Algorithm	Sampling Rate (SPS)	Sensitivity (%)	+ve Predictivity (%)	Current (µA)	Parameter	Delay
Amplitude Thresholding & First Derivative [3]	125	86.7	90.2	15.2	β	2
	250	87.4	89.5	27.4		
	500	88.1	88.9	50.5		
	1000	88.2	82.5	95.7		
First & Second Derivative [1] [18]	125	71.2	93.4	10.0	primary β, secondary β	2
	250	91.6	97.3	18.5		
	500	96.3	94.2	32.9		
	1000	95.8	88.8	58.7		
Okada (M = 6) [19]	125	93.1	99.7	23.1	β, M	12
	250	95.4	99.7	43.2		
	500	88.4	97.8	84.1		
	1000	78.9	95.0	161.0		
Pan-Tompkins [20]	125	96.3	94.5	37.3	β, N	24 + N/2
	250	99.2	99.7	60.2		
	500	98.1	99.1	182.9		
	1000	98.2	95.6	474.0		
Hamilton-Tompkins [21]	125	94.6	94.7	40.0	β, N	24 + N/2
	250	99.4	99.7	86.4		
	500	98.7	99.2	203.5		
	1000	96.3	95.7	527.4		
Hilbert Transform [22] [23]	125	98.5	99.1	30.1	β, Kaiser Window Size (M)	M/2
	250	98.9	98.7	55.4		
	500	98.9	97.2	108.0		
	1000	98.7	91.5	210.5		
Mathematical Morphology [24]	125	97.4	91.7	39.7	β, structure element	32
	250	98.2	99.1	77.3		
	500	97.3	96.3	150.5		
	1000	96.1	87.8	295.9		
Linear Prediction [25]	125	96.6	90.0	21.0	β, N	5 + N/2
	250	98.3	96.1	42.6		
	500	98.5	98.4	113.2		
	1000	98.6	97.3	311.7		
Digital Filters [26]	125	92.1	87.7	53.1	β, filter orders	45
	250	99.8	97.9	104.4		
	500	99.4	99.3	206.4		
	1000	99.3	96.8	401.3		

β = Threshold Ratio, M = Filter Window Size, N = Moving Average Window Size

Sensitivity is not the sole indicator of a particular algorithm's QRS detection performance. Before we proceed to analyse further, the effect of sampling rate on the positive predictivity of each of the nine algorithms is presented as. Fig. 4 visualises such a comparison, in a similar way.

Of the nine algorithms, five peaks at 250 SPS, two peaks at 125 SPS, and the other two peaks at 500 SPS. Among

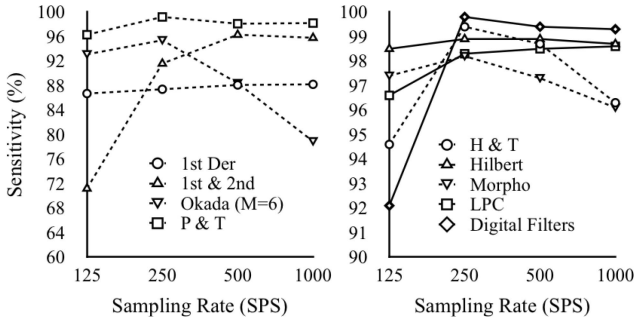

FIGURE 3. Sensitivity vs Sampling Rate of Different QRS Detection Algorithms.

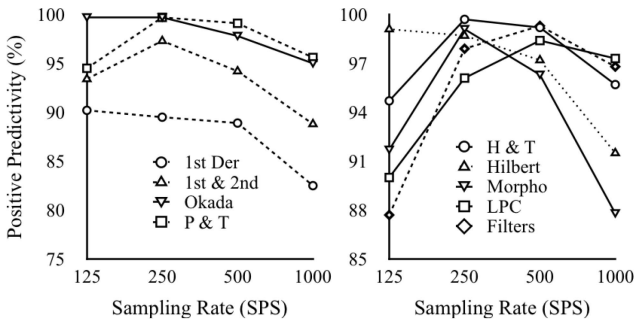

FIGURE 4. Positive Predictivity vs Sampling Rate of Different QRS Detection Algorithms.

the four algorithms whose best positive predictivity does not occur at 250 SPS, three of them demonstrate the second-best positive predictivity values at 250 SPS. Besides, six out of the nine algorithms have the best or second-best positive predictivity at 500 SPS, while only three have it at 125 SPS. The algorithm based on linear prediction is the only one that has the second-best positive predictivity at 1000 SPS.

Based on the measurements summarised in Table 4, power consumption surges as the sampling rate increases. Coupled with the observations from Fig. 3 and 4, sampling at 1000 SPS should almost always be avoided, as it has a low performance at significant power consumption.

QRS detection performances are comparable at 250 SPS and 500 SPS, with slightly better detection results observed at 250 SPS. Besides, sampling at 250 SPS consumes much less power as compared to the case at 500 SPS, which makes 250 SPS a more favourable sampling rate for implementation.

One would expect the QRS detection techniques to have higher detection performance at a higher sampling rate. There are two reasons for the reduction in detection performance at a higher sampling rate. Firstly, the various parameters in these algorithms are optimised by the respective authors with 360 Hz MIT-BIH database. These algorithm parameters have to be re-worked based on the sampling rate used for optimal performance. This study was focused on analysing the power efficiency of techniques presented, and therefore re-working the algorithm parameters, such as threshold values and filter parameters are considered out of scope for this work. Secondly, the accuracy numbers are calculated using MIT-BIH database re-sampled at different frequencies. Since this database is originally sampled at 360Hz, re-sampling it to higher speed doesn't add any new information in the data.

At 125 SPS, power consumption is at the lowest and most ideal scenario. However, choosing such a low sampling rate risks having an insufficient amount of data for an accurate ECG signal representation, which could potentially compromise an algorithm's QRS detection performance. Therefore, other than for systems designed for non-critical usage and with power consumption as its major constraint, sampling at 125 SPS is not recommended. The power numbers given in Table 4 excludes the power consumed by the data acquisition chip, ADS1292R. Due to the integration of multiple components and its implementation characteristics, ADS1292R consumed roughly the same current at different sampling rates.

B. DISCUSSIONS ON OPTIMISATION OF QRS DETECTION ALGORITHMS

Based on the comparative study done previously, sampling at 250 SPS delivers the most ideal QRS detection results at a reasonably low cost of current consumption, among the four sampling rates examined. Therefore, in this comparison, the characteristics of each individual algorithm is studied with the sampling rate fixed at 250 SPS.

From Table 4, it is observed that the QRS detection algorithm based on Hamilton-Tompkins method, which is an improved version of Pan-Tompkins', achieves sensitivity and positive predictivity of over 99%, ranking first in terms of feature extraction performance at 250 SPS among all nine algorithms. Its performance indicator values are slightly higher than its predecessor's, but so does the current consumption. The delay introduced by these two algorithms, though, is relatively high as compared to their counterparts.

It is also observed that both Hilbert transform based algorithm and linear prediction based algorithms manage to deliver reasonably good detection measurements (considering non-representative strange abnormalities exists in the MIT-BIH database at times), with lower current consumption. One characteristic that the two algorithms share is that they both collect data into segments of certain lengths and only conduct data processing by batches, as opposed to the implementation of other algorithms which processes by data points. It could be possible that processing by batch would be more power-efficient, but this needs further investigation.

Compared to the current consumed by the detection algorithm based on amplitude thresholding & first derivative, the one based on first & second derivative is observed to draw less current, despite having a slightly more complicated enhancement stage. This is likely due to its less complicated QRS detection stage where only one conditional statement is required for each iteration. The same applies to algorithms based on Pan-Tompkins and Hamilton-Tompkins methods.

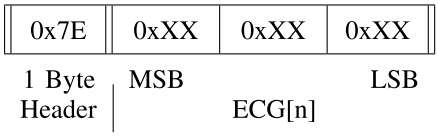

FIGURE 5. Data Format for ECG Signal Samples.

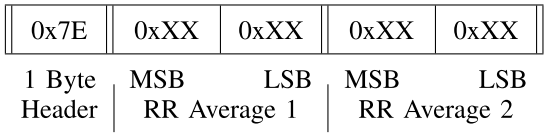

FIGURE 6. Data Format for RR Interval Averages.

The latter has a detection stage involving more comprehensive conditional statements, and thereby drawing higher current.

V. OPTIMISATION OF DATA TRANSMISSION

The data is wirelessly transmitted over BLE as it is a low energy wireless protocol. The hardware platform acts as GAP (Generic Access Profile) peripheral and GATT (Generic Attribute Profile) server whereas the BLE receiver is a GAP central and GATT client. To enable transmission of large amounts of data efficiently, a custom Nordic UART Service (NUS) is used. NUS is provided by Nordic Semiconductor and emulates a UART / serial port over BLE. NUS allows up to 244 bytes of useful data to be transmitted in a single packet over BLE. The PHY is set to 2 Mbps and the connection interval to 400 ms. The ECG data is acquired using SPI at a data rate of 8 Mbps, a sampling rate of 250 SPS and DMA buffer list size of 1000 employing DMA list approach. The ECG data is processed and/or sent every 4 seconds (= 1000 samples/250 SPS).

Firstly, the off-board processing case is simulated where the processing is performed only after receiving the data wirelessly. The whole ECG signals are sent directly over BLE without being processed, i.e., without running any QRS detection algorithm. Each ECG signal sample is a signed 24-bit number, so an ECG sample value along with a single byte header takes four bytes in a BLE packet as shown in Fig. 5. One BLE packet of 244 bytes contains 61 such samples. Thus, it takes 17 BLE packets with 16 packets of 244 bytes and the last packet of 96 bytes size to transmit 1000 samples of the ECG signal.

In the second scenario, ECG data is processed using Pan-Tompkins algorithm to detect peaks of the R waves and eventually the RR interval averages. In this case, only the RR interval averages are transmitted. Pan-Tompkins algorithm is chosen out of all the QRS detection algorithms as it achieves excellent performance without drawing considerable current at 250 SPS. In Pan-Tompkins algorithm, two RR interval averages are maintained; one is the average of the eight most recent beats and the other is the average of the eight most recent beats having RR intervals that fall within certain limits [20]. The data packet in this case consists of a single byte header and two bytes each of the two RR interval averages as shown in Fig. 6. Thus, it takes a single BLE packet of only 5 bytes of data to transmit the processed outcome.

Power overhead of these two transmission approaches with respect to BLE is given in Table 5.

TABLE 5. Current consumption and time taken with respect to BLE.

Scenario	Current	Time		
		Processing	Transmission	Total
No BLE	60.2 µA	25.9 ms	0	25.9 ms
ECG signal sent	91.9 µA	0	25.9 ms	25.9 ms
RR interval sent	73.1 µA	25.9 ms	2.0 ms	27.9 ms

Based on the values in Table 5, it takes 25.9 ms to process 1000 samples of ECG signals using Pan-Tompkins algorithm while consuming just 60.2 µA of current. Transmitting only the RR interval averages over BLE results in slight increase in the current consumption and time. On the contrary, when all the 1000 samples of ECG signals are transmitted over BLE, it consumes a considerable amount of current. Therefore, it is evident that on-sensor processing results in improved power efficiency of wearable sensor designs. The BLE transmission power is automatically controlled by the default firmware of nrf52 chipset. The transmitter and receiver boards were placed approximately one meter apart during the measurement. The power difference between transmission of ECG signal and RR interval would potentially be even greater for a higher BLE transmission power.

A plethora of works have been reported in literature regarding development wireless wearable sensors for health monitoring [27], [28], [29], [30], [31]. One of the main focus in existing works is to accurately record and display the data. In [27], a wearable sensor which can record seismocardiogram, single lead ECG is developed. TI ADS1291 and an accelerometer was used for collecting single lead ECG and activity data and an ATMEL micro-controller was used for data processing. There was no support for wireless transmission. The work focused on data collection and accuracy of recording and didn't report any power optimisation techniques. A low-power wearable ECG monitoring system for remote patient monitoring is reported in [28]. A TI ADS1246 chip is used as the ECG frontend and a Freescale MC13224 is used for Zigbee wireless transmission. A number of design optimisations including data compression is implemented for power reduction. The authors report a power consumption of 12 mW and 160 hours of lifetime for the sensor during continuous operation. In [29], a low power wrist worn ECG monitor is proposed. This design uses discrete op-amps for data acquisition, an ADX362 accelerometer and a TI CC2650 for BLE transmission. The system was optimised for low power by directing sensor data over BLE advertisement channel. For continuous ECG transmission at 128Hz,

the system consumes an average current of 9 mA which is way higher than what is measured in our work. A portable tele-ECG monitoring system is proposed in [30]. The design uses discrete op-amps for data acquisition and filtering and a TI CC2650 SoC for BLE transmission. The work focuses on establishing the accuracy of the measured data and reports a power consumption of 1.85 mA at 250 Hz sampling. In [31], an integrated wearable sensor platform for neonatal monitoring is reported. The work uses a graphene based flexible material for electrodes. A TI MSP430 micro-controller for data conversion and processing and TI CC2564 for BLE transmission. The design reports a battery life of 8 hours using a 600 mAH battery. It can be seen that the main focus in existing works is to accurately record and display the data. Compared to the existing literature on the topic which focus on overall system design and data accuracy, this work discussed and established several power optimisation strategies for IoT embedded wearable sensors.

VI. CONCLUSION

This paper investigates techniques and algorithms to reduce the power consumption of wearable ECG devices. We explore potential optimisations in the data acquisition, processing, and transmission stages. An ECG data acquisition using the DMA list approach is proposed, which reduces the current consumed during ECG data acquisition over SPI by a factor of three over the traditional DMA approach at 250 SPS. By comparison studies, it is noted that the higher SPI data rate results in more energy-efficient data transfer. Also, the non-linear increase in current consumption with increasing ECG signal sampling rate is affirmed. Besides, it is noted that the DMA buffer list size has no significant effect on power consumption.

The paper also conducts a thorough comparative study of nine selected QRS detection algorithms. The algorithms are implemented and compared with respect to sensitivity, positive predictivity, current consumption, parameter choice, delay, and potential deficiencies. Pan-Tompkins and Hamilton-Tompkins algorithms for QRS classification are found to have the best detection performance at a relatively low power consumption.

It is also noted that current consumption decreases at lower sampling rates. However, too low a sampling rate leads to insufficient ECG signal representation, which has a negative impact on the detection performance, as observed. A sampling rate of 250 SPS offers a good compromise between power consumption and QRS detection performance. It is also observed that algorithms that perform batch processing on a segment of data consume less power than those that perform feature detection at each data point, and that algorithms with a more complicated detection stage generally require higher power than those with more complex enhancement stage.

Finally, the effect of wireless transmission with and without data processing is studied. As expected, sending raw ECG signals over BLE is less efficient than sending the processed data. Transmitting the RR interval averages over BLE results in lesser current consumption as compared to transmitting raw ECG signals. It consumes only 73.1 μA to process 1000 samples of ECG signal at 250 SPS with SPI data rate of 8 Mbps and transmit the RR interval averages over BLE every 4 seconds.

The influence of benchmarks on the detection performance measurements of QRS detection algorithms remains unknown in this project, as MIT-BIH database is mainly used for performance evaluation. Further research needs to be conducted to investigate whether the relative performance of QRS detection algorithms would vary with respect to benchmarks.

The BLE parameters can be fine-tuned to achieve the best performance and efficiency. Further work can be done in compressing the data prior to transmission [32] to improve power efficiency and performance at the data transmission stage.

REFERENCES

[1] M. Elgendi, B. Eskofier, S. Dokos, and D. Abbott, "Revisiting QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems," *PLoS ONE*, vol. 9, no. 1, pp. 1–18, Jan. 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0084018

[2] G. S. Furno and W. J. Tompkins, "A learning filter for removing noise interference," *IEEE Trans. Biomed. Eng.*, vol. BME-30, no. 4, pp. 234–235, Apr. 1983.

[3] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 1, pp. 85–98, Jan. 1990.

[4] C. J. Deepu and Y. Lian, "A joint QRS detection and data compression scheme for wearable sensors," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 1, pp. 165–175, Jan. 2015.

[5] T. De Cooman, G. Goovaerts, C. Varon, D. Widjaja, and S. Van Huffel, "Heart beat detection in multimodal data using signal recognition and beat location estimation," in *Proc. Comput. Cardiol.*, Cambridge, MA, USA, 2014, pp. 257–260.

[6] B. S. Chandra, C. S. Sastry, and S. Jana, "Robust heartbeat detection from multimodal data via CNN-based generalizable information fusion," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 3, pp. 710–717, Mar. 2019.

[7] A. G. Ramakrishnan, A. P. Prathosh, and T. V. Ananthapadmanabha, "Threshold-independent QRS detection using the dynamic plosion index," *IEEE Signal Process. Lett.*, vol. 21, no. 5, pp. 554–558, May 2014.

[8] D. Pandit, L. Zhang, C. Liu, S. Chattopadhyay, N. Aslam, and C. P. Lim, "A lightweight QRS detector for single lead ECG signals using a max-min difference algorithm," *Comput. Methods Programs Biomed.*, vol. 144, pp. 61–75, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169260716302735

[9] T. Chanwimalueang, W. von Rosenberg, and D. P. Mandic, "Enabling R-peak detection in wearable ECG: Combining matched filtering and hilbert transform," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Singapore, 2015, pp. 134–138.

[10] C. Liu *et al.*, "Signal quality assessment and lightweight QRS detection for wearable ECG smartvest system," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1363–1374, Apr. 2019.

[11] J. M. Bote, J. Recas, F. Rincón, D. Atienza, and R. Hermida, "A modular low-complexity ECG delineation algorithm for real-time embedded systems," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 2, pp. 429–441, Mar. 2018.

[12] C. Ieong *et al.*, "A 0.83-μw QRS detection processor using quadratic spline wavelet transform for wireless ECG acquisition in 0.35-μm CMOS," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 6, pp. 586–595, Dec. 2012.

[13] F. N. Najm, "Power estimation techniques for integrated circuits," in *Proc. IEEE Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 1995, pp. 492–499.
[14] C. J. Deepu, X. Y. Xu, D. L. T. Wong, C. H. Heng, and Y. Lian, "A 2.3 μW ECG-on-chip for wireless wearable sensors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 10, pp. 1385–1389, Oct. 2018.
[15] C. J. Deepu, X. Zhang, W. Liew, D. L. T. Wong, and Y. Lian, "An ECG-on-chip with 535 nW/channel integrated lossless data compressor for wireless sensors," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2435–2448, Nov. 2014.
[16] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.
[17] A. L. Goldberger *et al.*, "PhysioBank, physioToolkit, and physioNet," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215
[18] R. Kher, D. Vala, T. Pawar, and V. K. Thakar, "Implementation of derivative based QRS complex detection methods," in *Proc. 3rd Int. Conf. Biomed. Eng. Informat.*, vol. 3. Yantai, China, Oct. 2010, pp. 927–931.
[19] M. Okada, "A digital filter for the QRS complex detection," *IEEE Trans. Biomed. Eng.*, vol. BME-26, no. 12, pp. 700–703, Dec. 1979.
[20] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.
[21] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH Arrhythmia database," *IEEE Trans. Biomed. Eng.*, vol. BME-33, no. 12, pp. 1157–1165, Dec. 1986.
[22] M.-E. Nygårds and L. Sörnmo, "Delineation of the QRS complex using the envelope of the ecg," *Med. Biol. Eng. Comput.*, vol. 21, no. 5, pp. 538–547, 1983.
[23] N. M. Arzeno, Z. Deng, and C. Poon, "Analysis of first-derivative based QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 2, pp. 478–484, Feb. 2008.
[24] P. E. Trahanias, "An approach to QRS complex detection using mathematical morphology," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 2, pp. 201–205, Feb. 1993.
[25] Z. E. H. Slimane and F. B. Reguig, "Detection of the QRS complex by linear prediction," *J. Med. Eng. Technol.*, vol. 30, no. 3, pp. 134–138, 2006. [Online]. Available: https://doi.org/10.1080/03091900500286468
[26] C. J. Deepu, X. Zhang, C. H. Heng, and Y. Lian, "A 3-lead ECG-on-chip with QRS detection and lossless compression for wireless sensors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 12, pp. 1151–1155, Dec. 2016.
[27] M. Etemadi, O. T. Inan, J. A. Heller, S. Hersek, L. Klein, and S. Roy, "A wearable patch to enable long-term monitoring of environmental, activity and hemodynamics variables," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 2, pp. 280–288, Apr. 2016.
[28] E. Spanò, S. Di Pascoli, and G. Iannaccone, "Low-power wearable ECG monitoring system for multiple-patient remote monitoring," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5452–5462, Jul. 2016.
[29] C. Beach *et al.*, "An ultra low power personalizable wrist worn ECG monitor integrated with IoT infrastructure," *IEEE Access*, vol. 6, pp. 44010–44021, 2018.
[30] H. Ozkan, O. Ozhan, Y. Karadana, M. Gulcu, S. Macit, and F. Husain, "A portable wearable tele-ECG monitoring system," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 173–182, Jan. 2020.
[31] H. Chen *et al.*, "Design of an integrated wearable multi-sensor platform based on flexible materials for neonatal monitoring," *IEEE Access*, vol. 8, pp. 23732–23747, 2020.
[32] C. J. Deepu and Y. Lian, "A low complexity lossless compression scheme for wearable ECG sensors," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Singapore, 2015, pp. 449–453.