Event-Triggered Sensing for High-Quality and Low-Power Cardiovascular Monitoring Systems

Grégoire Surrel[1] , Tomás Teijeiro[1] , Matthieu Chevrier[2],

Amir Aminifar[1] , *Member, IEEE,* and David Atienza[1] , *Fellow, IEEE*

[1]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

[2]Texas Instruments Deutschland GmbH, Freising, Germany

In the context of wearable medical systems, resources are scarce while the performance requirements are high. Traditional sampling strategies create large amounts of data, which hinder the device's battery lifetime. Energy savings are possible when relying on an event-triggered strategy, following the brain example. In this paper, we explore the use of non-Nyquist sampling for cardiovascular monitoring systems, with an in-depth analysis of the performance of a knowledge-based adaptive sampling strategy. By reducing the average sampling rate from 360 Hz down to 13.6 Hz, we can increase the battery lifetime by 4× with a marginal impact on the accuracy of heart-rate analysis.

Index Terms—Non-Nyquist sampling, Event-triggered systems, Time to Digital Converter, TDC, Medical systems, Personalized healthcare, Electrocardiogram, ECG, Wearable systems, Online computation, Polygonal approximation

I. INTRODUCTION

SIGNAL sampling is the root of signal processing. Ever since digital signal processing has existed, uniform sampling has been the most common approach. However, recent work have shown that this is not always the optimal solution and alternatives are possible.

A. The Data Overflow Problem

As global health has improved, the World Health Organization evaluated that coronary (or ischemic) heart disease combined with stroke account for 26.6% deaths worldwide, motivating the need for long-term heart monitoring. Such a personalized healthcare is impossible without the existence of non-intrusive devices. Thus, low-power wearable cardiac monitoring systems are the future of global health. However, digitizing the electrocardiogram (ECG) is the most expensive process for such optimized systems [1]. The functional structure of wearable cardiovascular monitoring systems is shown in Figure 1, with different possibilities for signal acquisition, processing and wireless data transmission.

Like in many industries, most bio-medical applications rely on collecting data according to the Nyquist-Shannon theorem. Indeed, this theorem guarantees that all the frequencies contained in the signal are perfectly acquired provided that the uniform sampling frequency for the Voltage Analog-to-Digital Converter (V-ADC, commonly called ADC) is at least twice higher than the highest frequency in the signal. Conversely, an insufficient sampling rate creates ghost frequencies in the sampled signal due to the spectral folding phenomenon.

However, uniform sampling has two main issues. First, the sampling is done at constant rate, ignoring the signal's temporal properties. This can lead to over-provisioned sampling during certain intervals without high frequencies. Secondly, defining a high-enough sampling rate is not always possible because
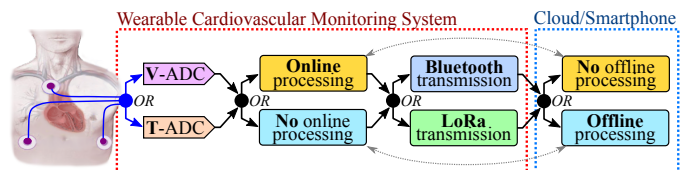
Figure 1: Structure of the cardiovascular monitoring system, along with the final data sink on a smartphone or in the cloud. In this work, different options of signal acquisition (uniform or event-triggered sampling), data processing (online in the sensor or offline on a remote device) and transmission (short-range Bluetooth Low-Energy (BLE) or mid-range LoRa) are considered. In total, eight scenarios are envisioned. The light-dotted arrows are ECG processing constraints: it needs to happen only once, either online or offline.

certain signals, e.g., sound waves, do not have definite frequency bounds.

The over-sampling effect of uniform sampling is critical in the context of resource-constrained medical systems [2], [3] because more data samples require more energy to process, store, or transmit the acquired data. Such wearable medical devices are extremely limited in terms of processing power, communication bandwidth, memory storage, and energy density. Once the device's software is optimized to handle the amount of data uniformly sampled with an efficient on-node processing, the signal digitization becomes the main energy expenditure [1], [2]. Changing the sampling paradigm is a solution for better energy efficiency and, *in fine*, longer battery lifetime for end-users.

B. An Old-New Hope for Efficient Data Sampling

From the *Animalia* kingdom, the brain is an event-triggered system, down to it's smallest component. Neurons react to action potentials they receive, because this is an energy-efficient approach. However, event-triggering is not exclusive to biological systems. In the computing world, external interrupt
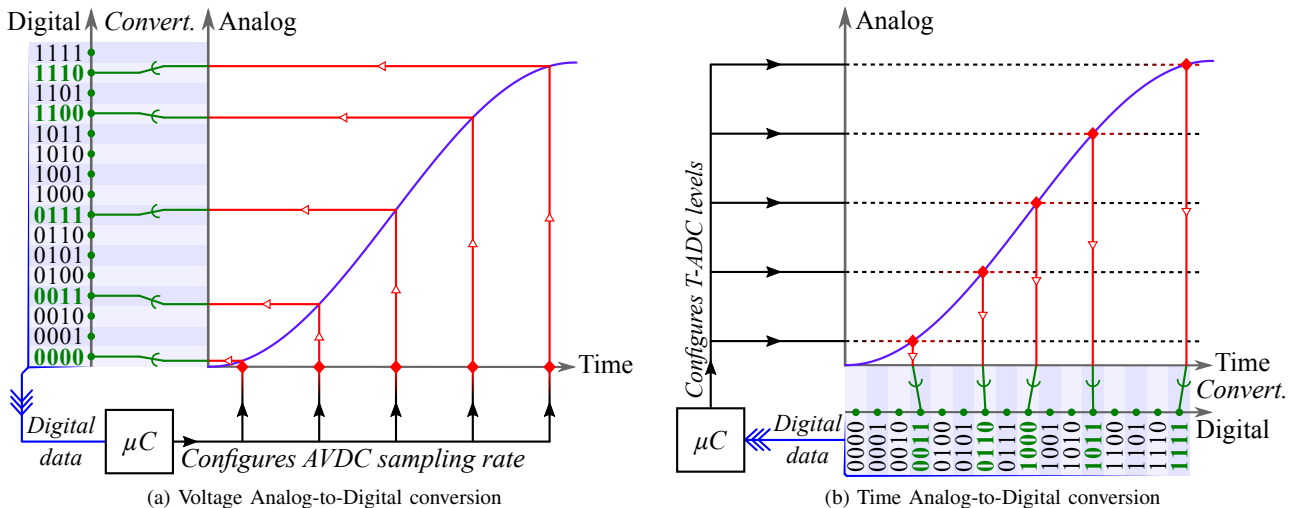
Figure 2: Comparison of the working principle of the usual V-ADC (2a) and the event-triggered T-ADC (2b).
As a first step, the micro-controller μC configures the chip responsible for the analog-to-digital conversion, visible as a **black arrows** ➝. In the case of both V-ADC and T-ADC, the **red lozenges** ◆ shows the trigger of the measurement. In the V-ADC case, the trigger happens on regular time intervals, whereas in the T-ADC case, the signal itself triggers the sampling by crossing the configured thresholds. The measured value is visible following the **red arrows** ⟶▷ until they reach the axis. Whether it is voltage (V-ADC) or time (T-ADC), this analog value needs to be digitized, using a given number of bits, which establishes the resolution of the measure. This conversion is showed with the **green arrows** ⟶. It illustrates the origin of digital noise, making visible the rounding of each sample towards discrete digital values, which are represented as **green dots** ●. Finally, each sample's **binary value** (in green) is returned to the micro-controller μC following the **blue arrows** ⟫.

signals on micro-controllers/processors are mimicking this approach.

However, less have been done in the context of event-driven sensing. In particular, constant signals do not need new samples as the absence of samples carry the information about the signal being invariant. Hence, new data points are required only if the signal is changing, whether the frequency is high or low. Time Analog-to-Digital Converters (T-ADC), or also called in the literature as Time to Digital Convertor (TDC), measure time between two events, such as the output of a multi-threshold comparator in case of a level-crossing approach. Rather than sampling the signal value with regular time intervals (Figure 2a), level-crossing T-ADCs sample the time needed for the signal to change from one value to another (Figure 2b). When comparing V-ADCs and T-ADCs in Figure 2, there is a fundamental change in the process. It is, however, not easy to claim that one solution outperforms the other, as the performance of each sampling scheme is signal-dependent.

C. Contributions

In the context of non-intrusive health monitoring, wearable medical devices are defining the concept of personalized medicine. Such devices must be both very efficient and not cumbersome to use, and therefore provide a long battery life. Since signal digitization becomes the main energy expenditure [1], [2], improving the sampling strategy is crucial. Targeting ECG-based devices, we present in this article event-triggered solutions for wearable medical systems.

As a high-level explanation, our simulations collect a new sample as soon as the error between signal and the linear interpolation of the previous measurements reaches a certain threshold. Lowering the error threshold refines the digitized signal and is a design parameter, in the same way a higher sampling frequency captures more information.

II. EVENT-TRIGGERED PARADIGM IN SENSING AND SAMPLING FOR ELECTROCARDIOGRAMS

Event-triggered sampling covers multiple approaches. In the following subsections, we explain these different paradigms applied to the specific case of ECGs.

ECG signals combine periods of high frequency when the beat happens, and lower frequencies otherwise. Each heartbeat in an ECG is observed as a sequence of three wave components (annotated in Figure 3):

1) P wave: electrical activation of the atria,
2) QRS complex: electrical activation of the ventricles,
3) T wave: electrical recovery of the ventricles.

A. Level-Crossing Event Triggering

When comparing different sampling strategies, we consider the distance between the linearly interpolated signal and the source signal. In the two cases illustrated in Figure 3, we highlight the trade-offs of both solutions. On one hand, for the traditional V-ADC, a low sampling rate can miss the main R peak from the QRS complex. On the other hand, the T-ADC measures an accurate timing of the R peak, while accumulating error on the smaller T wave.

Compressed sensing [4] helps lower the average sampling rate relying on the signal's sparsity. This technique is based on the use of a sensing matrix transforming the original signal into an underdetermined linear system of lower dimensionality.

(a) Error for Voltage Analog-to-Digital conversion
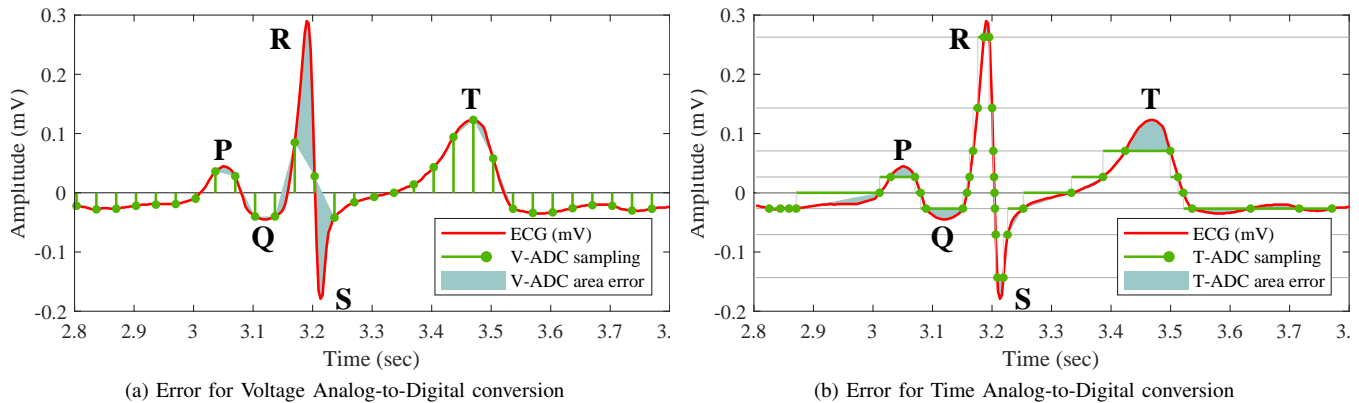
(b) Error for Time Analog-to-Digital conversion

Figure 3: Defining the reconstruction error between the classical V-ADC (3a) and the event-triggered T-ADC (3b). Both the V-ADC and T-ADC scenarios show the same number of samples. Due to the low sampling frequency, the V-ADC signal is missing almost completely the R peak from the QRS complex, whereas the T-ADC, relying on an exponential scaling of levels, captures the most important feature. [Source: Record A00848 from the PhysioNet Challenge 2007 database, between $t = 2.8s$ and $t = 3.8s$.]

Then, the reconstruction step implies finding solutions of this system.

The main counterparts of Compressed Sensing are two. First, the computational complexity of the reconstruction, hindering its application to low-power systems. Second, the sampling decision is taken in advance, which leads to some statistical loss of relevant values. Understandably, irregular heart beats linked to pathologies can be missed. Using event-triggered strategies brings the benefits of lower sampling rate without the risk of speculation.

Even though the ECG contains a lot of information, full details about the P, QRS, and T waves is optional. Depending on the application, desired accuracy and use-case, partial data is enough. For instance, Obstructive Sleep Apnea (OSA) can be detected relying heart beat timings [2]. The accuracy is improved when the peaks' amplitude is also used. Because all the processing is performed on the device, it must save as much energy as possible.

This aforementioned OSA detection uses a V-ADC, thus lowering the sampling frequency for coarse acquisition, two problems arise. First, it is impossible significantly reduce the sampling frequency as we need to accurately detect heart-beats. Secondly, the algorithm needs accurate timing between the heart-beats, otherwise the quality of the results decrease dramatically. Lowering the sampling frequency lowers the temporal resolution of the heart-beat detection. As a consequence, any energy saved from lowering the sampling frequency is paid with a lowered performance.

However, switching to a T-ADC signal acquisition process is beneficial due to two reasons. First, it becomes difficult to miss the heart-beat as it is the highest peak in the signal. It will quickly cross multiple thresholds, clearly flagging its presence in the triggers received. Secondly, even with a coarse configuration (ie. low number of thresholds), we only lose precision in the peak height but the time of when the heart-beat happened is preserved.

B. Error-Based Event Triggering

Putting the focus on signal reconstruction error, event-triggered sampling is a minimization problem. In our case, we look for the minimum number of samples allowing us to obtain a digital representation of the analog signal sufficient for ECG processing.

A family of methods well suited for this problem is polygonal approximation, also called piece-wise linear representation or linear path simplification [5], [6]. These methods assume the input signal can be represented as a sequence of linear segments, and they apply different techniques to obtain the minimum number of segments satisfying some error criterion.

Within this family, one method especially suitable for sampling time series is the Wall-Danielsson algorithm [6]. This method has linear complexity. Moreover, it works on a data stream (online), and only needs one signal sample in advance to estimate the approximation error. Conversely, it cannot guarantee optimality neither in the number of points nor in the selected samples. This method follows a bottom-up approach, in which points are merged to the current linear segment until an error threshold is reached, and then a new segment is created. The error is measured as the area deviation between the original signal and the current segment, as illustrated in Figure 3.

This algorithm overcomes two main shortcomings of the classical level-crossing method. First, an almost constant signal oscillating near a level generates more events than required. Second, fast linear changes generate numerous events. With polygonal approximation, the number of samples is not affected by constant displacements of the signal level, and linear changes are always represented by just two samples, no matter the slope value. In this work we have used the Wall-Danielsson algorithm as the basis for a novel adaptive-sampling method for ECG signals, which is explained in Section II-C.

Higher order approximation functions (quadratic or cubic splines), are not suitable for this specific problem. The main reasons are the absence of linear methods for building the approximation, the complexity of the reconstruction procedure and the higher dependency on floating-point precision. Since

Figure 4: Event-triggered adaptive sampling of an ECG fragment using polygonal approximation. Top: Original signal, sampled at 360 Hz. Bottom: Resulting signal of the adaptive sampling method. The detection of a regular rhythm enables a substantial reduction of the sampling frequency by getting a much coarse representation of the signal. After a rhythm change (second vertical line), the sampling frequency is increased to allow getting the details of the abnormal area. [Source: Record 119 of the MIT-BIH Arrhythmia DB, lead *MLII*, between 17:10 and 17:24].

our final objective is not only data volume reduction, but mainly energy saving, we require methods not compensating the savings derived from a lower sampling rate with an increased computational cost.

C. Knowledge-Based Adaptive Sampling for ECG Signals

The physiological regularity observed in the ECG can be exploited to reduce the amount of sampled data. In particular, under a normal sinus rhythm situation, the same heartbeat pattern is repeated between 60 and 100 times per minute. Thus, if this situation is detected on a signal fragment, from there it would be enough to capture just the information needed to identify a change in the rhythm.

This idea is illustrated in Figure 4, showing a 24-second ECG segment. As long as we observe three regular P-QRS-T heartbeat patterns with a normal distance between them, we drastically reduce the detail of the signal, checking that the regularity is maintained. This results in a rougher signal, but detailed enough to observe the regular heartbeats at the expected time points. When an unexpected event breaks this regularity, the procedure is able to increase the detail and support a more precise analysis of the new situation.

The knowledge-based adaptive sampling pseudocode is available as Algorithm 1. The four main variables, sig', $llim$, $ulim$ and $thres$, correspond to the linearly interpolated subsampled signal, the lower and upper temporal limits for the next expected QRS complex according to the current rhythm, and the threshold for the polygonal approximation algorithm. Two constants, mRR and MRR represent the minimum and maximum possible RR interval.

Each cycle of the algorithm has four main steps: 1) obtain the signal up to $ulim$ (line 8), 2) subsample it with the current threshold and interpolate it (lines 8-9), 3) look for the next QRS complex (line 10), and 4) update the approximation threshold and the time limits for the next QRS complex (lines 12-32). The subsampling function is called WD(), and it is a direct implementation of the Wall-Danielsson algorithm [6]. It receives as input a signal and an error threshold, and returns the time and value of the selected points, that are subsequently interpolated by means of the LINTERP() function. The error threshold is updated as a fraction of the area of the largest wave of the observed QRS complex, which is estimated from its amplitude and duration (line 18). The delineation of the QRS complex and its constituent waves is made using the *qrsdel* algorithm [7] (line 16). After this, the time limits for the following QRS complex are updated considering the regularity of the observed rhythm (lines 24-30). The adopted definition of regularity has been taken from the *Construe* algorithm's implementation [8], which sets a maximum instant variation of the RR interval of a 20% or 200 ms (lines 28-30).

In order to avoid one-time increases that could lead to an excessive signal deformation, a conservative approach has been adopted for the updating of the polygonal approximation threshold. Thus, if the new threshold $nthres$ is lower than the previous one, the new one is immediately applied (line 20). However, if the new threshold is higher and the QRS was observed on time (between $llim$ and $ulim$), then the increase is applied in a 90%-10% fashion, requiring some time to get a large increase (line 22). The initial value selected for the threshold is 0.5 mm^2 in common ECG scale (25 mm/s, 10 mm/mV). According to the experimental results shown

Algorithm 1 Adaptive sampling algorithm. It takes as input an ECG signal sig and returns an event-triggered sampled version sig' obtained by polygonal approximation with dynamic thresholding. For compatibility with the QRS detector, sig' is linearly interpolated at the same frequency of sig.

```
 1: function ADAPTIVE_SAMPLE(sig)
 2:     const mRR = 0.28s, MRR = 2.4s
 3:     var sig' = []                    ▷ Resulting subsampled signal
 4:     var llim, ulim = 0, MRR
 5:     var thres = 0.5mm²
 6:     var beats = []                   ▷ Dynamically detected QRSs
 7:     while len(sig') < len(sig) do
 8:         x, y = WD(sig[len(sig') : ulim], thres)
 9:         segment = LINTERP(x, y)
10:         qrs = GQRS(segment)[0]
11:         sig' = CONCATENATE(sig', segment[: qrs])
12:         if qrs is null then
13:             thres = 0.5mm²
14:             ulim = 2 · ulim − beats[−1]
15:         else
16:             delineation = QRSDEL(qrs)
17:             w = MAX(delineation.waves)
18:             nthres = 0.25 · (w.amp · w.dur/2)
19:             if nthres < thres then
20:                 thres = nthres
21:             else if qrs ≥ llim then
22:                 thres = 0.9 · thres + 0.1 · nthres
23:             end if
24:             llim, ulim = qrs + mRR, qrs + MRR
25:             APPEND(beats, qrs)
26:             rr₀ = beats[−1] − beats[−2]
27:             rr₁ = beats[−2] − beats[−3]
28:             if |rr₀ − rr₁| ≤ 0.2s ∨ 0.8 ≤ rr₀/rr₁ ≤ 1.2 then
29:                 llim = qrs+MIN(0.8 · rr₀, rr₀ − 0.2s)
30:                 ulim = qrs+MAX(1.2 · rr₀, rr₀ + 0.2s)
31:             end if
32:         end if
33:     end while
34:     return sig'
35: end function
```


Figure 5: Evolution of the QRS detection error with the average sampling frequency and the area fraction used for threshold update.

below, a fraction of 0.25 for threshold update has shown a good compromise between the resulting sampling frequency and the QRS detection performance.

III. EXPERIMENTAL EVALUATION

To demonstrate the potential of this knowledge-based adaptive sampling approach for ECG signals, we performed a series of experiments comparing the performance of a standard QRS detection algorithm using different sampling methods. The details of these experiments are explained next.[1]

A. Dataset and Detection Algorithm

The dataset used is the MIT-BIH Arrhythmia database available on PhysioNet, which is widely used in the bibliography to evaluate QRS detection algorithms. This database contains 48 ECG records of 30 minutes duration sampled at 360 Hz. In our case, the event-triggered sampling procedure is single lead.

We decided to use the *MLII*, as it is the most repeated lead in the database. Since records 102 and 104 do not have this lead available, they were discarded for the experiments. Hence, the total run-time of the database is 23 hours.

The implementation of the *gqrs* algorithm used for QRS detection is from the WFDB software package from PhysioNet. This enables us to use a openly available and trustworthy resource for consistent performance assessment.

B. Compared Methods

Comparing each approach requires a systematic process. As a reference, we considered the performance using the 360 Hz uniform sampling (US) from the database. We then applied the same methodology for the three approaches we considered, namely:

- Level-Crossing (L.C.) Sampling: to convert the database to a level-crossing scheme, we considered a regularly spaced set of levels. Whenever the signal crossed a level, both the time and level value were reported. Finally, we benchmarked the performance of a linear interpolation resampled at 360 Hz.
- Compressed Sensing (C.S.): we compressed the database using the exact same approach as defined in [4], with a compression ratio of 50%. After decompression, we obtained the initial sampling frequency of 360 Hz, not requiring additional processing.
- Knowledge-Based (K.B.) Sampling: similarly as the level-crossing sampling, we obtained series of times and values. This approach also required a linear interpolation at 360 Hz to run the performance benchmark.

C. Performance Validation

For benchmarking the influence of the event-triggered sampling strategy on the QRS detection algorithm's performance, Algorithm 1 was applied to all records with the *MLII* lead. Then,

---

[1]With the aim of supporting reproducible research, the full source code of the algorithms to reproduce the experiments presented in this paper has been published under an Open Source License, along with the individual results: https://c4science.ch/diffusion/8502
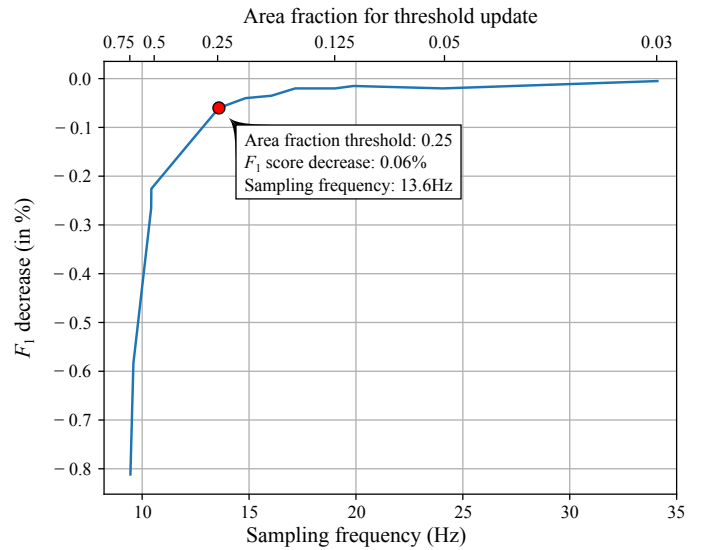
Sampling Strategy	Se	+P	F_1	fs (Hz)
Uniform Sampling (U.S.)	99.73	99.85	99.79	360.0
Compressed Sensing (C.S.)	99.64	99.82	99.73	180.0
Level-Crossing (L.C.)	99.66	99.83	99.74	43.7
Knowledge-Based (K.B.)	99.62	99.84	99.73	13.6

Table I: QRS detection performance comparison among different sampling strategies and resulting average sample rate for the 46 selected records from the MIT-BIH Arrhythmia DB.

Data Flow	Stream [J]		Online [J]	
Strategy (from Table I)	V-ADC U.S.	T-ADC L.C. / K.B.	V-ADC U.S.	T-ADC L.C. ≈ K.B.
ADC [9]	27.7	0.18	27.7	0.18
Sig. proc. [10]	-	-	23.9	23.9
BLE tx. [11]	56.7	11.8 / 9.94	7.06	7.06
LoRa tx. [12]	660	88.0 / 66.0	22.0	22.0
Total (BLE)	84.4	12.0 / **10.1**	58.7	31.2
Total (LoRa)	**687**	88.2 / 66.2	73.7	46.1

Table II: Model of the total energy consumption for the scenarios considered, with different sampling strategies (Uniform-Sampling V-ADC, Level-Crossing T-ADC, or Knowledge-Based T-ADC), signal processing (streamed to another device or local online analysis), and wireless communication protocols (BLE or LoRa). The values are reported for 23h of run-time.

the *gqrs* algorithm was run both on the original signal and in the resultant subsampled and interpolated signal. The results of both runs were then compared with the manual beat annotations contained in the .atr files using the *bxb* application from the WFDB toolkit, using the F_1 score as the performance metric.

Figure 5 shows how the overall F_1 and the average sampling frequency varies depending on the fraction constant we apply to update the approximation threshold (line 18 in Algorithm 1). As we can see, reducing the sampling frequency more than 10× (down to 34.10 Hz) has no noticeable performance penalty, and reducing it up to 37× (9.59 Hz) has less than 1% of F_1 decrease. From this curve, we select a constant fraction of 0.25 as a good compromise between a 26× reduction of the sampling frequency (13.60 Hz) and only a 0.06% of F_1 penalty. This is an acceptable performance decrease to qualify for medical certification [4], while lowering the most the sampling frequency. In this scheme, each sample is larger, since it has to encode both the time and amplitude of the signal.

Then, Table I shows a performance comparison between the proposed method and the other sampling strategies, including ordinary uniform sampling, compressed sensing and level-crossing. The considered performance metrics are specificity, positive predictivity and the combined F_1 score. The compressed sensing method has been applied as explained in [4], while the adopted level-crossing scheme is linear with a threshold every 200 µV. The results show that for a similar F_1 score, compressed-sensing halves the sampling frequency while level-crossing divides it by more than eight. The knowledge-based approach outperforms the two other strategies. Because of the simpler model of level-crossing for hardware implementation, it must be considered.

D. Energy Consumption for Wearable Cardiac Monitoring Systems

We modeled the consumption of a full cardiac monitoring system active for 23 hours to match the database considered in Section III-A. As illustrated in Figure 1, we studied multiple approaches to sampling, processing and transmitting data. All our results in Table II consider V-ADC or T-ADC (level-crossing or knowledge-based) sampling, embedded online processing or remote offline processing, and finally BLE or LoRa for wireless data transmission.

In all cases considered, the medium-range low-power LoRa wireless communication is higher-power than BLE, even though it becomes very competitive with smaller payloads. This is

therefore a solution to consider when no Bluetooth-enabled smartphone is available.

Focusing on the data acquisition and processing, the most straightforward approach of streaming the V-ADC sampled ECG signal is the least efficient. Our results match the conclusions in [1], where ECG streaming consumes more than on-device delineation. However, a paradigm shift occurs with state-of-the-art T-ADCs devices. Indeed, the much lower sampling frequency enables savings both at the sampling level and at the transmission level. Paying attention to the results, the L.C. T-ADC is very close to the most advanced K.B. T-ADC, even though the sampling frequency is higher (43.7 Hz vs 13.6 Hz). This is because in the K.B. strategy, each sample has 10 bit for time and 12 bits for voltage, whereas the L.C. strategy only requires one bit to encode the direction of the signal change as the levels are known. Therefore, a direct streaming of the T-ADC values for remote processing is the most efficient solution. From these results, we envision devices with a battery lifetime multiplied by at least four.

IV. CONCLUSION

In this work, we demonstrated that the sampling rate required to perform a widespread task in current wearable medical systems (heart-rate monitoring) can be reduced between one and two orders of magnitude using an event-triggered sampling strategy, which retaining the same performance as a compressed-sensing approach with 50% compression ratio. Hence, the observed magnitude of the difference allows for optimism, with high energy-savings for a hardware implementation, along with a lowered processing and storage for micro-controllers thanks to a lower amount of data to process. When considering full systems, such as wearable cardiac monitoring devices, further developing and the use of T-ADCs opens the way to a promising future of long-lasting solutions for a non-intrusive and personalized healthcare.

References

[1] F. Rincón et al. Development and evaluation of multilead wavelet-based ecg delineation algorithms for embedded wireless sensor nodes. *IEEE TITB*, 15(6):854–863, 2011.

[2] G. Surrel et al. Online obstructive sleep apnea detection on medical wearable sensors. *IEEE TBioCaS*, 12(4):762–773, 2018.

[3] D. Sopic et al. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE TBioCaS*, (99):1–11, 2018.

[4] H. Mamaghanian et al. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE TBME*, 58(9):2456–2466, 2011.

[5] E. Keogh et al. Segmenting time series: A survey and novel approach. In *Data mining in time series databases*, pp. 1–21. World Scientific, 2004.

[6] K. Wall and P. E. Danielsson. A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, & Image Processing*, 28(2):220–227, nov 1984.

[7] T. Teijeiro et al. A Noise Robust QRS Delineation Method Based on Path Simplification. *cinc.org*, 42:209–212, 2015.

[8] T. Teijeiro et al. Heartbeat Classification Using Abstract Features from the Abductive Interpretation of the ECG. *IEEE JBHI*, 22(2):409–420, nov 2018.

[9] G. Rovere et al. A 2.2-μw cognitive always-on wake-up circuit for event-driven duty-cycling of iot sensor nodes. *IEEE JESTCaS*, 8(3):543–554, 2018.

[10] S. Murali et al. A wearable device for physical and emotional health monitoring. In *2015 Computing in Cardiology Conference (CinC)*, pp. 121–124. IEEE, 2015.

[11] Online power profiler - nordic devzone.

[12] M. N. Ochoa et al. Evaluating lora energy efficiency for adaptive networks: From star to mesh topologies. In *IEEE WiMob*, pp. 1–8, 2017.