EPFL

Thèse n° 8052

System-level Design of Adaptive Wearable Sensors for Health and Wellness Monitoring

Présentée le 21 mai 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire des systèmes embarqués
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Elisabetta DE GIOVANNI

Acceptée sur proposition du jury

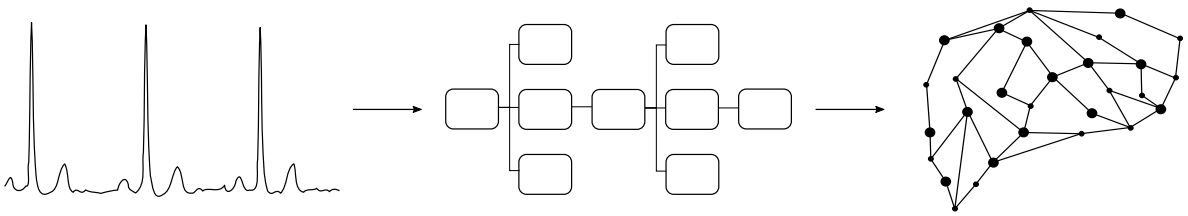
Prof. K. Aminian, président du jury
Prof. D. Atienza Alonso, directeur de thèse
Prof. F. Leporati, rapporteur
Dr M. Lemay, rapporteur
Prof. S. Carrara, rapporteur

École polytechnique fédérale de Lausanne

2021

"Wit beyond measure is man's greatest treasure."
— Luna Lovegood

To me, first, and to the reader, second,



Acknowledgements

It is very difficult to acknowledge everybody who helped me get to this exact moment. So, I will try to be brief and thank as many people as possible and, if I forget to mention anybody, I apologize. I also apologize as I will occasionally change language to address directly the people I am thanking.

First of all, I want to thank my jury members, Prof. Kamiar Aminian, Prof. Sandro Carrara, Prof. Francesco Leporati and Dr. Mathieu Lemay, who read thoroughly this thesis. They helped me to improve it with their insightful comments and very interesting discussion we had during my private defense. I enjoyed being challenged and reminded how research is continuously growing, and the debate of one's hypothesis and theories with the community is the best part of it. Thank you!

Then, I want to thank my thesis director, Prof. David Atienza. During my years of PhD, I felt that the area of my brain dedicated to research was synchronized and working at the same wavelength as his. This made my PhD life easier as I rarely had troubles liking what I was working on because I was mostly working on what I like. I also have to thank him for encouraging me in helping younger students, presenting demos and projects for the lab as it made me like even more the topic. Through my PhD years and these challenging tasks, I have realized what research means for me: thinking outside the box and coming up with innovative solutions to share with the world.

Next, I have to thank my daily supervisors, or the postdocs, that by nature are never staying in one place for long but always leave a mark. In chronological order, I thank Franscisco Rincon and Srinivasan Murali who helped me during my master thesis and showed me how interesting the research in the lab could be. Then, I thank Amir Aminifar and Adriana Arza Valdes who followed me during the first steps of my PhD. Finally, I thank Miguel Peón Quirós and Tomás Teijeiro who followed me during my last year or two of PhD and really helped me find the last pieces and build the puzzle that was my thesis (with a little sprinkle of life and food discussions).

Don't think, though, that my PhD life was sunshine and rainbows. Whose is? I have had my share of doubts and existential crisis, as well as joyful moments. Fortunately, the lab, or better the people in it, helped me through it and I have to thank everybody for that. From the moment I stepped in, a new master student, shy and quiet, I was welcomed with the joy and powerful singing voice of the most eccentric figure I could imagine, Dionisije. Then, in that downstairs lab/lounge room, a quiet creature was sitting just beside me and I thought that he was the most focused and serious person I have seen, and I didn't want to spook him. Little that I know, Gregito was actually listening to some rock 'n roll. He is one of the best people to talk to about everything, and I learned so much from (bothered) him and still continue to learn (bother)! On another corner of the lab, another person was too quiet and I knew something was off. I remember guessing his age and being completely shocked in learning he was not as young as he looked. And that was Fabio. If there is somebody who reminds me of my family the most, it's him. There is nothing that he could not turn into a funny moment. Thanks for sharing a laugh and the occasional pain au chocolat with me. Time to time, there was also a weird russian guy who talked about politics and was doing lots of pull-ups. The lab is actually a tapestry of cultures and people and I really have to thank everybody for just making this journey better, from administrative assistants (Homeira), to IT people (Rodolphe, Mikael, John), to office mates (Ruben, Farnaz), to work mates (Benoit, Simone, Lara), to visiting students and postdocs (il Fabio salentino, Alberto, Federico), basically to everyone with whom I shared either a laugh, a drink, a walk (with dogs) or just a nice conversation (Arman, Ali, Renato, Halima, Andrew, Una, Joshua, Silvio, and everybody else that I cannot list just because it would take too

much, sorry!). Oh, and I cannot forget former lab mates who like green tea and metal (Pableras).

Outside of the lab, I could find a very nice group of people who revived the last couple of years. I am talking about la chorale de Lep's Go! La seule chose qui me manquait ici était la possibilité de chanter dans une chorale et lorsque je vous ai rencontrés, j'ai senti que je l'avais enfin trouvé. Chanter avec vous est l'une des choses les plus joyeuses que j'ai vécues et j'espère que nous continuerons jusqu'à ce que je sois là !

Going back in time, I would have never arrived where I am without my friends in Pavia. First, Elisa, whom I met the very first day of university. I am so glad she tapped me on the shoulder that day because I would have not survived a day without her. We have experienced the university journey together, and she did help me till the end with her brilliant comments on my presentation for the private defense even if she does not like my field at all! Grazie, Eli, per esserci sempre anche sei in capo al mondo (letteralmente!). In later years, I also met Adelina e Lea who taught me to relax and enjoy my life. Grazie! Moving from the university to the warmth of my apartment in Pavia, I need to thank from the bottom of my heart Paola. I think she is the closest thing to a sister I could get. And that's probably one of the few reasons leaving Pavia was very hard. She also introduced me to the magic world of Harry Potter and I couldn't thank her more for it. Grazie, Paola! Finally, an honorable mention has to go to the always present Matteo. L'onnipotente Matteo. Dal liceo all'università, è quasi confortevole sapere che esiste una persona, fuori dalla mia famiglia, che mi conosce da più tempo di tutti. E, a prescindere da quanto tempo passi, una conversazione con quella persona mi fa ridere e riflettere allo stesso tempo, un privilegio che non tutti hanno. Quindi, Grazie!

Going even further in time, but always present, I have to thank my family. All my family. And again, I apologize but this needs the intimacy of my mother tongue. È arrivato il momento di ringraziare la mia famiglia. TUTTA la mia famiglia. Chi mi conosce sa (e si confonde spesso) che la mia è una famiglia molto numerosa. E la parte migliore e di cui vado fiera è che siamo tutti vicini, anche se non fisicamente. Loredana, la mia madrina, sempre pronta a farmi ridere e a regalarmi un libro. Tutte le mie zie, zii e cugini dalla parte di mio papà (anche quelli che non sono piu' con noi), è sempre un piacere

passare del tempo con voi. Grazie ai miei nonni che non ci sono più ma che sono sempre nei miei pensieri e a mia nonna che ancora mi prepara le pittule per Natale. Grazie agli zii, alle zie e ai cugini da parte di mia mamma che hanno contribuito tutti in parte al mio essere qui, con ispirazione e supporto. Tra loro, Manuela ha ovviamente un posto particolare, essendo cresciute insieme tra archi fatti di rametti e fili di raso, trappole per lucertole e piante di finocchio selvatico. Grazie, Maggie.

Il concetto di famiglia, nel mio caso, ha due significati. C'è la famiglia grande che ho appena menzionato e, poi, la famiglia con un significato più profondo. Il fatto di aver scelto ingegneria deriva probabilmente dal fatto che mia mamma è la persona più pratica che conosco. Dall'altra parte è anche una delle persone più solari che esista e il motivo per cui i nostri battibecchi finiscono sempre in risate. Mio papà è l'artista della famiglia. È lui che mi ha trasmesso la passione per la musica e per il canto, e, in generale, per la creatività che, a mio avviso, è parte della ricerca. Entrambi mi hanno sempre lasciato libera di decidere il mio futuro ed è per questo che mi ritrovo qui, con un dottorato in ingegneria, ma con un pezzo di me sempre dedicato alla musica e alla letteratura. Grazie. E poi ci sono i miei due fratelli, compagni di risate continue, di videogiochi, qualche volta di wrestling. Essendo la seconda figlia, guardavo a mio fratello maggiore con ammirazione (soprattutto da piccola quando lo seguivo ovunque) e a mio fratello minore con protezione (soprattutto da piccolo quando lo seguivo ovunque). Senza di loro non sarei mai arrivata fino a qui, quindi grazie!

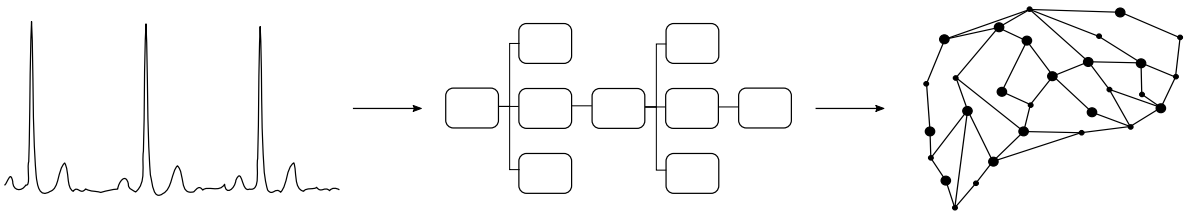
Finally, there is only one person left to thank. The one person who helped me the most going through my PhD, who was honest and caring, and who knows I don't need to write down here what I feel for him and how much I am grateful he's in my life.

So, this is the end and I leave you with one thought:

"For humanity to progress, research and improvement should never stop. However, at some point, you need to share it with the world to make at least the smallest positive impact."

— Me

Ecublens, April 30, 2021 Elisabetta De Giovanni



Abstract

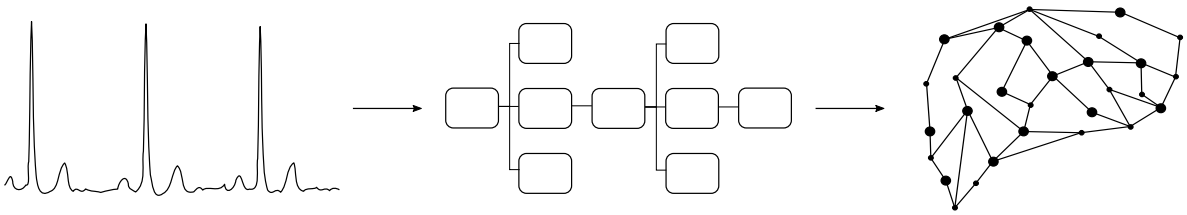
In recent years, population aging and the consequent higher incidence of noncommunicable diseases have increased the need for long-term health monitoring. Moreover, as healthcare cost is projected to grow substantially by 2030 in the Organisation for Economic Co-operation and Development (OECD) countries, the demand for portable, easy-to-use and low cost ultra-low power means of monitoring, diagnosis and prevention rises. Wearable sensors technology for remote health and wellness monitoring is an optimal candidate to tackle this problem and has advanced drastically in the last years. However, wearable sensors impose several design constraints. They must process data in real-time and provide highly accurate diagnosis and adapt to different cases. At the same time, to perform long-term monitoring, they must maximize battery lifetime, hence, usability. However, the need for more accurate algorithms and the need to obtain energy-efficient implementations can work against each other. For this reason, enhancing the energy-accuracy trade-off is essential.

Several works in the literature have addressed the energy-accuracy trade-off problem. The general approach is to first develop offline methodologies that maximize the algorithm's accuracy, using signal processing and machine learning. Then, these methods are optimized to be implemented as online designs in resource-constrained ultra-low power platforms. However, different problems can occur in these two steps. Most methods use highly variable datasets (mainly having different subjects); others use fixed parameters tailored to specific conditions, which overall decreases the robust-

ness of the algorithm. In traditional single-core devices, some optimizations whose goal is to lower the algorithms' complexity and computational burden, such as downsampling or features reduction, can lead to a loss in precision. With the advances of ultra-low power platforms and new machine learning strategies, even more challenges arise. However, these advances allow to exploit the growing capabilities of the platforms and use innovative and more complex strategies that achieve high levels of accuracy, robustness and energy-efficiency.

In this thesis, I propose a set of adaptive strategies in the context of remote health and wellness monitoring for an enhanced energy-accuracy trade-off in wearable sensors. First, I present three methodologies for multi-biosignal monitoring and pathology detection, which adapt to the specific physiological conditions by means of personalization to the subject and knowledge acquired from the signal. Second, in the context of modern heterogeneous wearable platforms, I propose a modular approach to software parallelization and hardware acceleration for biomedical applications to maximize the attainable speed-up and, therefore, minimizing energy consumption. Moreover, I propose an approach to scale computing resources and independent memory banks based on the specific characteristics of the patient in modern wearable sensors. Finally, in the context of intense physical exercise, I propose an online design that adapts to the sudden physiological changes occurring in the signal. This method combines a lightweight algorithm with a more robust though more complex one to reduce energy consumption while maintaining a very high accuracy. Moreover, this adaptive strategy exploits the heterogeneity of modern platforms by matching the complexity of each algorithm with the capabilities of each core, which further enhances the energy-accuracy trade-off.

Keywords: Adaptive Wearable Sensors, Personalized Healthcare, Multi-Biosignal Monitoring, Modular Design, Scalable Computation, Memory Management, Heterogeneous Processing Nodes, Parallel Computing, Energy-Accuracy Trade-Off, Green Internet of Things (IoT), Ultra-Low Power Computing



Résumé

Ces dernières années, le vieillissement de la population et l'incidence accrue des maladies non transmissibles qui en résulte ont accru la nécessité d'une surveillance de la santé à long terme. En outre, comme les coûts des soins de santé devraient augmenter considérablement d'ici 2030 dans les pays de l'Organisation de coopération et de développement économiques (OCDE), la demande de dispositifs de surveillance, de diagnostic et de prévention portatifs, peu contraignant, peu coûteux et à très faible consommation d'énergie augmente. La technologie des capteurs portatifs pour le suivi du bien-être et de l'état de santé est un candidat idéal pour s'attaquer à ce problème, d'autant plus que ce secteur a considérablement progressé ces dernières années.. Cependant, les dispositifs portatifs imposent plusieurs contraintes de conception. Ils doivent traiter les données en temps réel et fournir un diagnostic très précis tout en s'adaptant aux différentes situations. En même temps, pour effectuer un suivi de long terme, ils doivent maximiser la durée de vie de la batterie pour diminuer la contrainte qui incombe au patient de recharger l'appareil régulièrement. Cependant, la nécessité d'avoir des algorithmes performants et le besoin d'économies énergétiques peuvent aller à l'encontre l'un de l'autre. C'est pourquoi il est essentiel d'améliorer le compromis entre la précision énergétique et l'efficacité.

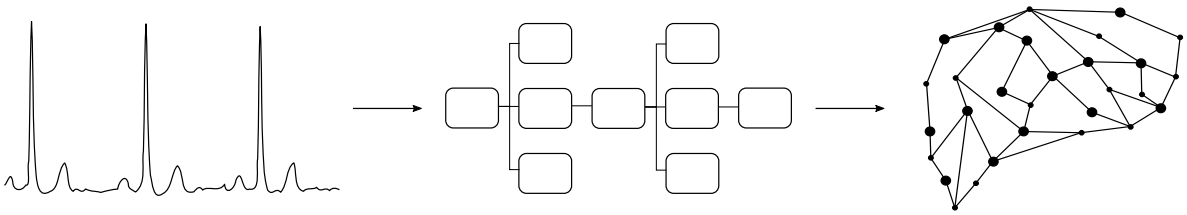
Plusieurs travaux dans la littérature ont abordé le problème du compromis énergie-précision. L'approche générale consiste à développer d'abord des méthodologies hors ligne qui maximisent la précision de l'algorithme, en utilisant le traitement du signal et l'apprentissage machine. Ensuite, ces mé-

thodes sont optimisées pour être mises en œuvre avec des données arrivant en flux, dans des plateformes à très faible consommation d'énergie et aux ressources limitées. Cependant, différents problèmes peuvent survenir au cours de ces deux étapes. La plupart des méthodes utilisent des ensembles de données très variables (utilisant principalement données des sujets différents) ; d'autres utilisent des paramètres fixes adaptés à des conditions spécifiques, ce qui diminue globalement la robustesse de l'algorithme. Dans les dispositifs monocœurs traditionnels, certaines optimisations dont le but est de réduire la complexité des algorithmes et la charge de calcul, comme le sous-échantillonnage ou la réduction des caractéristiques, peuvent entraîner une perte de précision. Avec les progrès des plateformes à très faible puissance et les nouvelles stratégies d'apprentissage machine, les défis sont encore plus nombreux. Toutefois, ces progrès permettent d'exploiter les capacités croissantes des plateformes et d'utiliser des stratégies innovantes et plus complexes qui atteignent des niveaux élevés de précision, de robustesse et d'efficacité énergétique.

Dans cette thèse, je propose un ensemble de stratégies adaptatives dans le contexte du suivi à distance du bien-être et de la santé avec un meilleur compromis énergie-précision dans les capteurs portatifs. Tout d'abord, je présente trois méthodologies de surveillance multi-bio-signaux et de détection de pathologies, qui s'adaptent aux conditions physiologiques spécifiques par le biais de la personnalisation individuelle et des connaissances acquises depuis les signaux. Deuxièmement, dans le contexte des plates-formes portables hétérogènes modernes, je propose une approche modulaire de la parallélisation logicielle et de l'accélération matérielle pour les applications biomédicales afin de maximiser l'accélération réalisable et, par conséquent, de minimiser la consommation d'énergie. De plus, je propose une approche pour le passage à l'échelle les ressources de calcul et les banques de mémoire indépendantes en fonction des caractéristiques spécifiques du patient dans les capteurs portatifs modernes. Enfin, dans le cadre d'un exercice physique intensif, je propose une conception qui s'adapte en continu aux changements physiologiques soudains qui se produisent. Cette méthode combine un algorithme léger mais moins précis avec un algorithme plus robuste mais plus complexe pour réduire la consommation d'énergie tout en maintenant une très grande précision. De plus, cette stratégie adaptative exploite l'hétérogé-

néité des plateformes modernes en faisant correspondre la complexité de chaque algorithme aux capacités de chaque nœud capteur, ce qui améliore encore le compromis énergie-précision.

Mots-clés : Capteurs Adaptatifs Portatifs, Soins de Santé Personnalisés, Surveillance Multi-Biosignaux, Conception Modulaire, Calcul Évolutif, Gestion de la Mémoire, Nœuds de Calcul Hétérogènes, Calcul Parallèle, Compromis entre Précision et Énergie, Internet des Objets (IdO) Vert, Calcul Économes en Énergie



Sommario

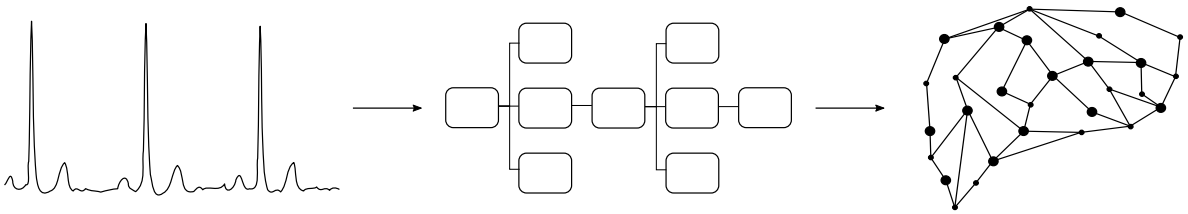
NEGLI ultimi anni, l'invecchiamento della popolazione e la conseguente maggiore incidenza di malattie non trasmissibili hanno aumentato la necessità di monitoraggio della salute a lungo termine. Inoltre, poiché si prevede che il costo dell'assistenza sanitaria crescerà sostanzialmente entro il 2030 nei paesi dell'Organizzazione per la cooperazione e lo sviluppo economico (OCSE), la domanda di mezzi di monitoraggio, diagnosi e prevenzione portatili, facili da usare, a basso costo e a bassissimo consumo aumenta. La tecnologia dei sensori indossabili per il monitoraggio remoto della salute e del benessere è un candidato ottimale per affrontare questo problema ed è avanzata drasticamente negli ultimi anni. Tuttavia, i sensori indossabili impongono diversi vincoli di progettazione. Devono elaborare i dati in tempo reale per fornire diagnosi altamente accurate e adattarsi a casi diversi. Allo stesso tempo, per eseguire il monitoraggio a lungo termine, devono massimizzare la durata della batteria, quindi l'usabilità. Tuttavia, la necessità di algoritmi più accurati e la necessità di ottenere implementazioni efficienti dal punto di vista energetico possono lavorare l'una contro l'altra. Per questo motivo, migliorare il trade-off energia-accuratezza è essenziale.

Diversi lavori in letteratura hanno affrontato il problema del trade-off energia-accuratezza. L'approccio generale è quello di sviluppare prima metodologie offline che massimizzano la precisione dell'algoritmo, utilizzando l'elaborazione del segnale e l'apprendimento automatico (i.e., machine learning). Poi, questi metodi sono ottimizzati per essere implementati online in piattaforme a bassissimo consumo con risorse limitate. Tuttavia, in queste due fasi pos-

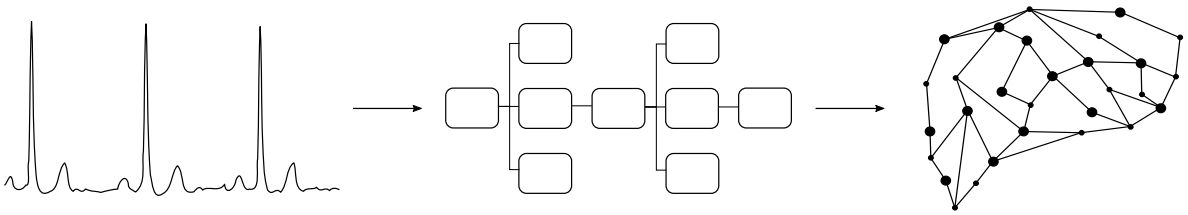
sono verificarsi diversi problemi. La maggior parte dei metodi utilizza set di dati altamente variabili (principalmente con soggetti diversi); altri utilizzano parametri fissi adattati a condizioni specifiche, che nel complesso diminuisce la robustezza dell'algoritmo. Nei tradizionali dispositivi single-core (i.e., un processore), alcune ottimizzazioni il cui obiettivo è quello di abbassare la complessità e il carico computazionale degli algoritmi, come il downsampling o la riduzione dimensionale delle caratteristiche nel machine learning, possono portare a una perdita di precisione. Con i progressi delle piattaforme a bassissimo consumo e le nuove strategie di apprendimento automatico, sorgono ancora più sfide. Tuttavia, questi progressi permettono di sfruttare le crescenti capacità delle piattaforme e utilizzare strategie innovative e più complesse che raggiungono alti livelli di precisione, robustezza ed efficienza energetica.

In questa tesi, propongo una serie di strategie adattive nel contesto del monitoraggio remoto della salute e del benessere per un migliore trade-off energia-accuratezza nei sensori indossabili. In primo luogo, presento tre metodologie per il monitoraggio di molteplici biosegnali e la rilevazione di patologie, che si adattano alle specifiche condizioni fisiologiche attraverso la personalizzazione al soggetto e la conoscenza acquisita dal segnale. In secondo luogo, nel contesto delle moderne piattaforme indossabili eterogenee, propongo un approccio modulare alla parallelizzazione software e all'accelerazione hardware per applicazioni biomediche con lo scopo di massimizzare la velocita' di computazione raggiungibile e, quindi, minimizzare il consumo di energia. Inoltre, propongo un approccio per scalare le risorse computazionali e i banchi di memoria indipendenti in base alle caratteristiche specifiche del paziente nei moderni sensori indossabili. Infine, nel contesto dell'esercizio fisico intenso, propongo un metodo online che si adatta ai cambiamenti fisiologici improvvisi che si verificano nel segnale. Questo metodo combina un algoritmo a basso carico computazionale con uno più robusto anche se più complesso per ridurre il consumo di energia mantenendo un'accuratezza molto elevata. Inoltre, questa strategia adattiva sfrutta l'eterogeneità delle piattaforme moderne facendo corrispondere la complessità di ogni algoritmo con le capacità di ogni processore, il che migliora ulteriormente il trade-off energia-accuratezza.

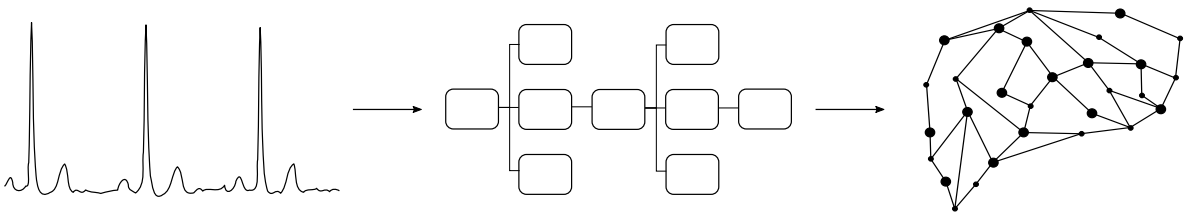
Parole chiave: Sensori Adattivi Indossabili, Assistenza Sanitaria Personalizzata, Monitoraggio Multi-Biosegnale, Design Modulare, Computazione Scalabile, Gestione della Memoria, Nodi di Elaborazione Eterogenei, Computazione in Parallelo, Trade-Off Energia-Accuratezza, Green Internet of Things (IoT), Calcolo a Bassissimo Consumo



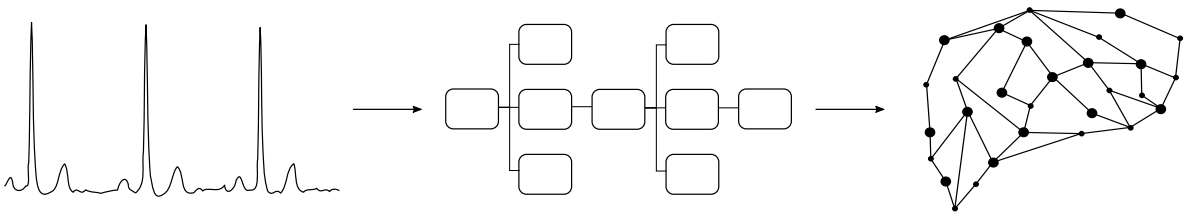
Contents



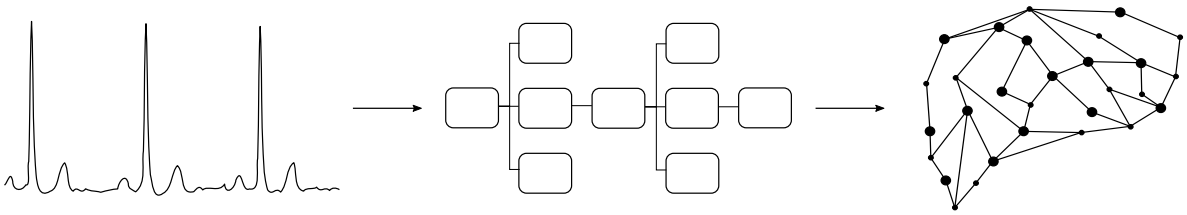
List of Figures



List of Tables



List of Algorithms



Acronyms

AAE Average Absolute Error. 47–49

AF atrial fibrillation. 10, 13, 52, 70, 74, 110, 118

BLE Bluetooth Low-Energy. 6, 68, 155–157

BPF band-pass filtering. 20, 29, 30, 32, 34

BPM beats per minute. 16, 17, 24, 36, 38, 39, 41–45, 49, 54, 58, 64, 109

CGRA coarse-grained reconfigurable array. 11, 72, 78, 80, 87, 93, 94, 98, 99, 103, 105, 116

CL cluster. 78, 79, 81–84, 87, 93, 98, 99, 102, 109, 110, 114, 131–135, 143, 146–148

CV cross-validation. 54, 58, 59

CVD cardiovascular disease. 1, 4, 13, 14, 18, 51, 120

DMA direct memory access. 72, 78, 79, 87–89, 91, 106, 131, 133

ECG electrocardiogram. 1–6, 9, 12, 14–16, 18–23, 26–28, 31, 32, 34, 35, 46, 47, 51–54, 56–58, 60–62, 64, 65, 67–70, 72, 73, 75–77, 79, 85, 88–90, 92, 95–97, 101–103, 106, 109, 116–118, 120, 122–128, 130, 132–134, 137, 138, 141, 147–150, 153, 155–157, 159, 161

Acronyms

EEG electroencephalography. 51, 75

FC fabric controller. 78, 82, 87, 91, 93, 98, 99, 102, 109, 110, 114, 131–135, 146, 147

FFT fast Fourier transform. 16, 36, 38, 39, 42, 45, 46, 69, 80, 151

FIR finite impulse response. 20, 39

HR heart rate. 5, 9, 13, 14, 16, 17, 24, 36, 38, 40–50, 54, 58, 64, 69, 70, 90, 109, 118, 122, 128, 148, 150, 151, 153, 156, 157

HRV heart rate variability. 3–5, 15, 120, 122, 147, 148, 152, 153, 156, 157

HW hardware. 6, 7, 10, 15, 16, 18, 28, 34, 50, 72, 77, 78, 83, 86, 87, 93, 94, 98, 103–105, 115, 150

ICG impedance cardiogram. 77

ICT information and communication technology. 2

IoT Internet of Things. 6, 74, 116

iPPG imaging photoplethysmography. 75

LED light emitting diode. 9, 16, 36, 38

LOO leave-one-out. 125, 126

MAs motion artifacts. 10, 16, 17, 36–44, 69, 70

MCU microcontroller unit. 22, 27, 32, 45, 47, 67, 156

MF morphological filtering. 20, 29–34, 55, 75, 89, 95, 99, 101, 114, 124, 132, 134

NCD noncommunicable disease. 1, 3, 4, 14, 51, 72, 76, 77, 149

PAF paroxysmal atrial fibrillation. 10, 11, 17, 51–54, 57–60, 62–65, 67–71, 73, 75, 104, 106, 109, 110, 116, 147, 148, 150, 152, 153

PPG photoplethysmography. 4, 5, 9, 10, 14, 16, 17, 35–47, 49, 50, 69, 77, 90, 150, 151, 153, 155–157, 159–161

PPV positive predictive value. 25, 30, 31, 67, 138, 139

PTT pulse transit time. 4

RC reconfigurable cell. 93, 94

Rel-En Relative-Energy. 9, 14, 16, 21–23, 28, 56, 67, 69, 75, 89, 95, 99, 100, 106, 107, 124, 128, 131, 134, 153

RMS root-mean-square. 75, 76, 91, 92, 95, 99, 100, 116

SCM standard cell memory. 93

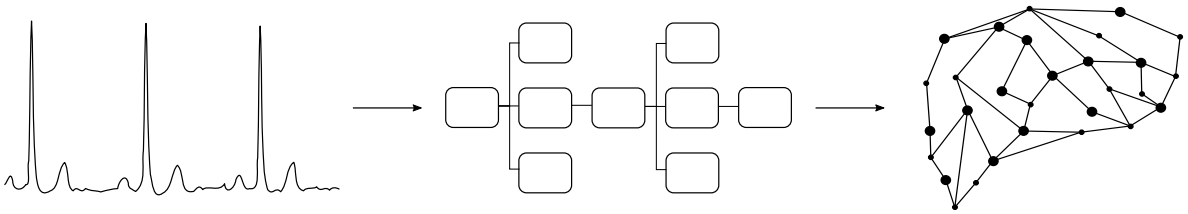
SVM support vector machine. 54, 58–60, 64, 70, 156, 157, 159

SW software. 10, 11, 65, 74, 77, 83, 86, 87, 98, 105, 150, 159

ULP ultra-low power. 2, 7, 8, 10–12, 14, 16–19, 21, 28, 45, 51, 65, 67, 69–74, 77, 78, 106, 110, 115–118, 147–150, 153

WSN wearable sensor node. 1–6, 9, 13, 14, 16, 17, 36, 45, 51, 52, 69–72, 74, 75, 77, 79, 80, 83, 88, 91–93, 104, 115–118, 123, 144

WT wavelet transform. 20, 21, 55, 56, 66



Introduction

INCREASING healthcare costs [1] and hospital overcrowding call for new technological advances that improve remote wellness monitoring and enable self-diagnosis, early intervention, and prevention [2]. In addition, population aging and the consequent higher incidence of noncommunicable diseases (NCDs) create the need for long-term health and wellness monitoring. Within NCDs, cardiovascular diseases (CVDs) in particular—which are characterized by abnormal events that need to be detected in real-time—are the major cause of death globally [3]. To prevent, predict and detect NCDs, there is an increasing need for automatic applications that continuously and remotely monitor biosignals, such as the electrocardiogram (ECG) [4], and extract relevant characteristics from them. Moreover, to prevent CVDs a daily physical activity is highly recommended [5]. However, movement contaminates the signal, which leads to a need for optimizing algorithms to reduce artifacts and improve inference accuracy for better daily monitoring. While the need for medically acceptable accuracy is of most priority when dealing with health and wellness monitoring, the energy consumption of the wearable sensor nodes (WSNs) employed for continuous remote monitoring is a parallel concern. In fact, a more extended battery life and, hence, improved usability, can lead to improved results in terms of accuracy. To achieve extended battery life and improve accuracy, one effective solution is adaptivity of algorithms and platforms by means of personalization to the subject, online multibiosignal-based knowledge acquisition, modular and scalable

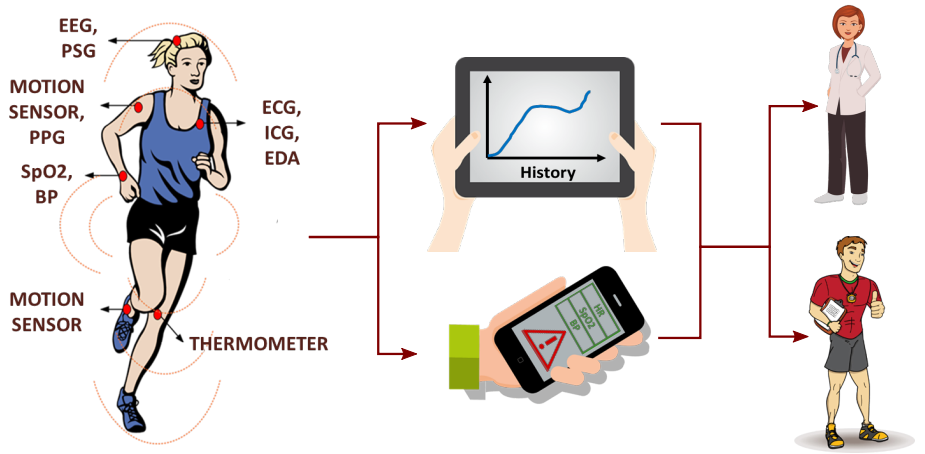


Figure 1.1 – Representation of a remote health and wellness monitoring system with different WSNs

optimizations. New adaptive approaches to optimize the energy-accuracy trade-off in modern ultra-low power (ULP) wearable devices in the context of continuous remote health and wellness monitoring is the center topic of this thesis discussion and contributions.

1.1 Remote Health and Wellness Monitoring

Telemedicine describes a set of healthcare services to provide diagnosis, treatment and prevention of disease to a physically distant individual using information and communication technologies (ICTs) [6]. The history of modern telemedicine as a means of diagnosis is fairly recent, with the transmission of an ECG in 1905 by Einthoven, who is also known for providing the standard model for ECG electrode placement [7]. He was able to combine his galvanometer, the first high-quality ECG machine, with a telephone and succeed in transmitting and receiving heart beats through it about 1.5 km away [8]. With the advent of modern communication technologies and specifically faster and wireless transmissions, a new era for remote patient monitoring started, and with it the introduction of wearable sensors. Fig. 1.1 shows a typical system for remote health and wellness monitoring with multiple wearable sensors. The purpose of the system is to gather physiological and biome-

chanical information, transmit it to a portable smart device, such a tablet or a phone, and make it available, usually through the cloud, to doctors and trainers. Then, they can evaluate the data and send feedback to the individual. The data processed by the wearable sensors and then transmitted to the phone can be either signals or more advanced diagnostics with a final goal of transmitting only an alarm in case of anomalies. No matter the configuration, the system is a revolutionary idea to solve the problems of distance and slow intervention, as well as to promote self-awareness of health and wellness.

Wearable non-invasive technology for telemedicine emerged with the first portable though bulky Holter ECG monitor in 1949 [9, 10]. The first prototype was an approximately 35 kg backpack transmitting the ECG via radio and it was tested while cycling on a stationary bike, a major breakthrough since most ECG devices at the time required the subject to lie still. Nowadays it is a miniaturized, battery-powered portable device that doctors give to the patient to continuously monitor their heart's activity for 24-48 hours recordings. Needless to say, the standard Holter monitor does not perform any complex processing but only stores and transmits the signal, though recent developments in long-term Holter monitors include ECG waves detection, heart rate variability (HRV) analysis, and even additional biosensors, such as an oxygen saturation (SpO2) sensor [11]. Nevertheless, the evolution of health and wellness wearable devices varies from standard ECG monitors to smart health and physical activity monitors with multiple biosignals and processing capabilities [12–14], smart clothing [15], and the recent stretchable strain sensors [16].

1.2 Energy-Accuracy Trade-Off in Modern Wearable Sensors

The main goal of this thesis is to propose more targeted, adaptive, and personalized solutions in the context of WSN-based biomedical applications. These innovative solutions aim at improving the detection and prevention of NCDs with the use of advanced computing, such as machine learning. However, with the development of new complex algorithms comes the question of constrained resources management in WSNs, and the consequent toll on

energy consumption. In this section, I explore how to maximize accuracy and minimize energy consumption for an enhanced energy-accuracy trade-off.

1.2.1 Maximizing Accuracy and Robustness

Remote patient monitoring through WSNs has the advantage of a more targeted diagnostic and personalized medicine [17,18]. In fact, the development of new algorithms for physiological parameters and pathology detection, specifically NCDs, increased significantly in the last years [19]. Some algorithms can detect vital parameters, such as blood pressure, which can be estimated by measuring a surrogate marker of it, pulse transit time (PTT) [20]. This can be extracted by combining relevant information from the ECG and the optical photoplethysmography (PPG) signal (i.e., the pulse) to estimate the time for the blood to transit from the heart to the PPG measurement location. Others apply more complex machine learning-based approaches to detect or classify different types of CVDs, such as arrhythmias [21–23]. Moreover, some works use the ECG information related to the autonomic nervous system (e.g., low frequency and HRV analysis), to detect different types of pathologies, such as obstructive sleep apnea and epilepsy [24,25]. These algorithms achieve good performances in terms of detection accuracy, however, it is in general far from what doctors can achieve, specifically in terms of personalized medicine. In fact, patient-specific approaches have been proven to increase the accuracy of detection methods by eliminating the variability factor across patients [23–29]. Moreover, some pathologies show sudden changes that are manifested in the biosignals morphology that are not often captured by standard algorithms, making them less robust. Therefore, there is a need for the detection algorithms to adapt to the physiology and characteristics of the patient—even though the validation process requires collecting more data from the same patient—as well as their pathology, at design and run time in remote health monitoring.

In the context of wellness monitoring in healthy subjects, most smart wearable sensors on the market focus on giving information about the intensity of training sessions and rely on anthropomorphic data and assumptions to estimate parameters [30]. Only recently, there was an increase to promote awareness of the advantages and limitations of consumer wearable

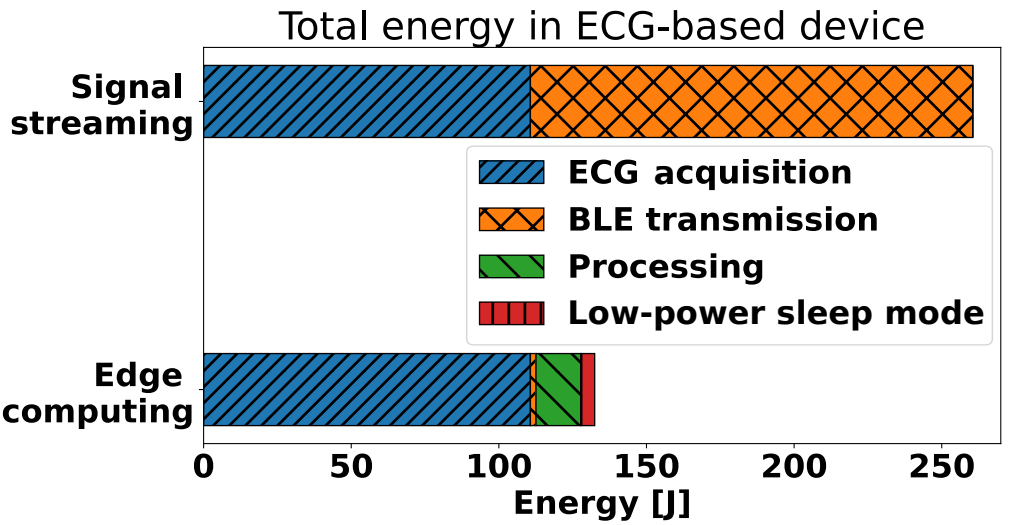


Figure 1.2 – Comparison of edge computing and signal streaming in terms of energy consumption for a well-known WSN-based biomedical application [38] in a real-life ECG-based device [24], divided in their main blocks of signal acquisition, transmission, low-power sleep mode and processing

devices and developing best-practice protocols for the evaluation of their validity [31]. Modern heart rate (HR) monitors, such as chest straps, can provide good or moderate reliable HRV analysis [32], which are helpful for fatigue detection [33] and estimation of oxygen uptake and VO_{2max} [34–36], as they affect the autonomous system. However, according to [31] only few chest straps were validated for reliable RR data, which is the basis for HRV analysis. PPG-based devices are also compelling WSNs, as they are low cost and more comfortable than electrode-based sensors. However, their parameter estimations or noise removal are not reliable for challenging conditions such as intense physical exercise [32, 37]. These analyses are subject-specific and are based on sudden changes in the HR series. Hence, there is a need for new algorithms that are personalized to the physiology of the subject and, more importantly, adapting their parameters in real-time to strive for medical standards of accuracy and robustness when deployed on WSNs.

1.2.2 Minimizing Energy Consumption

Advances in wearable technology for health and wellness monitoring have risen in parallel to the improvements in low power electronics [12], as more edge- and fog-based systems offload the data processing to the network edge [39]. Nowadays, edge computing is widely spread in the Internet of Things (IoT) domain for different reasons, from delays in transmission to offloading computational power, advantages in data security and privacy [39, 40]. In WSNs specifically, edge computing is preferable as signal transmission consumes more energy than processing an anomaly detection on the node. Fig. 1.2 shows an example comparison in terms of energy consumption between edge computing and continuously transmitting the signal via Bluetooth Low-Energy (BLE), in the context of a well-known wavelet-based ECG delineation algorithm [38] running for 24h on a Cortex-M3 core, considering the energy consumption numbers in [41]. As the figure shows, edge computing highly decreases the energy consumption, as it was demonstrated in previous works in the literature for different types of devices [38, 42, 43]. However, the energy consumed during signal acquisition is yet significant and edge computing inevitably has resource constraints that might affect the accuracy of biomedical applications.

To tackle the problem at the acquisition process, some works focus on its optimization at the hardware (HW) level. In fact, signal digitization is one of the main energy draining blocks in some biomedical applications [24, 38]. Therefore, event-triggered solutions have been explored to highly reduce the energy required by the sampling process [43]. These solutions abandon the paradigm of a constant sampling in time for one based on thresholds, or levels, reached by the signal, or abnormal events. For an ECG, this translates to acquiring only few samples belonging to its main waves (i.e., level crossing) or few abnormal QRS complexes (i.e., knowledge-based), highly reducing the sampling rate while maintaining the QRS detection accuracy. However, these solutions are still being explored, so this thesis focuses on the challenges in edge computing.

The evolution of WSNs from single-core systems [24, 42, 44] into multi-core parallel computing platforms [45–49] has opened new possibilities for health

and wellness monitoring. From parallel processing to independent memory banks, to heterogeneity of cores and HW accelerators, these platforms facilitate the implementation of more complex, accurate and adaptive biomedical applications. However, the question of how to design adaptive approaches that exploit the capabilities of modern platforms still remains. Fig. 1.3 shows the effect of the application duty cycle (here used as a measure of algorithm complexity), multi-core processing (assuming a 7 × speed-up) and memory banks scaling on the energy consumption of one of the aforementioned modern platforms [49]. The energy numbers are computed based on the power numbers in [48, 49]. The architecture has one main core and a cluster of up to eight cores, which are different from the main one, and memory banks that can be powered off independently. From the figure, I show the possibilities for optimization that the architecture can give. For example, for a low-duty cycle application memory management has more impact than for high duty cycle applications, which are affected more by the processing. Therefore, for applications with higher complexity, adapting the computing resources (i.e., the number and type of cores used) and maximizing speed-up should receive higher priority than memory management, whereas for applications with lower computational complexity the focus should be on the reduction and scaling potential of the number of memory banks.

The capabilities of modern platforms allow designers of biomedical applications to approach the energy-accuracy trade-off with different lenses, having the ability to implement optimal adaptive solutions from both the physiological and the platform perspectives.

1.3 Contributions

This thesis describes three main domains of contributions in the context of remote health and wellness monitoring. First, I propose new personalized and ULP algorithms for multi-biosignal monitoring in traditional wearable sensors, with the goal of maximizing accuracy and reducing energy consumption. Then, in the context of modern ULP heterogeneous nodes, I propose platform optimizations for the modular and scalable (i.e., personalized to the subject) use of the computing and memory resources to design more energy-efficient biomedical applications. Finally, I propose an adaptive design based

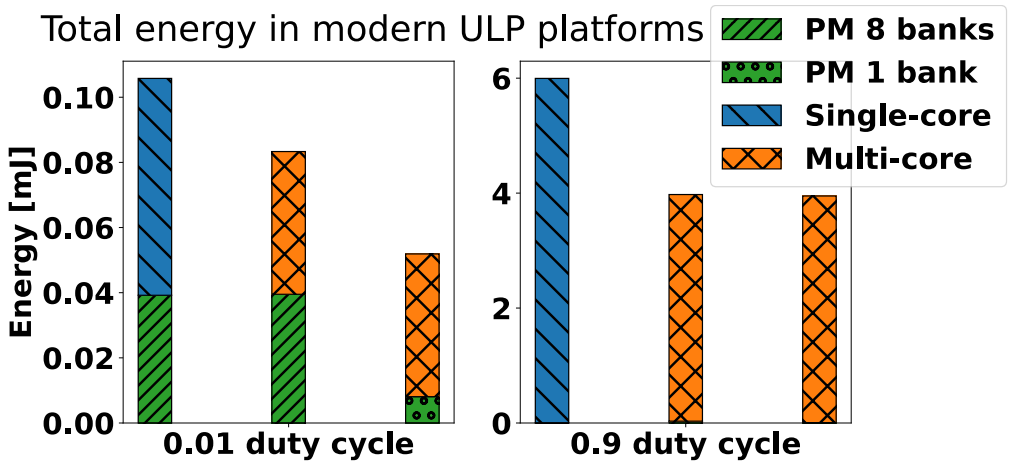


Figure 1.3 – Effects of duty cycle (i.e., algorithm complexity), multi-core processing (assuming a 7 × speed-up), and memory banks scaling on energy consumption in modern ULP wearable sensors [49] for health and wellness monitoring, divided in power and memory (PM) management, and processing in single-core and multi-core designs. For a low duty cycle application, multi-core processing with very good speed-up is advantageous compared to a single-core design. Moreover, the energy consumed for memory management (green) is approximately half in the multi-core design, hence, memory scaling from eight banks to one bank lowers the energy significantly. On the contrary, for a high duty cycle application, memory scaling does not affect the energy consumption, while multi-core processing can significantly lower the energy consumption. This analysis is relevant for designers of biomedical applications, which can focus on some optimizations rather than others considering the application computational complexity, attainable speed-up and, memory management

on a novel highly accurate though complex ECG R peak detection algorithm, called BayeSlope, and a lightweight and less robust method, called REWARD. The mail goal of this adaptive design is to achieve an optimal energy-accuracy trade-off by triggering the more accurate BayeSlope when REWARD fails and assigning different computing resources based on their complexity.

1.3.1 Personalized and Ultra-Low Power Multi-Biosignal Monitoring

In the context of multi-biosignal monitoring through WSNs, the challenges to maximize accuracy while reducing energy consumption lie first at the algorithmic level. Many approaches have been tackling this issue with acceptable levels of accuracy and energy efficiency [23–28]. However, a new perspective must be applied to improve the energy-accuracy trade-off. Therefore, in Chapter 2, I propose three new methods involving adaptive, personalized (i.e., to the patient's physiology) and knowledge-based strategies that are accurate and energy-aware.

The first contribution is presented in Section 2.2. Here I propose a novel lightweight real-time R peak detection method, called Relative-Energy-based WeArable R Peak Detection algorithm (REWARD). The method is based on a nonlinear filtering technique called Relative-Energy (Rel-En) [50], which amplifies the dominant peaks in the ECG. Rel-En can actually be applied to different types of biosignals other than ECG, making it a good candidate for multi-biosignal filtering. However, this thesis focuses only on its use on the ECG. REWARD applies adaptive hysteresis thresholds for the peak search and knowledge of physiological parameters to discern correct and incorrect peaks. REWARD is implemented on a traditional single-core Cortex-M3 device, and it is compared in terms of accuracy and energy to state-of-the-art algorithms running on the same platform.

Section 2.3 presents the second contribution of Chapter 2, which is a novel method for HR estimation using the PPG signal, a low-cost optical alternative to the ECG. A PPG-based device detects the blood volume changes in vessels by illuminating the skin with a light emitting diode (LED) and receiving a waveform that represents the light reflected by the tissues. The PPG waveform

gives information about the pulse rate, since the blood volume changes are caused by the heart pumping the blood to the periphery during each cardiac cycle and affect the sensor light absorbed. However, in conditions of intense physical exercise, PPG is highly affected by motion artifacts (MAs) that fall in the same range of frequencies as the pulse rate [51]. Therefore, I propose a novel method for MAs removal that compares the frequency domain of a 3-axis accelerometer and the PPG without the need for signal reconstruction. The knowledge of the movement analysis makes the algorithm more robust than other state-of-the-art examples [52–55]. Moreover, the method is highly energy-efficient by focusing on the frequency domain and implementing signal downsampling, integer arithmetic, and power management.

The third contribution of Chapter 2, presented in Section 2.4, describes a new online paroxysmal atrial fibrillation (PAF) prediction model for ULP wearable sensors, which scales the computation by considering the specific features of the individuals and their condition. In fact, atrial fibrillation (AF) is a type of arrhythmia caused by heterogeneous mechanisms in different patients and is one of the major causes of stroke and heart failure [56]. Therefore, personalization is essential for an accurate diagnosis and prediction. As mentioned, personalization has been proved to highly increase the accuracy of pathology detection and, specifically, in a previous work I proved that this is also the case for PAF prediction [57]. In this contribution, I exploit a patient-specific training phase to implement an optimized feature extraction and inference model to achieve scalable computation. The adaptive patient-specific training parameters affect the design in single-core platforms and the energy consumption by creating different energy levels based on the patient characteristics. Finally, power management allows for a scalable battery lifetime targeted to each individual.

1.3.2 Modularity and Patient-Specific Scalability in Multi-Core Heterogeneous Nodes

In the context of modern ULP heterogeneous platforms presented in Section 1.2, in Chapter 3, I propose two solutions that apply software (SW) parallelization and HW acceleration techniques on independent application modules. Moreover, I propose an adaptive and scalable solution for assigning

computing and memory resources in multi-core heterogeneous nodes based on the specific characteristics of the patient, in the context of the online PAF prediction that I present in Section 2.4. The two main contributions of Chapter 3 are presented in Section 3.4 and Section 3.5.

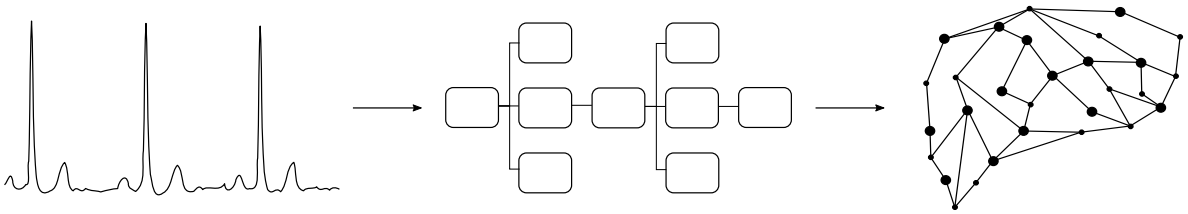
Most biomedical applications are organized in independent sequential modules, namely signal filtering, enhancement, feature extraction, and inference, as I explain in Section 3.2. Sometimes all the modules are present, such as [57], while other applications might apply only some of them, such as REWARD [41]. By exploiting the capabilities of modern ULP heterogeneous platforms, described in detail in Section 3.3, I propose a top-down approach to SW parallelization with techniques applied at different abstraction levels (c.f. Section 3.4). This strategy has the intent of maximizing the attainable speed-up of the parallelization and, hence, the potential energy savings. Additionally, I apply memory management to adaptively switch each memory independent memory bank between active, retentive and off mode based on the buffer acquisition memory usage, and reduce energy consumption. Finally, computationally intensive kernels are accelerated by a domain-specific coarse-grained reconfigurable array (CGRA) [46], consuming less energy than the general purpose cores available. Overall the three optimizations are orthogonal to each other and can be applied independently according to the application needs.

In Section 3.5, I present my design of a patient-specific parallelization technique targeting a multi-core platform, based on the online personalized PAF prediction algorithm that I present in Section 2.4. For each patient, the approach selects specific parameters and training models during the learning phase representing the number of cores used for the parallelization. In fact, as I analyze in Section 3.3.5, assuming the same attainable speed-up, assigning the lowest number of cores achieves higher energy savings than assigning more cores. Therefore, in the context of a personalized approach to PAF prediction, assigning computing resources based on the characteristics of each patient ensures significant energy savings. Moreover, I explore the effect on the energy consumption of memory bank scaling from 8 KiB to 4 KiB, 2 KiB and 1 KiB. By scaling to 1 KiB, the algorithm achieves the highest energy savings overall, although scaled to the specific characteristics of the patient.

In conclusion, modularity and scalability, by means of personalization at algorithmic and platform levels, are adaptive solutions that maximize accuracy while significantly reducing energy consumption.

1.3.3 Online Adaptive Design for Enhanced Energy-Accuracy Trade-Off

In the context of biomedical applications that monitor complex physical conditions, enhancing the energy-accuracy trade-off is challenging. As an example, during intense physical exercise (but not only) the cardiovascular and respiratory systems undergo significant changes that are manifested in signals like the ECG. Sudden changes in the heart rhythm, hyperventilation and the high demand in blood supply by the muscles translates in shorter RR intervals (i.e., the time between two heart beats), changes in the P wave (i.e., contraction of the upper chambers of the heart), smaller R peaks (i.e., main heart contraction or what mainly constitutes a heart beat) and higher T waves (i.e., lower chambers relaxation). In these conditions, standard R peak detection algorithms fail to properly discern the ECG main wave. Moreover, more complex algorithms can cause a substantial draining of platform resources leading to frequent device charging, not suitable for long-term remote monitoring. For this reason, in Chapter 4, I propose an adaptive design of a novel highly accurate and robust R peak detection algorithm, BayeSlope, paired with the lightweight but less robust REWARD. An error detection is applied to the output of REWARD that triggers BayeSlope if REWARD fails. Moreover, to exploit the advent of modern ULP heterogeneous platforms and their capabilities, BayeSlope is implemented in a more capable core and with more resources than the one where REWARD runs. By monitoring and adapting its performance and complexity, by means of two different algorithms, this strategy achieves an optimal energy-accuracy trade-off.



2

Personalized and Ultra-Low Power Multi-Biosignal Monitoring

Tackling the energy-accuracy trade-off in wellness monitoring applications that use wearable sensor nodes (WSNs) is a multifaceted challenge. The first step is to consider innovative optimizations at the algorithmic level for different biomedical applications using multiple biosignals. In fact, from vital parameters estimation in various physical conditions to pathology detection, the challenges of improving energy efficiency while maintaining accuracy in wearable sensors are similar. However, optimizations need to be targeted and adaptive to reach an optimal energy-accuracy trade-off specific to the application and/or the subject analyzed.

In this chapter, I propose several algorithmic optimization methods in the context of wellness monitoring, which achieve a high level of energy-accuracy trade-off, as a first aspect of the final optimal goal. First, I explore two methods that estimate, respectively, the instantaneous and average heart rate (HR) from two different biosignals (electrical and optical), and in distinct physical conditions (rest, pathology, and intense physical exercise). Then, I tackle the problem of atrial fibrillation (AF), a cardiovascular disease (CVD) that is one of the major causes of stroke and heart failure, and propose a patient-specific real-time approach to predict its onset.

2.1 Introduction

Remote wellness monitoring has become an essential branch of healthcare with the rapid development of wearable sensors technology. WSNs are unobtrusive and cost-effective means of continuous monitoring of vital parameters and noncommunicable diseases (NCDs) with the final goal of health enhancement and prevention [58]. WSNs can acquire multiple biosignals, such as the electrocardiogram (ECG) and the photoplethysmography (PPG) waveform, and be used in various settings, such as at rest and during intense physical exercises. Moreover, they can also include embedded algorithms for the detection and prevention of CVDs, which are the primary cause of death globally [3].

The vital parameters estimation and pathology detection algorithms for WSNs must provide a high level of accuracy according to medical standards. Many works that target remote wellness monitoring focus on reaching this medically acceptable level by optimizing their algorithms [21, 24, 26, 38, 42, 54, 57, 59]. However, energy efficiency must be taken into account for edge computing devices used for continuous monitoring, which are the object of this thesis [23–28]. In this chapter, I propose various methods that are energy-efficient while maintaining high levels of accuracy for ultra-low power (ULP) wearable devices.

For the first contribution of this chapter, I present the Relative-Energy-based WeArable R Peak Detection algorithm (REWARD), which is a novel real-time R peak detection mechanism based on a nonlinear filtering method called Relative-Energy (Rel-En) [50] applied to an ECG signal. The ECG describes the electrical activity of the heart during its contraction with three main waves[1] [60], shown in Fig. 2.1. The P wave represents the contraction of the upper chambers of the heart (i.e., atria). Then, the QRS complex, or the ECG main wave, represents the contraction of the lower chambers of the heart (i.e., ventricles). Finally, the T wave represents the relaxation state of the ventricles. The three waves combined make one cardiac cycle, or a heartbeat. The R peak is a parameter included in the ECG main wave and the frequency of its occurrence, i.e., HR, provides valuable medical information [61]. R peak detection

[1] In this thesis, U waves were not considered as they are not always observable.

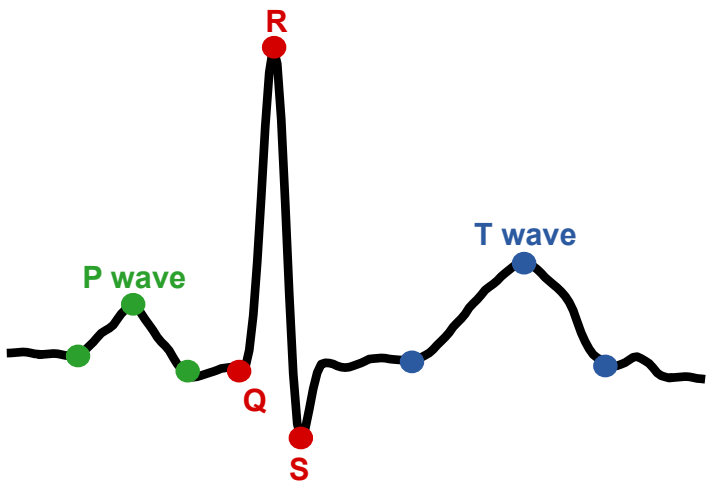


Figure 2.1 – ECG heart beat with its three main waves representing the phases of the heart electrical activity during its contraction

is essential to more complex algorithms that screen for serious medical conditions that affect the cardiac rhythm, such as myocardial infarction [26] and atrial fibrillation [42]. Moreover, it is the base for heart rate variability (HRV) analysis, measured as the variation in the beat-to-beat interval or RR time series. This analysis gives essential information on the effects of the autonomous nervous system in medical and sport settings [33, 62–65]. Current real-time R peak detection algorithms employ techniques such as signal derivative analysis [66], adaptive thresholds and parameters [67], and variations of the Wavelet Transform [38,68,69]. These algorithms have been widely compared in terms of R peak detection accuracy [70]. However, few of these works have performed a complete study on the energy and memory footprint trade-offs of the algorithms when implemented on resource-constrained real-time systems [38, 66, 71]. Furthermore, each work tests its proposed algorithm on a different hardware (HW) processor or simulator platform, which makes a comparative assessment of the algorithms difficult. Few works in the literature have compared the feasibility of implementing different real-time R peak detection techniques on embedded systems. For this reason, this contribution addresses the need for a comprehensive comparison of well-

established approaches for real-time R peak detection in wearable systems. The main objectives of this contribution are:

- Implementing the Rel-En method in real-time and designing a peak detection procedure to complement it.
- Optimizing the REWARD algorithm specifically for use on resource-constrained systems, maximizing its accuracy while minimizing its energy and memory footprints.
- Comparing this new algorithm against three state-of-the-art real-time R peak detection algorithms and showing that it performs comparably in terms of accuracy while consuming less energy and memory, measured on the same HW platform.

The second contribution to this chapter is an alternate method for HR estimation using an optical source (i.e., PPG) instead of an electrical one (i.e., ECG) and in conditions of intense physical exercise where the PPG signal integrity is most negatively affected. In fact, a PPG sensor gives information about the cardiac rhythm, by illuminating the skin with a light emitting diode (LED) and collecting the light reflected (i.e., the PPG waveform), which detects the blood volume changes in vessels. Specifically, by considering the PPG frequency spectrum, it is possible to estimate the HR within its standard range between 0.67 Hz and 3.67 Hz, i.e., 40 beats per minute (BPM) and 220 BPM, considered in this work, corresponding to a range of HR from a rest condition to intense physical exercise. However, during the latter the PPG signal is affected by strong motion artifacts (MAs) in the same range of frequencies [51]. There are various methods to estimate the HR from the PPG waveform [52–55], but they all present different problems from a low level of accuracy to a high level of complexity, which makes it not suitable for implementation on ULP WSNs for remote wellness monitoring. Therefore, I propose a method for monitoring HR in real-time which only analyzes the spectrum retrieved from the fast Fourier transform (FFT) and it is targeted to WSNs. The main outcomes of this contribution are:

- The method does not require a noise-free signal reconstruction, but only focuses on the detection of MAs as peaks within the standard range

of frequencies previously mentioned. This allows to gain computational time, speed and memory space and decrease power consumption in the embedded device.

- The method detects a wide range of MAs, and it manages to estimate the HR when PPG and MAs spectra overlap. Overall, it shows an average absolute error of only 1.27 BPM with a standard deviation of 0.91 BPM on the database analyzed, comparable to the performance of state-of-the-art offline algorithms.

- The method works on short windows of data, which makes it applicable to real-time processing on WSNs. It does not require a reference signal, as the history of estimated values is used to update the current one. Moreover, it employs integer arithmetic to reduce execution time. By computing the execution time of the algorithm on-board, i.e., 226 ms per second, it grants a battery autonomy of 9.37 days for the fully working device.

Finally, for the third contribution to this chapter, I propose a methodology to design a new online paroxysmal atrial fibrillation (PAF) prediction model targeting scalable computation on ULP wearable sensors, which considers the specific features of the individuals and their condition. In fact, in a previous work I proved that a patient-specific approach highly increases the accuracy of PAF prediction [72]. In this chapter, I tackle the challenge of designing a real-time version of this approach for ULP WSNs and exploit the patient-specific training phase to achieve scalable computation. The scalability is driven by the adaptive training parameters, which affect the design in single-core platforms to reduce energy consumption for each individual patient. The main outcomes of this contribution are the following:

- The online energy-efficient PAF prediction model is implemented on a single-core ULP wearable sensor INYU [24], which is personalized according to the characteristics of each patient. The optimized and personalized model allows to reduce the energy consumption and processing execution time, by considering the constraints of the wearable sensor.

- Additionally, by exploiting the existing low-power sleep modes between sample acquisition, the wearable device running my proposed method achieves a scalable battery lifetime of, at least, 37 days.

This chapter continues by presenting the details of the three contributions mentioned. First, Section 2.2 describes the related work, methods, experimental setup, and results of the REWARD algorithm, which has been published in the proceedings of Engineering in Medicine and Biology Conference (EMBC) [41]. Then, it presents the second main contribution in Section 2.3, which has been published in the proceedings of the 19th Euromicro Conference on Digital System Design (DSD) [57]. Then, it follows with the third contribution in Section 2.4, which has been published in a special issue of the IEEE Transactions on Emerging Topics in Computing journal, called "New Trends in Parallel and Distributed Computing for Human Sensible Applications" [73].

2.2 REWARD: a Real-Time Relative-Energy Wearable R Peak Detection Algorithm

An essential parameter detection for CVD monitoring and diagnosis is the detection of the ECG main wave, which includes the R peak. For this reason, I present a new lightweight real-time R peak detection algorithm, called REWARD, and I compare it with three state-of-the-art algorithms in terms of accuracy, robustness, memory footprint, and energy consumption using the same HW platform.

In this section, I first explore the state-of-the-art algorithms used in the comparison (c.f. Section 2.2.1). Then, I describe the complete method and the optimizations for resource-constrained ULP embedded devices (c.f. Section 2.2.2). Finally, I present the database and HW platform used for the comparison of the four algorithms (c.f. Section 2.2.3), resulting in the outcomes described in Section 2.2.4.

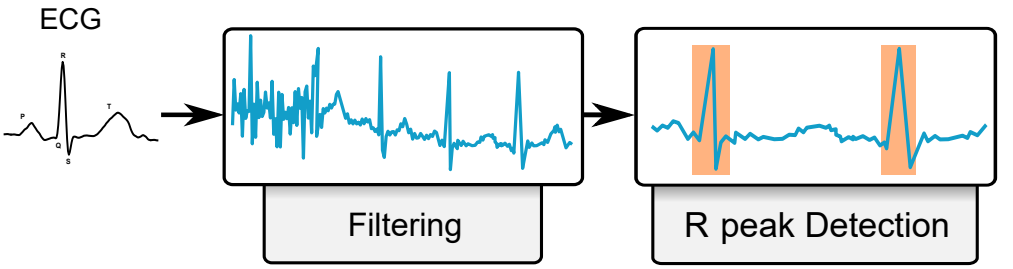


Figure 2.2 – Block diagram of a general real-time R peak detection

2.2.1 Related Work on Real-Time R Peak Detection Algorithms

Many R peak detection algorithms have been developed, but relatively few are designed for real-time implementation on ULP embedded systems. Three algorithms that meet these design constraints are the Pan-Tompkins (PT) algorithm [67], a wavelet transform delineation (WTD) [38], and a derivative-analysis-based delineation (DAD) [66]. These algorithms are well-known for their high accuracy and previous implementation on real-time wearable systems. They also represent diverse R peak detection methodologies.

An ECG R peak detection algorithm generally consists of a two-step procedure, as depicted in Fig. 2.2. First, a preprocessing step may include a filtering method to suppress noise in the ECG excerpt, as well as specific techniques to highlight the principal components of the ECG waveform. Then, a peak detection procedure locates the R peaks.

This section provides an overview of the aforementioned three state-of-the-art R peak detection algorithms, detailing the real-time implementation of each of them. All filters and algorithms are implemented in the C programming language using primarily 16-bit integer arithmetic, a sampling frequency of 250 Hz, and minimal buffer sizes to ensure a fair energy and memory consumption comparison.

2.2.1.1 Real-Time Preprocessing Methods

Every algorithm contains its own preprocessing method for filtering and highlighting the principal components of the ECG waveform. These methods

are often used to remove baseline wander, high-frequency noise, and muscle artifacts. Two frequently-used real-time filtering approaches, a morphological filtering (MF) [74] and a band-pass filtering (BPF) [66], are analyzed to determine the benefits they provide to the accuracy of each algorithm versus the drawbacks of additional energy and memory consumption. The MF performs two operations, opening and closing, which respectively remove the peaks and valleys of the signal. These operations produce the baseline, which is then subtracted from the original signal to remove any baseline drift. An opening window of 0.2 s is used, along with a closing window of 0.3 s. The filter is coded in C in real-time and introduces a delay of 0.49 s. The finite impulse response (FIR) filter design of the BPF, with a passband of 0.3-40 Hz and order 32, is done offline, which suppresses both the baseline and high-frequency noise. It is coded in C using mainly 16-bit integer operations and implemented using symmetry criteria for the filter coefficients, as described in [66].

2.2.1.2 Real-Time R Peak Detection Algorithms

The Pan-Tompkins Algorithm (PT) Pan and Tompkins proposed a real-time ECG R peak detection algorithm in 1985 [67], which has since been widely used in the literature. The 4-step preprocessing method of the algorithm consists of a 5-12 Hz bandpass filter, a derivative of the filtered signal, squaring the derivative to amplify the QRS complex, and a moving-window integrator. Then, the peaks of the ECG signal are identified by applying adaptive thresholds on the filtered and integrated signals. These thresholds use the past eight peak amplitudes and RR intervals to identify peaks and ensure that they have an RR interval above 0.2 s.

The initial delay and buffer size of this algorithm include 5 s of signal, namely, 2 s to initialize the peak detection thresholds, plus 3 s to compute the initial RR interval and search back for missed peaks. The algorithm is implemented in real-time in C using primarily 16-bit integer arithmetic.

Wavelet Transform Delineation (WTD) A widely implemented algorithm that performs full ECG delineation is the wavelet transform (WT). The WT of a signal is proportional to the derivative of the signal with a smoothing

impulse response at different scales. Therefore, the zero-crossings of the WT function correspond to the local maxima or minima of a signal at a given scale, and the peaks correspond to its maximum slopes. Five dyadic scales are chosen (i.e., 2^1 to 2^5), since most of the ECG signal energy lies within these scales [68]. Once the WT is applied to the signal, the R peaks are identified as the zero-crossings that are common across scales 2^1 through 2^4, and which are preceded by a positive peak and followed by a negative peak.

The WTD algorithm analyzed in this work is the optimized, single-lead, offline ECG delineation algorithm presented in [68], implemented by [69], and extended in [38]. This delineator detects all characteristic points of an ECG waveform using a quadratic spline WT. The algorithm is implemented in real-time with a buffer size of 1.024 s. It is coded in C using primarily 16-bit integer operations.

Derivative Analysis Delineation (DAD) Recently, Bote et al. proposed a derivative-based, low-complexity algorithm for ECG delineation [66], which has a modular design. It can perform either full ECG delineation, or operate in a low-power mode that only detects R peaks, the latter of which is analyzed in this work. First, the signal is preprocessed with a 14 Hz lowpass filter. Next, the first and second derivatives in a 2 s window are analyzed to identify the R peaks as points at which 1) there is a zero crossing of the first derivative, 2) the RR interval is higher than 0.25 s, and 3) the magnitude of the second derivative exceeds 0.33 × the average of the past five minimum/maximum window values. This algorithm is implemented in real-time with a buffer length of 2 s. It is coded using primarily 16-bit integer arithmetic.

2.2.2 REWARD Design and Real-Time Optimization

The REWARD algorithm includes two main components. Firstly, the ECG signal is preprocessed to highlight its peaks and suppress its baseline using the Rel-En nonlinear filtering method proposed in [50]. Secondly, the R peaks are detected from the filtered signal. In this section, I present the first real-time implementation of the Rel-En preprocessing method and the optimizations for use on ULP wearable systems. Then, it is paired with the R

peak detection procedure that was designed within the same publication [41], all of which is described next.

2.2.2.1 Rel-En Preprocessing Implementation and Optimization

The Rel-En preprocessing method considers the energies of a long sliding window l_{win} (0.95 s) and a short sliding window s_{win} (0.14 s), both centered at sample n. l_{win} describes the long-term behavior of the ECG signal x, while s_{win} can capture an R peak occurrence, resulting in a larger short-term energy than when no peak occurs. The ratio between the energies of these windows, the coefficient *c(n)*, is multiplied by *x(n)*, resulting in a signal x_{RE}, in which the peaks are amplified, as depicted in (2.1) and (2.2). The parameter w in (2.1) represents a Hamming window function, and $p = 2$.

$$c(n) = \frac{\sum_{i=n-swin/2}^{n+swin/2} |x(i)|^p}{\sum_{j=n-lwin/2}^{n+lwin/2} |w(j) \times x(j)|^p} \tag{2.1}$$

$$x_{\text{RE}}(n) = c(n)x(n) \tag{2.2}$$

The effect of Rel-En on a filtered ECG is shown in Fig. 2.3, where it is evident how the method amplifies the dominant peaks of the ECG in the final output x_{RE}.

In this contribution, the Rel-En preprocessing method is ported from MATLAB to C and optimized for single-lead, real-time ECG R peak detection on resource-constrained wearable systems. First, the computation is changed from floating point arithmetic (32-bit) to short integer (16-bit) to consume less energy and memory on the microcontroller unit (MCU). Subsequently, the Rel-En method is implemented using circular buffers to minimize RAM usage and processing delays. It considers a centered sliding window of 0.95 s of the ECG signal for preprocessing to obtain the x_{RE} signal.

Finally, the Rel-En preprocessing method is simplified to reduce its energy consumption. In [50], the Hamming window function is used to smooth the long-term energy of each coefficient *c(n)*, which represents a significant number of operations performed per coefficient output. Specifically, suppressing

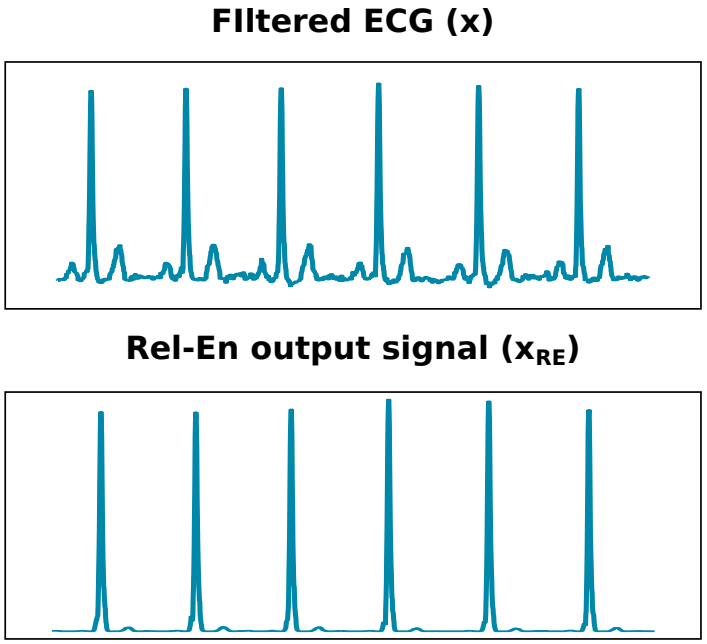


Figure 2.3 – The output of the Rel-En method applied to a filtered ECG. Rel-En isolates and amplifies the dominant peaks in a signal.

this step reduces the computational load by a factor of N=fs*l_{win} where fs is the sampling frequency, so for each coefficient there must be N calculated Hamming window coefficients and N multiplications. In order to reduce the algorithm's complexity and consequent energy consumption, the method does not use the Hamming window function from the long-term window calculation, i.e., $w(j) = 1$, whose removal did not result in any significant cost to the algorithm's performance (c.f. Section 2.2.4.1).

2.2.2.2 REWARD Peak Detection

To complete the REWARD algorithm, a second step is paired to the Rel-En method. This is a real-time peak detection procedure that is both adaptable and computationally simple. The algorithm is based on the hysteresis comparator [75], and several optimizations are applied to improve its detection accuracy. The R peaks are detected using a window of 1.75 s. Consequently, the initial delay of this algorithm is $(0.95/2 + 1.75)$s. For each R peak detection window, the algorithm first checks if the dominant peak is positive

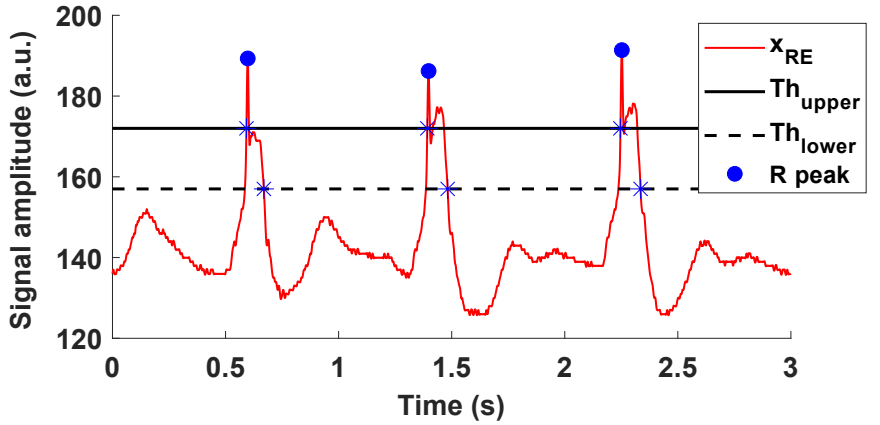


Figure 2.4 – The hysteresis comparator detects the peaks of sel102 of the QTDB.

or negative to change the values of its parameters accordingly. Then the hysteresis comparator is applied to identify possible peaks. Next, the algorithm selects the peaks that meet a set of criteria, such as representing an HR between 30-240 BPM and a peak width in the same range as that of the previously-selected R peak.

The first step of the algorithm is a real-time variation of the negative peak identification procedure described in [76]. Within the 1.75 s peak detection window, if the minimum amplitude relative to the mean value of x_{RE} is greater than 70% of the maximum amplitude relative to the mean, it is presumed that the R peak is negative. The method applies the same steps for both positive and negative peaks, though the values of its parameters change depending on the polarity. For simplicity from this moment on, I will describe the positive case.

Next, the hysteresis comparator method is implemented to identify the locations of the peaks based on two adaptive thresholds, as illustrated in Fig. 2.4. Segments in which the signal goes above the upper threshold, Th_{upper}, and subsequently below the lower threshold, Th_{lower}, are considered active peak regions. The maximum of the signal between these two points is considered a peak candidate. The purpose of having two thresholds is to eliminate false R peak candidates due to high-frequency oscillations or false peaks at the

Algorithm 1 REWARD: R peak selection within a long window

1: **function** PEAKSEL($\mathbf{x_{RE}}, R_{last}, T_u, T_l$)
2: $(Avg, Max, Min) = statSigInWindow(\mathbf{x_{RE}})$;
3: $Th_{upper} = Avg + T_u \times |Avg - Max/Min|$;
4: $Th_{lower} = Avg + T_l \times |Avg - Max/Min|$;
5: $\mathbf{pks^{loc,wid}} = findPks(\mathbf{x_{RE}}, Th_{upper}, Th_{lower})$;
6: **for** $n = 2 : length(\mathbf{pks^{loc}})$ **do**
7: **if** $0.25s < pks_n^{loc} - R_{last}$ **then**
8: $discardPeaks(pks_n^{loc})$;
9: **else if** $pks_n^{loc} - R_{last} > 0.5s$ **then**
10: $keepPeaks(pks_n^{loc})$;
11: **else**
12: **if** $0.65 < pks_n^{wid}/pks_{n-1}^{wid} > 1.35$ **then**
13: $discardPeakWithLargerWidth()$;
14: **else**
15: $keepPeaks(pks_n^{loc})$;
16: **end if**
17: **end if**
18: **end for**
19: **end function**

threshold boundary. This procedure is less complex than initially searching for all local maxima in a window and then applying a threshold to select the peaks, since false peaks often exist at the threshold boundary due to signal noise.

In order for the thresholds to adapt to changes in the signal's amplitude from one window to the next, they are defined as shown in Lines 2-4 of Algorithm 1. Avg and Max denote the mean and maximum values in one window of $\mathrm{x_{RE}}$. To determine the optimal thresholds, the unfiltered REWARD algorithm was tested on the QTDB with every combination of threshold constants T_u and T_l. Thresholds that are too high result in missed peaks and low sensitivity, whereas thresholds that are too low result in a low positive predictive value (PPV). Thus, after a careful experimental validation, the chosen pair of threshold constants are the ones that produce the highest G-mean (98.61%) is $T_u = 0.4$ and $T_l = 0.15$.

False peaks often occur due to enlarged T-waves that exceed Th_{upper}. To solve this issue, it is necessary to employ the physiological characteristics of the RR interval and the T-wave: The RR interval lies between 0.25-2s, while the T-wave is typically wider than the QRS complex and occurs within 0.5 s after the R peak. Accordingly, the peak selection procedure is described in Lines 5-18 of Algorithm 1. First, the algorithm computes the peak widths in Line 5 and checks the corresponding RR interval. The peak width is the time interval between the peak onset and offset, determined by Th_{upper} and Th_{lower}. If the RR interval is longer than 0.5 s, the current peak is kept, and if it is less than 0.25 s, the peak is discarded. Finally, if the RR interval is between 0.25-0.5s, the widths of both peaks are compared to determine if both peaks are valid or one is an enlarged T-wave. If one peak is more than 35% wider than the other, the peak is discarded. This percentage value was chosen empirically (cf. Section 2.2.3.1 for details about the used databases), considering the physiology of the ECG waveform and taking into account the different types of morphologies included in the database. The aforementioned approach is robust against premature beats because whenever the condition on the RR intervals is not satisfied, the algorithm checks the widths of the two peaks and if they are within the same range, it keeps both peaks.

The real-time peak detection procedure is more adaptable and computationally simple than the peak detection proposed in [50]. In particular, in the original offline method, the entire signal was first normalized based on its minimum and maximum values, the mean of the signal was subtracted, and then a fixed threshold was applied. This normalization and mean subtraction introduces significant computational complexity, since each function performs at least one mathematical operation on every sample in the signal. In contrast, the hysteresis-based procedure described in Algorithm 1 includes peak detection thresholds, $\mathrm{Th}_{\mathrm{upper}}$ and $\mathrm{Th}_{\mathrm{lower}}$, which are adapted based on the average and maximum values of each peak detection window.

2.2.3 Experimental Setup

2.2.3.1 Standard Databases and Metrics for Accuracy Evaluation

In order to quantify the detection accuracy of the algorithms, I use two public databases provided by Physionet [77]. The first is the QT Database (QTDB) [78], which consists of 105 two-channel ECG Holter recordings with a wide variety of ECG morphologies, including various arrhythmias and sudden death cases. Each 15-minute recording contains two leads, sampled at 250 Hz. At least 30 beats of each recording have been manually annotated by an expert to identify the locations of several ECG fiducial points. Out of the 3622 annotated beats in the database, a total of 3587 annotations include the R peak.

Next, to test the algorithms' robustness to noise, the Noise Stress Test Database (NSTDB) [79] is used. The NSTDB consists of two clean 30-minute-long signals of the MIT-BIH Arrhythmia Database (MITDB), to which five varying amounts of noise were added such that the Signal-to-Noise Ratio (SNR) decreased by 6 dB for each noise addition. The signals were originally sampled at a frequency of 360 Hz and re-sampled to 250 Hz to maintain consistency when testing the algorithms. The original R peak annotations from the MITDB are used as ground-truth.

To assess the performance of the analyzed algorithms, the true positives (TP), false positives (FP), and false negatives (FN) of the detected peaks were computed using 150 ms of tolerance from the annotated peak [80]. Accordingly, I show the performance metrics of sensitivity (SE = $\frac{TP}{TP+FN}$), positive predictive value (PPV = $\frac{TP}{TP+FP}$), geometric mean (G-mean = $\sqrt{SE * PPV}$), and detection error rate (DER = $\frac{FP+FN}{TP+FN}$). Finally, the mean error (m) and its standard deviation (σ) are measured.

2.2.3.2 HW Platform for Real-Time Implementation

To measure the energy consumption of the state-of-the-art and proposed algorithms and filters on resource-constrained wearable devices, they were implemented on the Silicon Labs EFM32 Leopard Gecko 32-bit MCU [81]. This board contains a 48 MHz ARM Cortex-M3 CPU, 32 KiB of RAM, and

256 KiB of flash memory. These HW specifications are similar to those found on wearable ECG sensor nodes; the aforementioned INYU device uses the same processor. Every algorithm runs using the -O3 compiler optimization level, which is the best optimization tolerated by the EFM32. The board, along with its development environment Simplicity Studio, includes an energy profiler that measures the execution time, as well as the total energy consumed by the algorithms within a specified execution window. The board is placed into a sleep mode before and after each algorithm execution to ensure that the energy consumption and execution time only reflect those of the algorithm. The Simplicity Studio development environment also measures the amount of memory, both RAM and Flash, consumed by each algorithm. Flash memory permanently stores the variables and instructions of a program, whereas RAM performs run-time operations on the variables it retrieves from Flash.

The algorithms were tested independently on the EFM32 board, considering 12 s (3000 samples) of four different recordings of the QT Database, which were chosen to contain varying degrees of R peak detection accuracy.

2.2.4 Experimental Results

2.2.4.1 Accuracy and Energy Impact of REWARD Real-Time Design and Optimization

As REWARD was designed for use on real-time, ULP systems, several optimizations were performed to minimize its energy and memory footprints while increasing its R peak detection accuracy. This involved multiple changes to the original Rel-En preprocessing method proposed in [50], as well as our design of a paired peak detection algorithm.

First, the original Rel-En preprocessing method was ported from MATLAB to C, changed from 32-bit floating point to 16-bit integer arithmetic, and run on the EFM32 using the -O3 optimization level. Then the Rel-En method was optimized by removing the Hamming window function $w(j)$ in Equation (2.1), as described in Section 2.2.2.1. Multiplying each value of the long window by its corresponding Hamming window coefficient significantly increased the number of operations performed per coefficient output. Consequently, the

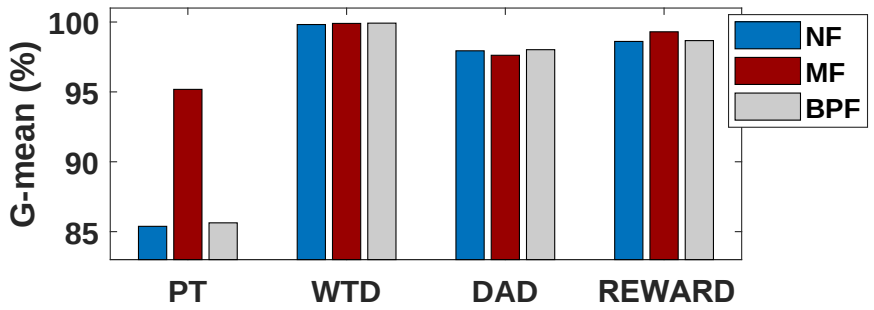


Figure 2.5 – The G-means of the four R peak detection algorithms with their respective filters applied to the QTDB.

Hamming window function consumed 96 % of the total energy of REWARD. This optimization only decreased the unfiltered REWARD G-mean by 0.13 %, but dramatically reduced the energy consumption by more than three orders of magnitude (1316x).

The original algorithm described in [50] was tested on the QTDB with an offline 4-40 Hz BPF and produced a G-mean of 99.97 %. The final G-mean of the new optimized, real-time REWARD algorithm with a real-time 0.3-40 Hz BPF was only 1.28 % lower than that of the original algorithm.

2.2.4.2 R Peak Detection Accuracy

The accuracy results of the four algorithms are displayed in Table 2.1, which lists the performance with no additional filters "NF", and with a MF or BPF applied. Moreover, the algorithms' performance in terms of G-mean is summarized in Fig. 2.5. First, the benefit of the two filters was assessed. Due to their design, using a BPF alongside the REWARD, WTD, DAD, and PT algorithms did not significantly improve their accuracy; their G-means increased by 0.08 % or less. This is because the REWARD preprocessing amplifies the peaks, and the hysteresis comparator counteracts the effects of high-frequency noise. Similarly, PT and DAD use aggressive lowpass filters, while their use of derivatives mitigates baseline drift. MF, on the other hand, resulted in higher G-mean increases in the algorithms. The use of MF increased PT's G-mean by 11.5 %, that of REWARD by 0.70 %, and that of WTD by 0.10 %. Overall, filtering does not significantly increase the accuracy of

Table 2.1 – R peak detection results on the QT database on 3587 evaluated beats

		Methods			
	Filters	**REWARD**	**PT [67]**	**WTD [38]**	**DAD [66]**
TP	**NF**	3550	2694	3574	3445
	MF	3563	3281	3581	3421
	BPF	3546	2602	3576	3443
FP	**NF**	63	129	0	4
	MF	26	32	0	3
	BPF	53	209	0	2
FN	**NF**	37	833	13	142
	MF	24	306	6	166
	BPF	41	985	11	144
SE (%)	**NF**	98.97	76.38	99.64	96.04
	MF	99.33	91.47	99.83	95.37
	BPF	98.86	72.54	99.69	95.99
PPV (%)	**NF**	98.26	95.43	100.0	99.88
	MF	99.28	99.03	100.0	99.91
	BPF	98.53	92.56	100.0	99.94
G-mean (%)	**NF**	98.61	85.38	99.82	97.94
	MF	99.30	95.18	99.92	97.62
	BPF	98.69	81.94	99.85	97.94
DER (%)	**NF**	2.74	26.31	0.36	4.07
	MF	1.38	9.34	0.17	4.71
	BPF	2.58	31.45	0.31	4.07
m$\pm\sigma$(ms)	**NF**	9.5±4	100±9.6	11±3.8	11±3.3
	MF	9.3±3.5	13±4.4	7.5±3.3	7.5±3.2
	BPF	9.7±4.3	100±10	11±3.7	7.6±3.2

these algorithms, but it is still advisable for medical applications in which precise results are indispensable.

Analyzing the SE and PPV of the algorithms reveals that REWARD produced high accuracy results both with and without filtering. In terms of SE, REWARD paired with MF was only 0.50 % less than WTD with MF. In terms of PPV, REWARD with MF was 0.72 % lower than WTD with and without filtering. The four achieved high PPVs with MF: over 99.0 %. WTD produced the highest G-mean, with the unfiltered G-means of REWARD and DAD trailing that of WTD by only 1.2 % and 1.89 %, respectively. With MF applied, the REWARD G-mean was only 0.62 % lower than that of WTD. Furthermore, REWARD produced comparable accuracy results to both WTD and other state-of-the-art algorithms, and MF further increased its performance [70].

The comparison of all these parameters gives a complete picture of the performance of the four algorithms, in terms of correct detection of R peaks but also maximum error rate. The performance results considering the use of filters describe the robustness of the four algorithms and the additional complexity introduced, which are two important factors to consider when tackling the energy-accuracy trade-off problem.

2.2.4.3 Robustness to Noise Evaluation

REWARD, WTD, and DAD were tested on the NSTDB to determine how signal quality degradation affects their R peak detection accuracy. In Fig. 2.6, the SNR of each signal in the NSTDB is plotted against the corresponding G-mean of each algorithm. For signal 118, G-mean of REWARD was an average of 3.77 % lower than that of DAD for the three highest SNRs, and an average of 7.67 % lower overall. REWARD performed poorly on signal 119, however, due to overly high hysteresis thresholds for this particular ECG morphology. The REWARD G-mean decreased fairly consistently with each 6 dB drop, while the DAD and WTD G-means stayed nearly the same for SNRs between 12 and 24 dB and then dropped sharply. These results indicate the algorithms' behavior in the presence of noise, which can occur in daily environments using a wearable sensor.

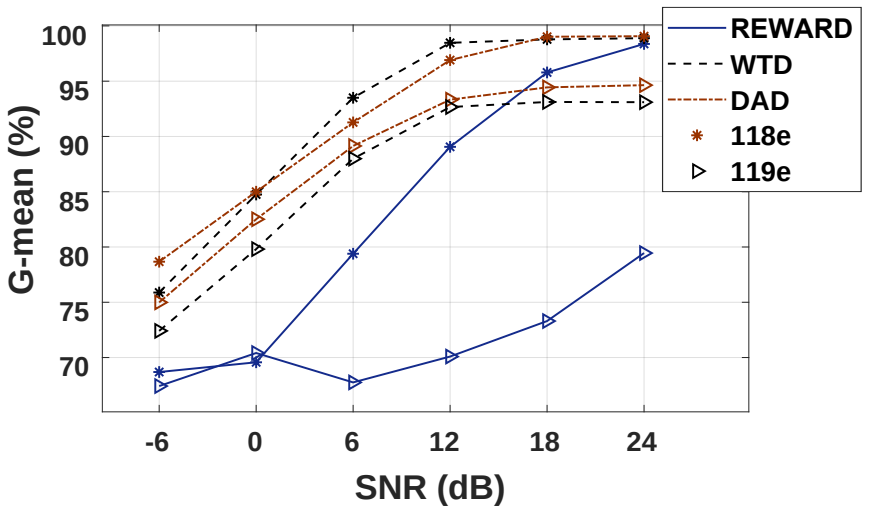


Figure 2.6 – The accuracy results of WTD, DAD, and REWARD algorithms with MF and BPF on the NSTSB signals with varying SNRs.

2.2.4.4 Energy Consumption and Memory Footprint Assessment

The energy metrics of the algorithms and filters on the four 12-second-long (3000 samples) QTDB signals, when run on the EFM32 MCU using -O3 compiler optimization, are averaged and presented in Table 2.2. First, the table displays the computational burden, which is the total code execution time divided by the total ECG signal acquisition time (i.e., 12 s). This metric indicates what percentage of the ECG sampling period (i.e., 4 ms) is spent performing the algorithm's functions. The table also shows the total energy that it takes to process the 12 s. REWARD consumed the least energy: only 916 µJ, which corresponds to 305 nJ per processed sample. DAD and WTD consumed 2.72 × and 3.6 × more energy than REWARD, respectively. Furthermore, when the code was run using no compiler optimizations (-O0), DAD consumed 2.29 × more energy than REWARD, indicating that increased optimization levels lead to a comparatively better performance for REWARD. PT consumed over 98 × as much energy as REWARD.

Fig. 2.7 depicts the energy consumed by each algorithm when it processes 12 s of data from the QTDB, when both filters are applied. It shows that three

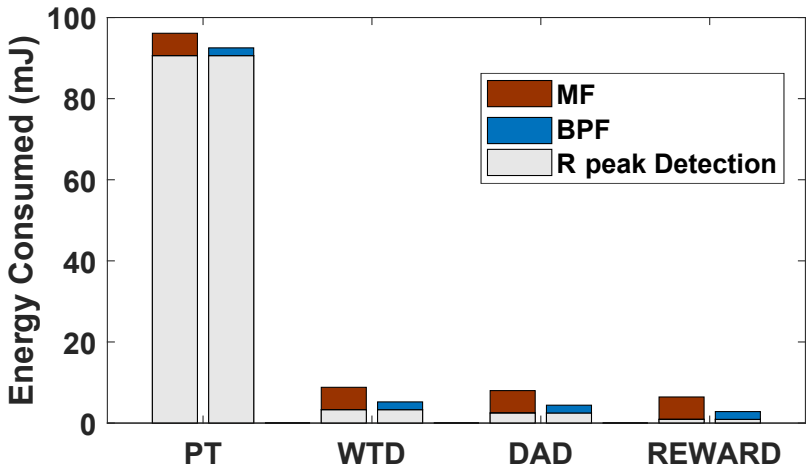


Figure 2.7 – The energy consumed by four R peak detection algorithms with and without filters on 12 s of samples from the QTDB.

of the four algorithms (REWARD, WTD, and DAD) had comparable energy consumptions, while PT consumed much more.

Table 2.2 also displays the memory footprints of the algorithms and filters. REWARD consumed by far the least amount of RAM; 1.51 × less than DAD. WTD consumed 3.96 × more RAM than REWARD, while PT consumed nearly all of the available 32 KiB of RAM. REWARD and DAD consumed less Flash than the other algorithms, followed by PT. Finally, WTD consumed 1.89 × more Flash than REWARD. Overall, REWARD and DAD were the most memory-efficient algorithms.

2.2.4.5 Energy and Memory vs Accuracy Analysis

Fig. 2.8 displays the energy and RAM consumption of each algorithm-filter combination, except those of PT, plotted against their G-means from Table 2.1. This figure shows that without filters, REWARD achieves the lowest energy and RAM footprint, while WTD has the highest accuracy but also the largest amount of RAM and energy consumption. It also shows that applying a MF to REWARD increases both the accuracy and energy consumption, whereas applying filters to WTD and DAD increases their energy consumption without a significant impact on their accuracies.

Table 2.2 – Energy profile and memory footprint of R peak Detection Algorithms and Filters

		Computational burden (%)	Total Energy (mJ)	RAM (KiB)	Flash (KiB)
Filtering	MF	0.95	5.52	1.88	15.8
	BPF	0.29	1.92	0.228	7.50
R peak detection	PT	16.4	90.6	28.6	16.7
	WTD	0.56	3.30	6.26	23.10
	DAD	0.37	2.49	2.34	8.51
	REWARD	0.16	0.916	1.58	12.20

REWARD has the lowest energy consumption and a small memory footprint, but its current implementation is less robust than for WTD and DAD. WTD produces the most accurate, robust results, and consumes little energy, but its memory footprint is significantly higher than that of DAD and REWARD. Similarly, DAD exhibits the best robustness to noise and lowest Flash memory consumption, but its SE when tested on the QTDB is low compared to the WTD and REWARD, which implies that it does not correctly identify R peaks of various ECG morphologies present in the QTDB. Finally, for the filters, Table 2.2 shows that BPF is more energy-efficient than MF, using 65 % less energy, whereas Table 2.1 shows that MF leads to higher G-means. Though the use of MF only slightly improved the accuracy results of the four algorithms, it should still be considered for processing noisier signals from wearable sensors.

While each algorithm's specific computational burden and energy consumption may vary depending on the selected HW, testing all four algorithms on the same ARM Cortex-M3-based platform provides a comparative analysis of the algorithms relative to each other. This assessment enables wearable technology designers to select the algorithm and filter that fits the accuracy, energy consumption, and memory footprint constraints of their device. All the options and results presented in Table 2.2 in terms of memory footprint and computational burden fit in any state-of-the-art microcontrollers [82–85].

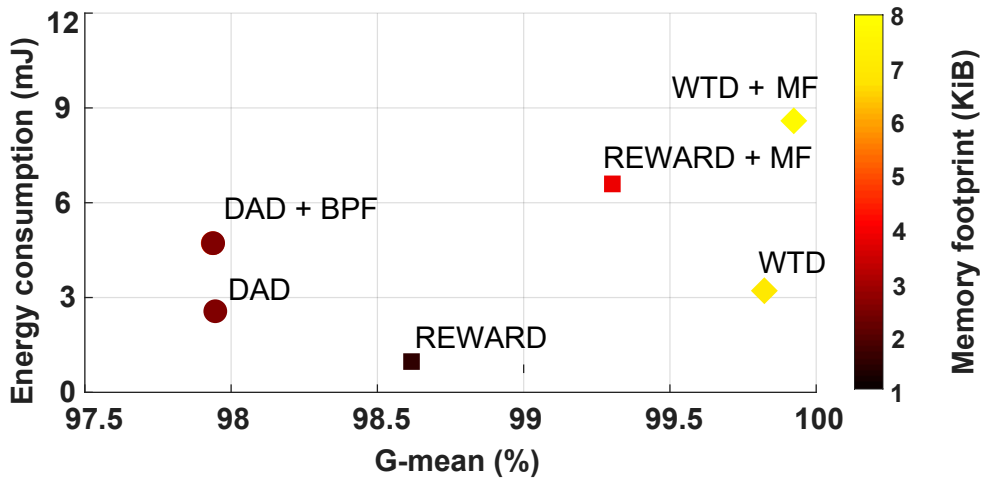


Figure 2.8 – The energy and RAM consumed by the R peak detection algorithms with their respective filters versus their accuracies tested on the QTDB.

Moreover, the results indicate that the new optimized REWARD algorithm is the optimal choice for wearable medical devices in which on-board machine learning is necessary, since its low energy consumption and memory footprint leave room for additional processing capabilities. For example, using REWARD for R peak detection would leave RAM available for complex cardiological analysis, such as the HR variability analysis algorithm described in [86] (which consumes 8 KiB of RAM), and the atrial fibrillation detection algorithm in [42] (which uses 2 KiB of RAM). The implementation of such complex analyses on-board enables real-time feedback to the user in case of complex pathologies.

2.3 Ultra-Low Power Heart Rate Estimation Using a Wearable Photoplethysmographic System

On top of ECG analysis and R peak detection, another relevant biosignal for wellness monitoring is the PPG waveform. This signal is acquired using an optical sensor that illuminates the skin and collects the light reflected by the tissues. The changes in blood volume within the vessels represent the subject pulse and affect the light absorbed so that the pulse is visible in the

PPG waveform. PPG analysis is very useful for blood pressure detection [87], oxygen saturation, and HR estimation [88] in various physical conditions. However, the PPG signal is strongly affected by MAs, which must be removed for accurate vital parameters estimation. In this section, I present a new approach for HR estimation from a PPG waveform during intense physical exercise. This method uses FFT analysis without signal reconstruction.

First, I describe the background of PPG-based WSNs (c.f. Section 2.3.1). Then, I describe the complete method (c.f. Section 2.3.2) and subsequent optimizations for resource-constrained embedded devices (c.f. Section 2.3.3). Then, I present the accuracy of the algorithm based on the database described in Section 2.3.4. Finally, I measure the execution time and estimate the energy consumption on a real-life device and present the results in Section 2.3.5.

2.3.1 Background and Related Work

Wearable systems based on PPG apply a non-invasive, low-cost, optical technique, which detects the blood volume changes in vessels. They contain a sensor that consists of a LED illuminating the skin and a photodetector receiving the light reflected from the tissues. The alternating current component of the PPG waveform gives information about the cardiac rhythm, therefore, considering the PPG frequency spectrum, it is possible to estimate the HR within its standard range between 0.67 Hz and 3.67 Hz, i.e., 40 BPM and 220 BPM. This range corresponds to the standard values of HR from a rest condition to an intense physical exercise.

During exercises and physical activities, the PPG signal can be affected by strong MAs in the same range of frequencies, which must be removed to make accurate HR estimation. There are commercial wrist-based devices for fitness applying algorithms for MAs removal, such as Mio Alpha 2 [89], which presents a high accuracy but a considerable variation from subject to subject [90]. Additionally, different research studies present various methods for estimating HR in corrupted PPG signals. One of the common methods used is the periodic moving average filter, based on the quasi-periodicity of the PPG signal [52]. This filter segments the signal into periods and resamples each period. However, in-band noise occurs when the spectra of the MAs and

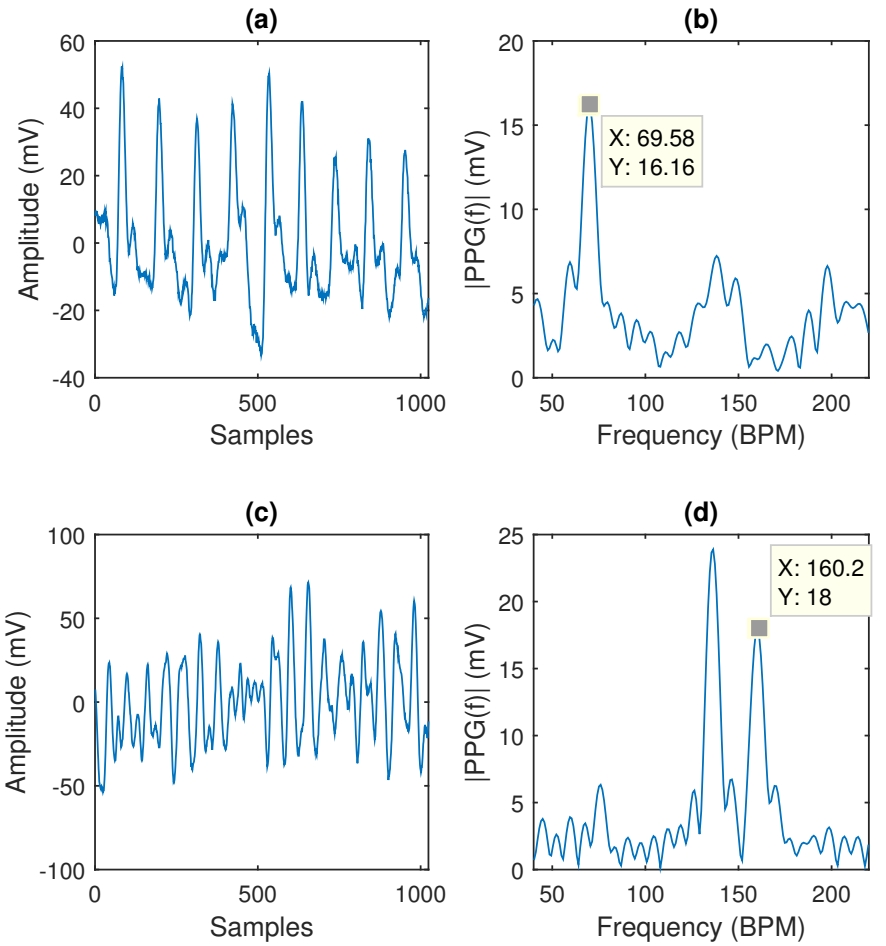


Figure 2.9 – PPG signals and corresponding spectra at a sampling frequency of 125 Hz. (a) represents a PPG signal of a subject at rest and (b) its single-sided amplitude spectrum. (c) represents a PPG signal while the subject is running with strong MAs and (d) the corresponding spectrum.

the PPG one overlap. Another technique is using acceleration-based adaptive filters. They require a reference signal to minimize the mean squared error between the filter output and the reference [53]. One relevant framework for motion removal from the PPG signal is TROIKA [54], which claims that the three steps of the method are necessary for this purpose. The method applies a signal decomposition, which partially removes MAs frequency components and reconstructs the noise-free signal. In addition, it applies a sparse signal reconstruction for high-resolution spectrum estimation, which requires solving an optimization problem to improve performance. The last step is a spectral peak tracking with verification, which analyzes the PPG spectrum to detect the HR as a peak and verifying it by looking at previous windows. The method shows an average absolute error (see Section 2.3.4) for HR estimation of 2.34 BPM, with a standard deviation of 0.82 BPM. Another work was proposed by the same authors [55], but it is not designed for embedded devices as we target in this contribution.

2.3.2 Proposed Algorithm for HR Estimation

During physical activity, the PPG signal is strongly affected by MAs. In fact, periodical MAs appear in the PPG frequency spectrum in addition to the pulse rate. Fig. 2.9 shows an example of PPG signals of a subject at rest (a) and during running (c), and the corresponding single-sided spectra obtained by applying the FFT (b, d). In the spectrum (d), the peak due to the MAs is highlighted and shows a high amplitude compared to the peak corresponding to the HR.

A simple method to detect and remove MAs is adopting a 3-axis accelerometer. This gives information about proper acceleration in a 3-axis reference system due to the movement of the body part where the system is worn. Moreover, the selection of the wavelength of the LED employed is relevant to decrease MAs in the PPG signal. The green light (530 nm) was shown to be more suitable than red, blue, and near-infrared light for monitoring the HR in daily life due to its relative freedom from MAs compared to other wavelengths [91, 92]. Therefore, I use a PPG system with a green LED in reflectance mode, since the chosen measurement position is the upper arm or the wrist.

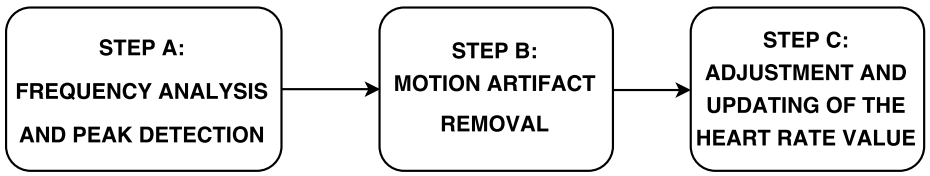


Figure 2.10 – Block diagram of the main steps of the proposed algorithm

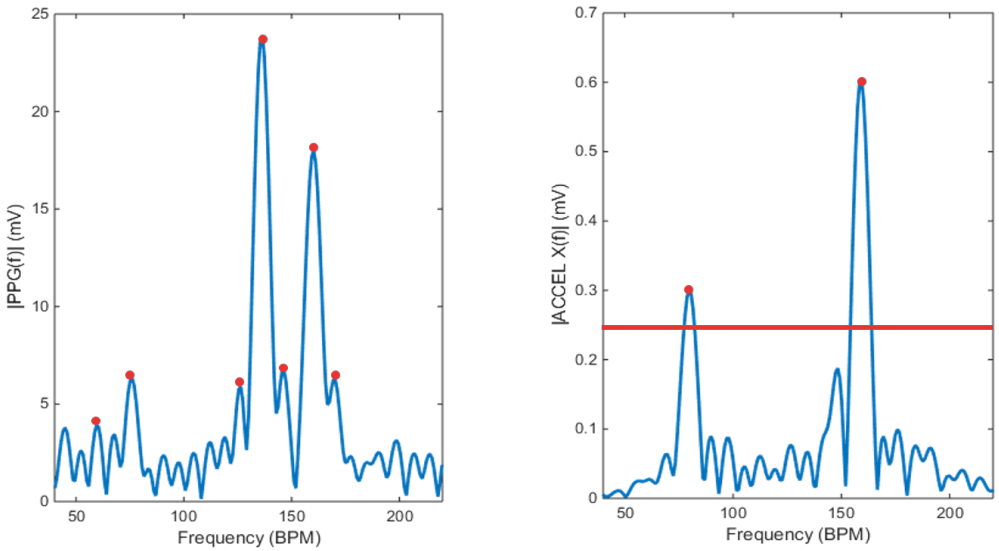
The developed algorithm is divided into three main steps, as the block diagram in Fig. 2.10 shows, which are described in detail in the following sections.

2.3.2.1 Frequency Analysis and Peak Detection

The first step of the method is a frequency analysis based on the FFT and subsequent peak detection. Searching for the pulse rate only within the FFT spectrum and not using additional complex signal decomposition and reconstruction can reduce significantly the computational time. Since the PPG signal is non-stationary and quasi-periodic, a Fourier series analysis is not directly applicable. It can only be applied on a cycle-by-cycle basis [93]. Therefore, I have chosen the use of the FFT applied to short windows of data (8 s) sliding by 1 s per iteration, assuming that the main frequency is stable.

Before computing the FFT, a band-pass FIR filter is applied to the PPG signal to remove low and high frequencies with cut-off values set at 0.5 Hz and 10 Hz, which account for the full range of frequencies needed. Next, the algorithm computes the FFT and the corresponding spectrum for both PPG and accelerometer signals, within the range of frequencies needed to find the pulse rate and MAs (40 - 220 BPM). In order to lower the FFT resolution, the actual window length of the input signal to the FFT is increased with a 7:1 ratio of zero-padding from the initial 8-second window of data. Given a sampling frequency of 125 Hz and a window of 8 s (1024 samples), the actual window length is set at 8192 samples so that the resolution is $\frac{125}{8192} = 0.0153$ Hz. The window length is set as a power of two because it is faster to calculate [94].

The pulse rate and MAs are represented as peaks within the FFT spectrum of the PPG. Similarly, the MAs are peaks within the FFT spectrum of the accelerometer signals within the same range. The method detects k biggest



(a) PPG spectrum

(b) Accelerometer spectrum

Figure 2.11 – In the PPG spectrum on the right, the k biggest peaks are chosen since the HR can be very low compared to the MAs. In the accelerometer spectrum (one for each axis), a threshold based on the maximum peak is applied to find the peak corresponding to the MAs

peaks within the PPG spectrum, as shown in Fig. 2.11a. In fact, in the PPG spectrum the pulse rate can correspond to a low peak compared to the MAs, hence, a higher number of peaks must be kept to account for its amplitude variability within the spectrum. In the accelerometer spectrum (Fig. 2.11b), the peaks related to the MAs to keep are chosen based on a threshold relative to the maximum peak. The peaks detected from the spectrum of each axis of the accelerometer are used as a reference to remove the MAs in the PPG spectrum, as well as extract the correct peak corresponding to the pulse rate. A complete analysis of what happens when the HR is synchronized with the cadence is provided later in this chapter.

2.3.2.2 Motion Artifacts Removal

To remove the MAs from the PPG spectrum and extract the HR, the algorithm takes into account different events occurring in the signals. As the PPG spectrum is a superimposition of different frequencies due to the pulse rate and MAs, the frequency at which the movement occurs is not exactly the same as the one appearing in the accelerometer spectrum. Therefore, in order to detect MAs in the PPG spectrum, the algorithm sets a tolerance for the frequency of the movement occurring in the PPG spectrum. This interval depends on the frequency in the accelerometer spectrum, and it is set at $\pm 2\,\%$ from it. Fig. 2.12 shows the spectra of both PPG and X-axis accelerometer. The peak highlighted in the figure, with value 160.2, is removed because it is lower than $159.3 * 1.02 \simeq 162.49$, as the condition requires.

The main peak of the PPG could sometimes correspond to the horizontal movement of the arm or the wrist, and it must be removed. In fact, if we take a 3-axis reference system on the upper arm or the wrist and consider the Z axis as the vertical movement, the frequency along Z corresponds to the step frequency, unlike the horizontal one, which corresponds to half of the step frequency, namely, to complete an arm swing two footsteps are necessary [95]. The step frequencies range reached during this activity starts from 2.2 Hz (fast walking, corresponding to 132 BPM) to 3.2 Hz (190 BPM) or even more, as extrapolation of previous studies [96]. While running, the HR can reach values from 70 % to 90 % of the maximum HR (from 105 BPM to 140 BPM as minimum value) [97], far from the maximum frequency of horizontal

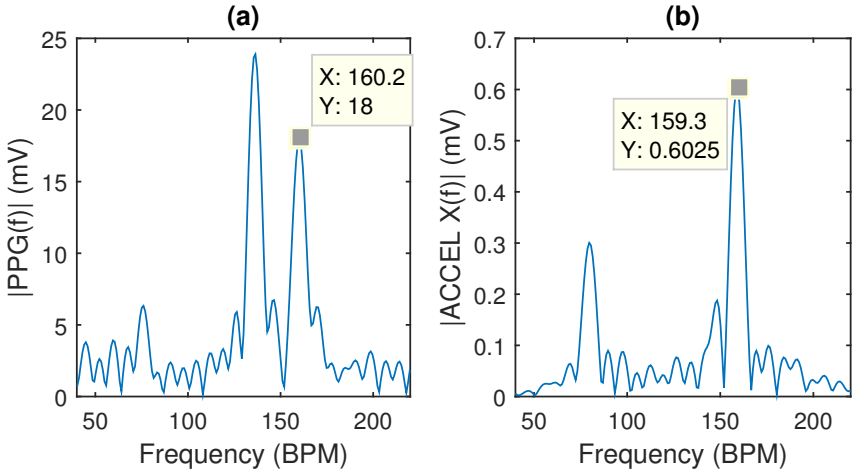
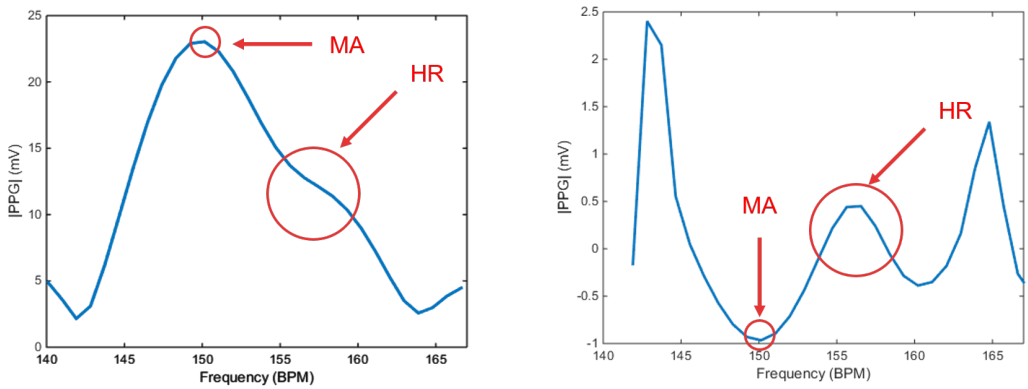


Figure 2.12 – Peak recognition in PPG spectrum when it is in the range of one of the accelerometer peaks. (a) PPG spectrum. (b) Accelerometer spectrum.

movement considered. Therefore, if there is a peak in the accelerometer at half of the frequency of the maximum considered (190 BPM, that is 95 BPM), it is removed from the PPG peaks.

If the pulse rate gets closer to the step frequency, there are two main strategies that the algorithm follows corresponding to two conditions. One strategy is used for discerning the peak corresponding to the HR that is close to the MAs, though with a low amplitude. Another strategy deals with the event where the HR is merged with the accelerometer peak. The first strategy mainly checks the neighbourhood of the dominant peak and the distance between this peak and each of the other peaks within the PPG spectrum. If one of the peaks of the neighbourhood is close to the dominant peak (less than 10 BPM) and tall enough (20 % of the maximum amplitude), it is likely that it represents the real HR while the dominant peak represents the step frequency and it is removed. The parameters of distance and threshold on the amplitude are set depending on the resolution of the FFT and the corresponding spectrum. The second strategy is used when close peaks within the neighbourhood are not detected. In this case, the HR value is likely merged with the PPG main peak. To detect it, the algorithm computes the 2nd order derivative of the



(a) PPG spectrum

(b) 2nd derivative

Figure 2.13 – Sometimes the HR can be merged with the MAs peak since their values are close to each other. To discern them, the method checks for peaks in the 2nd derivative that represent the inflection point in the PPG spectrum, namely, the HR

spectrum and then finds the peaks. As shown in Fig. 2.13, the potential HR in the PPG spectrum is represented by an inflection point within the MAs peak. In the 2nd derivative signal, the MAs corresponds to a zero-crossing while the potential HR is a maximum. Therefore, within the PPG spectrum, the MAs frequency point is removed while the inflection point is kept as potential HR. However, if there is only one discernible peak in the PPG spectrum that corresponds to MAs frequency, the peak is kept as potential HR as it is highly likely that the HR is synchronized with the cadence (i.e., step frequency).

After removing all the MAs, the algorithm checks that the remaining peaks, which are potential HR values (only the biggest two are kept), do not vary significantly compared to the previous five HR values. The maximum variation between the HR from one window to another is set to 5 BPM. This value assumes that in one second the HR does not vary more than 5 BPM. Therefore, the maximum variation from the 5th previous windows is 25 BPM, from the 4th 20 BPM, etc.

2.3.2.3 Adjustment and Updating of the HR Value

Algorithm 2 shows the third and final step of my proposed method. If the PPG spectrum main peak exceeds the maximum range of variation of HR set for the previous five windows ($flagHROK = 0$), then the peak needs to be adjusted. This can happen when the acquired PPG signal is noisy compared to the accelerometer one. Therefore, the peaks found in the PPG spectrum are mainly MAs related. This problem can lead to an accumulated error,

Algorithm 2 Adjusting and updating HR considering previous window

if $flagHROK = 0$ **then**
    $HR_{new} = HR_{old} + sign(mainPeak_{ppg} - HR_{old}) \times 2$
else
    if $lengthPeaks_{PPG} > 0$ **then**
        $dist_{toLastPeak} = 1000$
        for $i = 0$ to $lengthPeaks_{PPG}$ **do**
            $dist = dist(p_{PPG}(i), HR_{old})$
            if $p_{PPG}(i) \neq 0$ & $dist \leq dist_{toLastPeak}$ **then**
                $HR_{new} = HR_{old} + sign(p_{PPG}(i) - HR_{old}) \times min(dist, 5)$
                $dist_{toLastPeak} = dist$
            end if
        end for
    end if
end if
if $lengthPeaks_{PPG} > 0$ **then**
    $HR = HR_{new}$
    $HR_{old} = HR_{new}$
else
    $HR = HR_{old}$
end if
$hr_{prev}(pr) = HR$
▷ This code is executed after removing MAs and checking five previous windows

therefore, the algorithm decreases the maximum variation of the current HR from the previous window to 2 BPM. The direction of the variation is chosen as the position of the current PPG main peak from the previous HR value ($sign(mainPeak_{ppg} - HR_{old})$). The HR value in the current window

is updated as shown in the formula in Line 2, where HR_{old} is the HR value in the previous window.

If no error occurs, the algorithm checks the distance between every remaining peak and the previous HR. The closest one is chosen as the suitable HR value. In this case, the current window is updated as shown in the formula at Line 9, where $dist$ is the distance of every PPG peak from the old HR. If the distance is lower than 5 BPM, the algorithm updates the value using $dist$, otherwise it updates the HR by 5 BPM. Finally, if the algorithm does not detect any peaks, it assigns the value of the previous window as the current HR, as shown in Line 19.

2.3.3 Optimizations for Execution in Wearable Sensor Nodes

To better execute the presented algorithm on WSNs, I present two main optimizations. The first one is the use of integer arithmetic to lower the execution time of the FFT routine and to generally reduce the memory space allocated for the computation. In this contribution, the main portion of computation time gained is related to the FFT routine. Therefore, I chose to use a fixed-point short integer FFT [98] faster than the one using floating-points. As a second optimization, I downsampled the acquired data without losing accuracy and resolution of the FFT. In fact, the device used in this contribution contains a PPG sensor and a 3-axis accelerometer sampling data at 125 Hz and 250 Hz and an ULP 32-bit MCU with a 48 KiB RAM. Let us compute the initial memory footprint of the window of data and the sliding window used. First, to have the same sampling frequency for both PPG and accelerometer to compare their spectrum with the same resolution, I downsample the accelerometer to 125 Hz. Then, I compute the memory footprint of both PPG and accelerometer data considering their size in bytes. The final memory footprint of the window of data used by the FFT is:

$$mem_{wind} = 2 * 1024 * 4 + 2 * 3 * 1024 * 2 \simeq 20\,\text{KiB} \tag{2.3}$$

where 1024 represents the length of the window buffer of 8,192 s of data (power of two used for the speed of the FFT) at the specified sampling frequency, in samples. The value is multiplied by two because I use a circular buffer to update the window and a sorted buffer as actual input to the algorithm. The first addend represents the PPG data, which consists of 22-bits per sample; therefore, we store integer values, each occupying four bytes of memory. The second addend represents the 3-axis accelerometer data, short integer values, each occupying two bytes of memory. The memory footprint of the sliding window of data used for updating the HR value is:

$$mem_{slide} = 2 * 128 * 4 + 2 * 3 * 128 * 2 \simeq 2.5\,\text{KiB} \tag{2.4}$$

where 128 represents the length of the sliding buffer of 1,024 s of data at the specified sampling frequency, in samples. The value is multiplied by two because we use two buffers in the interrupt routine in order to execute the algorithm routine while the signal is sampled.

Since the memory portion stored at 125 Hz is almost half of the memory available, I reduce the sampling frequency to 31.25 Hz that, according to the Nyquist-Shannon sampling theorem, can represent a bandwidth of $\frac{Fs}{2}$, that is 16 Hz, greater than the maximum frequency of 3.67 Hz considered in the algorithm. With this second optimization, I manage to reduce the memory footprint of approximately 25 % using 31.25 Hz compared to 125 Hz of sampling rate, that is 5.8 KiB.

2.3.4 Experimental Setup

The first validation of the algorithm was conducted in Matlab R2014b and involved twelve datasets provided by the 2015 IEEE Signal Processing Cup [99]. They were recorded when subjects performed various physical exercises. Two-channel PPG signal and 3-axis accelerometer were recorded from the subject's wrist and one-channel ECG from the subject's chest as ground-truth of the HR, each sampled at 125 Hz. As the ECG-based HR is updated every 2 s, the output of the algorithm is also plotted every 2 s, even if computed every

second. I used only one of the two PPG channels and the 3-axis accelerometer data. The algorithm is applied to signals sampled at 125 Hz and downsampled at 31.25 Hz, on a data window of 8 s. For the validation, the output of the algorithm is plotted compared to the ECG ground-truth value. Two types of analysis are conducted to show the behaviour of the algorithm: Average Absolute Error (AAE) and median value.

$$AAE = \frac{1}{N} \times \sum | HR_{alg}(i) - HR_{true}(i) | \tag{2.5}$$

where N is the number of window steps considered, $HR_{alg}(i)$ is the output of the algorithm at each step and $HR_{true}(i)$ is the ground-truth value from the ECG. The AAE is used in order to compare it with the existing algorithms, while the median shows the difference of the two values not biased by small or big values. It is computed considering the values estimated and the ECG after 15 steps, that is 30 s, giving the algorithm time to reach stability.

The on-board processing was implemented in C, using the device mentioned in Section 2.3.3. The sampling of PPG and accelerometer signals were simulated storing static arrays of 30 s of data for the twelve subjects considered and filling the buffers in the interrupt routine. The execution time of the algorithm routine is computed on one of the signals on a 30 s window of data. The power consumption of the device is computed considering the duty cycle of the PPG sensor for both receiving and transmission modes, the accelerometer, the MCU active and sleep mode, and the execution of the proposed HR estimation algorithm.

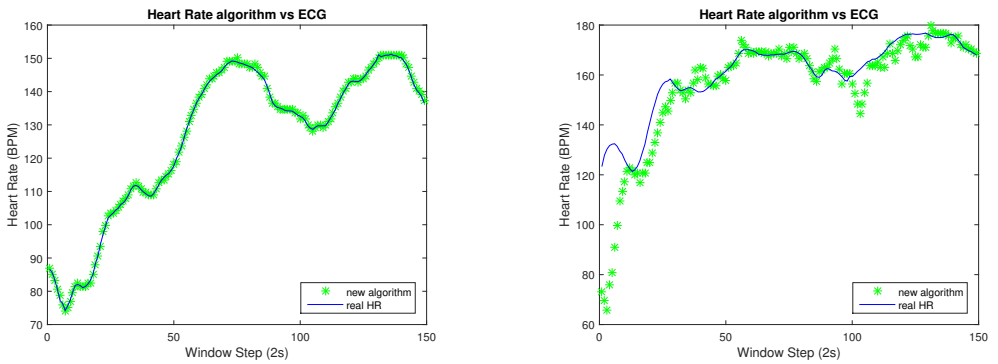
2.3.5 Results and Validation

Table 2.3 shows the AAE of the twelve subjects and the median value for both sampling frequencies mentioned in Section 2.3.4. It also shows the average and standard deviation of the AAE and median value for the twelve subjects.

Fig. 2.14a and Fig. 2.14b show the results for two of the twelve subjects, the best and worst case, for a sampling frequency of 125 Hz. As shown in Ta-

Table 2.3 – Analysis on twelve subjects of SPC 2015 - AAE and median in BPM for signals sampled at 125 Hz and 32.15 Hz

	Analysis in BPM			
	125 Hz		31.25 Hz	
	AAE	Median	AAE	Median
S1	1.62	0.91	1.87	1.40
S2	1.42	0.71	2.97	2.19
S3	1.26	0.65	2.08	1.82
S4	1.40	0.49	2.53	1.73
S5	0.61	0.39	1.56	1.48
S6	1.55	0.60	2.00	1.30
S7	0.47	0.45	1.24	1.18
S8	0.43	0.39	2.10	1.7
S9	0.36	0.31	1.43	1.23
S10	3.78	2.03	4.99	3.04
S11	1.14	0.80	1.49	1.31
S12	1.16	0.77	2.60	2.20
MEAN	**1.27**	**0.7**	**2.24**	**1.71**
STD	**0.91**	**0.46**	**1.01**	**0.54**



(a) Best Case *(b)* Worst Case

Figure 2.14 – Heart rate estimated compared to ECG true HR for best case subject (a) and worst case subject (b), for a sampling frequency of 125 Hz

ble 2.3, at 125 Hz, the AAE for the twelve subjects is 1.27 BPM ± 0.91 BPM, which compared to the TROIKA firmware (2.34 BPM, c.f. section 2.3.1) is 1 BPM lower and the maximum value (1.27 + 0.91 = 2.18 BPM) of AAE is 0.16 BPM still lower than the TROIKA average result. Eleven subjects out of twelve show an AAE lower than 1.7 BPM. In contrast, an outlier shows an AAE of 3.78 BPM, which compared to the TROIKA is only 1 BPM more, acceptable considering the advantages of the method. The outlier is shown in Fig. 2.14b: after reaching stability the algorithm follows the ground-truth pretty well, even in the worst case subject. The best subject is presented in Fig. 2.14a, which has an AAE of 0.36 BPM. The median value in the worst case is 2.03 BPM and in the best case is 0.31 BPM, showing that the unbiased difference between the two values is very low. At 31.25 Hz, the AAE for the twelve subjects is 2.24 BPM ± 1.01 BPM, 1 BPM more than the one at 125 Hz, because of the resampling precision. The value range is still comparable to the performance of the TROIKA framework. The median value is lower than the AAE because it is not biased by small or big values. The outlier has an AAE higher than the mean value, but it is still acceptable considering the advantages for real-time processing. The results show clearly that it is possible to avoid signal reconstruction and focus directly on the spectra of the PPG and accelerometer and relative peaks, obtaining high performance in terms of accuracy.

Therefore, the method is suitable to be implemented on embedded systems and Table 2.4 shows the absolute error between the post-processing results and the same data streamed on an actual wearable device. The average error has a mean value of 0.42 BPM which is due to the approximation precision for using integer arithmetic implementation on device, while in Matlab the HR is computed as a floating-point value.

I computed the execution time of the algorithm routine, obtaining an average of 226 ms per second, which is the time between two outputs of the algorithm. Table 2.5 shows the total average current computed considered the features mentioned in Section 2.3.4 to retrieve the total amount of power consumed while the device is fully working.

Table 2.4 – Absolute Error (AE) between the HR value computed in post-processing and the one computed in the embedded device considering a sampling frequency of 31.25 Hz

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**
AE (BPM)	0.41	0.37	0.3	0.2	0.26	0.27	0.36	0.24	0.31	0.40	0.24	1.72

Table 2.5 – Average current consumed by HW components and algorithm processing with a sampling frequency of 31.25 Hz

	Current (mA)	Time (%s)	Average current (mA)
PPG Tx On-Rx On	1.225	10%	0.123
PPG Tx Off-Rx On	0.6	10%	0.06
Accelerometer	0.5	100%	0.5
Baseline current	0.045	100%	0.045
HR processing	10.5	23%	2.415
Sleep mode	0.018	77%	0.014
Total			**3.16**

Considering the battery rating of 710 mA h the device can successfully achieve a battery lifetime of 9.37 days.

2.4 Real-Time Personalized Atrial Fibrillation Prediction on Single-Core Wearable Sensors

After describing a set of algorithmic optimizations for vital parameters estimation in multiple biosignals, the next step in achieving an optimal energy-accuracy trade-off in remote wellness monitoring is to focus on relevant pathology detection through personalized methods. As described in Section 1, within NCDs, CVDs are the major cause of death globally. One CVD, the PAF, a type of arrhythmia, is one of the key causes of stroke and heart failure [56]. In this section, I propose a new online PAF prediction model targeting ULP single-core WSNs, which considers the specific features of the individuals and their condition. First, I describe the background of the pathology (c.f. Section 2.4.1) and a preliminary analysis of how this patient-specific algorithmic optimization achieves high accuracy compared to state-of-the-art algorithms that consider inter-patient variability (c.f. Section 2.4.2). Then, I present the optimization for ULP single-core WSNs and show how the method also reduces energy consumption and processing execution time (c.f. Section 2.4.3 and Section 2.4.5). Finally, I describe the scalable battery lifetime achieved by implementing the method, personalized to the characteristics of each patient, on a real-life ULP ECG monitoring device, INYU [24], (c.f. Section 2.4.5).

2.4.1 Background and Motivation

Wearable sensors allow monitoring specific characteristics of a pathology by measuring bio-signals with non-invasive sensors, e.g., ECG and electroencephalography (EEG). Therefore, they can enable accurate detection and prediction of major NCDs and allow people to self-assess their health status [20–23,25,26]. Despite recent advances in wearable technologies, essential challenges exist to exploit such systems fully. In particular, energy efficiency and scalability (i.e., according to the specific pathology characteristics of each patient) are fundamental factors to take into account in any wearable sensor design [100] for personalized remote long-term health monitoring [23–28]. In

the context of PAF prediction, I propose a new online personalized method targeting single-core WSNs, which scales the computation based on the characteristics of each patient.

AF is a type of arrhythmia, defined as a quivering or irregular heartbeat [56]. The majority of patients suffer from an initial paroxysmal form (PAF), which progresses into a persistent or permanent arrhythmia. Moreover, a significant proportion of patients affected by PAF are initially asymptomatic, and PAF events may be undiagnosed, risking complications, such as stroke and heart failure [56]. According to the European Society of Cardiology (ESC) 2016 guidelines [56] for the management of AF, the aim is to reduce the frequency of episodes, prevent complications and alleviate symptoms, in the case of PAF. Furthermore, PAF patients may be suitable for domiciliary self-treatment, using the so-called "pill in the pocket" approach [56], that is, a single oral dose pharmacological cardioversion. Therefore, continuous personalized monitoring of PAF patients by predicting a recent-onset episode may shorten the initiation of the treatment, as well as the time to resolution of symptoms. Additionally, this prediction is particularly relevant for treating asymptomatic episodes, which remain unseen otherwise.

Different studies describe the prediction of PAF onset, by analysing changes in the surface ECG from few minutes to few hours before the onset. The classical approach is to consider premature atrial complexes (PACs) and P-wave variability [101, 102] in the 30 minutes prior to the onset. PACs are premature beats originating in the atria from ectopic pacemaking tissue active before the sinoatrial node. Zong et al. [101] detect the PACs and predict the PAF onset based on a measurement of PAC rate weighted for different windows in the signal, favouring the closest to the onset. Schreier et al. [102] analyse the P-wave morphology of both regular and premature beats. Then, they extract the probability that a specific degree of P-wave variability is associated with a PAF episode. Other approaches consider the P-wave non-linear dynamics to achieve higher accuracy in the prediction two hours prior to the onset [59]. However, AF is caused by heterogeneous mechanisms in different patients, and the therapeutic strategies should derive from the individual conditions. Different works report patient-specific modelling, in particular for ECG signal analysis. Indeed, two works report automatic patient-specific classification

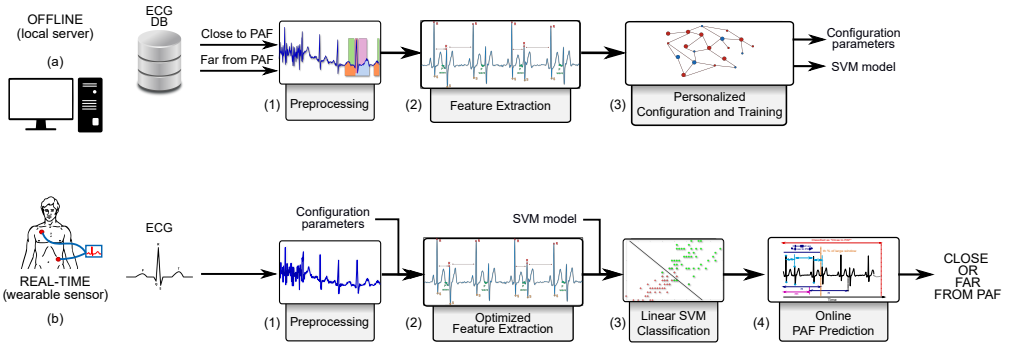


Figure 2.15 – Block diagram of the personalized PAF prediction. On top (a), the blocks of the configuration and training phase. At the bottom (b), the real-time prediction process executed on a wearable sensor.

of normal or premature beats considering swarm optimization feature selection [103, 104]. In order to draw a comparison with my method, I report two key cases of the recent literature [59, 105], describing offline methodologies, which include inter-patient variability and achieve higher accuracy compared to other methods [101, 106, 107]. I apply my method on the same dataset used in these works, the PAF prediction challenge [108], which is described in detail in Section 3.5.3. In this case, the prediction of a PAF onset is defined as a classification method of a 30-minute ECG excerpt as one of the two labeled classes, far from or close to the onset. Based on this definition, I compare the accuracy results of the online design with the two state-of-the-art methods and my preliminary work on an offline PAF prediction [72].

2.4.2 Personalized PAF Prediction Method for Long-Term Monitoring on Wearable Sensors

The personalized PAF prediction approach presented in this section is designed to be used for long-term monitoring. First, the method trains a personalized model on a set of ECG signals previously acquired from a single patient. Then, the model is applied to a newly acquired ECG signal from the same patient, producing an output at least every 15 s to 45 s.

Fig. 2.15 shows the methodology for a single patient. In Fig. 2.15a, I show the offline method running on a local server, which performs the personalized analysis on previously acquired ECG data for the target patient. These data need to include one excerpt of ECG signal before the onset of a PAF event (i.e, "Close to PAF"). Additionally, the approach needs one excerpt of normal sinus ECG signal at least 45 min far from any event (i.e., "Far from PAF"). The data is split into separated training and test set for each patient. I choose a training window of at least 350 R peaks for both excerpts, which corresponds to 3–9 min considering a HR varying from 40 BPM to 110 BPM, as done in [72]. For the excerpt "Close to PAF", the training window is extracted right before the PAF onset, while the test window starts from 30 minutes away and stops before the training window starts. Moreover, to avoid overfitting and ensure robustness and generalization of the training model, the approach includes a learning curve analysis [109] on the number of training samples and a 5-fold cross-validation (CV).

Referring to Fig. 2.15a, the ECG signals are, first, preprocessed by filtering and delineating their main waves, in Step 1. Then, the methodology extracts features from small windows of consecutive beats to capture short events that can happen before a PAF onset (Step 2). The features are the input to Step 3, called "personalized configuration and training". The training is done using a linear classifier based on support vector machines (SVMs) and a 5-fold CV. The output of the offline process is a subset of optimal features specific to the patient and the corresponding classification model. After the personalized configuration parameters, the selected features and the classification model are loaded in the embedded device, which runs the automatic PAF prediction, shown in Fig. 2.15b. Once the device acquires a small window of consecutive beats of ECG signal, the algorithm starts the preprocessing (Step 1) and extraction of a reduced set of features personalized to the patient (Step 2). Finally, using a linear SVM classification model (Step 3), the algorithm performs an online PAF prediction producing a binary output, as shown in Step 4. Additionally, the preprocessing and feature extraction steps are suitable for parallelization (cf. Section 3.5.1).

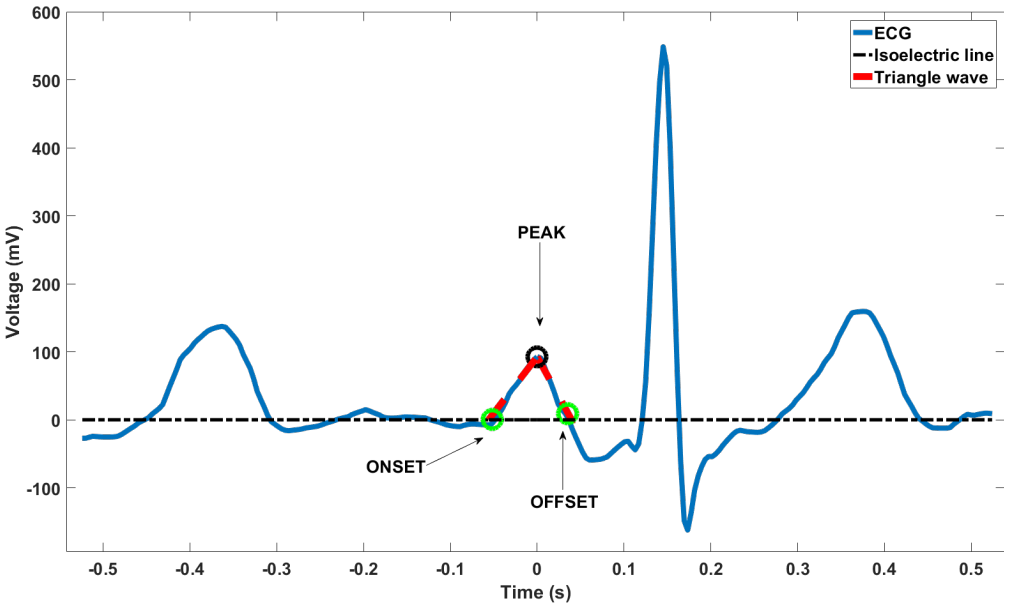


Figure 2.16 – Detection of the onset (offset) of the P-wave based on the minimum Euclidean distance between the P-wave and a triangle wave with a slope depending on the peak and the onset (offset).

2.4.2.1 Preprocessing -- Filtering and Delineation

In both phases of the methodology, the first step is preprocessing the signal. The common step to both phases is filtering, which consists of removing the baseline wandering by applying a MF. I use an implementation proposed by Sun et al. [74] and optimized for an embedded system by Braojos et al. [71]. The MF removes the peaks and valleys of the signal with operations related to the shape or morphology of the signal features. Then, the signal baseline is finally subtracted from the original signal.

For the delineation of both offline and online real-time processing, I use sequentially three methods. First, I use a real-time implementation of the WT delineation to detect the R peak [38, 68] for a design on standard platforms. This method uses the WT of a signal, which is proportional to its derivative with a smoothing impulse response at different scales. By applying the WT to the signal, the R peaks are detected as the zero-crossings that are common

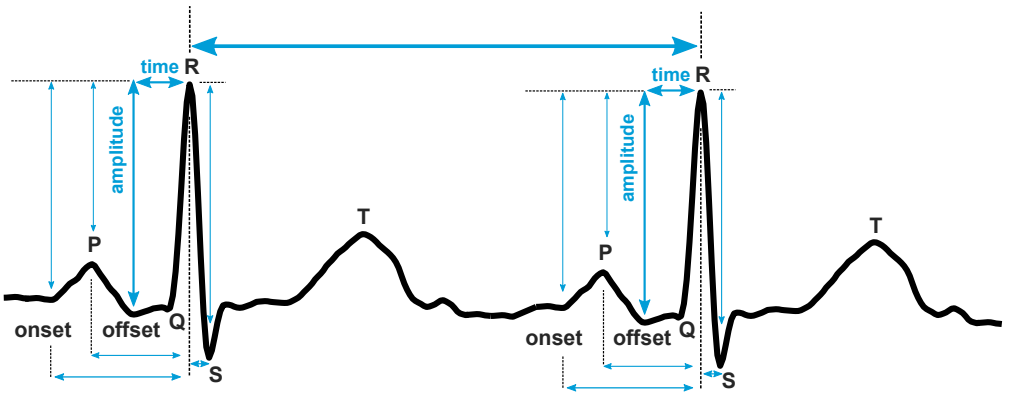


Figure 2.17 – Feature extraction on a small window of two consecutive beats.

across scales 2^1 through 2^4 (where most of the ECG signal energy lies [68]). These zero-crossings must be preceded by a positive peak (i.e., increase in the original ECG main wave) and followed by a negative peak (i.e., decrease in the original ECG main wave) to be detected. Moreover, for the single-core design, I consider a more lightweight R peak detection, called REWARD [41], which is more suitable for personalization (and parallelization, c.f. Section 3.4.1). REWARD includes two main steps: signal enhancement and R peak detection. The signal enhancement method is called Rel-En, and it uses the signal energy to amplify dominant peaks. The R peak detection step generates a set of adaptive hysteresis thresholds to isolate the highly dominant peaks and performs a subsequent check on their widths to eliminate false positives. WT and REWARD are described extensively in Section 2.2 and compared in terms of accuracy and energy consumption running on the same platform. Second, I apply a method described in my previous work [72] to detect the onset and offset of the P wave by comparing it with a triangular wave starting at the P peak and finishing at the isoelectric line, as shown in Fig. 2.16. Finally, the third method delineates the S wave as the minimum point after the R peak, within the standard physiological duration of the QRS complex (approximately 80 ms to 100 ms).

2.4.2.2 Personalized Feature Extraction

The features considered in the prediction algorithm depend directly on the fiducial points delineated, as shown in Fig. 2.17. Specifically, I consider the distance in time from the main fiducial points analyzed (P wave onset, peak and offset, and S) to the R peak within the same beat, and beat-to-beat RR intervals if the fiducial point is the R peak itself. Additionally, I consider the signal amplitude of the five fiducial points related to the amplitude of the closest R peak. These features were chosen since they can be affected by the changes that occur before a PAF event, as depicted in Fig. 2.18 on one ECG segment extracted from the dataset analyzed (c.f. Section 2.4.4.1). Moreover,

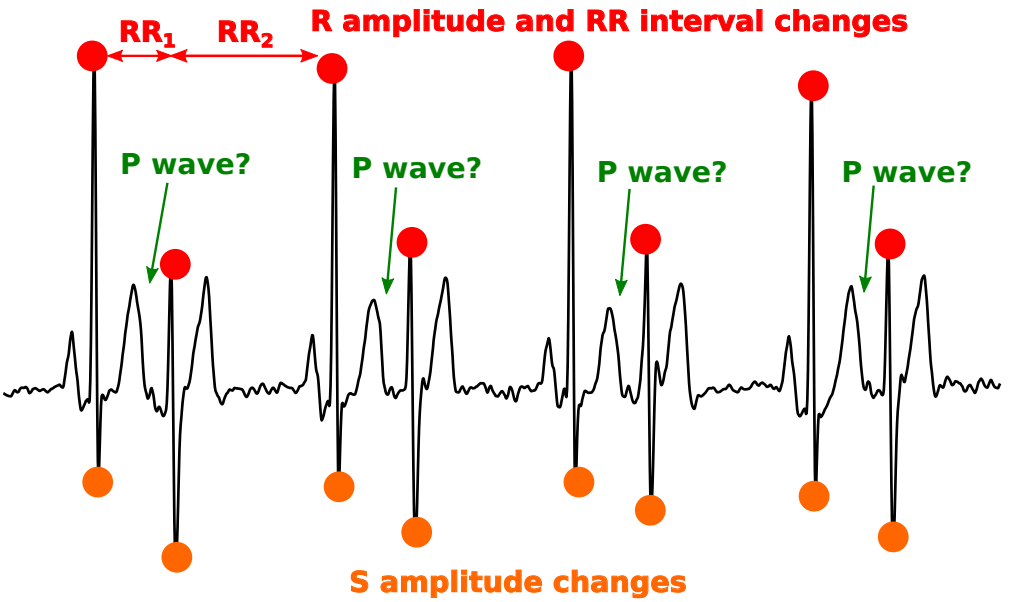


Figure 2.18 – Example of ECG before a PAF onset where sudden changes occur: variable RR intervals and R peaks amplitude, missing P waves and variable S waves amplitude.

the algorithm selects a personalized combination of features in a specific amount of consecutive beats that defines a small window of processing. In this way, the approach analyzes sudden events occurring within the window, as shown in my previous work [72]. Additionally, I include an overlapping window adjusted to the patient. By considering groups of features depending on the main ECG waves, the algorithm trains and chooses the set that gives

Algorithm 3 Offline: personalized feature selection

1: **Input**: windows of n consecutive beats of the filtered ECG; m beats sliding window
2: **Output**: selected $group$ of fiducial points
3: **for** $window = 1, 2, ..., \frac{ECGlength}{m}$ **do**
4: **for** $beat = 1, 2, ..., n$ **do**
5: Extract R peaks
6: Extract P, P onset, P offset, S
7: **end for**
8: **end for**
9: Consider 5 groups of fiducial points: (R); (P, R); (P, R, S); $(P, Pon, Poff, R)$; $(P, Pon, Poff, R, S)$
10: **for** $group = 1, 2, ..., 5$ **do**
11: Train and test SVM with 5-fold CV
12: Compute mean F-score on folds
13: **end for**
14: Choose $group$ with best mean F-score
15: **return** $group$

the best 5-fold CV performance for each patient (F-score), as shown in Lines 9–14 of Algorithm 3. Then, in the online phase, the algorithm only delineates the fiducial points related to the selected group of features. The personalized features selection has the advantage of scaling the computation of the ECG delineation in the wearable sensor by performing a selective delineation for each patient.

2.4.2.3 Personalized Classification Parameters

The offline personalized configuration and training, shown in Fig. 2.15a, includes several steps, other than the feature selection. The personalization of the model for each patient captures the heterogeneity of the PAF pathology and optimizes its implementation on wearable sensors. I use a minimum time resolution window for the prediction of 25 beats. This value varies within 15 s and 45 s, considering a resting HR in the range of 40 BPM to 110 BPM, which is a fair prediction window length considering the physiology of the PAF event occurrence. The first set of configuration parameters includes the training model coefficients. I choose to use a supervised learning model

Algorithm 4 Offline: personalized configuration and training

Input: filtered ECG; $n = 3:7$ consecutive beats; $m = 1:n$ sliding window; selected $group$; $predwlen = 25beats$
Output: selected (n,m); selected th; SVM model (mdl)
function SMALLWINDOWPARAMETERS
    for $(n,m) = (3,1),(4,2),...,(7,n)$ **do**
        Compute fitting and F-score on learning curves
    end for
    Choose (n,m) with best fitting and F-score
end function
function THPARAMETER
    $nw = \frac{predwlen}{m}$ ▷ small windows in $predwlen$
    for $th = 0.1, 0.2, ..., 0.9$ **do**
        Train and test SVM with 5-fold CV
        Count small windows "Close to PAF" ($swpaf$)
        if $\frac{swpaf}{nw} > th$ **then**
            "Close to PAF"
        else
            "Far from PAF"
        end if
        Compute mean F-score on folds
    end for
    Choose th with best mean F-score
end function
Train SVM mdl with selected $group$, (n,m), th
return (n,m); th; mdl

based on SVMs to classify an excerpt of signals labeled "Close to" or "Far from" PAF events. An SVM-based model is selected since at inference time it has very low complexity [110], and it achieved a high accuracy and robustness in the preliminary offline approach [72]. The second set of personalized configuration parameters includes the number of consecutive beats (n) within a small window and the length of the sliding window in beats (m). These two parameters vary within the following ranges: $n = 3,4,...7$ and $m = 1,2,...n$ and affect the number of training samples. The best combination (n,m) was chosen by selecting the most robust model performance using leaning curve analysis [109] and F-score. In my methodology, summarized in Lines 3–8 of Algorithm 4, I analyze the F-score against the number of samples with an increment of 10 samples at each iteration, applying a 5-fold CV for better generalization of the results. Next, within the predefined minimum prediction window, the algorithm cross-validates a threshold on the set of small windows the model classifies correctly, based on (n,m), described in detail in Lines 9–22 of Algorithm 4. The final step of the learning phase consists in choosing the number of minimum prediction windows of ECG samples needed to process and predict the PAF onset, where abnormal events may occur. Considering the heterogeneity of the occurrence of small and sudden events [108], my algorithm learns on both normal sinus rhythm (i.e., signal far from any PAF event) and close to a PAF event to choose the prediction time length for each patient. The algorithm computes the maximum amount of consecutive prediction windows (each of 25 beats) in the normal sinus rhythm misclassified as "Close to PAF", using excerpts from a different set than the training and test ones. Therefore, the updated minimum window of prediction is a multiple of 25 beats, which corresponds approximately to the range from 15 s to 45 s.

During the online PAF prediction monitoring, shown in Fig. 2.15b, once the personalized features within the small window (of n consecutive beats) are extracted, they are fed to the linear SVM model. Then, the algorithm classifies each small window as either "Close to PAF" or "Far from PAF". Next, my algorithm is optimized to stop the processing once the precomputed personalized threshold of small windows classified as "Close to PAF" is reached, reducing the energy consumption and scaling it for different patients. The final output

of the classification is transmitted at least every 15 s to 45 s, according to the minimum window of prediction selected.

2.4.3 Patient-Specific Optimizations for Single-Core Ultra-Low Power Platforms

The parameters and model selected during the learning phase, described in Section 2.4.2.3, are the configuration inputs to apply in the algorithm implemented on a wearable embedded device. With the final goal of saving energy for personalized continuous monitoring, I apply a set of optimizations to implement an online method for a single-core platform.

2.4.3.1 Patient-Specific Online Design for Single-Core Platforms

Considering the model selected at training time for each patient, I present two main optimization techniques that decrease the computation within a window of analysis.

Selective Feature Extraction As described in Section 2.4.2.2, the algorithm only extracts for each patient the group of features within n consecutive beats with a sliding window of m beats, which were selected in our personalized patient configuration phase (see Section 2.4.2.2). Aiming to save and scale computation, hence energy, the algorithm delineates within each consecutive beat only the fiducial points needed for the ECG waves selected at training time.

Fig. 2.19 describes the strategy used to optimize the delineation within a beat for two different patients, and it receives as input their corresponding selected group of features. In Fig. 2.19a, the trained group of features are within the P and R wave, while in Fig. 2.19b they are within the R and S wave. As presented in my previous work [72], the method to compute the P wave onset/offset compares the corresponding wave and a set of triangular waves with origin in the peak, which end in different points of the isoelectric line. This is computationally more expensive than finding a minimum such as S. Therefore, if for the patient in Fig. 2.19b the algorithm would delineate the three main ECG waves the computation load will be much higher. However,

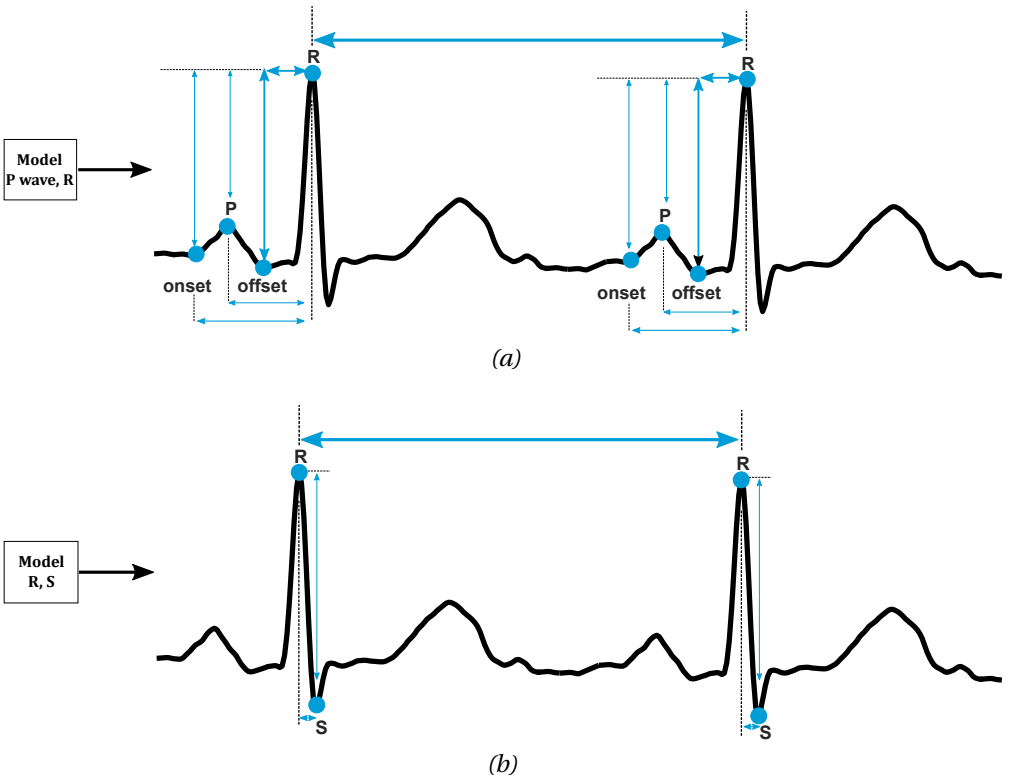


Figure 2.19 – Proposed selective online feature extraction within a beat for two patients with different training models.

if the personalized model for a patient includes full P wave feature set, as shown in Fig. 2.19a, then the algorithm will compute them.

Optimization on the Classification Threshold By applying the threshold on the small windows processed within a minimum window of prediction, defined in Section 2.4.2.3, my approach increases the computational savings in the online PAF prediction process. As an example, Fig. 2.20 describes the real-time prediction on an excerpt of ECG signal before a PAF onset for Patient 1 of the chosen dataset [108] (cf. Section 2.4.4.1). As described in Section 2.4.3.1, the algorithm extracts patient-specific features based on the fiducial points for each beat of the small window of consecutive beats. For this patient, the small window consists of three consecutive beats (n) with a sliding window of two (m), and the features extracted are the time and amplitude of the R peaks and the S wave. Once the features are extracted, they

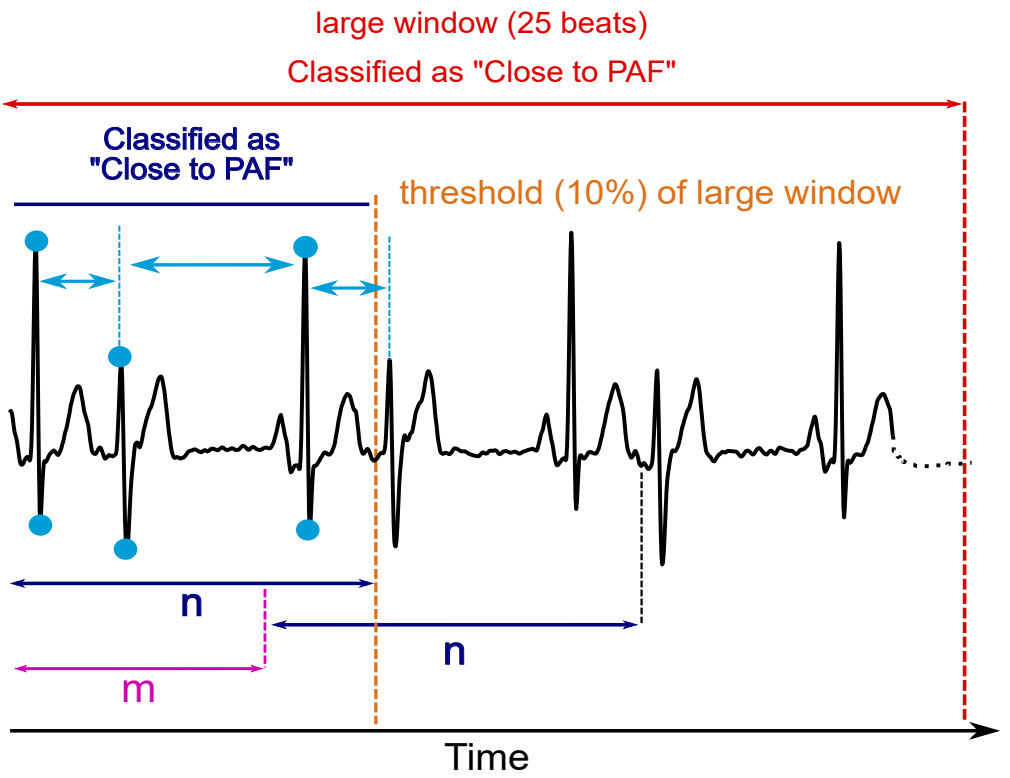


Figure 2.20 – Proposed optimized online prediction of a PAF event for Patient 1 of the chosen dataset (cf. Section 2.4.4.1). The configuration parameters are $(n,m) = (3,2)$, $th = 10\%$.

are fed to the linear SVM model, which classifies the small window as "Close to PAF". Then, using the training parameter (th), the algorithm computes and checks the percentage of small windows classified as positive over a large window of 25 beats to define the whole large window as positive. In the case in Fig. 2.20, the algorithm reaches th with only one small window, i.e., approximately 10 % of 25 beats. Till the end of the current large window, the algorithm stops extracting features and performing the classification. It starts over at the next large window. This implementation allows to save energy within a window of analysis, i.e., 25 beats, by stopping the feature extraction and classification. Finally, this process is different for each patient enabling scalability, thus the adaptability of the device lifetime (cf. Section 2.4.5.2).

2.4.4 Experimental Setup

In this section, I describe the database used for training the models and the test bench for the single-core design.

2.4.4.1 Database for Offline Training and Online Testing

The framework has been tested on the PAF Prediction Challenge (2001) Physionet database [108], containing 53 patients affected by PAF over the two learning and test sets described. The database includes for each patient two 30-minute ECG signals close to and far from (at least 45 min) a PAF event. The signals are acquired at a sampling frequency of 128 Hz and resampled at 250 Hz. In this work, I did not consider the signals acquired from healthy subjects with a normal sinus rhythm, as the part of the challenge related to prediction did not include them. However, the signal far from any event includes for the most part normal sinus beats. For both signals, the personalized model is trained on the last 350 beats of the recording (approximately 3–9 min considering a HR range from 40 BPM to 110 BPM). For the signals close to a PAF event, the method tests on the remaining of the recording. For the signal far from any event, two-thirds of the remaining signal are used to configure the minimum window of prediction (cf. Section 2.4.2.3), and then testing is done on the remaining third.

2.4.4.2 Test Bench and Platforms for Single-Core Design

The single-core design is a sample-by-sample method. Two different R peak detection algorithms are used for testing, a wavelet-based as done in my previous work [72] and REWARD [41]. I report the overall accuracy on the full database with both wavelet and REWARD algorithms. Then, I choose six cases that vary in terms of configuration parameters to evaluate one window of analysis when a PAF event occurs. I consider the window length-related parameters n and m, the selected group of features, and the classification threshold, described in Section 2.4.2.2 and Section 2.4.2.3. Specifically, I select from a worst to a best case of computation in the context of the target multi-core architectures exploration, considering the sum of each configuration parameter computational cost. Finally, the window of analysis varies from 15 s to 45 s, depending on the patient. I measure the energy consumption of the single-core wavelet-based design using the Simplicity Studio software (SW) energy profiler on the Cortex-M3 based EFM32LG-STK3600 [111]. Then, I use this energy measurement to show the battery lifetime estimation on the real-life SmartCardia INYU ECG-based wearable device [24]. Finally, I run the energy profiling on the single-core REWARD-based design for the six cases on the Cortex-M3 platform.

2.4.5 Experimental Results

In this section, I first report the results on the prediction performance of the real-time personalized approach, after the optimizations described in Section 2.4.3, and I compare it with the state-of-the-art offline algorithms. Then, I report the energy consumption of my online single-core design. Finally, I present the scalable battery lifetime achieved by the method while running on a real-life ULP ECG monitoring device [24].

2.4.5.1 Accuracy of the PAF Event Prediction

In Table 2.6, I compare the accuracy of the two inter-patient variability approaches presented by Martínez et al. [59] and Ebrahimzadeh et al. [105], the offline personalized algorithm presented in my previous work [72], my real-time optimized personalized single-core approach considering the wavelet-based R peak detection [38], and the optimized version with the

Table 2.6 – Performance scores for patient-specific (offline and real-time) and inter-patient variability approaches: accuracy, F_1 score (F_1), sensitivity (Sens), specificity (Spec).

		Evaluation parameters			
		Accuracy	F_1	Sens	Spec
Inter-patient variability	[59] Martínez et al. (%)	93.0	–	–	–
	[105] Ebrahimzadeh et al. (%)	98.2	–	100.0	95.5
Personalized	[72] Offline (%)	97.1	97.1	96.2	98.1
	Real-time with WT [38] (%)	93.4	93.6	96.2	90.6
	Real-time with REWARD [41] (%)	91.5	91.6	92.5	90.6

REWARD algorithm [41]. Moreover, I report the F_1 score, sensitivity, and specificity of the personalized approaches over the full dataset. The F_1 score is a measure of the classification accuracy that focuses on the positive rate and it is defined as the harmonic mean of precision (i.e., PPV) and recall (i.e., sensitivity). The sensitivity is defined as the proportion of ECG segments classified correctly as "Close to PAF", while the specificity is the proportion of ECG segments classified correctly as "Far from PAF". In all the cases presented, I define as "prediction" the classification of the two types of segments mentioned before and, consequently, the performance scores refer to this classification.

The online single-core implementation using the wavelet-based delineation reduces the accuracy by 4 % compared to the reference offline algorithm [72], while keeping the same sensitivity. However, the accuracy is comparable with the state-of-the-art offline algorithms. I also used a more lightweight and suitable for personalization R peak detection [41] with a 2 % reduction in accuracy due to the misdetection of peaks. REWARD relies on amplitude thresholds applied to the Rel-En signal within a window of analysis (1.75 s) to detect its peaks. However, if sudden changes in amplitude occur within a window, namely, a small peak follows a tall peak, REWARD can fail in detecting the small peak, hence, the accuracy loss. I explore this issue in Chapter 4, and I propose an adaptive solution that detects failures in REWARD and triggers a more robust algorithm. However, in this case, the accuracy loss of 2 % is acceptable as it is still within the state-of-the-art range.

2.4.5.2 Energy Consumption in Standard Single-Core Platforms

In the EFM32LG-STK3600 Cortex-M3 based sensor, the energy consumed within one window of analysis varies between 16 mJ and 9 mJ for the worst case and best case scenario. The personalized training of the optimized PAF prediction model (cf. Section 2.4.3.1) allows energy savings between patients with 84 % to 56 % difference compared to the worst case of the six selected cases. Then, using the energy results of EFM32, I estimate the battery lifetime of the different components and execution modes of the real-life ULP ECG monitoring device, INYU [24], which uses the same MCU family to run my PAF prediction algorithm. I show the worst case in terms of computational

Table 2.7 – Average current consumed by different modules of the INYU in the worst case

		Current (mA)	Duty cycle (%)	Average current (mA)
ECG acquisition	ADS1291 active [112]	0.427	100	0.427
Accelerometer (idle)	MPU600 [113]	0.005	100	0.005
Data processing	PAF prediction process	10.5	3.2	0.336
Power saving	Low-Power sleep mode [82]	0.018	96.8	0.017
BLE	NRF8001 [114]	0.014	100	0.014
Total				**0.799**

burden, calculated as the execution time of the processing part over the actual prediction time. The maximum operating frequency used on the EFM32 was 48 MHz. However, the STM32 on the INYU device can operate only up to 32 MHz. Therefore, I apply a factor of 1.5 × to the execution time measured in EFM32 to use it for the battery lifetime estimation on the STM32.

Table 2.7 shows the worst case of total average current consumed among the six cases chosen. The duty cycle represents the percentage of computational load of the five main modules of the INYU over a 30-minute window. The ECG signal acquisition is always on for the 30-minute window, as well as the Bluetooth Low-Energy (BLE) module to send the output of the classification every 15 s to 45 s. In addition, by computing the total average current of 0.799 mA consumed within 30 min and a battery of 710 mA h, the battery lifetime of the worst case is 889 hours (approximately 37 days) in the six cases analyzed. In the best case, the battery lifetime is 41 days. This is an excellent result because the system can last more than a month with a single battery recharge.

The results demonstrate how a personalized online PAF prediction algorithm in single-core can save energy depending on the patient features. Moreover, to the best of my knowledge, there are no other algorithms that tackle this problem in embedded devices. In Chapter 3, I show how my personalization can also be applied to adapt an ULP multi-core architecture to the characteristics of each patient.

2.5 Conclusion

Wearable technologies provide accurate, energy efficient means of health and pathology monitoring, prevention, and diagnosis. The first step towards achieving an optimal energy-accuracy trade-off in ULP WSN platforms is focusing on algorithmic optimizations. In this chapter, I provided many examples in various domains and levels of problem complexity, which already achieve a significantly high energy-accuracy trade-off.

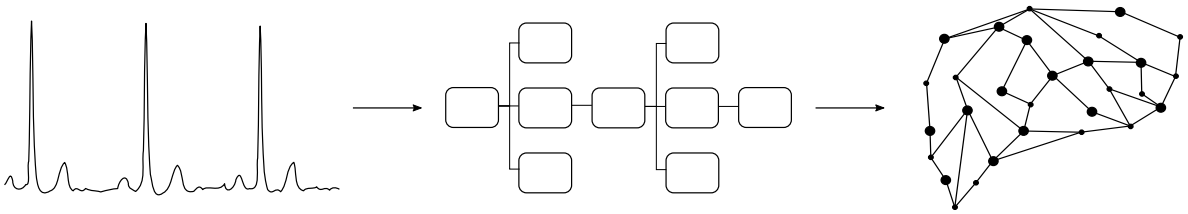
First, I detailed the real-time implementation and optimization, in the context of resource-constrained wearable devices, of the low-complexity Rel-En preprocessing method, as well as the design of a novel R peak detection algorithm to complement it. Furthermore, this contribution has addressed the need for a comprehensive comparison of three state-of-the-art real-time R peak detection algorithms (PT, WTD, DAD), as well as the REWARD algorithm, to determine each algorithm's feasibility of implementation on ultra-low power real-time embedded systems. REWARD was the most efficient compared with state-of-the art R peak detection algorithms, using at least 63 % less energy and 32 % less RAM than the other algorithms while producing comparable accuracy results. This algorithm is highly suitable as a base for more complex cardiovascular analysis algorithms, as illustrated throughout this thesis.

Next, on top of ECG analysis and its R peak detection, I have shown an accurate and energy-efficient HR estimation though using wearable non-invasive low-cost PPG systems. The challenge of using these optical sensors is to accurately detect the HR during physical activities from a signal strongly affected by MAs. Therefore, I proposed a method which applies the FFT on short windows of data and removes MAs depending on the spectra of the PPG

and 3-axis accelerometer signals, avoiding signal complex reconstruction or adaptive filtering. The results showed that the algorithm removes a wide range of MAs, thus achieving a high degree of accuracy. The method was suitable to be implemented in a wearable embedded device running 22.6 % of the time between two HR values. The device can fully work for 9.37 days, including idle and processing time.

Finally, in the context of a more complex problem regarding one of the major causes of heart failure, AF, I have proposed an online, energy-efficient and personalized PAF prediction method targeting emerging ULP wearable sensors. In fact, patients affected by PAF are at risk of the arrhythmia progression into a sustained AF. Therefore, predicting the onset of PAF episodes is required for progression prevention and lower stroke risk. In a preliminary analysis of this contribution, I demonstrated that considering the specific profile of each patient, highly improves PAF onset prediction. By training a linear SVM classifier and considering features related to the P-wave and the QRS complex, the real-time patient-specific method predicts PAF onset with an F_1 score of 93.6 %, a sensitivity of 96.2 %, and a specificity of 90.6 %. Moreover, my method enables energy savings for a continuous PAF event monitoring in single-core resource-constrained wearable sensors, and scales its energy consumption depending on the patient's characteristics. By considering my algorithm running on the INYU wearable ECG-monitoring sensor, I estimated a battery lifetime of at least 37 days.

These contributions show how personalized and ULP multi-biosignal approaches highly improve accuracy while consuming very low energy in real-life devices. However, the resources and constraints of new platforms need to be taken into account when designing WSNs for remote wellness monitoring. In the next chapter, I will focus on the challenges posed by the resources of modern ULP platforms for wearable sensors and exploit their power-saving capabilities to design WSN-based biomedical applications.



3

Modular and Patient-Specific Optimizations in Modern Wearable Sensor Nodes

The next step in the design of a wearable sensor node (WSN) for wellness monitoring is looking at modern platforms and the challenges to overcome and reach an optimal energy-accuracy trade-off. Modern ultra-low power (ULP) platforms provide parallel computing capabilities, clock- and power-gating of independent blocks to reduce power during idle time and the possibility to connect accelerators to further decrease energy consumption.

In this chapter, I propose two methods to tackle the challenges of energy savings in these platforms, ensuring the high accuracy required by biomedical applications in the context of remote wellness monitoring. First, I expose the modularity of parallelization and power-saving capabilities of modern ultra-low power platforms. Then, in the context of paroxysmal atrial fibrillation (PAF) prediction, I propose a method to scale the computational resources (i.e., number of cores) and memory banks, according to a patient-specific model.

3.1 Introduction

WSNs have already proven capable of attaining accurate inference with minimal power consumption [18]. In this way, WSNs have evolved from single-core systems [24, 42] into ULP [44] and multi-core parallel computing platforms [45–49]. Most of the typical WSN-based biomedical applications in the state of the art have been implemented on single-core proces-

sors [24, 25, 42, 57]. To exploit the new parallel capabilities of modern WSN platforms in the context of biomedical applications, per-lead (i.e., channel) multi-core computation is a natural option to achieve low-power operation, as in the case of multi-lead electrocardiogram (ECG) analysis [46, 115]. However, more general WSN-based biomedical applications for monitoring of noncommunicable diseases (NCDs) typically include several building blocks which often are not amenable to standard parallelization techniques (e.g., per-lead parallelization) [21, 24, 25, 47, 57, 72, 116–119]. Modern platforms have also evolved into hybrid systems with a main core and an additional cluster of cores [45] that allow flexible design of efficient single-core and parallel modules, in applications where several modules cannot be parallelized easily.

In addition to parallelization, modern platforms offer clock- and power-gating mechanisms to reduce both dynamic and static (leakage) power when the system is not actively computing (e.g., when waiting for new samples to arrive in an input buffer considering the usual low sampling frequency of biomedical applications). Some platforms include specialized direct memory access (DMA) engines that execute data capturing tasks within tight power budgets while the rest of the system is clock-gated or executing other tasks [48, 49]. Additionally, other platforms contain SRAMs structured in independent banks that can be power-gated depending on the application needs [48, 49]. Moreover, application modules typically include computationally expensive kernels that can be accelerated with domain-specific hardware (HW), such as coarse-grained reconfigurable arrays (CGRAs) [46]. Thus, HW acceleration is an orthogonal benefit to the parallelization, and it can benefit both single-core and multi-core application design.

For my first contribution to this chapter, I propose modular optimizations for ULP heterogeneous platforms with the following main outcomes:

- I show how the parallelization of the typical modules in WSN-based biomedical applications at different levels of abstraction (i.e., lead, sample analysis-window, heart beat, or data-level) maximizes speed-up and consequently reduces energy consumption up to 41.6 %.

- I explore the reduction of static power by exploiting power management and SRAM-bank memory scaling with additional energy savings of up to 16.8 % for a state-of-the-art application.
- I investigate the use of programmable domain-specific accelerators to perform intensive computations at lower power than with general-purpose processors obtaining energy savings up to 46.7 % in the multi-core implementation of the state-of-the-art application.
- Finally, I show the orthogonality of the previous optimizations achieving accumulated energy savings of up to 51.3 %.

Although modularity highly reduces energy consumption for a general biomedical application, more challenges are introduced if we consider personalized medicine. As proved in Chapter 2, patient-specific methods highly improve the accuracy of pathology detection, and it is as well advantageous for energy reduction in traditional single-core platforms [23–28, 73]. However, modern ULP heterogenous platforms offer better resources to tackle the challenge of energy efficiency and scalability (i.e., according to the specific pathology characteristics of each patient) [100]. For this reason, as a second contribution, I propose a methodology to design a new online PAF prediction model targeting scalable computation on modern ULP wearable sensors, which considers the specific features of the individuals and their condition. The scalability is driven by the adaptive algorithm and architecture parameters, which affect the design in multi-core platforms to reduce energy consumption for each individual patient. I have already shown the scalability in single-core platforms and, specifically, in a real-life ECG monitoring device in Section 2.4.5. The main outcomes of this method are the following:

- I develop a personalized parallelization technique for new open-source multi-core RISC-V based computing architectures that can be included in wearable sensors. This technique scales with the number of cores, i.e., distributing the computation among cores, according to a patient-specific model. My proposed parallelization achieves energy savings of up to 24 % compared to the single-core design.

- I explore the memory design space to execute personalized atrial fibrillation (AF) algorithms in multi-core Internet of Things (IoT) and wearable platforms [48], by scaling the size of the memory banks (8 KiB, 4 KiB, 2 KiB, and 1 KiB) and storing buffers of different lengths according to the number of cores. Also, I highlight the energy consumption reduction thanks to deep sleep modes that exploit the specific characteristics of the patient, as done in the single-core design in Chapter 2. Overall, the personalized multi-core design provides up to 34 % energy savings in comparison to recent single-core wearable sensors.

This chapter initially presents the typical modules of a biomedical application in Section 3.2. Next, Section 3.3 describes the capabilities of modern ULP platforms and the motivation analysis to sustain the two contributions. Then, it presents the first main contribution mentioned before in Section 3.4, which has a double publication in the ESWEEK 2020 (CODES+ISSS) conference and in the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems journal [120]. Then, it follows with the second contribution in Section 3.5, which has been published in a special issue of the IEEE Transactions on Emerging Topics in Computing journal, called "New Trends in Parallel and Distributed Computing for Human Sensible Applications" [73].

3.2 Typical Biomedical Modules

Considering the characteristics of modern ULP platforms, I propose a modular design approach for biomedical applications that combines different types of software (SW) parallelization to achieve optimal speed-up. This approach is applied to the two contributions presented in this chapter. Let us consider a typical WSN-based biomedical application for long-term health monitoring, described in Fig. 3.1. First, the single or multi-channel signal is filtered to remove high or low frequency noise, baseline wandering, or muscle noise. The second module typically includes some additional preprocessing of the signal to enhance specific characteristics or combine different channels. The third module is the extraction of patterns or features, such as the signal main waveforms and time or frequency-domain parameters. The final step, inference, includes any kind of classification or regression technique that uses the information of the extracted features to predict an outcome,

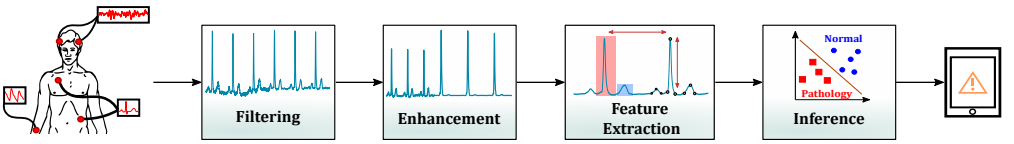


Figure 3.1 – Typical modules of a general WSN-based biomedical application.

such as the occurrence of a pathology. In this chapter, I apply the energy-saving capabilities of modern platforms to an optimized single-core version of well-known instances of each of those modules. Then, I evaluate them as part of complete state-of-the-art applications. Moreover, I apply the same modularity approach to the PAF prediction method described in Section 2.4.

3.2.1 Filtering

Digital filtering in biomedical applications is used to remove undesired noise at specific frequencies or isolate the frequencies of interest. In biosignal processing there exist different types of filtering [121]. In this chapter, I analyze the morphological filtering (MF), which extracts the signal baseline based on the shape of the original signal and then subtracts it. This method was originally used in image processing and then modified to be used on a single or a multi-lead ECG in embedded systems [71]. Additional techniques to filter the raw ECG input data that are suitable for embedded systems are described in [71].

3.2.2 Enhancement

Several techniques, such as the signal derivative or the root-mean-square (RMS) combination, are available to enhance a biosignal or combine different leads. I study a lightweight example of short-term event amplification: Relative-Energy (Rel-En) [41]. In the context of an ECG signal, this technique extracts the energy of specific windows of analysis to amplify the R peaks, since the signal energy is larger when an R peak occurs. The Rel-En method is also used for K-complex detection in electroencephalography (EEG) and pulse extraction in imaging photoplethysmography (iPPG) [50]. Additionally,

I consider the RMS lead combination as part of a complete application in Section 3.4.6.

3.2.3 Feature Extraction

This module enables the biosignal abstraction through the extraction of the most relevant features, from its waveforms or points to time and frequency domain parameters [21, 24, 47, 57, 72, 116, 118, 119]. In ECG analysis, for example, a common technique, called "delineation," abstracts the signal main waves (i.e., QRS complex, P and T waves [68]) with three "fiducial points" representing the onset, offset and peak. These points are the input to the inference module or can be further processed extracting additional features (e.g., QRS complex duration, QT interval, etc.). In this chapter, I analyze the ECG delineation since it is a relevant and well-known method for long-term monitoring of NCDs. The process of delineation can be divided into two parts. First, the R peak or QRS complex are detected, often independently from the other ECG waves, since they describe the heart rhythm and are relevant for the detection of many arrhythmias [21]. As R peak detection technique, we choose to implement REWARD [41] for its claimed low computational load and described in Section 2.2. REWARD uses amplitude thresholds to isolate the R peak. Moreover, it analyzes physiological peak-to-peak distance and peak width to filter false positives, such as dominant T-waves. The remaining fiducial points can be delineated in different ways. I choose a low-complexity method [72], described in Section 2.4.2.1, which assumes that the signal's main waves are positive. This can be ensured by an RMS combination of leads or choosing lead II of the 12-lead ECG technique [122]. Under this assumption, the Q and S points are identified as a minimum within a physiological interval near the R peak. The P and T peaks of the two other main waves are computed as a maximum within physiological windows between two R peaks. Finally, the onset/offset of the P and T waves are computed considering the minimum Euclidean distance between the original waves and their piece-wise linear approximation. The point with the minimum Euclidean distance that intersects the isoelectric line is the onset/offset.

3.2.4 Inference

The last module is commonly a classification or regression problem applied to a set of features that performs automatic events and pathology detection, such as the occurrence of abnormal beats. Several types of arrhythmia can change the heart electrical signal, thus causing abnormalities in the ECG main waves. Therefore, automatically detecting abnormal beats and their nature helps to treat them and prevents further complications [42,117]. Other biosignals (e.g., photoplethysmography (PPG), respiration, impedance cardiogram (ICG), etc.) also contain relevant features to classify NCDs, such as sleep apnea [116], to monitor a subject state in stressful environments [119] or for gesture recognition [47]. In this chapter, I analyze a classification module for detection of abnormal beats from an ECG signal using random projections and a neuro-fuzzy classifier [123].

3.3 Modern Ultra-Low Power Platforms for Wearable Sensors

The main goal of modern ULP platforms is reducing energy consumption to maximize battery lifetime, while still running complex algorithms on the nodes. Multiprocessing has been proved effective in reducing energy consumption—through lower operating frequencies and supply voltages—while preserving performance in the biomedical [115] and multimedia [45] domains. However, SW tasks must be divided into parallel subtasks or organized as independent parallel ones, i.e., application modules, targeting an energy-efficient management of resources. Often, a major obstacle to achieve adequate speed-ups is the overhead of synchronization. Fast HW event managers offer single-cycle synchronization and enable clock-gating the processors while waiting for events, hence saving significant amounts of energy even with fine-grained parallelization [49,124]. A novel architecture that can overcome these obstacles and ensure the flexible design of modular and personalized WSN-based biomedical applications, is the open-source RISC-V based PULP platform [45]. In this section, I describe the power saving capabilities of parallelization on multi-core platforms based on PULP. Moreover, I describe the power and memory management possibilities in modern

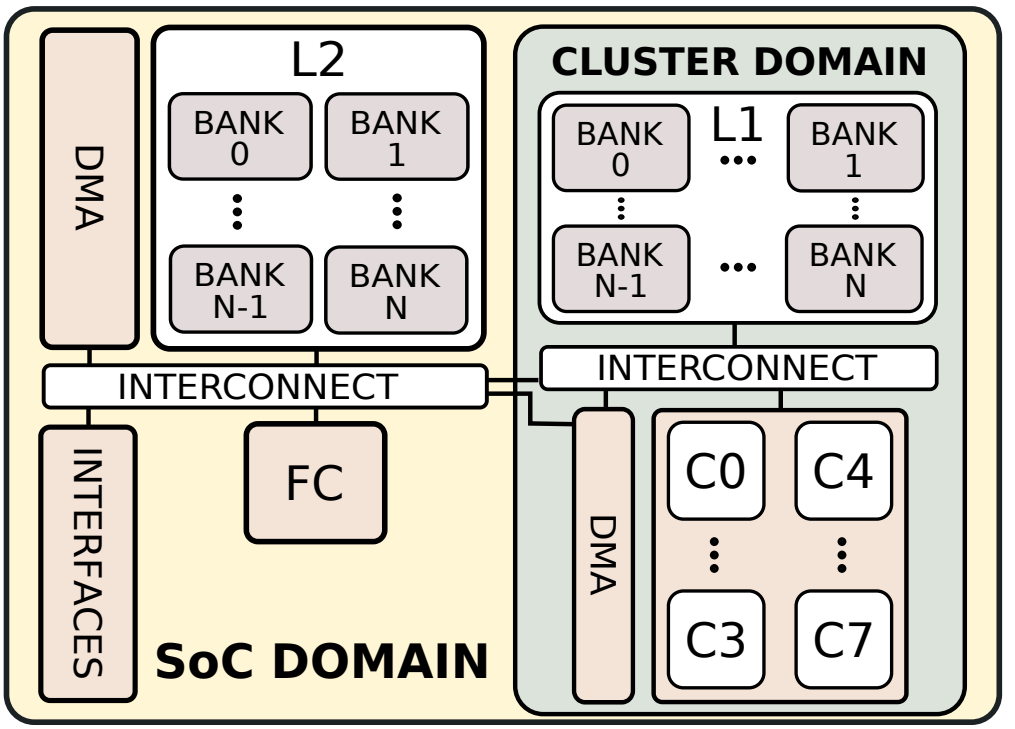


Figure 3.2 – Main architecture of PULP-based platforms, divided into a main streamlined processor, the fabric controller (FC), and an 8-core parallel compute cluster (CL). PULP includes a multi-banked L2 memory, a DMA, and a shared multi-banked L1 memory in the cluster side.

ULP platforms. Finally, I explore the architectural heterogeneity of adding CGRAs to accelerate computationally intensive kernels.

3.3.1 Parallelization in the PULP Platform

In this chapter, I target the PULP platform [45], whose main architecture is shown in Fig. 3.2. PULP is divided into a main streamlined processor, the fabric controller (FC), and an 8-core parallel compute cluster (CL). It includes a multi-banked 512 KiB L2 memory, a HW event synchronizer, and a shared multi-banked 64 KiB L1 memory with single-cycle latency in the cluster side. Both FC and CL are power-gated while the DMA fills the required L2 memory bank during sample acquisition. Each of the cores in the CL

can be independently clock-gated to reduce dynamic power. For example, the CL cores become clock-gated after reaching a synchronization point. This flexibility allows to easily implement parallel and single-core modular applications with adaptable resources assignment.

3.3.2 Power and Memory Management

In addition to parallelization, WSN-based biomedical applications need power management to ensure continuous remote monitoring. A common technique to save energy is clock-gating, which reduces dynamic power. In the context of the PULP platform, architecture-level clock-gating is applied at different levels. The SoC is clock-gated when waiting for an event, such as a DMA transfer or the end of a computation on the CL. Additionally, if no workload is assigned to some cores of the CL, they are automatically clock-gated. This is relevant in the context of modular WSN-based biomedical applications, because an optimal assignment of resources to the modules reduces energy consumption. Conversely, power-gating interrupts the power supply to parts of the circuit that are unused for longer periods, hence suppressing leakage current. Power-gating has a larger physical overhead than clock gating—due to the power switches and controllers around the power gated area. Thus, it is applicable only for large blocks (e.g., a cluster of processors). Moreover, the recovery period for power-gating can be in the order of tens of thousands of cycles, particularly if clock generators are affected, making it suitable only for applications that undergo long idle periods. Typical WSN-based applications are characterized by low sampling frequency (e.g., ECG acquisition is in the standard range of 250 Hz–500 Hz), hence, the main SoC can be power-gated, while waiting for the following sample. Additionally, the division of the platform SRAM memories into smaller banks that can be independently power-gated, or placed into retention mode, enables fine-grained control on memory energy use. In the context of healthcare wearable sensors, this feature enables the processing of the acquired input biosignals in “windows” that drive which banks are written by the DMA (active), which ones contain data to retain until the next processing interval, and which banks can remain off. Furthermore, the use of banks of different sizes enables an appropriate data placement for input data of accessed biosignals into smaller banks, which consume less energy per access (cf. Section 3.5.4.1).

3.3.3 HW Acceleration

Finally, domain-specific accelerators, either programmable (e.g., CGRAs [46]) or task-specific (e.g., for fast Fourier transform (FFT) or sample-rate conversion [44]) are added to accelerate intensive application kernels. In this case, energy savings stem from the shorter execution times and the specialized implementations of the accelerators. Hardware accelerators can be introduced at the end of the optimization process to offload kernels assigned to particular cores. In this chapter, I explore the possibility of integrating a CGRA into the PULP platform, designed to execute small loop-based kernels with high numbers of iterations. I describe in detail the architecture of the CGRA and the computational kernels accelerated in Section 3.4.4.

3.3.4 Motivational Analysis for Optimizations in PULP

Considering the low duty cycle of WSN-based biomedical applications, we conduct an analysis of the impact of the application duty cycle and the attainable speed-up in an 8-core parallelization on the energy savings in the PULP platform. In this analysis, I assume the eight cores are all used during the active part of the duty cycle, while during idle periods they are power-gated. In contrast, in many biomedical applications or its modules, as the ones I present in Section 3.4.6, it may happen that only some of the cores are active. At the same time, the remaining ones are clock-gated (i.e., unused). Moreover, I show how activating one bank (of 64 KiB) or the full memory (i.e., eight banks for a total of 512 KiB) affects the energy savings. Finally, this analysis shows that the percentage of energy consumed during idle time is proportionally inverse to the duty cycle. Consequently, platforms that execute very low duty cycle applications need to optimize energy consumption during idle periods (e.g., turning off unused memory banks). In contrast, with higher duty cycles, the energy consumed during active time prevails. Therefore, it becomes more relevant to optimize computation (e.g., increasing the speed-up to reduce active time) in order to lower the total energy consumption.

Figure 3.3 shows the previous analysis on one evolution of the PULP platform, Mr.Wolf [49]. For each platform, the graph reports the energy savings compared to a single-core implementation of a generic application in the

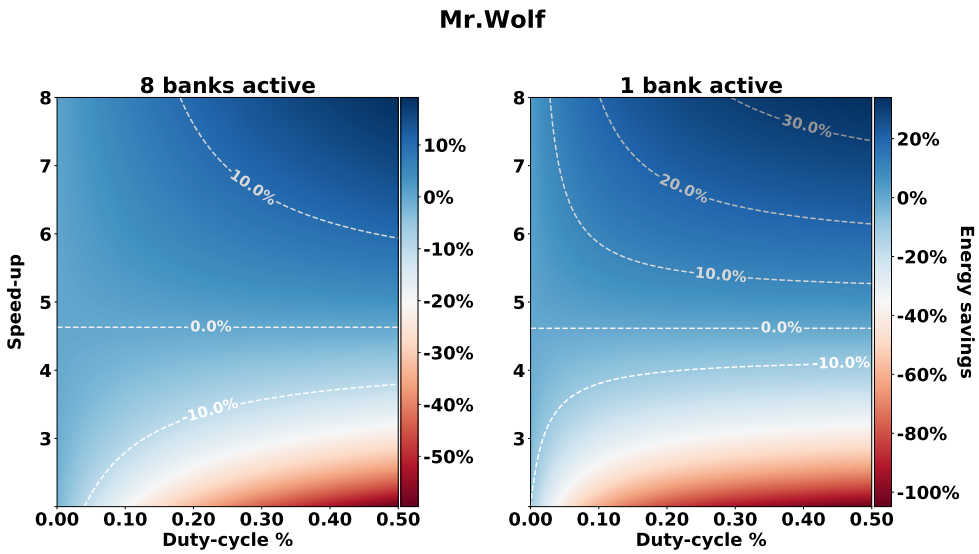


Figure 3.3 – Potential energy savings in Mr.Wolf, an implementation of the PULP platform, according to the application duty cycle and the attainable speed-up through an 8-core parallelization in the CL. On the left, I present the analysis on Mr.Wolf with its eight memory banks active. On the right, I show the analysis on Mr.Wolf with only one bank active. The dotted lines mark different levels of energy savings.

FC. Mr.Wolf includes a core for the FC (Zero-riscy [125]) that is simpler than the RI5CY cores of the CL [126] and runs at a higher frequency (170 MHz for FC and 110 MHz for the CL) but has a lower IPC. Moreover, Mr.Wolf is more efficient for higher duty cycles because it was designed to handle high computational load, and the deep sleep mode is not optimized for long idle periods—different PULP implementations with a core optimized for deep sleep exist, though. Therefore, my analysis is applied to the Mr.Wolf architecture with a more optimized deep sleep mode based on other PULP implementations. The graphs in Fig. 3.3 are generated using the energy models in (3.1) and (3.2) for the single-core (E_{SC}) and the multi-core (E_{MC}) configurations, respectively, where dc is the duty cycle of the application, FC_P_{dyn} and FC_P_{leak} are the dynamic and leakage power of the FC, respectively, DS_P is the power in deep sleep, CL_P_{dyn} and CL_P_{leak} are the dynamic and leakage power of the CL, and f_{cr} is the frequency correction ratio ($\frac{170\,\text{MHz}}{110\,\text{MHz}}$) for the FC and CL.

$$E_{SC} = dc \times (FC\_P_{dyn} + FC\_P_{leak}) + (1 - dc) \times DS\_P \tag{3.1}$$

$$E_{MC} = \frac{dc \times f_{cr}}{speedup} \times (FC\_P_{leak} + CL\_P_{leak} + CL\_P_{dyn}) + (1 - \frac{dc \times f_{cr}}{su}) \times DS\_P \tag{3.2}$$

Finally, the ratio (in percentage) of potential energy savings attainable by a multi-core configuration against the single-core one is computed using (3.3).

$$E_{\%} = (1 - \frac{E_{MC}}{E_{SC}}) \times 100 \tag{3.3}$$

On the left side of Fig. 3.3, I show the analysis for Mr.Wolf with the full memory active (i.e., eight banks). It shows that the energy overhead of the multi-core

CL is recovered when a speed-up of 4.6 × is reached and becomes more energy efficient compared to the single-core implementation for higher speed-ups. Additionally, each of the Mr.Wolf eight memory banks of 64 KiB can be powered-off depending on the application. Consequently, on the right side of Fig. 3.3, I show how the analysis changes if there is only one bank active. Whereas the threshold of speed-up does not change, for lower duty cycles it is possible to achieve higher energy savings.

I have also run the analysis on the full scale of duty cycle values to explore the benefits attainable under higher duty cycles. The architecture is able to achieve energy savings up to 42 % for 100 % duty cycle and maximum speed-up with the eight cores and eight banks always active. An interesting result is that, for high duty cycle applications, memory management has less impact than for low duty cycle ones. Nonetheless, in this chapter, I focus on the energy savings attainable on low duty cycle, which is a characteristic of typical biomedical applications.

From this previous analysis, I can conclude that, for this implementation of PULP, the speed-up required by the parallel application has to be at least 4.6x. This shows the importance of suitable optimizations (e.g., parallelization techniques) to achieve energy efficiency on modern low power heterogeneous platforms, which is the main motivation for this chapter. To achieve optimal speed-up, a modular approach to SW parallelization is necessary considering the typical modules of WSN-based biomedical applications described in Section 3.2. Then, to maximize the speed-up of the overall application, I consider different parallelization techniques and HW acceleration. Power management is also a significant factor in low duty cycle applications. Finally, memory bank management plays an important role in energy saving and, specifically, for applications with low memory footprint. In Section 3.4, I refer to a general conceptual architecture that takes advantage of all the benefits of the PULP platform discussed in this analysis.

3.3.5 Energy Savings versus Resources Assigned

One of the two main goals of this chapter is to show how a patient-specific assignment of resources in a multi-core platform, like PULP, achieves signifi-

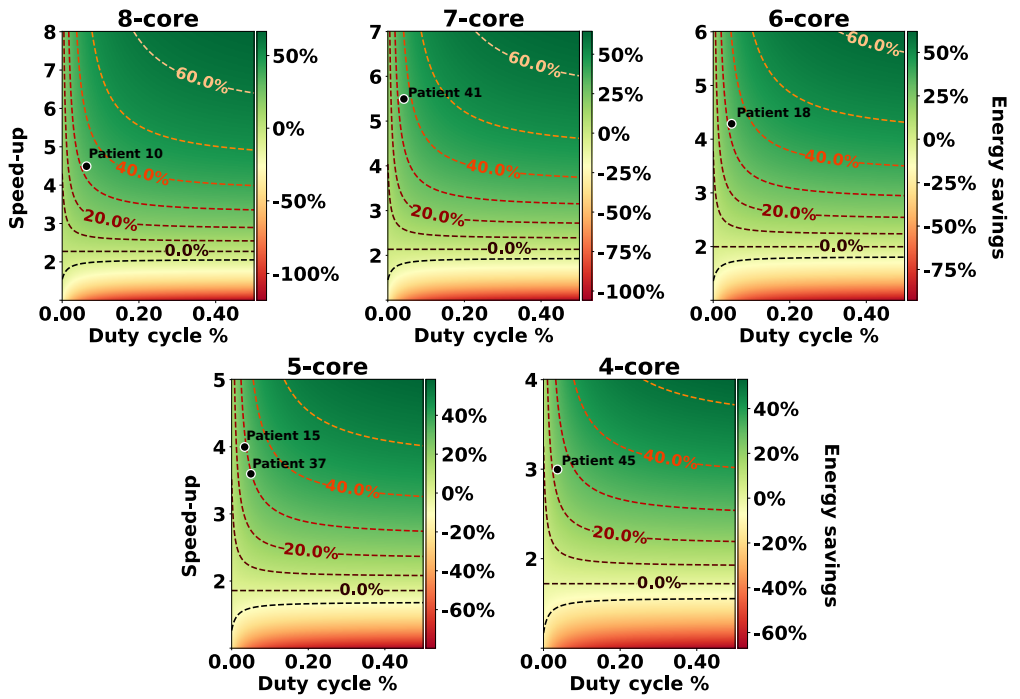


Figure 3.4 – Potential energy savings in the multi-core GAP8 platform [48] according to the duty cycle of the application and the attainable speed-up compared to the single-core implementation. I report five configurations with a varying number of active cores in the CL from eight to four. The dotted contour lines represent different levels of energy savings, which include the efficiency border between single and multi-core computation for GAP8 at 0 %. Additionally, I mark the working point for the six subjects [108] studied in Section 3.5.4.

cant energy savings. To this end, I did a preliminary analysis on the estimated energy consumption of one commercial implementation of PULP, GAP8 [48], by varying the resources assigned in terms of the number of active cores, application duty cycle (i.e., percentage of application computational load in 1 s), and parallel implementation speed-up. In this analysis, the energy consumption estimation accounts for the processing and idle time with one memory bank active out of the four banks available in GAP8, and it is computed as done in Section 3.3.4. Fig. 3.4 shows the energy savings compared to the single-core implementation in the five possible multi-core configurations of my design space with a varying number of active cores from eight to four. Each configuration reports the percentage of energy savings achieved at different attainable speed-up figures from 1 × to 8 ×, while the application duty cycle values can vary between 0 % to 0.5 %, according to the typical intervals that a single-lead ECG-based biomedical application has. Thus, the attainable speed-up for a specific number of cores is a measure of efficiency of the parallel application. However, as Fig. 3.4 shows, there is a threshold of speed-up that the parallel application needs to reach to start achieving energy savings compared to the single-core implementation. Nonetheless, interestingly enough, this threshold significantly varies with the number of active cores, namely, from 2.3 × for an 8-core implementation to 1.7 × for a 4-core implementation. Therefore, below the speed-up threshold, the single-core implementation is more efficient than a particular multi-core implementation. In contrast, above the threshold, the multi-core option is more efficient.

In addition, in Fig. 3.4, I report six cases of the analyzed dataset [108] to cover a wide range of computational requirements (i.e., including best, average and worst case scenarios) with different patients (cf. Section 3.5.3.1 and Section 3.5.4). In my design space, the duty cycle of the parallel application varies from 0 % to 0.07 %. At the same time, I assign between four and eight cores depending on the window length (i.e., number of consecutive beats), thus achieving different speed-ups from 3 × to 5 ×. As a result, Fig. 3.4 indicates that the absolute energy savings increase by adding more active cores. Nonetheless, if the parallel application achieves its highest speed-up for a certain number of cores, for example, 3 × in the 4-core configuration, it is

then more energy-efficient to assign the lowest possible number of cores (i.e., it is better to use four cores instead of five or more for Patient 45). Moreover, within one configuration, for example the 5-core implementation in Fig. 3.4, in the context of limited duty cycle values, the case with a higher duty cycle but lower speed-up (Patient 37) achieves higher energy savings than the case with a lower duty cycle and higher speed-up (Patient 15). I further discuss the implications of this analysis on the six reported cases in Section 3.5.4.1.

3.4 SW and HW Optimizations in Modular Biomedical Applications

In this section, I do a top-down exploration of parallelization techniques at different abstraction levels. This strategy helps to compose a modular biomedical application. Therefore, it can exploit the energy-saving platform characteristics and maximize it taking into account the analysis done in Section 3.3.4. Additionally, I apply memory and power management according to general characteristics of the application (e.g., duty cycle, memory needed for acquisition, etc.). Finally, I integrate a domain-specific accelerator that can execute intensive kernels faster (and consuming less energy) than the general purpose cores available. In Fig. 3.5, I draw a conceptual architecture, based on the analysis reported in Section 3.3.4, which can be used to apply the SW and HW optimizations described in this section. For each main optimization, I report their mapping to the component used in the architecture. Then, I present the experimental setup (c.f. Section 3.4.5) used to apply the optimizations and achieve the results and conclusions of this first contribution presented in Section 3.4.6.

3.4.1 Modular SW Optimizations

Considering the characteristics of the algorithms described in Section 3.2, I present several techniques to extract parallelism. I also propose a top-down order for exploring them, as follows. These techniques are mapped to the 8-core cluster, shown in Fig. 3.5. The first choice of parallelization is by lead (or channel). In fact, if leads are processed independently throughout the application, it is the simplest and most efficient implementation. However,

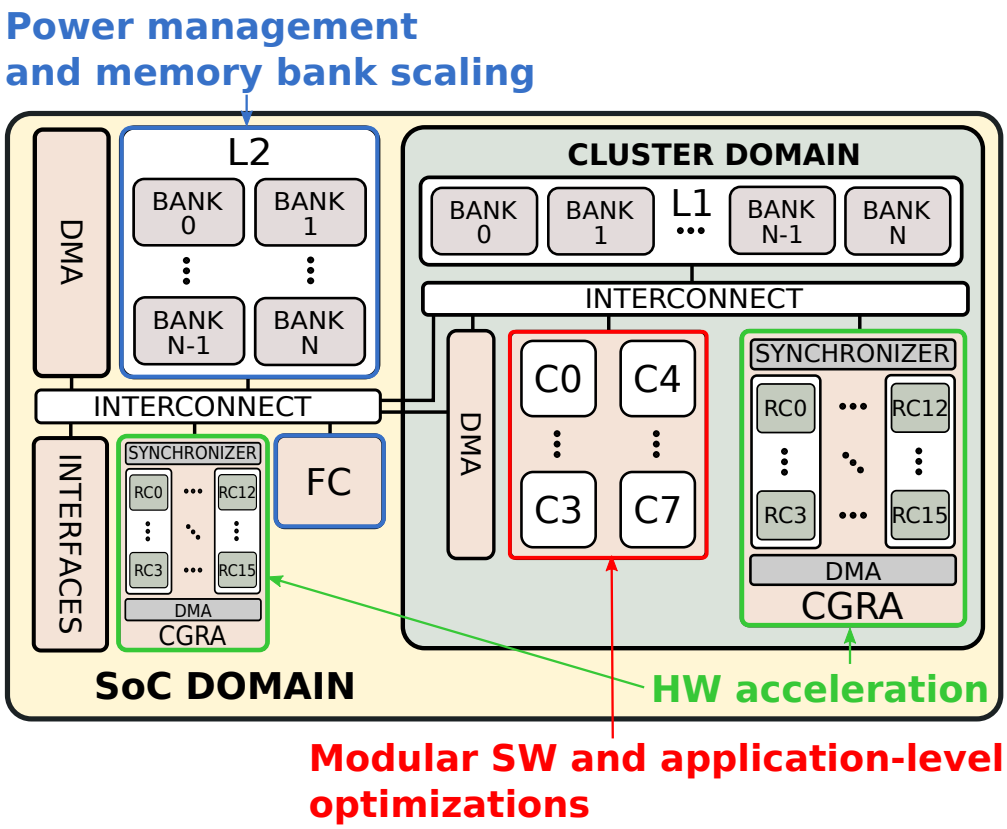


Figure 3.5 – Conceptual architecture following the PULP platform. From left to right: SoC domain containing the main processor (FC), the L2 memory, the DMA, and potentially a CGRA. The cluster domain contains the multi-core CL; the L1 memory; a CGRA. Additionally, I report each of the main optimizations presented in the section that are mapped to the components used to implement them: modular SW and application-level optimizations (red), power management and memory bank scaling (blue), HW acceleration (green).

many biomedical applications and their modules only work on single-lead or a combination of multiple leads into one. Then, a window parallelization should be considered where the cores work on subsegments of the signal. In some cases, the characteristics of the signal and the application make it necessary to consider a more specific type of parallelization, such as a beat parallelization for cardiovascular-based signals. This method can be extended to any kind of periodic or pattern signals where the features within a period or pattern need to be captured. When the previous methods cannot be applied, a general data-level parallelization should be considered. Finally, if none of the previous methods can be applied, or if the obtained speed-up is not satisfactory, a pipelining strategy can be considered, where a subset of the cores is assigned to each of the pipeline stages. The cores at one stage process segments of input data and produce segments of output, which are processed by the cores in the next stage in a parallel consumer/producer pattern. However, for accuracy and standardization purposes, biomedical applications often include checks or feature combinations that need to be executed once the complete output of a module has been generated [41, 57, 62]. Given this limitation, and the fact that the effort to implement pipelining is larger, I consider only the first four types of per-module parallelization in the proposed top-down order. Table 3.1 summarizes the different kinds of parallelization techniques applied to each module described in Section 3.2.

3.4.1.1 Lead Parallelization

WSN-based biomedical applications commonly acquire multi-lead signals (e.g., 3–12 ECG leads) to extract more information for highly accurate monitoring. Multi-lead parallelization, where each core processes the data corresponding to one lead in parallel, should be applied first as it typically offers almost linear speed-ups. The most common application is the filtering module, which often works on multiple leads or channels [46, 47] or even on multiple signals [57]. Another example from the literature where this parallelization is applied is a multi-lead delineation using multi-scale morphological derivatives (MMD) [46].

As shown in Fig. 3.5, in the PULP architecture the DMA can access both the L2 and L1 memories. It can be used to transfer the samples of each lead from

Table 3.1 – Summary of parallelizations applied to each analyzed module

Module	Algorithm	Parall.	Notes
Filtering	Morph. Filt. (8L-MF)	Lead	Data-dependent
Enhance.	Relative-Energy (Rel-En)	Window	Homogen., overlap
Enhance.	Lead combination (RMS)	Data	Homogen., 1/8 samples
Feat. Extr.	R peak (REWARD)	Window	8×1.75 s windows
Feat. Extr.	Fiducial points	Beat	Data-dependent
Inference	Beat classification	Beat	Data-dependent

L2 into separate areas of L1, thus allowing the cluster of cores to implement the per-lead filtering without interference. The MF algorithm analyzed is data-dependent; hence, the workload of each core depends on the amount of noise of each lead (e.g., due to problems in the electrode positioning). For the modular analysis to compute the maximum attainable speed-up of an 8-core parallelization against a single-core design, I consider an 8-lead ECG (8L-MF), one lead per core.

3.4.1.2 Window Parallelization

For subsequent modules in the processing chain, or in the case of applications that obtain data from a single lead, the data to be processed can be divided in multiple windows [47,72]. In this way, each window is processed in parallel by a different core. Furthermore, if the samples are directly collected by the DMA module, this method enables power-gating of the platform cores over larger periods. Energy savings stem from operating at lower frequency and voltage than a single core and by a more aggressive application of power-gating than possible when operating on a sample-by-sample basis.

In my example, I apply this technique to the signal enhancement (Rel-En) and the feature extraction (R peak detection) modules. In the case of Rel-En, I divide the window into smaller windows, with each core starting from the first sample of each sub-window as explained in [127]. Since the Rel-En algorithm computes the signal energy at the sample n using information starting from $(t(n) - \frac{0.95}{2})$ s, a small window overlapping is necessary. Therefore, the

computational workload is, in this case, homogeneous among the cores, but the speed-up is reduced by the introduced overhead.

On their side, R peak detection techniques usually consider fixed windows of analysis to extract the peaks based on physiological characteristics. In our case, the REWARD algorithm [41] uses a fixed window of 1.75 s. Therefore, considering 8 cores, my method collects a buffer of (8×1.75) s so that each core will compute one fixed window.

3.4.1.3 Beat Parallelization

The ECG and other cardiovascular-based signals (e.g., PPG) are characterized by beats, each representing one cycle of the heart contraction as a periodic wave. Applications often perform the same operation for each beat in which there is essential information. Therefore, beat parallelization is the next step to explore in the top-down proposed order. This technique can be applied to any upper-level feature, time series, or excerpt of relevant information from the signal. There are several examples of classifiers and feature extraction techniques in the literature where this type of parallelization can be applied [72, 117, 123, 128]. However, for simplicity I only analyze two of them, as they represent standard techniques, similar to other examples in the literature, which apply the same type of computation for each beat and are targeted and implemented for embedded devices. They correspond to two of the modules described in Section 3.2, namely the beat classification [123] and the delineation of fiducial points [72]. In the former case, the beat is centered to the R peak; in the latter, it comprises the signal between two R peaks. Again, to match the characteristics of our platform, we collect eight beats, one per core. During beat classification, the workload is data-dependent and also varies with the window length (which may be fixed). In the case of the fiducial points delineation, each core's workload is linked to the natural variability of the RR intervals (i.e., heart rate (HR)). In Section 3.4.6, I show the effect of the different workloads on speed-up and energy consumption.

3.4.1.4 General Data-Level Parallelization

General-purpose parallelization techniques can be applied to the inner kernels of each module. Good candidates at this stage common to multiple

applications are sorting algorithms, RMS combination [47, 123], training algorithms running on node [47], or several filtering techniques, such as those presented in [24]. In this contribution, I study the RMS combination algorithm, which is also used in the complete application that I analyze in Section 3.4.6. RMS is a signal enhancement technique that computes the root-mean-square of a buffer of data. In WSN-based biomedical applications, this is used to combine a multi-lead signal into a single-lead one. Following the work presented in [123], the implementation first computes the sum of squares of the samples of the different leads and then applies a square-root to the result. Since RMS works on sample i_{th} from each lead independently of the other samples, each core receives a similarly-sized subset of the samples from all the leads.

3.4.2 Power Management and Memory Bank Scaling

When combining the modules into a full application, I apply SoC and SRAM power-management considering the two components shown in Fig. 3.5, the FC and the L2 memory with its independent banks. The FC in the PULP platform is power-gated whenever the data is acquired, while it needs to be clock-gated when the DMA stores the data in L2. Considering the low duty cycles of typical WSN-based biomedical applications, such as the one reported in Section 3.4.6, the time spent during acquisition and storing is significantly high compared to the processing. The power management strategy of power- and clock-gating during idle time allows to significantly reduce the energy consumption. Moreover, during the acquisition phase, banks not containing new data (nor application code) can be powered off. Banks that contain captured samples waiting to be processed can be placed in retention mode. Finally, only the bank currently receiving samples needs to be active. However, since the memory needed for the analyzed biomedical applications is significantly lower than 512 KiB, I explore the possibility of reducing the overall memory to 128 KiB and assuming eight banks scaling each bank size to 16 KiB. This strategy allows a smaller resolution in bank size and a better management of the activated banks depending on the specific application, hence, reduced energy consumption. For example, let us consider an application that needs to process a signal window of 30 s integer 16 bit acquired at a sampling frequency of 250 Hz. Since the buffer to store

is 30 s ∗ 250 Hz ∗ 2 = 15 KiB, only one bank needs to be active, on top of the banks needed for the code. As shown in Fig. 3.5, the scaling strategy can be pushed to the limits of feasibility and significantly lower energy consumption, especially for applications with low memory footprint. Memory scaling and management is a relevant design factor orthogonal to parallelization for typical low duty cycle biomedical applications. I will explore this concept further in Section 3.5.

3.4.3 Application-Level Optimizations

In addition to general-purpose power and memory management, specific algorithmic-level optimizations for WSN-based biomedical applications need to be applied. These optimizations are related to the usage of computing resources. They are mapped in the cluster of cores in Fig. 3.5. For example, one of the applications I evaluate is the beat classifier discussed in Section 3.2, which requires several of the modules described previously. The single-core implementation of this algorithm adapts its computational complexity based on the outcome of the classification. First, it analyzes a single-lead ECG and performs only R peak detection to save energy. Then, if the algorithm detects an abnormal beat via a neuro-fuzzy classifier based on random projections, it performs an RMS combination of a 3-lead ECG[1] and a full delineation, as shown in the original paper [123]. However, this approach can be counter-productive in multi-core platforms because the direct execution of the 3-lead ECG analysis on three cores consumes roughly 50 % less time than the "1+2" analysis approach. In particular, with the database used in the experiments (MITDB, c.f. Section 3.4.5), approximately 27 % of the patients experience abnormal beats more than 50 % of the time, thus requiring the full 3-lead processing. This can be exploited at run time by determining the frequency of execution of the full analysis: if a certain threshold is exceeded, the system switches to the parallel version. Another application that I evaluate is the delineation of a complete set of 12-lead ECG. The resources assigned in this case include the full 8-core cluster. However, after processing eight leads with an approximately equal distribution of

[1] Using a 3-lead electrode positioning is a medical standard in mobile ECG measurements

computation, four cores are automatically clock-gated while the other four process the remaining leads.

3.4.4 HW Acceleration for Intensive Computational Kernels

The last optimization that I propose is a HW acceleration of intensive computational kernels, and it is mapped in a CGRA that can be connected to the FC or the CL, as shown in Fig. 3.5. MorphoSys [129] is one of the earliest examples of CGRAs originally proposed to accelerate multimedia applications with strong computational demands. Later works showed how a CGRA can be used in the domain of biomedical applications to reduce power by both accelerating common operations and reducing the energy cost of executing those operations [46]. I consider the open-source PULP platform [130] extended with a CGRA following the design presented in [46] for biomedical applications, which is composed of 16 reconfigurable cells (RCs) forming a 4×4 torus interconnect. The CGRA can be integrated with the SoC-domain (i.e., connected to the FC), or in the cluster domain (i.e., connected to the cores of the CL), accessing the L2 or L1 memories directly, respectively, as shown in Fig. 3.5. In this contribution, I use a CGRA divided into four independent columns of RCs; each kernel may use 1, 2 or 4 columns. Unused columns remain clock-gated. The configuration memory is implemented as a 2 KiB standard cell memory (SCM). The cores make acceleration requests by writing a kernel ID to the CGRA peripheral registers (one per core). The CGRA synchronizer maps the request to the number of columns necessary to execute the specified kernel. When a core requests an acceleration, it becomes clock-gated until the request is completed. The RCs of the CGRA have a 16-bit datapath, which is suitable for most WSN-based biomedical applications whose input data is typically limited by ADC resolution. However, several modules, such as the signal enhancement, require 32-bit accumulation; thus, it cannot be accelerated with the current platform design.

3.4.4.1 Kernel Selection

The kernel selection procedure for the CGRA follows the steps described in [131, Chap. 3]. LLVM is used to analyze the application from the C code and generate an execution profile report. This enables the identification of

Table 3.2 – Computational kernels executed on the CGRA

Algorithm	Kernel	Notes
Morph. Filt.	dblmin / dblmax	Linear 1st and 2nd min./max. search in a vector
Fiducial points	maxpeak	Linear peak (absolute max.) search in a vector
Beat classification	min_max	Circular min. and max. search in a vector

computationally intensive loops that are good candidates for CGRA acceleration.[2] Finally, kernels that do not meet the design constraints of our CGRA are discarded. In that sense, the main limiting factor is the small instruction memory of the CGRA (16 32-bit instructions per RC), which restricts the selection to short kernels. Table 3.2 lists the kernels executed on the CGRA.

3.4.4.2 Kernel Mapping

To map kernels on the CGRA, the C code disassembly is inspected to identify operations that can be parallelized. Then, these operations are translated to the CGRA instruction set and distributed over the RCs and columns. This last step is done manually to fully exploit the torus interconnect of the CGRA—each RC is connected to its neighbours—generating the data flow execution that is one of the advantages of this CGRA design.

3.4.5 Experimental Setup

3.4.5.1 Test Benches for Biomedical Modules

I designed a test bench for each module that includes appropriate input signals. For the filtering, signal enhancement, and signal delineation modules, I consider excerpts of signals from the Physionet QT database (QTDB) [78]. This database was used to analyze the three single-core benchmarks presented in Section 3.2 by [41]. I chose four signals from the QTDB, as four examples that

[2] If LLVM is not available for the target platform, cycle-accurate simulators, such as those available in the PULP SDK, can be used in combination with processor HW counters to profile the main blocks of the application.

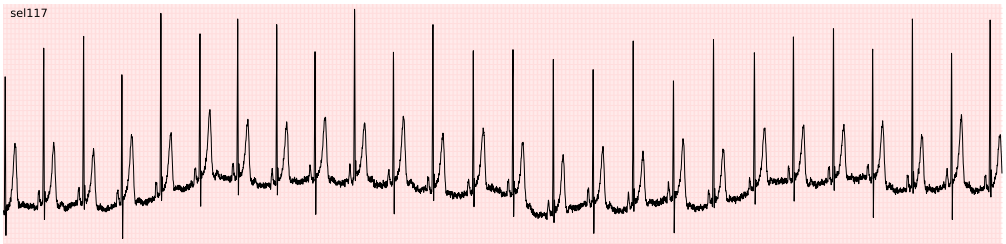
represent worst, best, and two average cases in terms of a combination of noise and shape of the three ECG waves (Fig. 3.6). For the inference module, I consider the MIT-BIH Arrhythmia Database (MITDB) [132], as reported in [123]. I chose four signals as worst, best and two average cases in terms of percentage of abnormal beats over the total number of annotated beats (Fig. 3.7). Its output is a label classifying the beat depending on the pathology: "N" for normal beats, "V" for premature ventricular contraction, "L" for left branch block and "U" for unknown. For all the modules, the choice of four cases should describe most of the design space in terms of complexity and energy consumption due to data-dependent variability. The performance in terms of accuracy of all the methods was not affected by the parallelization process.

3.4.5.2 Test Benches for Biomedical Application

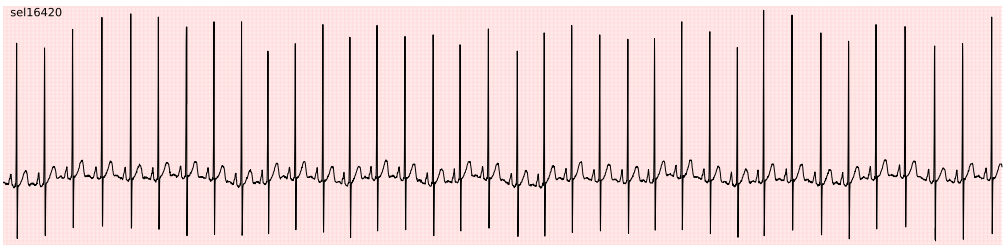
To better evaluate the impact of the proposed optimizations, I evaluate two applications, with data capturing periods, using our biomedical modules. First, I consider a 3-lead heartbeat classifying application [123]. This application applies MF, Rel-En, and R peak detection on one lead (lead I). If the heartbeats are classified as normal, the algorithm goes to the next window of analysis. However, if any abnormality is detected (e.g., the beat is classified as "V","L" or "U"), then it applies the same methods and fiducial points detection to the other two leads (leads II & III) to supply additional information. Second, I implement an application processing the complete set of 12-lead ECG signals. Such application is required for medical compliance and used in intensive care units of hospitals, or in athletic or military training supervision. It combines the modules MF, RMS (to combine all the signals into a single one), R peak, and fiducial points detection. Both applications capture ECG samples during 15 s; then, the system becomes active to process. The performance in terms of accuracy of both applications was not affected by the parallelization process.

3.4.5.3 Multi-Core WSN Platform: PULP+CGRA

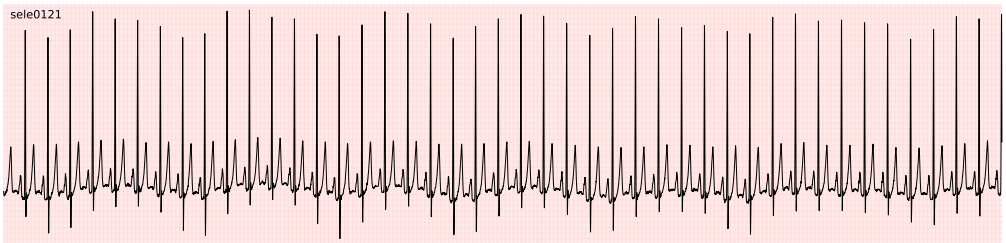
To measure the execution time of both independent modules and complete applications I used the open PULP platform [130]. PULP provides the RTL description of the multi-core platform and an SDK to run RTL simulations,



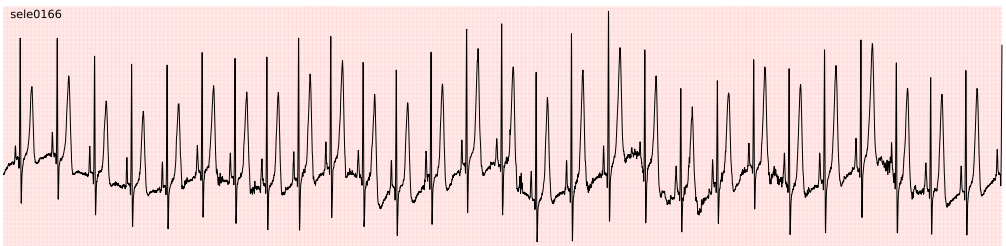
(a) 1



(b) 2

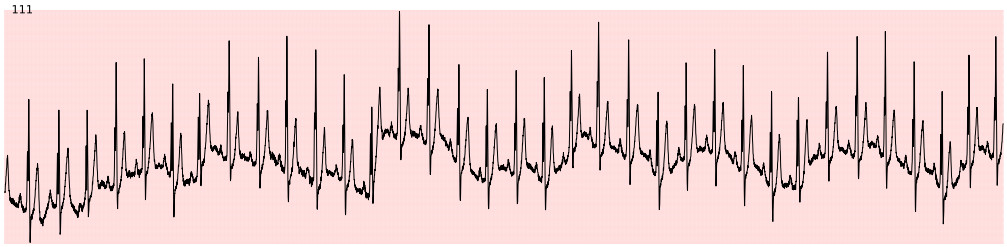


(c) 3

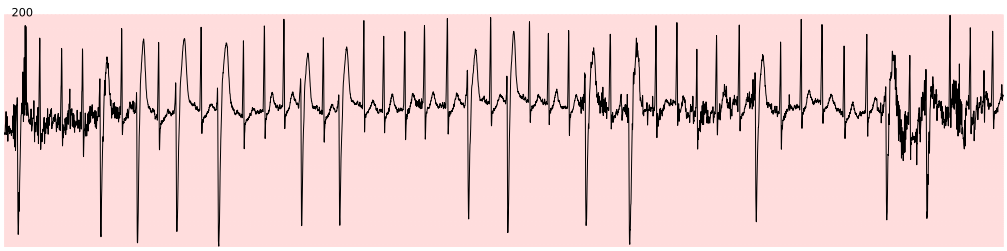


(d) 4

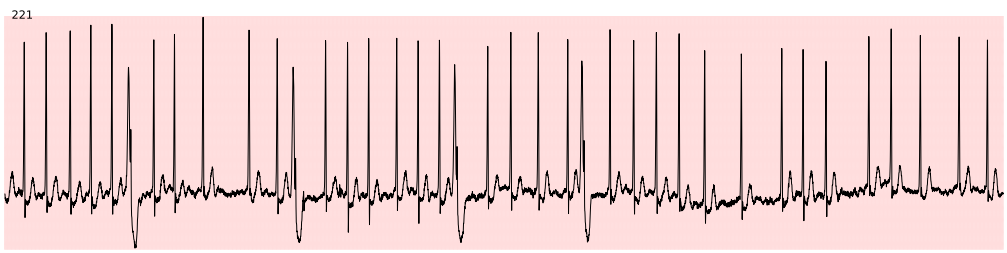
Figure 3.6 – Four signals selected from the Physionet QT database (QTDB) [78] representing best, worst, and two average cases in terms of a combination of noise and shape of the three ECG waves. The four signals are shown on standard ECG sheets containing small squares of $1\,\mathrm{mm}{\cdot}1\,\mathrm{mm}$ corresponding to $40\,\mathrm{ms}$ (horizontal) and $0.1\,\mathrm{mV}$ (vertical) [60]. They also include big squares of $5\,\mathrm{mm}{\cdot}5\,\mathrm{mm}$, and correspond to $200\,\mathrm{ms}{\cdot}0.5\,\mathrm{mV}$.



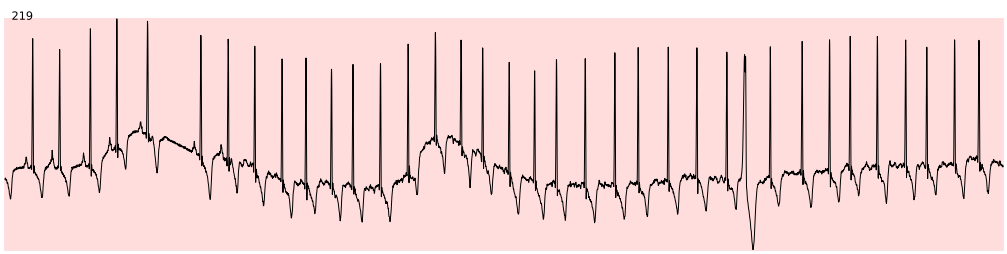
(a) 1



(b) 2



(c) 3



(d) 4

Figure 3.7 – Four signals selected from the MIT-BIH Arrhythmia Database (MITDB) [132] representing worst, best, and two average cases in terms of percentage of abnormal beats over the total number of annotated beats. The four signals are shown on standard ECG sheets containing small squares of $1\,\mathrm{mm}{\cdot}1\,\mathrm{mm}$ corresponding to $40\,\mathrm{ms}$ (horizontal) and $0.1\,\mathrm{mV}$ (vertical) [60]. They also include big squares of $5\,\mathrm{mm}{\cdot}5\,\mathrm{mm}$, and correspond to $200\,\mathrm{ms}{\cdot}0.5\,\mathrm{mV}$.

using Modelsim, in order to obtain cycle-accurate timings. Additionally, to further explore the advantages of heterogeneous platforms, I consider the CGRA that was added to the cluster domain integrating it in the existing cycle-accurate simulation flow. I used the power numbers reported for a chip based on the PULP architecture implemented in TSMC 40 nm LP CMOS technology, Mr.Wolf [49]. This SoC features a streamlined 12 k-gates RISC-V main processor (Zero-riscy [125]) (i.e., FC) and an 8-core compute cluster (i.e., CL) with DSP extensions (RI5CY). This platform includes eight physical memory banks for the 512 KiB L2 memory. I picked the lowest energy point of the platform, at 0.8 V. The platform requires 3.6 µW when power-gated[3] and 12.6 µW with full L2 retention—since typical biomedical applications require small amounts of memory, the size of the L2 was reduced to one fourth (i.e., 128 KiB), while maintaining the same bank number, and correspondingly reducing its power requirements. When the SoC is active, it requires 0.98 mW with its main processor clock-gated, and 6.66 mW with it operating at 170 MHz. Once the CL is activated, it requires 0.61 mW with all cores clock-gated and 18.87 mW with the eight cores running at 110 MHz. The power estimations for the CGRA are obtained through pre-layout netlist simulation with the TSMC 40 nm LP CMOS technology. The CGRA requires 104 µW when idle, with an average power of 669 µW when active. The CGRA and the CL are power-gated together.

First, I performed the RTL simulation and estimated the energy consumption on the test benches for biomedical modules to show the impact of the modular SW optimizations, as shown in Section 3.4.6.1. Then, I ran the RTL simulation and estimated the energy consumption on the complete applications to report in Section 3.4.6.2 the impact of parallelization, memory scaling, and HW acceleration.

[3] As reported for GAP-8 [48], which is an industrial version of PULP with SoA deep sleep optimizations not yet included in its academic counterpart.

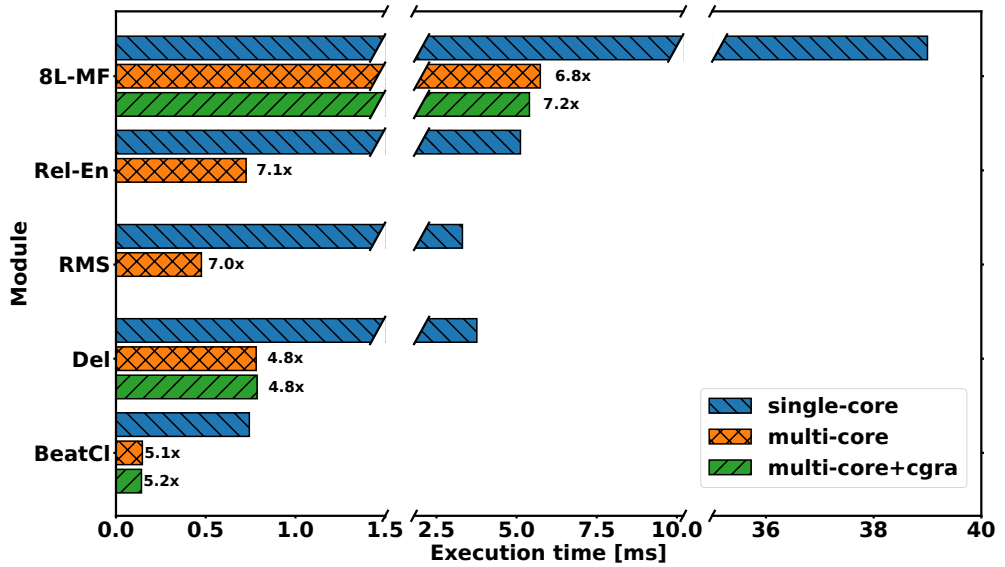


Figure 3.8 – Computation time and corresponding speed-up of each analyzed module for multi-core and multi-core + CGRA implementations versus a single-core one.

3.4.6 Experimental Results

3.4.6.1 Per-Module Speed-Ups and Energy Savings on PULP

Figure 3.8 shows the execution time of each module with the single- and multi-core implementations and the geometric mean of the obtained speed-ups. The maximum speed-up (7.1 ×) is reached in the Rel-En module, despite its small overhead due to the window overlapping scheme. For the remaining modules, the speed-up varies between 4.8 × and 7.0 ×, which is above the threshold of speed-up for the PULP platforms discussed in Section 3.3.4. The RMS module, which applies a data-level parallelization, reaches a speed-up of 7.0 ×, since the eight cores work independently on similar workloads. The MF module is executed on the same trace repeated for the eight leads to have the same workload and show a data-independent multi-core processing. This module achieves a similar speed-up of 6.8 ×, which is justified by two factors: the eight cores in the CL run at a lower frequency than the FC (i.e., ≈ 0.65 ×), but they have a higher IPC.

Table 3.3 – Execution time of the delineation module for different subjects from the Physionet QTDB [78] and the subsequent varying speed-ups

SUBJECT	SINGLE-CORE (ms)	MULTI-CORE (ms)	SPEED-UP
1	2.77	0.67	4.13 ×
2	3.66	0.78	4.69 ×
3	4.72	0.93	5.08 ×
4	3.88	0.75	5.17 ×

Table 3.4 – Energy savings in the delineation module on four subjects from the Physionet QTDB [78] for the single-core and multi-core platforms

SUBJECT	SINGLE-CORE (µJ)	MULTI-CORE (µJ)	SAVINGS (%)
1	18.4	11.3	38.6
2	24.3	13.6	44.0
3	31.4	16.4	47.8
4	25.8	13.2	48.8

The minimum speed-up (4.8 ×) is obtained for the delineation module (Del) because the workload cannot be divided evenly among the cores: first, the R peak detection algorithm has several data-dependent conditional branches that change the execution path for different cores; second, the beat parallelization used during the delineation depends on how many peaks are detected; finally, the beat length (i.e., the RR interval) is variable and, hence, the size of the input varies for each core. This effect can be observed in the time spent in the delineation module (Table 3.3) for four different subjects from QTDB.

The previous speed-ups translate neatly into energy savings. Figure 3.9 reports the geometric mean of the energy consumption for each module over the four chosen subjects of [78] and [132]. The maximum energy savings of the multi-core design correspond to the RMS (60 %) and Rel-En (58 %)

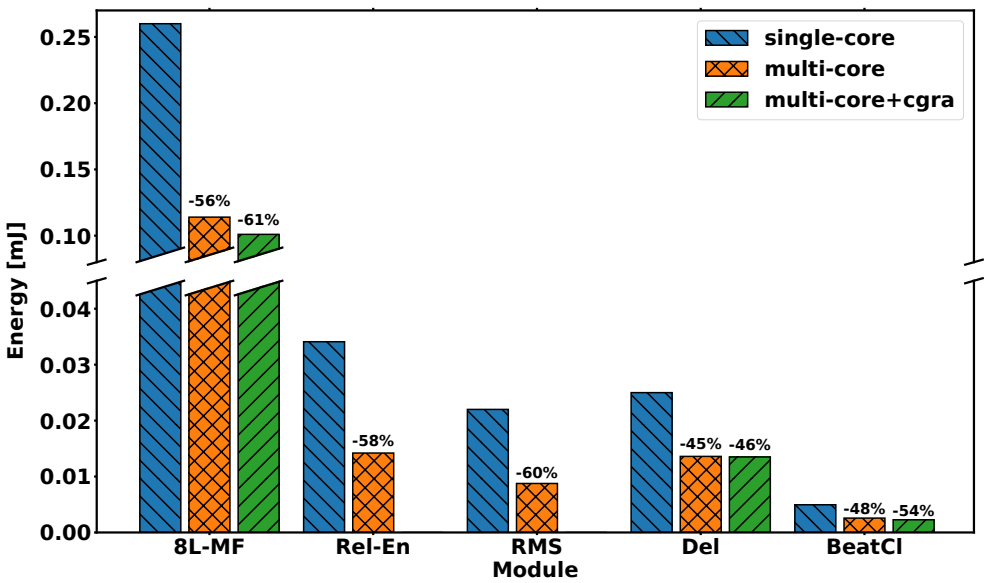


Figure 3.9 – Per-module energy consumption and savings (geometric mean) compared to the single-core design.

modules, which are also the modules with the highest speed-up. Again, the minimum energy savings (45%) are obtained for the delineation module due to the variability in the load of each core (Table 3.4).

3.4.6.2 Application-Level Energy Savings on PULP

I evaluate the impact of the previous optimizations on two different modular applications, including the energy spent during data capturing periods. First, I consider a 3-lead heartbeat classifying application [123] in three different configurations depending on the optimizations discussed in Section 3.4.3. Then, I consider the 12-lead ECG delineation application. Table 3.5 shows the energy and time results for these applications. The values reported include memory scaling to banks of 16 KiB on both single- and multi-core implementations.

The multi-core configuration of the platform is the most efficient option in the four cases analyzed. Even for the 1-lead application, where MF is the most expensive module (i.e., 81.6 % of the active time) and it is executed

Table 3.5 – Average results of energy consumption (including data capture) and execution time on PULP (with memory scaling) for the complete applications on four subjects

# **of leads**	**Single core**		**Multi-core**			
	Energy (mJ)	Time (s)	Energy (mJ)	Savings (%)	Time (s)	Speed-up (×)
1 lead	0.326	0.025	0.302	7.3	0.019	1.28
1+2 leads	0.611	0.068	0.588	3.7	0.041	1.66
3 leads	0.611	0.068	0.493	19.3	0.023	2.95
12 leads	1.78	0.238	1.00	43.5	0.046	5.14

on the FC, by parallelizing the other modules on the CL, I obtain modest energy savings (7.3 %). The total speed-up is low (1.28 ×) due to the small percentage of parallel code. However, the average speed-up of all the other parallel modules (approximately 5.6 ×) and the memory scaling are enough to achieve fair savings. However, when the application detects abnormal beats the following strategies (1+2 leads and 3 leads) can be applied. In the first case, which follows the optimizations of [123], processing the additional two leads *after* the first one limits the energy savings since the obtained speed-up is not enough to offset the energy of the cores of the CL during the extended period. However, if the beat classifier detects abnormal events often enough, the application can use the second strategy and process the three leads in parallel. In that case, the parallel version would achieve a reduction in computation time of 66 % and 19.3 % in energy. In this way, the three leads are analyzed simultaneously on three active cores of the CL while the others are clock-gated, enabling better energy savings.

Considering the low computational load of this application, the energy savings of the multi-core optimization are modest but still significant. However, applications requiring medical compliance, such as in intensive care units of hospitals, or in athletic or military training supervision, must process the complete set of 12-lead ECG signals, which generates higher computational load. The last row of Table 3.5 shows that the parallel version achieves, in this case, a speed-up of 5.14 × and energy savings of 43.5 %.

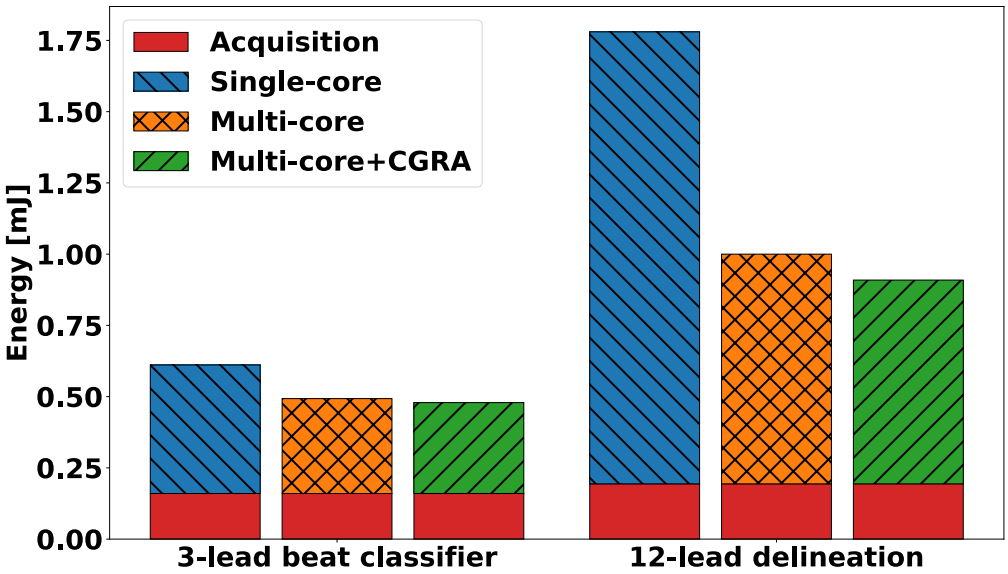


Figure 3.10 – Decomposition of energy consumption for the 3-lead and 12-lead ECG applications for PULP, including memory scaling.

I investigate the use of HW acceleration for the cases of 3-lead and 12-lead ECG signals, which can be observed in Fig. 3.10. The savings achieved by accelerating some intensive computational kernels in the 3-lead beat classifier application are 67 % in time and 2.9 % in energy compared to the multi-core implementation. The reason for the modest energy saving is the low computational load of the 3-lead application. Moreover, the minimalist CGRA design covers only a small amount of the total number of executed instructions, limiting its impact. Compared to the single-core implementation, it represents 21.6 % of energy savings. For the 12-lead application, the impact is more significant due to the higher computational load, with 9.6 % of additional energy savings compared to the multi-core implementation. However, as the figure shows, for low duty cycle applications, such as the 3-lead beat classifier, the energy consumed by the memories during sampling, although not dominant, is significant. In the case of the 12-lead application, the energy consumed during computation is much higher than the energy consumed by the memories during the sampling period (Fig. 3.10), hence the higher savings achieved. In fact, the energy during memory management was highly reduced by applying size scaling of each memory bank from the original

64 KiB of [49] to 16 KiB and memory management to keep only the bank needed by the application in active or retentive state. In applications with low computation load, one possible solution would be to design the SRAMs with a more significant number of banks and scale to the feasible resolution to enable a more aggressive power management during data sampling periods. I will explore different scaling factors for the second contribution of this chapter, described in Section 3.5.

Finally, in Table 3.6, I show a summary of the energy savings compared to the single-core configuration applying the optimizations described in Section 3.4. The three main optimizations, including parallelization, memory scaling and HW acceleration, can be applied orthogonally and significantly reduce the energy consumption compared to the traditional single-core implementation. For example, by applying memory scaling directly to the single-core implementation, the energy savings reach up to 23.45 % (this result corresponds to the value of the first column of Table 3.5 within a small rounding error). Additionally, it is possible to apply HW acceleration not only on the multi-core implementation but on the single-core design, achieving energy savings from 9.03 % up to 27.05 %. Therefore, the designer of WSN-based biomedical applications should take into account modularity and parallel implementation, memory scaling and, HW acceleration.

3.5 Patient-Specific Optimizations for Multi-Core Ultra-Low Power Platforms

With the final goal of saving energy for a personalized continuous monitoring and explore further the optimizations already presented in Section 3.4, I design a patient-specific parallelization technique targeting a multi-core platform and I apply memory and power management. As a starting point to apply these optimizations, I consider the online PAF prediction algorithm tested in a single-core platform analyzed in Section 2.4.2. The parameters and model selected during the learning phase, described in Section 2.4.2.3, are the configuration inputs to assign the resources in a multi-core wearable embedded device considering the specific characteristics of the patient.

Table 3.6 – Summary of energy savings applying the SW parallelization techniques, the HW acceleration and the memory scaling for the analyzed applications on the PULP platform. "1 lead", "1+2 leads", and "3 leads" represent different configurations of the heartbeat classifier. The "12 leads" corresponds to the 12-lead delineation.

		Heartbeat classifier			Delineation
		1 lead	1+2 leads	3 leads	12 leads
Energy (mJ)	Single-core	0.43	0.71	0.71	1.86
Energy savings (%)	Multi-core	6.51	3.18	16.56	41.57
	Single-core + CGRA	3.61	4.68	4.68	3.61
	Single-core + Memory scaling	23.45	14.05	14.05	4.53
	Multi-core + CGRA	6.59	6.37	18.58	46.73
	Multi-core + Memory scaling	29.95	17.23	30.60	46.10
	Single-core + CGRA + Memory scaling	27.05	18.73	18.73	9.03
	Multi-core + CGRA + Memory scaling	**30.03**	**20.42**	**32.62**	**51.26**

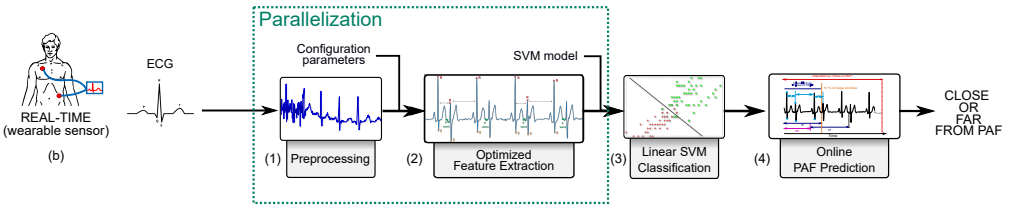


Figure 3.11 – Block diagram of the real-time personalized PAF prediction highlighting that steps that are parallelized.

3.5.1 Patient-Specific Parallelization for Multi-Core Platforms

Considering the number of consecutive beats selected at training time for each patient (n), I apply a patient-specific parallelization by varying the number of cores depending on n. The method consists of a three-step parallelization process by using $\#cores = n+1$, where n changes depending on the patient. Figure 3.11 shows the blocks of the real-time PAF prediction that are parallelized. The filtering step was designed to be parallelized on leads [46], as shown in Section 3.4. However, since I use a single-lead ECG, this step is not parallelized in this application. The three steps that can be parallelized are the signal enhancement preprocessing, Rel-En, the R peak detection steps of REWARD [41], and the selective feature extraction (cf. Section 2.4.3.1 and Section 3.2.3). Since the selective feature extraction includes the computation of the RR interval, the algorithm needs $n+1$ consecutive beats to extract all the necessary features. Considering that $n = 3,4,...7$, I can assign from four to eight cores for the multi-core implementation. In order to parallelize on the feature extraction, $n+1$ consecutive beats are required, hence, the multi-core implementation needs to collect a specific window of analysis and then process the data. In this way, I exploit the characteristics of modern ULP architectures, by storing a buffer using the DMA until it reaches the desired length, processing at the lowest voltage, but also at the maximum operating frequency, and enabling a faster computation.

Fig. 3.12 shows the three steps of the algorithm that are parallelized. Let us consider an example where at training time, the method chose $n = 3$ for one specific patient, therefore $\#cores = 4$ are assigned. Since the first two steps are part of the REWARD algorithm for R peak detection, I can define the

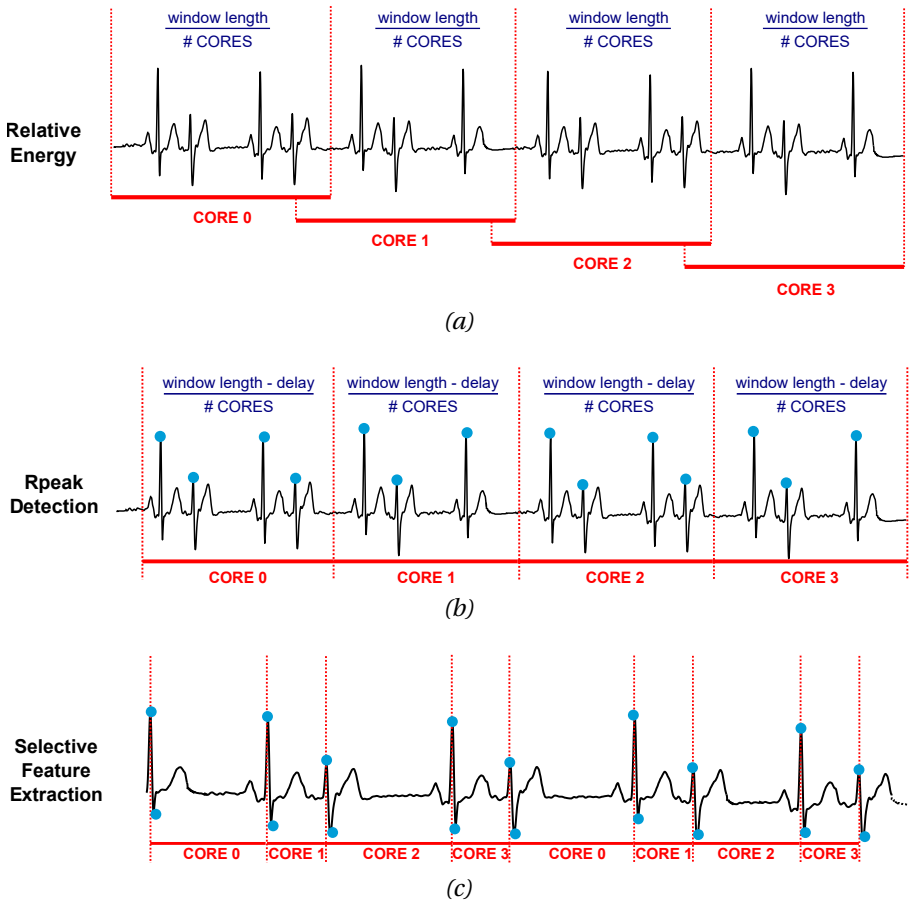


Figure 3.12 – Proposed personalized parallelization by choosing the number of cores based on n number of consecutive beats to analyze, so that $\#cores = n + 1$.

window length of analysis based on how many peaks the R peak detection can detect. The R peak detection needs a minimum window length of 1.75 s to detect at least one peak. Since I need four peaks the window of analysis is $(1.75 * 4)$ s. Moreover, the Rel-En step computes the energy of the signal for each sample by using the information of a specific window of 0.95 s [41]. Therefore, the final window length for this specific case is $(1.75 * 4 + 0.95)$ s. Fig. 3.12a shows the parallelization applied to the Rel-En step. For this case, I can parallelize on four cores by dividing the window of analysis in four parts and assigning different windows to different cores, with an overlap. Fig. 3.12b

shows the parallelization applied to the R peak detection step. In this step, the cores are assigned to four windows of 1.75 s, subtracting the delay of 0.95 s. Both steps use the window parallelization technique described in Section 3.4.1.2. Finally, the last step is the selective feature extraction, which uses the beat parallelization technique described in Section 3.4.1.2. This is performed within a peak-to-peak interval, and each core is assigned to one of the beats within the small window of analysis, containing in this case four peak-to-peak intervals. Since within 1.75 s there might be more than one peak, the feature extraction will reassign the cores to the next small window of three consecutive beats. All in all, the proposed parallelization exploits the patient-specific model parameters to save energy for an optimal personalized continuous monitoring in an ULP multi-core platform.

3.5.2 Memory and Power Management

Considering the modern ULP multi-core wearable sensors, in particular the GAP8 architecture [48], I apply memory management to the multi-core design to save energy during the signal buffering, which depends on the patient model. Considering the GAP8 L2 memory characteristics, I explore the possibility of reducing the total L2 memory size and, therefore, the bank size to meet the conditions of the application. First, for the single-core design, I consider a total of 64 KiB as a baseline, increasing the number of banks to eight and reducing the bank size to 8 KiB, since up to 48 KiB are required. Then, for the multi-core design, I explore lower bank sizes of 4 KiB, 2 KiB, 1 KiB with a total of 32 KiB, and increasing the number of memory banks. Considering smaller bank sizes, it is possible to power off the unneeded banks, based on the window length of analysis, thus on the patient model. Moreover, since the buffer requires more banks I alternate retention/active mode for buffer slices already stored. Hence, if for two patients I use windows of analysis from 4 KiB to 8 KiB, by reducing the bank size resolution (for example, to the minimum of 1 KiB) I can have a more efficient patient-specific memory management and an overall reduction in energy consumption. Finally, my design includes switching to deep sleep mode between the acquisition and storage of two samples since the signal sampling frequency is low.

3.5.3 Experimental Setup

In this section I describe the database used for training the models, the test bench and the platform analyzed for the multi-core design.

3.5.3.1 Database and Test Bench for Multi-Core Design

I apply the framework to the PAF Prediction Challenge (2001) Physionet database [108], which contains 53 patients affected by PAF. For each patient two 30-minute ECG signals close to and far from a PAF event are acquired at a sampling frequency of 128 Hz and, then, resampled at 250 Hz. The personalized training process has been described in Section 2.4.2. The training data includes the last 350 beats of the recording (approximately 3–9 min considering a HR range from 40 beats per minute (BPM) to 110 BPM). The testing process is done on the remaining of the recording for the signals close to a PAF event. For the signal far from any event, the method configures the minimum window of prediction on two-thirds of the remaining signal (cf. Section 2.4.2.3) and then tests on the remaining third.

I chose six cases that vary in terms of configuration parameters to evaluate one window of analysis when a PAF event occurs (the same as the ones reported in Section 2.4.4.1). I consider a set of input parameters trained for each patient, namely, the window length-related parameters n and m, the selected group of features, and the classification threshold, described in Section 2.4.2.2 and Section 2.4.2.3. Specifically, I select from a worst to a best case scenario, considering the sum of each configuration parameter computational cost. Finally, the window of analysis varies from 15 s to 45 s, depending on the patient. The multi-core method is designed to collect a window buffer depending on the number of cores (i.e., number of consecutive beats different for each patient).

3.5.3.2 Platform for Multi-Core Design: The GAP8 Sensor

GAP8 [48] is a commercial RISC-V implementation based on the PULP project [45] and built on a 55 nm technology. Its structure is similar to the main PULP architecture described in Section 3.3, with a main core (i.e., FC), and a cluster of eight cores (i.e., CL). In order to observe the energy savings,

I implement a window-based single-core design, similar to the multi-core one but running on the FC of GAP8 [48], instead of the CL. I use an open source SDK that simulates a RISC-V PULP platform [130] to profile the window-based single-core and multi-core designs. I profile the active cycles of the different cores in the RTL simulation and, then, I estimate the energy consumption using the power numbers for the GAP8 platform provided by [48], running at the lowest possible voltage supply 1 V and maximum operating frequency of 150 MHz in the SoC and 90 MHz in the CL. Moreover, during the idle time the platform is set at the lowest power leakage of 3.6 µW, since the deep sleep mode mostly dominates the consumption in the analyzed application. Furthermore, the GAP8 platform contains a 30 µW fully retentive memory of 512 KiB that can be divided into four banks of 128 KiB. Then, since my personalized and parallelized AF prediction algorithm does not need more than 64 KiB of storage, the overall memory was reduced to 64 KiB; thus, reducing the consumption to 10 µW, accounting for the leakage and the memory retention. Additionally, I explore bank size scaling from 8 KiB, 4 KiB, 2 KiB to a minimum of 1 KiB. To account for the memory and power management I estimate the energy spent in storing the window buffer with different bank sizes and entering deep sleep mode between the samples.

3.5.4 Experimental Results

In this section, I report the energy consumption and corresponding savings of the real-time personalized window-based PAF prediction approach on the analyzed multi-core. First, I show the energy savings achieved by the personalized parallelization technique in active mode. Then, I report the energy savings achieved by applying power and memory management, specifically the personalized bank size scaling. Finally, I show the comparison of the total energy consumption between the single- and the multi-core designs.

3.5.4.1 Energy Savings in Personalized Multi-Core Design

The ULP multi-core architecture described in 3.5.1 assigns a different number of cores depending on the patient model. In this section, I discuss the energy savings derived by applying the patient-specific parallelization.

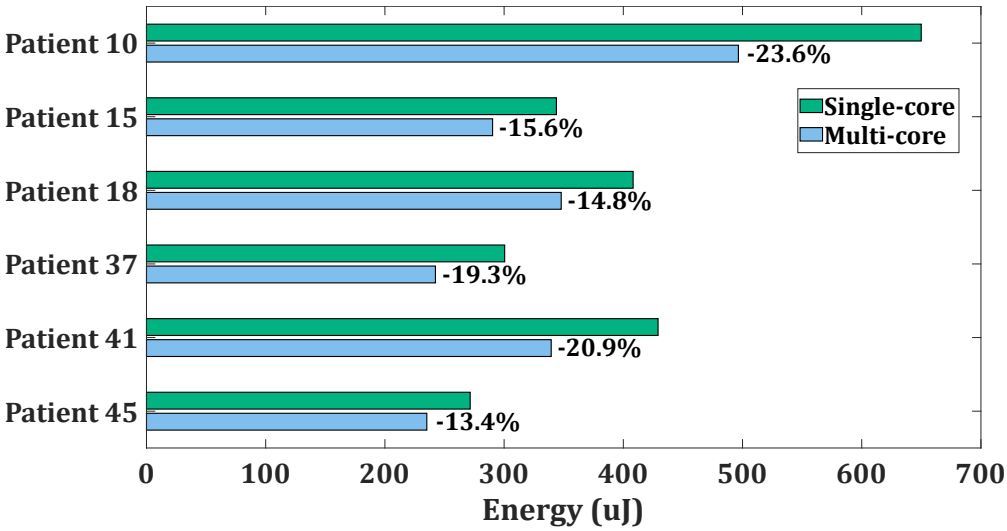


Figure 3.13 – Energy consumption in µJ during processing for the selected six cases for single-core and multi-core.

Fig. 3.13 shows the energy consumption of the six cases of the test bench. I report the energy consumed in the single-core window-based design compared to the multi-core. For the six cases, the multi-core design performs better than the single-core design with energy savings from approximately 13 % up to 24 %. The difference between the patients is directly depending on the corresponding training model and patient-specific parallelization. Since the number of cores depends on the small window length of n consecutive beats, the personalized selection of the window length directly affects the energy consumption. To describe this effect, I refer to the different configurations of the model described in Fig. 3.4 (cf. Section 3.3.5). In this figure, I report the cases analyzed by considering only the potential energy savings of the parallel implementation, while Fig. 3.13 shows the energy savings of the full implementation. Moreover, in Fig. 3.4, I also consider the energy consumed in idle time with one bank active. However, the final results in energy savings reflect the analysis previously mentioned. Furthermore, Table 3.7 reports a summary of the application features for the six analyzed cases and the corresponding energy results during processing, according to the analysis done in Section 3.3.5.

Let us consider two examples within the 5-core configuration, namely, Patient 15 and Patient 37, which have a duty cycle of 0.038 % and 0.044 %, respectively, for the parallel implementation. Even though the case of Patient 15 reaches a higher speed-up of 3.96 × (3.63 × for Patient 37), the case of Patient 37 achieves higher energy savings. Moreover, if we consider the case of Patient 45, which achieves a speed-up of 3.02 × with only four cores active, the parallel implementation (that only represents 30 % of the total computation) achieves better energy savings compared to a higher number of active cores for the same speed-up. Finally, if we consider the case of Patient 10, implemented in an 8-core configuration, it has a duty cycle of 0.069 % (approximately 43 % of the total computation) and it reaches a speed-up of 4.53 ×. This case achieves the highest overall energy savings of 23.6 %, which is explained by the higher value of duty cycle compared to the other cases. In fact, the speed-up is comparable to the case of Patient 18, implemented with a 6-core configuration, but its duty cycle is double. Moreover, if we compare the cases of Patient 10, Patient 37, and Patient 41 (duty cycle of 0.041 % and highest speed-up of 5.46 × with a 7-core configuration), it is possible to observe the effects of the trade-off between the three variables analyzed in Section 3.3.5. Indeed, as shown in Table 3.7, the three cases achieve high and comparable energy savings due to a varying number of active cores, speed-up, and duty cycle. This effect is also visible in the cases of lower energy savings in Fig. 3.4 and the corresponding values in Table 3.7.

The results show how the patient-specific multi-core design allows to scale the computational load by assigning the number of cores depending on the personalized application model and, therefore, scale and save energy.

3.5.4.2 Energy Savings with Memory Banks Management

As described in Section 3.5.2, I apply memory management scaling the bank sizes of the GAP8 L2 memory to 8 KiB, 4 KiB, 2 KiB and 1 KiB. In Table 3.8, I show the results of the energy savings for the memory and sleep mode in a window of analysis compared to the single-core window-based design. The single-core bank size is 8 KiB with the necessary number of banks powered on depending on the patient. The results show two orthogonal levels of energy savings. The first one is the buffer length which depends on the number

Table 3.7 – Summary of the energy savings during processing for the six analyzed cases compared to the application features

		Patient					
		10	15	18	37	41	45
Application features	# Active cores	8	5	6	5	7	4
	Speed-up (×)	4.53	3.96	4.20	3.63	5.46	3.02
	Duty cycle parallel (%)	0.069	0.038	0.036	0.044	0.041	0.039
	Parallel code (%)	43.05	28.95	27.04	38.15	33.73	30.43
Results energy processing	Single-core energy (µJ)	649.9	343.9	408.2	300.5	429.1	271.5
	Multi-core energy (µJ)	496.6	290.3	347.9	242.4	339.5	235.2
	Energy savings (%)	**23.6**	15.6	14.8	19.3	20.9	**13.4**

Table 3.8 – Energy savings with memory management considering different buffer lengths and bank sizes

		Bank size (KiB)			
		8	4	2	1
Buffer size (KiB) (patient-specific)	7.5	18.2%	26.1%	27.8%	**29.8%**
	4.8	10.3%	19.6%	24.0%	24.9%
	5.7	10.3%	19.2%	21.2%	23.4%
	4.8	10.3%	19.6%	21.5%	23.7%
	6.6	10.3%	18.93%	20.87%	23.03%
	4	0%	11.1%	15.9%	**16.9%**

of cores, hence, varying from 4 KiB to 7.5 KiB in this application. On this level, by keeping the bank size fixed for all the buffer size cases the memory management strategy achieves energy savings up to 18 % compared to the single-core design. The second level is fixing the buffer size and varying the bank size. By scaling from 8 KiB to 4 KiB, the strategy reaches energy savings up to approximately 10 %. A further scaling to 8 KiB or 1 KiB results in a significant improvement, up to 17 % (i.e., for the smallest buffer length of 4 KiB). Overall, the platform can get up to 30 % of savings during buffer storage and sleeping between samples compared to the single-core design by scaling the bank size to the minimum of 1 KiB.

3.5.4.3 Comparison of Total Energy Consumption in Single and Multi-Core Design

I combine the results of the energy consumed during the processing and in storage and deep sleep mode to show the overall savings of the multi-core design compared to the single-core design. Additionally, I show the energy consumed in the single-core original sample-by-sample design. Fig. 3.14 shows the energy consumed in µJ for the sum of processing (green) and memory and sleep (pink). In all the cases, the energy consumed during the processing is up to 6 × the energy consumed in the memory and sleep. Therefore, the impact of the memory management on the overall energy consumption is reduced to 5–6% from the values presented in Table 3.8, although still significant. The multi-core design reduces the energy consumption more in the cases with more computational load (i.e., Patient 10) up to 33.9 %. The minimum value of energy savings reached in the six cases analyzed is 14.5 % compared to the single-core designs. In this case, the single-core sample-by-sample design can be more or less efficient than the window-based one because of different conditions. First, since the MF is always running on the FC this step accounts for 57 % to 73 % of the total computation, depending on the patient model. Second, the model itself varies the computational load considering the different configuration parameters for each patient (i.e., the features extracted, the small window of consecutive beats and sliding window, and the classification threshold). Finally, the multi-core design uses the L1 memory to store the buffers and variables used by the parallelization steps. Accessing the L1 memory from the CL is more efficient than accessing the L2 from the SoC. The

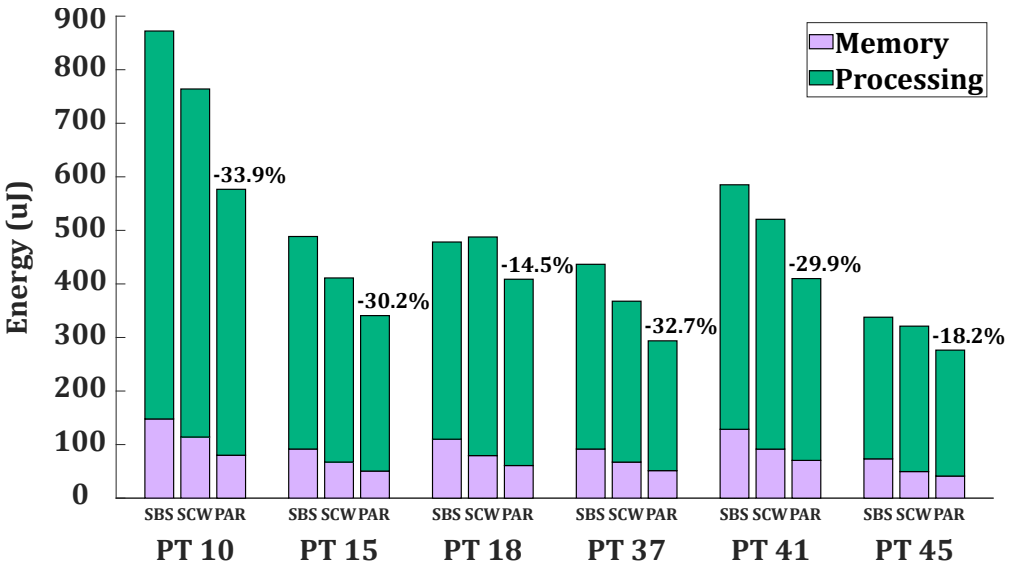


Figure 3.14 – Energy consumption in µJ during processing for the selected six cases ("PT" as "Patient") for single-core sample-by-sample (SBS), single-core window-based (SCW) and parallel (PAR).

results show how a multi-core design where the number of cores and memory bank sizes are scaled according to personalized models for biomedical applications is advised in modern ULP platforms. From the results, even if the parallelization step counts for only 30 % of the total computation, the energy savings of the two combined optimizations show a sufficient improvement compared to the single-core design.

3.6 Conclusion

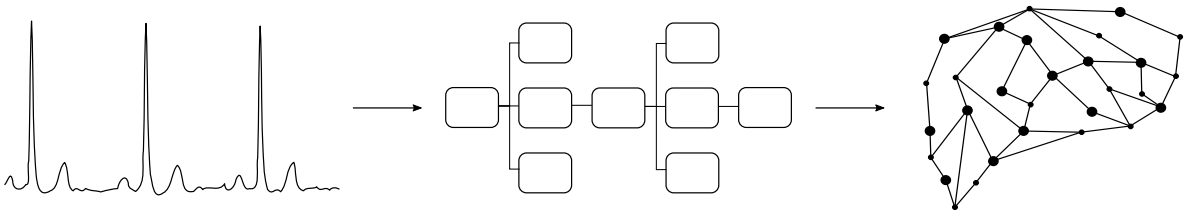
Modern ULP platforms for wearable sensors offer characteristics such as multiprocessing, clock- and power-gating that enable power and memory management and HW acceleration. In this chapter, I have proposed two methods for platform optimizations in the context of WSN-based biomedical applications.

In the first method, I propose a top-down approach of parallelization techniques to improve the mapping of modular biomedical applications. Addi-

tionally, I have shown how heterogeneous platforms can benefit from domain-specific accelerators, such as CGRAs, and memory scaling and management to further reduce energy consumption. I have demonstrated my proposal on a set of independent modules typical of WSN-based biomedical applications and on two composed multi-lead ECG-based applications. The results demonstrate energy savings of up to 60 % for the RMS module and up to 41.6 % for a complete multi-core application processing 12-lead ECG signals for a general PULP platform. Furthermore, I demonstrated that memory scaling is an orthogonal optimization that can be exploited to achieve additional energy savings up to 23.45 %. Finally, the experiments have also established that the domain-specific accelerator used can increase the energy savings to 46.7 % for the 12-lead delineation and 18.6 % for the complete heartbeat classifier. Thus, the overall combined energy savings reach up to 51.3 %.

In the second contribution, I have proven the portability, energy saving and scalability of an online, energy-efficient, and personalized PAF prediction method (proposed in Section 2.4) on the new generation of ULP multi-core architectures, based on the ULP GAP8 IoT architecture. In detail, I proposed a parallelization technique that assigns the number of cores depending on each patient's characteristics. Moreover, I explored the use of memory bank sizing (from 8 KiB to a minimum of 1 KiB) and buffer length sizing for a different number of cores, specific to each patient. My final ULP multi-core design combining processing and memory scalability achieves up to 34 % of energy savings with respect to the original single-core sensor design. A dynamic reconfiguration of the personalized parameters and resource assignment after the occurrence of new PAF events is a highly suitable extension of this work in future research (c.f. Chapter 5).

These two approaches show how platform optimizations are as relevant and needed as algorithmic optimizations to reduce energy consumption though maintaining the high accuracy required by WSN-based biomedical applications. The final step in the design of WSN for remote wellness monitoring is combining the two sets of optimizations described in Chapter 2 and Chapter 3, to design an adaptive system that achieves an optimal energy-accuracy trade-off.



Online Adaptive Design for Enhanced Energy-Accuracy Trade-Off

4

After the exploration of algorithmic and platform optimizations, the last step to reach an optimal energy-accuracy trade-off is through an online adaptive design of algorithms in modern ultra-low power (ULP) platforms. In fact, in biomedical applications for wellness monitoring, the conditions often change quickly (e.g., different physical activity intensities, sudden events in pathologies, etc.), which require adapting in real-time the complexity and accuracy of the algorithms and the use of platform resources.

In this chapter, I propose an online adaptive design of an R peak detection algorithm using an electrocardiogram (ECG), in the context of an incremental physical stress test. When the algorithm detects at run time an accuracy error on a less robust but less complex algorithm that runs by default on the main core, it triggers a more complex but more accurate method on a second energy-efficient core.

4.1 Introduction

In complex biomedical applications for remote wellness monitoring, the output accuracy is of most importance. However, implementing these applications in traditional wearable sensor nodes (WSNs) can cause a substantial draining of platform resources leading to frequent device charging [133]. Moreover, different algorithmic optimizations to lower the device energy consumption can lead to a decrease in the algorithm output accuracy [134]. With

the advent of modern ULP platforms and their capabilities, optimizing the energy consumption of the device resources maintaining a highly accurate output has become more attainable [73, 120, 135, 136]. Nevertheless, in the context of complex biomedical applications for WSN-based wellness monitoring, the designer faces new challenges to reach an optimal energy-accuracy trade-off. First, there exist different pathologies or physical conditions where sudden events occur in the acquired and analyzed biosignals that traditional algorithms can miss or misinterpret (e.g., atrial fibrillation (AF) or intense physical exercise) [56, 137–139], hence, their robustness in these cases is compromised. Second, another problem is the static nature of traditional algorithms and the need for handling and adapting the platform resources at run time according to the complexity of the application. New algorithms tackle self-aware applications at the algorithmic level applying a multi-layer classification or detection system with increasing complexity [123, 140, 141]. Based on the confidence of the low complexity classifiers or the detection of pathological events, the algorithm decides if it will run a more complex layer and therefore consume more energy. However, these algorithms are targeted to traditional homogeneous platforms, and some do not consider the error in the pathological events detection. There are examples reporting the advantages of adaptive design in terms of energy-accuracy trade-offs, such as in [142]. However, it is in the context of activity recognition in mobile phones. For ECG-based applications where sudden events occur, the need for adaptive and robust strategies starts with the R peak detection, as it is the base for most ECG analyses.

For these reasons, in this chapter, I propose an online adaptive design of a new ECG R peak detection algorithm, which exploits the capabilities and heterogeneity of modern ULP platforms. The proposed design introduces for the first time, BayeSlope, a slope-based R peak detection that uses a Bayesian filter, non-linear normalization, and a clustering technique. In the literature, the use of slope-based QRS detectors has been extensive [61, 68]. There are examples of the use of the Kalman filter for smoothed estimation of the heart rate (HR), different than R peak detection, and using multiple signals [143]. However, many of these works target ambulatory monitoring. Therefore, to the best of my knowledge, this is the first time that an R peak detection like

BayeSlope is used in the context of intense physical exercise. In fact, I apply the proposed method to a dataset collected in collaboration with the Institut des sciences du sport de l'Université de Lausanne (ISSUL), where the subjects performed an incremental stress test on a cycle ergometer till exhaustion. The outcomes of this contribution are:

- I propose a new highly accurate slope-based R peak detection, called BayeSlope, based on unsupervised learning. My new R peak detection method applies a Bayesian filter and a non-linear normalization to the input signal to enhance and correctly detect the next R peak in the expected position on a peak-to-peak resolution.

- I pair the newly proposed algorithm with the REWARD algorithm, presented in Chapter 2, which is less complex though more prone to error if sudden events occur. To ensure the adaptive nature of the design, I propose an error detection routine applied to REWARD that triggers BayeSlope if REWARD fails.

- The heterogeneity of the platform allows to run BayeSlope on a more capable core than the one where REWARD runs, which is simpler. In fact, the R peak detection step of REWARD is approximately 104 × less complex than BayeSlope when running on the same core. Therefore, a simpler and faster processor can handle it better, while a more powerful core handles better the more complex BayeSlope.

- The fully adaptive process has an F_1 score of up to 99.0 % compared to 92.5 % when running only REWARD, across five different exercise intensities. Moreover, the adaptive process loses less than 1 % in accuracy compared to always running BayeSlope, which achieves an F_1 score up to 99.3 %, across the five exercise intensities. However, the adaptive method implemented in modern heterogeneous platforms can reach energy savings up to 38.7 % compared to always executing BayeSlope. Therefore, the newly proposed adaptive design is the best solution for an optimal energy-accuracy trade-off.

In Section 4.2, I describe the background of the application analyzed and the relevance of a highly accurate R peak detection in such conditions. In

(a) *(b)*

Figure 4.1 – Standard protocol for gas analysis during incremental exercise stress test on a cycle ergometer

Section 4.3, I present the new R peak detection algorithm and its adaptive design. In Section 4.4, I describe the protocol of the experiments and the platform used. Finally, in Section 4.5 and Section 4.6, I present, respectively, the results and the conclusion of my analysis.

4.2 Background

In Chapter 2, I described the medical relevance of the R peak detection in an ECG for the diagnosis of cardiovascular diseases (CVDs) and for the analysis of the heart rate variability (HRV) in wellness monitoring. In this chapter, I focus on the importance of the latter in the context of assessing the respiratory and cardiovascular state during intense physical exercise. To capture this state, there exists a gold standard protocol where the subject performs an incremental exercise stress test on a cycle ergometer or a treadmill till exhaustion wearing a gas mask (c.f., Fig. 4.1) that measures the volume of O_2 and CO_2 (VO_2, VCO_2) inhaled and exhaled [144, 145]. Additionally, the

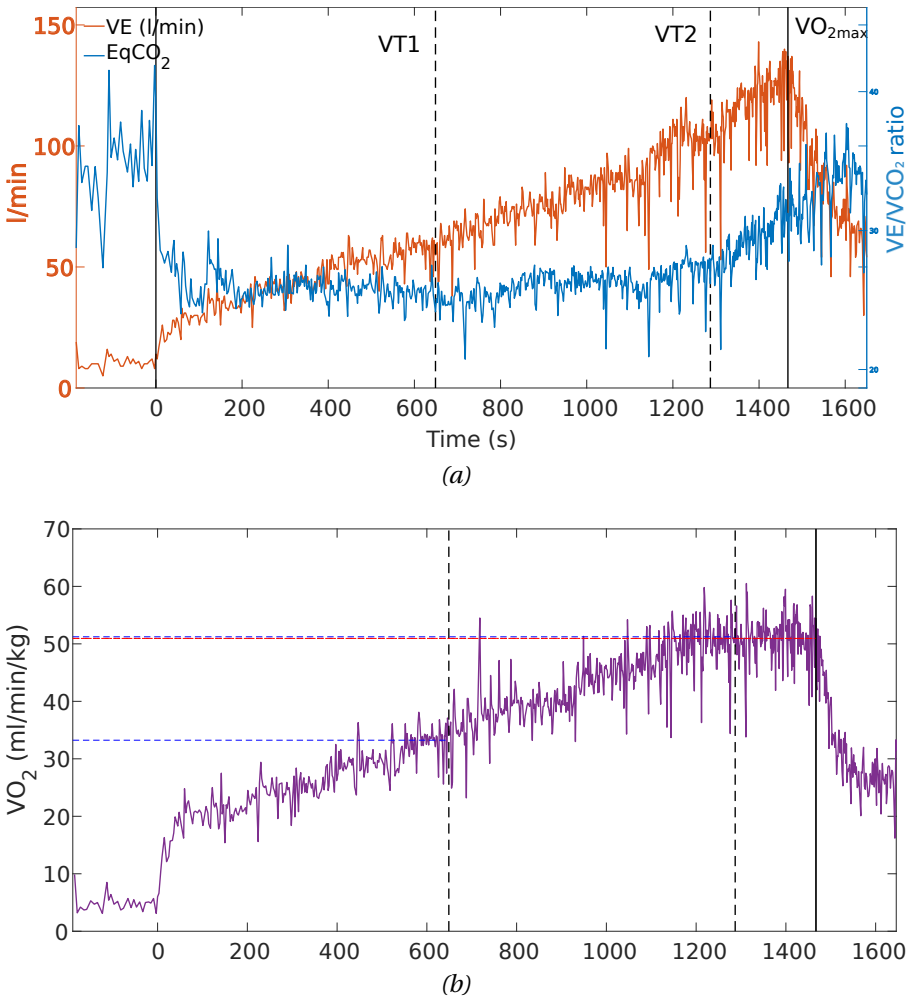


Figure 4.2 – Ventilatory thresholds estimation and agreement on the VE, VE/VCO_2 (a) and VO_2 (b) measurements from a gas analysis during incremental physical stress test on a cycle ergometer. These measurements are related to Subject 3 of the dataset analyzed in this chapter 4.4.1. After a resting period, the subject starts cycling till it reaches the hyperpnea (VT1) resulting in a non-linear increase in VE/VO_2 (orange in (a) and purple in (b)). Then, at VT2 the hyperpnea is not enough to eliminate the CO_2, which remains constant, leading to a sharp increase of VE/VCO_2 (blue). Finally, VO_{2max} is the maximum oxygen uptake at the moment of exhaustion.

protocol includes a single-lead ECG acquisition and analysis, from which specific HRV parameters can be extracted to estimate the so-called ventilatory thresholds (VT1, VT2 and VO_{2max}) with certain success [34–36, 146, 147]. These three parameters describe the cardiovascular and respiratory state during intense physical exercise. Fig. 4.2 shows an example of the gas analysis output and the ventilatory thresholds outputs on one of the subjects analyzed in this chapter. VT1 measures the hyperpnea (i.e., faster breathing) caused by the increased production of CO_2 for exercise intensities above the anaerobic threshold resulting in a non-linear increase in the ratio between ventilatory flow (VE) and VO_2. VT2 represents a phase where the hyperpnea is not enough to eliminate the CO_2, which remains constant, leading to a sharp increase of VE/VCO_2. Finally, VO_{2max} is the final stage where exhaustion is reached and, consequently, a maximum oxygen uptake and HR. However, the ventilatory thresholds extraction usually relies on an agreement between medical experts who evaluate the gas analysis and the HRV parameters and agree on where the thresholds are.

The HRV analysis uses the peak-to-peak (RR) time series of an ECG signal to extract time and frequency domain features, which are a measure of the autonomous nervous system. Within this, the respiratory trend is represented by the low frequency range within an ECG and its RR time series. Once the HRV features needed are extracted, they can be used for a direct estimation of VT1, VT2 and VO_{2max} [35]. The current methods to estimate these thresholds from HRV features are performed in post-processing with the help of medical experts and, usually, the RR time series is often interpolated and corrected. To ensure the correct comparison between ventilatory measurements and the RR time series, the R peak detection needs to be accurate and robust. Moreover, in future works (c.f. Section5.2 and Appendix A), the ventilatory threshold detection based on HRV parameters could be performed in real-time on wearable sensors. In this case, the R peak detection needs to be energy-efficient and adapt at run time to the sudden changes that affect the ECG during intense physical exercise [137].

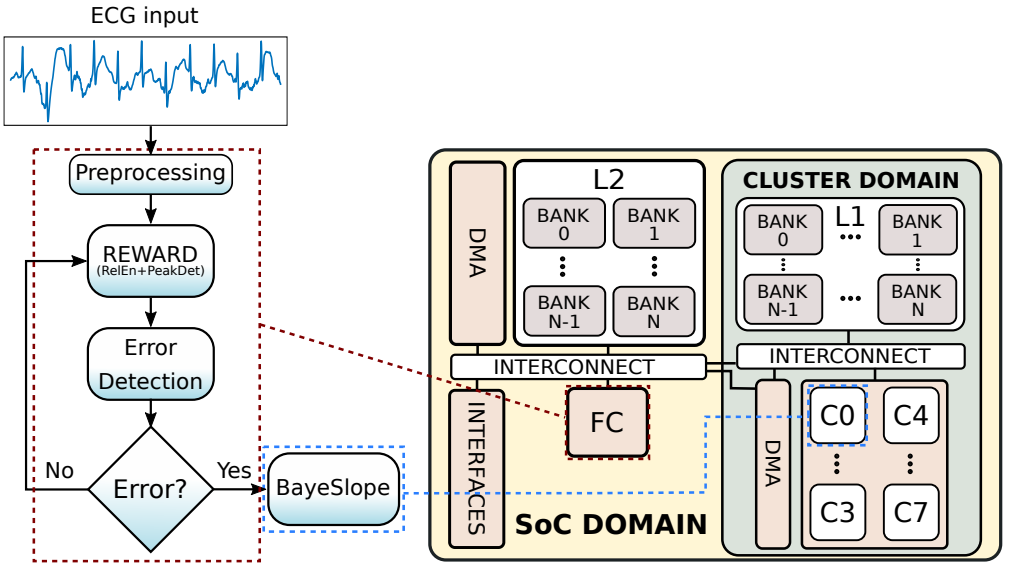


Figure 4.3 – On the left, data-flow diagram of the adaptive R peak detection algorithm with a raw ECG input. REWARD refers to the low complexity R peak detection presented in [41] and described in Section 2.2. BayeSlope is a new slope-based R peak detection algorithm presented in this Chapter. On the right, the PULP-based [45] architecture used for the analysis. Preprocessing, REWARD, and error detection run on the SoC domain in the fabric controller (FC), while BayeSlope runs on the cluster domain, in one core of the cluster (CL) of eight cores.

4.3 Adaptive R Peak Detection in Modern Wearable Sensors

One of the main problems in the context of edge computing in WSNs is minimizing energy consumption while maximizing output accuracy. In this section, I propose a method to detect R peaks from a single-lead ECG that optimizes the energy-accuracy trade-off with a two-level adaptive method. Fig. 4.3 shows the data-flow diagram of the full process and the architecture where the algorithm is implemented [45]. The lower level of adaptivity consists in the two different R peak detection algorithms, namely REWARD and the newly proposed algorithm, BayeSlope. REWARD was presented in [41]

and described in Chapter 2, and it uses hysteresis thresholds that are adapted to each window of 1.75 s. However, this window resolution is too small to capture peak-to-peak sudden changes. For this reason, I introduce BayeSlope that uses peak normalization by logistic function and a Bayesian filter to enhance and compute the expected position of the next R peak, and the k-means clustering to compute two centroids, one of which represents the R peak. The higher level of adaptivity consists in the data-flow shown in Fig. 4.3, where the output of REWARD is fed to an error detection method that checks when REWARD fails to properly detect R peaks and, in this case, triggers the more accurate BayeSlope. Moreover, BayeSlope is a more complex algorithm, hence, it benefits from execution on the cluster of cores in the platform, which includes eight cores with higher IPC and floating point units. Secondly, the main core, which is a simpler and faster core, can handle better running the less complex REWARD. In the next sections, I describe first the different blocks shown in Fig. 4.3 and then the higher level design within the heterogeneous platform used.

4.3.1 Preprocessing, REWARD and Error Detection

A standard R peak detection algorithm requires several steps of preprocessing of the ECG input signal. In this case, the input is a single-lead ECG where a morphological filtering (MF) is applied to remove the baseline and high frequency noise [71]. Then, the signal is enhanced by applying the Relative-Energy (Rel-En) method, which amplifies the most dominant peaks [41]. This preprocessing method is part of the REWARD algorithm presented in [41] and described in Chapter 2. The second part of the algorithm searches for the R peak in a window of 1.75 s using hysteresis thresholds (c.f., Fig. 2.4) based on the ECG morphology within the window. However, during intense physical exercise, the interval between two R peaks (i.e., RR interval) decreases significantly and sudden changes in amplitude occur. Therefore, within a window of analysis, many peaks can be missed, as shown in Fig. 4.4. Moreover, towards exhaustion during an incremental exercise stress test, the T wave—the wave after the R peak that represents the repolarization of the heart ventricles—can be significantly more dominant than the R peak itself while the P wave—the wave before the R peak that represents the depolarization of the atria—disappears, decreasing the RT interval. In these conditions,

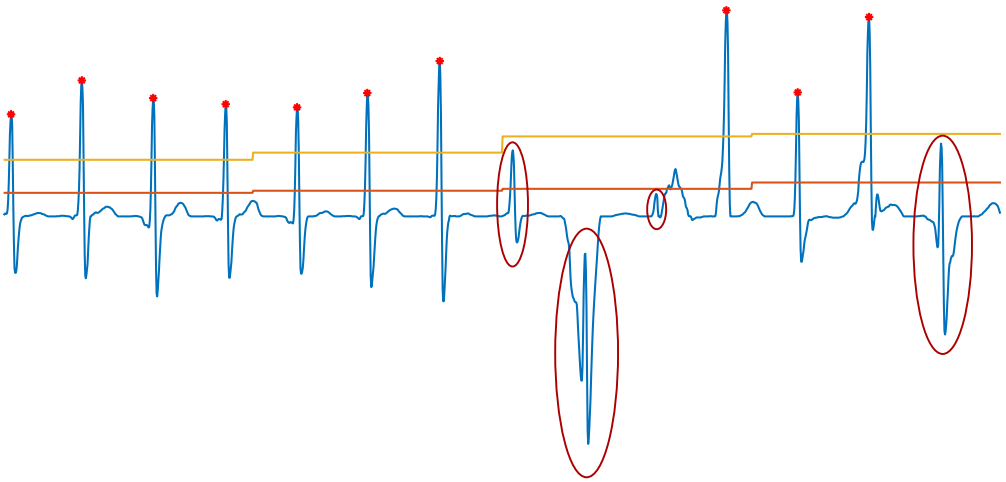


Figure 4.4 – Missed peaks in REWARD R peak detection using hysteresis thresholds (in orange and yellow) based on the ECG window morphology. The excerpt was extracted from Subject 3 of the dataset used (c.f. Section 4.4.1).

REWARD fails in detecting very small peaks as the hysteresis thresholds are skewed by the higher amplitude variability of the peaks within the window. However, it performs extremely well if these events do not occur as demonstrated in [41].

For this reason, I propose a method to identify errors in the R peak detection within a window of 1.75 s analyzing the distribution of the ratio $\frac{RR(n)}{RR(n-1)}$, where $n = 0, 1, 2...$, of all the data acquired. The distribution is computed offline using BayeSlope, since it is the most accurate (c.f. Section 4.5). However, to avoid data snooping, for each subject, the RR ratio distribution is computed with a leave-one-out (LOO) strategy, in which the analyzed subject is not included in the distribution. The RR ratio can capture sudden changes with a three-peak resolution, such as missing peaks, additional wrong peaks (e.g., T wave), and highly noisy signal excerpts.

First, the method computes offline the RR intervals and the corresponding RR ratio sequence used for the distribution from all the subjects, but the one analyzed. Then, for each subject, if at least one value of RR ratio computed within each window falls in the tails of the distribution (below the 0.5 or above

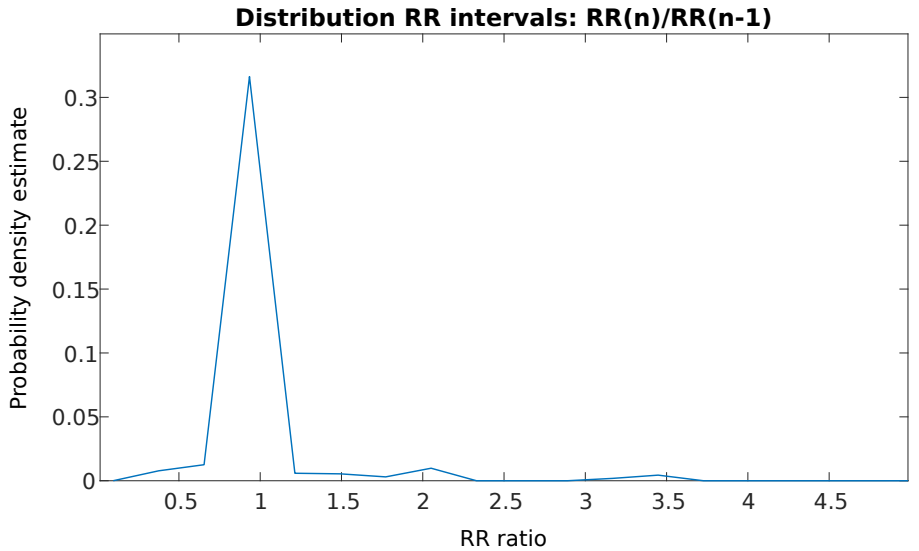


Figure 4.5 – RR ratio distribution for the full dataset acquired for this analysis

the 99.5 percentiles of the RR ratio distribution, respectively), the algorithm detects an error. This is performed in the online phase of the error detection applied to the output R peaks of REWARD. Fig. 4.5 shows the distribution considering all the subjects analyzed in this chapter. I report the overall distribution for convenience, although it is not the one used in the online error detection, as there is one specific distribution for each subject following the LOO strategy. The right tail is longer than reported on the figure as it is redundant considering that the percentile thresholds with the LOO strategy are:

$$P_{0.5} = 0.58 \pm 0.03; \qquad P_{99.5} = 1.66 \pm 0.06; \tag{4.1}$$

Therefore, if we consider the ECG example in Fig. 4.4, the result of the error detection are shown in Fig. 4.6, where the values of the RR ratio over the excerpt are reported. Considering the percentile thresholds $P_{0.5} = 0.64$ and $P_{99.5} = 1.47$ for the analyzed subject, the method can detect an error where REWARD fails. The last peak in Fig. 4.4 where there is an error will be detected in the next window.

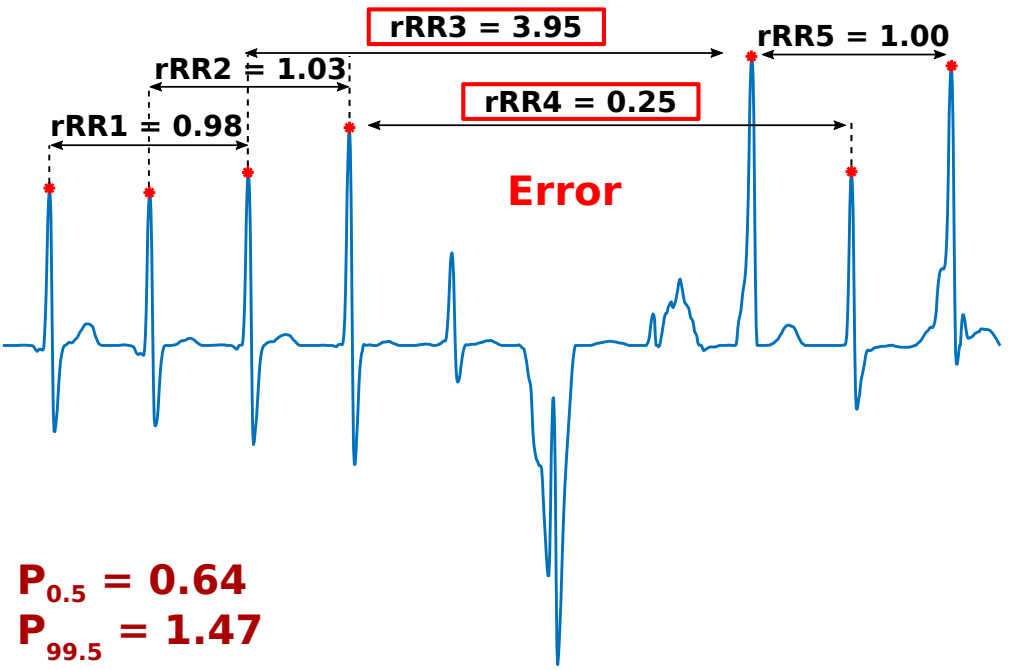


Figure 4.6 – Result of error detection on example ECG extracted from Subject 3 of the dataset used (c.f. Section 4.4.1). The values of the RR ratio are computed on a resolution of three R peaks. Considering the percentile thresholds for the analyzed subject (bottom left), the method can detect an error where REWARD fails (in the red boxes).

4.3.2 BAYESLOPE: Adaptive Slope-Based R Peak Detection

Once an error is detected, a more accurate adaptive R peak detection, BayeSlope, is triggered. This newly proposed method applies a non-linear normalization of the signal and a Bayesian filter to enhance high slope areas, which are assumed to belong to the QRS complex, and correctly detect the next peak in the expected position, which is based on the current HR. To distinguish low and high slope areas, the approach relies on a clustering method based on K-means.

Algorithm 5 describes the main steps of BayeSlope. The method takes as input the Rel-En signal window, s, and it outputs the vector of R peaks detected. The algorithm is derivative-based and considers two clusters that represent the high and low slope areas of the signal. The two centroids are initialized beforehand, as shown in Line 3, with $hcentr$ as the 99 percentile of the derivative of s and $lcentr = 1$. When a new sample is assigned to a cluster it is labeled as 1, if belonging to $hcentr$ cluster, or 0, if belonging to $lcentr$ cluster. Two windows of 1.75 s are used for the $hcentr$ initialization to account for enough peaks even at rest and avoid errors due to signal noise. Then, the algorithm initializes all the other parameters needed, constant and varying, in Lines 4–5. The values for these parameters were chosen based on physiological information and empirical tests.

The main process starts by considering the derivative of s and computing its absolute value, x, in Lines 7–8. The reason for applying this initial transformation is to enhance the maximum and minimum slopes of the original signal s, since the R peak is assumed to be located within the maximum upward and downward deflections within an ECG signal. Next, the method computes the Bayesian filter (Line 9), which is a Gaussian centered on the expected peak, mu, with standard deviation sd, two parameters computed based on the last five peaks. Then, in Line 10, the algorithm computes the generalized logistic function [148] with input x and its parameters computed based on the last $hcentr$ and $lcentr$. In fact, the sigmoid varies between 0 and the value of the higher k-means centroid, $hcentr$. The sigmoid and the Bayesian filter are used to normalize the peak or, specifically, to increase the amplitude of expected small peaks, as shown in Line 11 and Fig. 4.7. If the

Algorithm 5 BayeSlope R peak detection

Input: windows of RelEn signal, s
Output: R peaks, r
Initialize centroids: $hcentr$ = percentile(diff(s),99) and $lcentr = 1$
min_rr_dist = 240 ms; max_qrs_dur = 140 ms; ▷ Constant variables
Initialize: mu = 75 bpm; sd = 100 ms; $zeroctr = 0$; $qrs\_init = 0$; $label = 0$; in_qrs = false;
for $i = 2, ...$length(s) **do**
    $s2[i] = s[i] - s[i-1]$); ▷ Derivative approximation
    x = abs($s2[i]$);
    $bf[i]$ = gaussian($i - last\_peak, mu, sd$); ▷ Bayesian filter
    $bt[i]$ = genlogfun($x, param\_logfun$); ▷ Sigmoid normalization
    $st[i]$ = max($x, bt[i] * bf[i]$); ▷ Normalize signal
    Update $hcentr$ and $lcentr$ applying k-means clustering
    if in_qrs **then** ▷ Peak search
        if $label = 0$ **then**
            $zeroctr+ = 1$;
        else
            $zeroctr = 0$;
        end if
        if $zeroctr = 0$ OR $i - qrs\_init > max\_qrs\_dur$ **then**
            max_min_slope = argmaxmin($st * sign(s2)$);
            Search for new_peak within max_min_slope
            $r[i] = new\_peak$;
        end if
    else
        if $label = 1$ AND $i > last\_peak + min\_rr\_dist$ **then**
            in_qrs = true;
            $qrs\_init = 1$;
        end if
    end if
end for

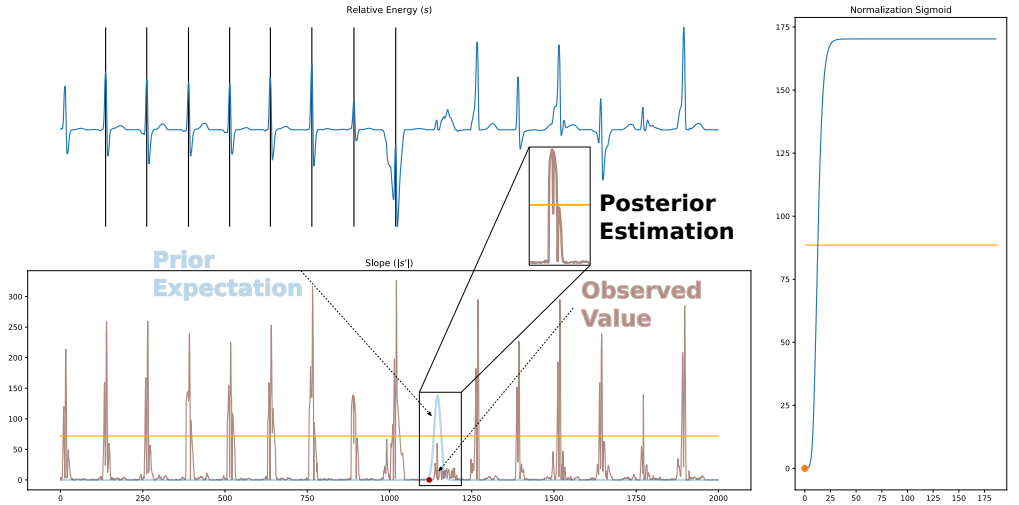


Figure 4.7 – Peak normalization in the expected location through bayesian filter (i.e., prior expectation) and generalized logistic function. Once the normalization is applied, the observed value is transformed into the posterior estimation, shown in the rectangle

analyzed sample does not reach the computed threshold, the function does not increase its value. When the input is approximately double the value of the lowest centroid, st (in Algorithm 5) reaches the threshold between the $lcentr$ and $hcentr$. In Fig. 4.7, the expected location of the peak (i.e., the prior expectation) is depicted with the Gaussian centered on it. In this case, the original peak (i.e., observed value) in x ($|s'|$) is small, and it will be enhanced by the Gaussian multiplied by the sigmoid function, st. This occurs at the values where st exceeds the threshold (in orange), with the result shown in the posterior estimation rectangle.

Once the signal is normalized, the algorithm starts a peak search within a QRS complex—the ECG main wave—in Lines 13–29. The distance between QRS complexes must be more than min_rr_dist, according to standard physiological characteristics and the sample that starts the QRS complex (qrs_init in Line 25) must belong to the cluster represented by $hcentr$ (i.e., $label = 1$). Within the QRS complex, the algorithm waits till it reaches its maximum duration according to physiology (max_qrs_dur) or for enough

samples ($zeroctr$ = 30) labeled 0 that represent the end of the QRS complex (Lines 14–19). Once within this interval (Lines 20–22), the algorithm computes the maximum and minimum of the function $st * sign(s2)$ representing the maximum upslope and downslope of the original signal. The sign function is used in case these values fall in the Q, S or T wave, which are not distinguished if only st is used, as it is positive by definition. Finally, the new_peak is found and stored in the vector r.

4.3.3 Adaptive Design in Modern Heterogeneous Platforms

As shown in Fig. 4.3, the algorithm modules run in different cores of the architecture depending on the complexity of the module. The architecture used in this work is based on the open-source PULP platform [45], and specifically on one of its evolutions, Mr.Wolf [49]. The PULP structure consists of a main streamlined processor, the fabric controller (FC), and an 8-core parallel compute cluster (CL). Moreover, PULP includes a direct memory access (DMA) that can transfer data to a multi-banked 512 KiB L2 memory during acquisition time or from L2 to a shared multi-banked 64 KiB L1 memory, which has a single-cycle latency in the cluster side. Both FC and CL are power-gated while the DMA fills the required L2 memory bank during sample acquisition. The FC is clock-gated when the CL is active, and each of the cores in the CL can be independently clock-gated to reduce dynamic power. Mr.Wolf includes a core for the FC (Zero-riscy [125]) that is simpler than the RI5CY cores of the CL [126] and runs at a higher frequency (170 MHz for FC and 110 MHz for the CL) but has a lower IPC. Mr.Wolf was designed to handle high computational load with a deep sleep mode not optimized for long idle periods. However, different PULP implementations do optimize deep sleep, consuming 3.6 µW when the platform is power-gated[1]. Therefore, the work in this chapter considers the Mr.Wolf architecture with a more optimized deep sleep mode based on other PULP implementations [48], as done in the work presented in Section 3.4.

Considering this design, the modules of preprocessing, REWARD (which includes Rel-En and R peak detection via hysteresis thresholds), and error

[1] As reported for GAP-8 [48], which is an industrial version of PULP with SoA deep sleep optimizations not yet included in its academic counterpart.

detection run in the Zero-riscy core (FC). The MF for a single-lead ECG was designed for single-core use, and it can only benefit from a parallelization per lead in the CL (c.f. Chapter 3), which is not possible in this case since the ECG acquired is single-lead. REWARD is a very lightweight integer-based algorithm, as demonstrated in [41]. In a preliminary analysis, considering the dataset (c.f. Section 4.4.1), I performed one test where the R peak detection step of REWARD was running on one core of CL and a second test where it was running on FC. The algorithm run on CL is 1.23 × slower (in terms of execution time) and consumes 1.35 × more energy. Therefore, REWARD benefits from running on the FC, which is a simpler and faster core. On the contrary, BayeSlope is a more complex floating point-based algorithm that benefits from running on a more capable core. In fact, since the FC does not have a floating-point unit, running BayeSlope on it requires a conversion to fixed-point representation of the complex operations in the algorithm, such as the Gaussian and the generalized logistic function. For this reason, I performed a test to convert BayeSlope, where the complex operations had to be divided into smaller ones. This can result in a slower execution time. Moreover, the clustering step requires an incremental variable that, in a fixed-point representation of 16-bit integer part and 16-bit decimal part, quickly reaches the maximum range representable (i.e., approximately 15 s of signal processing). In contrast, this does not occur in the floating-point representation as the maximum range is reached after a large number of hours of signal processing. Therefore, BayeSlope runs on one core of the CL (RI5CY), which has a floating point unit and higher IPC. BayeSlope has not yet been parallelized because of two reasons. First, the main goal of this chapter is to prove that the proposed adaptive design is able to assign heterogeneous resources based on the complexity of the algorithms, which can be done by only using two different cores. Second, the parallelization process requires several optimization steps to maintain the accuracy of the algorithm that take time to develop. In fact, I proposed this as a possible short-term future work (c.f. Section 5.2), which can use the parallelization techniques presented in Section 3.4.1.

After the signal filtering and REWARD running on the FC, the error detection (also running on the FC) checks the accuracy of the R peaks output. If an

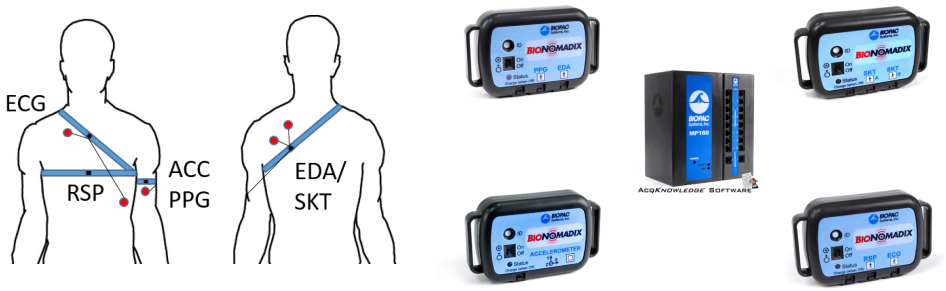


Figure 4.8 – Sketch of the BIOPAC [149] sensors positioning (on the left) during the experiment and the sensors themselves (on the right)

error is detected, the DMA transfers the necessary buffer of data from L2 to L1 ready for the core in the CL, while the FC is clock-gated. Since BayeSlope needs an initialization of the R peaks of two windows of 1.75 s, the previous window error needs to be checked. If the error in the previous window is 0, then the DMA transfers two windows, otherwise it transfers only one. This is an optimization applied in case REWARD fails more frequently and to avoid recomputing twice the same window. BayeSlope runs on the CL while FC is clock-gated. The final output is the combination of correct R peaks from REWARD and BayeSlope.

4.4 Experimental Setup

4.4.1 Database Acquisition Protocol

The database was acquired considering 22 subjects performing an incremental exercise stress test on a cycle ergometer for an average of 30 minutes each till exhaustion. The power of the cycle ergometer was increased every 3 min by 30 W, after initial 3 min of rest. A single-lead ECG sampled at 500 Hz was acquired using the BIOPAC system [149], together with other biosignals and oxygen uptake measurements that were not used for this work. Fig. 4.8 shows a sketch of the biosignals positioning and the equipment used. The ECG was downsampled to 250 Hz since REWARD was validated only for this frequency. Two of the 22 subjects were discarded because one did not complete the

protocol and for the second one the majority of the recording was corrupted. Therefore, the statistics and analysis are performed on 20 subjects. Next, five 20-second excerpts were extracted from the full ECG of each subject to be manually annotated by medical experts. These excerpts were chosen based on the different phases of the incremental stress test (i.e., before and after VT1, before and after VT2, and during the recovery after exhaustion). The segment at rest was ignored since REWARD performs very well in this condition, and there is no need to run BayeSlope. Only one out of 100 excerpts was not annotated. Therefore, the total number of excerpts considered for this analysis is $20 \cdot 5 - 1 = 99$. The input excerpts to the peak detection were extracted considering the 20 s given to the experts and going backward of 0.6 s + 0.95 s + 1.75 s, which represents, respectively, the initial delay of the MF, the initial delay of Rel-En, and one additional window of analysis for BayeSlope initialization, and going forward another 1.75 s to avoid missing the last peaks. Therefore, each excerpt is approximately 25 s long. The accuracy of the R peak detection is measured according to the standard tolerance of 150 ms as the time difference between the detected peak and the manually annotated one [80]. Moreover, I also report for each subject the mean and standard deviation of the time difference between the two. I compare the accuracy of the following three designs:

1. preprocessing (MF) and always running REWARD (Rel-En + peak detection);
2. preprocessing (MF + Rel-En) and always running BayeSlope;
3. adaptive design including preprocessing (MF), REWARD (Rel-En + peak detection), error detection and running BayeSlope only when REWARD fails.

4.4.2 Test Benches on Heterogeneous Platform

The three designs are mapped on the PULP platform to estimate their overall energy consumption and perform the energy-accuracy analysis. In all the test benches, the preprocessing always runs on FC. The first two test benches consists of 1) REWARD running on the FC with the CL power-gated, and 2) BayeSlope always running on the CL, which was first implemented in Python,

then translated to C and ported to PULP. The third test bench consists of the fully adaptive process, including the error detection, with REWARD running on the FC and BayeSlope running on CL when REWARD fails. Each of the test benches is applied to the 99 excerpts described in Section 4.4.1.

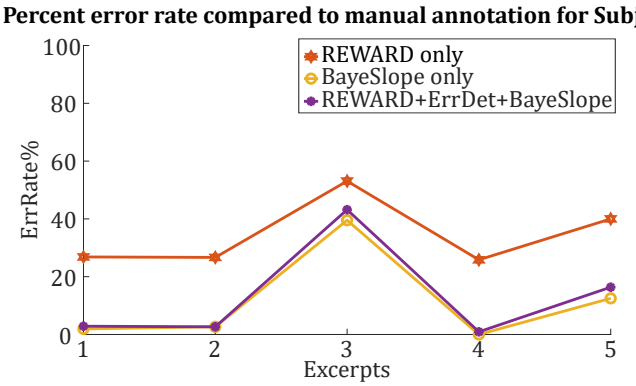
To measure the execution time of the three configurations, I used the open PULP platform [130]. PULP provides an SDK to run RTL simulations, using Modelsim, in order to obtain cycle-accurate profilings. To estimate the energy consumption I use the power numbers reported for a chip based on the PULP architecture implemented in TSMC 40 nm LP CMOS technology, Mr.Wolf [49], described in Section 4.3.3. I consider the lowest energy point of the platform, at 0.8 V. The platform requires 3.6 µW [48] when power-gated and 12.6 µW with full L2 retention. To implement better memory management of the activated banks (c.f., Section 3.4.2 and Section 3.5.2), I reduce the L2 size to 128 KiB, with a resolution of 16 KiB per memory bank, since the application does not need more memory. When the SoC is active, it consumes 0.98 mW with its main processor clock-gated, and 6.66 mW with it operating at 170 MHz. Once the CL is activated, it consumes 0.61 mW with all the cores clock-gated and 18.87 mW with the eight cores running at 110 MHz.

The three designs are compared first in terms of accuracy, then energy consumption of their mapping to the PULP platform and then in their energy-accuracy trade-off for all the subjects in the dataset and as a summary for worst, average and best cases.

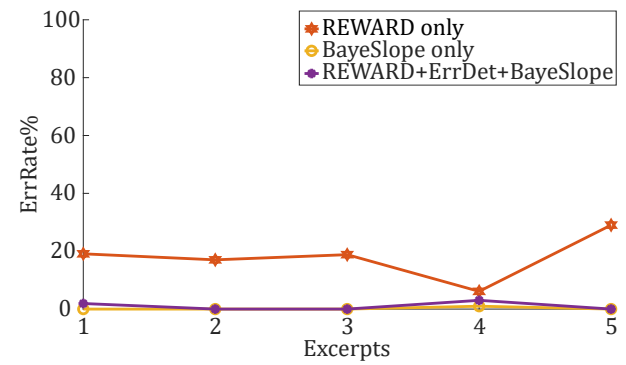
4.5 Experimental Results

4.5.1 Accuracy Analysis of Test Benches

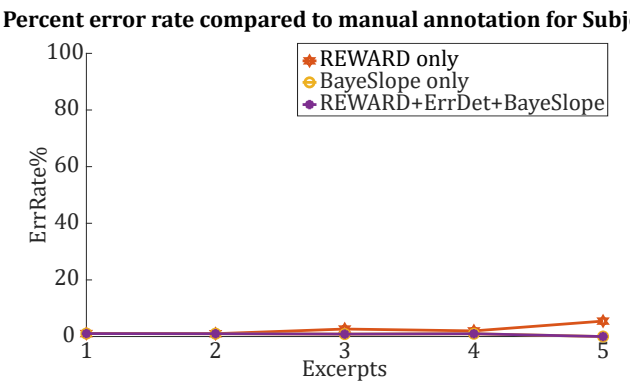
In Fig. 4.9, I report the percent of the error rate (ErrRate%) in the peak detection of the three designs, described in Section 4.4.1, and its evolution through the type of excerpts for three example subjects. These examples illustrate three cases within the worst, best, and average groups in terms of accuracy of the new algorithm, BayeSlope, and the fully adaptive design (REWARD+Error detection (ErrDet)+BayeSlope) compared to REWARD. ErrRate% is computed



(a) Worst case



(b) Average case



(c) Best case

Figure 4.9 – Percent error rate of the three designs described in Section 4.4.1 for three worst, average, and best case subjects along five increasing exercise intensities. Excerpts 1 and 2 represent the exercise before and after VT1, excerpts 3 and 4 before and after VT2, and excerpt 5 is the recovery right after exhaustion

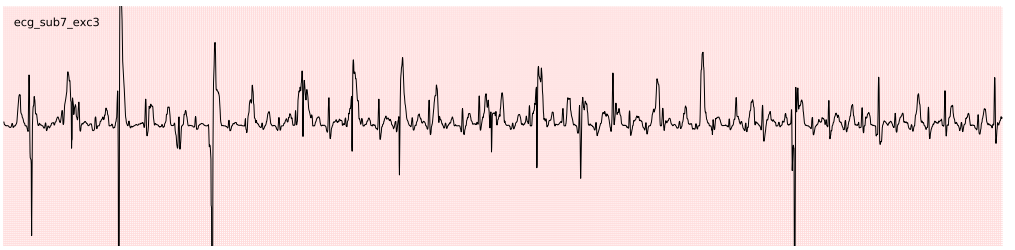


Figure 4.10 – ECG excerpt 3 (i.e., before VT2) for Subject 7. The amplitude of the peaks is highly variable due to the changes in the exercise intensities. The signal is shown on a standard ECG sheet containing small squares of 1 mm·1 mm corresponding to 40 ms (horizontal) and 0.1 mV (vertical) [60]. They also include big squares of 5 mm·5 mm and correspond to 200 ms·0.5 mV.

as (1-F_1)*100, where F_1 score is a measure of the peak detection performance defined as

$$F_1 = \frac{TP}{TP + \frac{1}{2} \cdot (FP + FN)} \tag{4.2}$$

TP is the set of the correctly detected peaks that match the manual annotations. FP represents all the misdetected peaks by the algorithm. FN is the set of all the peaks in the algorithm that do not match any manual annotation. The different excerpts shown in Fig. 4.9 represent increasing exercise intensities till the recovery after exhaustion (excerpt 5), as described in Section 4.4. In Fig. 4.9a, Subject 7 has one of the worst error rates for the new algorithm, and the reason is that excerpt 3 is quite noisy. The quality of the excerpt is shown in Fig. 4.10, where the amplitude of the ECG has a high variability due to changes caused by the exercise intensities near VT2 (excerpt 3 is before VT2). However, BayeSlope and its adaptive design, with an F_1 score at approximately 60.5 % and 56.8 %, respectively, gains within 13.5 % and 9.9 % in performance, compared to REWARD. In Fig. 4.9b, Subject 3 represents an average case where REWARD has a lower error rate compared to the worst case (Subject 7), though significant. In fact, the adaptive design performs significantly better, with an error rate up to 3 %, slightly worse than BayeSlope. In Fig. 4.9c, Subject 16 is one of the best cases where REWARD fails only

during more intense exercise, with an error rate up to 5.5 %, while BayeSlope has an error rate of only 1 %.

Considering the five exercise intensities, a relevant summary of the algorithms' performance is depicted in Table 4.1. Here, I report the F_1 score, sensitivity, and positive predictive value (PPV) of the three test benches for each of the five types of excerpt computed across the subjects, as well as the mean and standard deviation of the time difference between each test bench output and the manual annotations. BayeSlope is the most accurate of the designs over all the performance parameters. In fact, by adapting to all the changes in the ECG during intense physical exercise, it reaches an F_1 score up to 99.3 % with very low variability through the excerpts. On the contrary, the F_1 score and the sensitivity of REWARD during more intense exercise (before VT2 and right after exhaustion), where sudden changes in ECG occur, are significantly lower than the acceptable medical standard, compared to less intense exercise. However, combining both methods in an adaptive design is as accurate as BayeSlope (up to 1.7 % of difference in F_1 score).

Table 4.1 – F_1 score, PPV, sensitivity (%) for the three test benches and the five exercise intensities computed across the subjects

		Before VT1	After VT1	Before VT2	After VT2	Recovery	**Total**
F_1 (%)	REWARD (RW)	92.1	90.9	78.7	92.5	80.2	86.7
	BayeSlope (BS)	**99.0**	**99.1**	**97.9**	**99.3**	**98.8**	**98.8**
	RW + ErrDet + BS	98.9	99.0	96.2	98.5	97.1	97.9
PPV (%)	REWARD (RW)	98.2	98.2	97.1	98.1	96.3	97.6
	BayeSlope (BS)	**98.6**	**98.6**	**98.9**	**98.6**	**98.6**	**98.7**
	RW + ErrDet + BS	98.3	98.4	97.3	97.5	96.2	97.5
Sensitivity (%)	REWARD (RW)	86.8	84.5	66.1	87.4	68.7	78.0
	BayeSlope (BS)	99.3	99.5	**96.9**	**100.0**	**98.9**	**98.9**
	RW + ErrDet + BS	**99.4**	**99.6**	95.2	99.6	98.0	98.3
Time (ms) from manual annotation	REWARD (RW)	0.6±8.4	1.1±11.2	10.4±35.7	9.2±34.8	20.6±48.6	8.1±32.0
	BayeSlope (BS)	**0.5±6.5**	**0.3±4.6**	4.9±24.0	**1.0±10.3**	**6.8±29.1**	2.9±18.6
	RW + ErrDet + BS	**0.5±6.5**	**0.3±4.6**	**4.3±22.3**	1.1±11.3	7.3±30.1	**2.8±18.5**

Rarely, the adaptive design could perform better (less than 1 % difference in score) as it is shown in the sensitivity values. This is due to the initialization process of BayeSlope, which requires the signal to be stable as it does not use any prior information within this initial stage. Therefore, it happens rarely that the signal is more stable later in the excerpt where BayeSlope is triggered and will be initialized compared to the initialization at the beginning of the excerpt (when always running BayeSlope). This can also cause a delay in the adaptation and very few peaks missed and result instead in a slightly worse accuracy. Another reason for a lower performance in the adaptive design compared to always running BayeSlope, specifically for more intense exercise (before and after VT2, and recovery) as shown in Table 4.1, is due to an issue in the error detection. In fact, the RR ratio distribution used to compute the tail thresholds is performed on the full dataset and accounting for different exercise intensities. Within more intense exercises, as the RR intervals get smaller, it can happen that even if REWARD misses one peak, the RR ratio is still within the distribution. This is shown in Fig. 4.11, where the RR ratio computed on the small peaks not detected by REWARD is close to the $P_{99.5}$ of the distribution but not enough to trigger an error. This results in a lower accuracy for the adaptive design. One way to fix this problem is to compute different distributions for different exercise intensities. In the case of this dataset, it could be five distributions or two groups of low and high intensities. Another way is to adapt the distribution online by detecting the intensity type with a machine learning algorithm and choose the correct tail thresholds. Finally, as modern heterogeneous platforms allow, the distribution can be computed directly on the signal acquired through a small training process on BayeSlope and then adapting the tail thresholds.

In conclusion, the accuracy results show that always running BayeSlope is the most accurate and robust of the three designs. At the same time, REWARD's performance highly varies with the intensity of the exercise. However, it is approximately 104 × more complex than the R peak detection step of REWARD. Therefore, I proposed the adaptive design that combines both algorithms and has a similar accuracy compared to BayeSlope. In the next section, I will show the advantages in terms of energy consumption of the adaptive design mapping on the PULP platform.

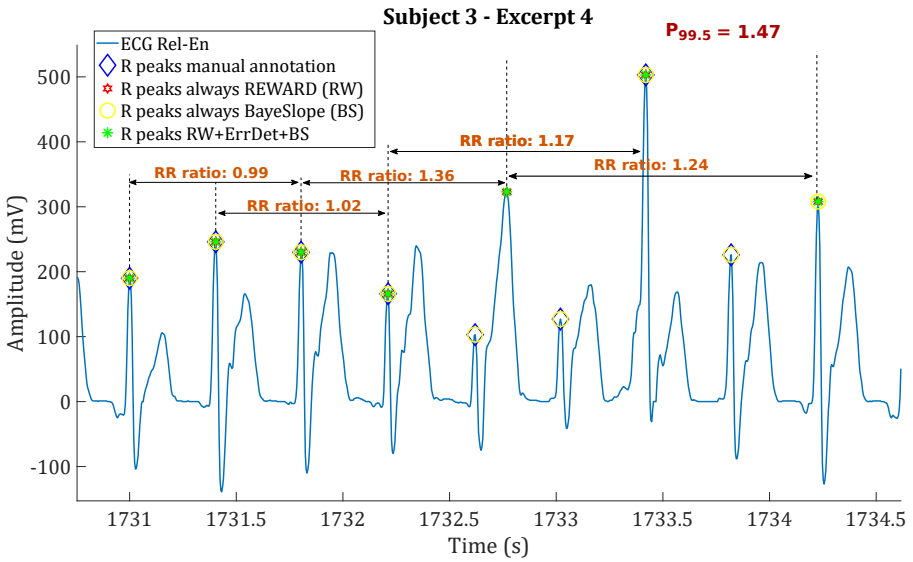
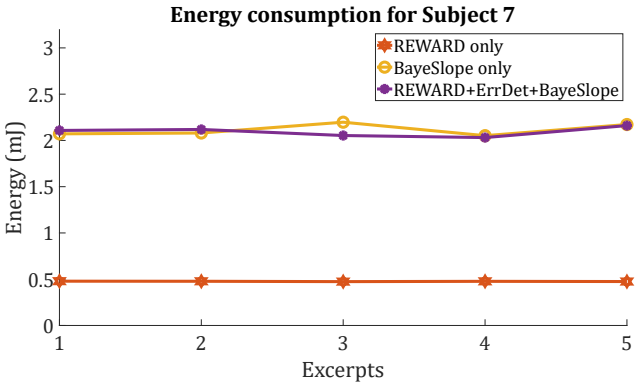


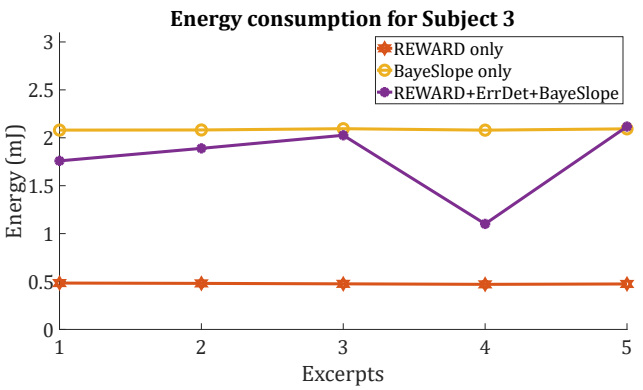
Figure 4.11 – ECG excerpt 4 (i.e., after VT2) for Subject 3 with the R peaks from the three designs. The RR ratio where the small R peaks are not detected by REWARD is close to $P_{99.5}$ but not enough to trigger an error

4.5.2 Energy Consumption of Test Benches in PULP

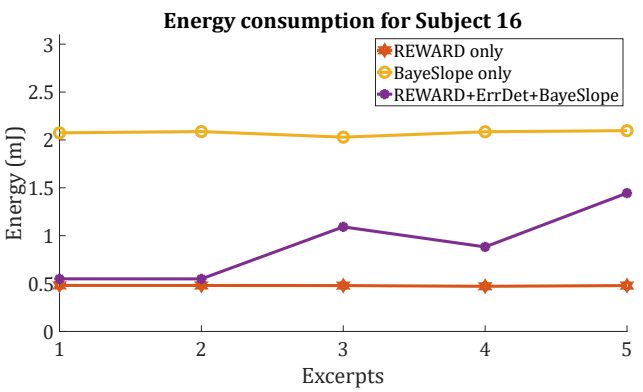
Figure 4.12 shows the three subjects described in Section 4.5.1. In Subject 7 (Fig. 4.12a), the worst case scenario, the fully adaptive design consumes the same amount of energy in almost all the windows. In excerpt 3, the adaptive design achieves 6.5 % of energy savings compared to always running BayeSlope, with a 3.7 % difference in F_1 score. However, the overall accuracy is far from the required medical standard. In Subject 3 (Fig. 4.12b), for all the exercise intensities except the last one, the fully adaptive design has energy savings up to 48 % compared to the BayeSlope with a loss in accuracy of only up to 2 % (c.f. Fig. 4.9b). For excerpt 3, even if the energy savings are one of the lowest at approximately 3.3 %, the fully adaptive design is as accurate as BayeSlope and 18.8 % more accurate compared to REWARD. Therefore, on average cases such as Subject 3, in most exercise intensities, choosing the fully adaptive design can improve the energy-accuracy trade-off. Subject 16, representing one of the best case scenario in Fig. 4.12c, highlights the adaptivity of the full design and its error detection through the excerpts, starting



(a) Worst case



(b) Average case



(c) Best case

Figure 4.12 – Energy consumption of the three test benches described in Section 4.4.2 for three worst, average, and best case subjects along five increasing exercise intensities. Excerpts 1 and 2 occur before and after VT1, excerpts 3 and 4 before and after VT2, and excerpt 5 is the recovery right after exhaustion

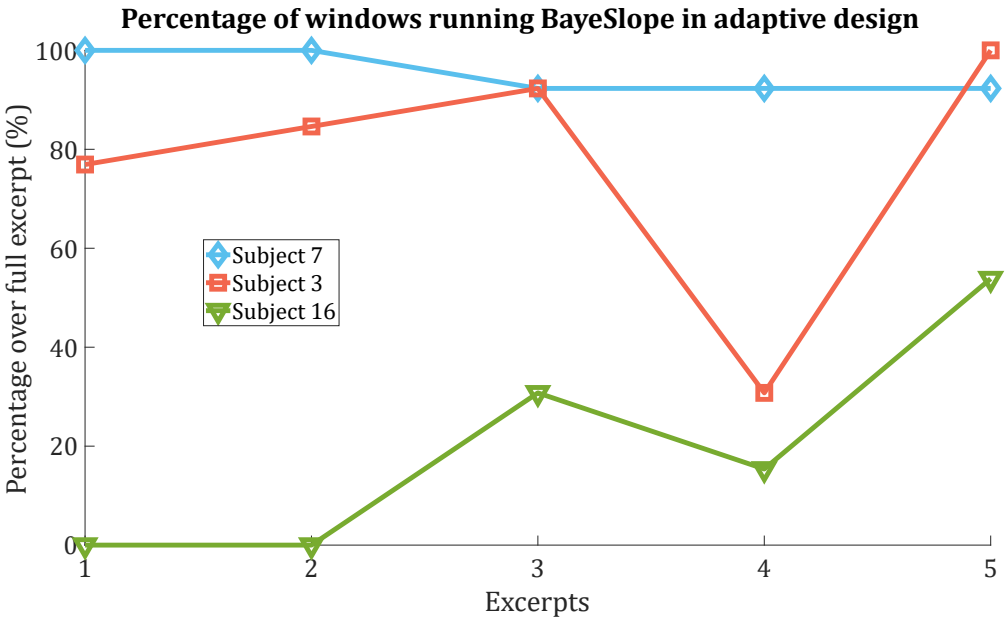


Figure 4.13 – Percentage of windows over the full excerpt where BayeSlope is triggered during the adaptive design for worst, average, and best case scenarios. Comparing these trends with the ones shown in Fig. 4.12, it is evident that the adaptive design reduces energy consumption by reducing the number of times BayeSlope runs on the CL

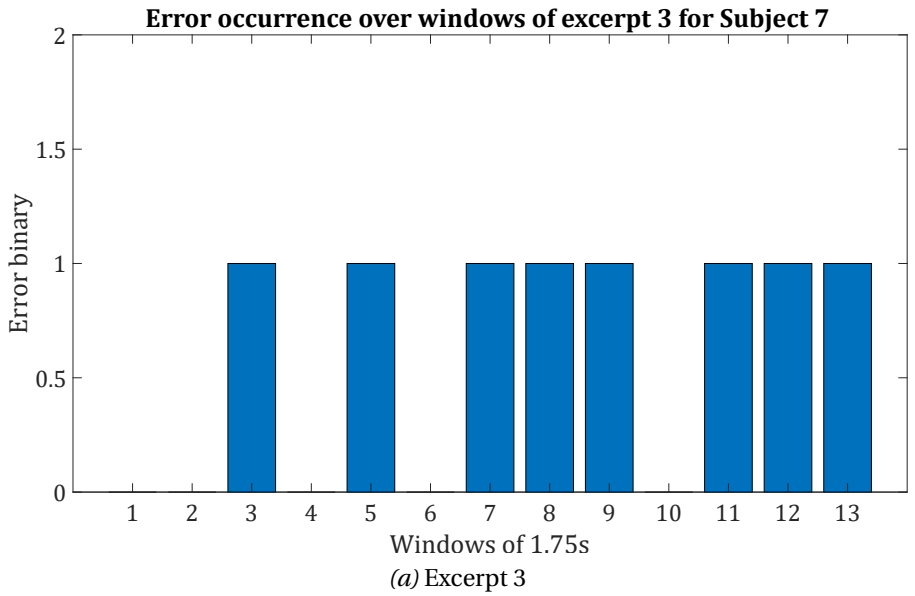
with a minimum energy consumption, since only REWARD is running, and maximum attainable accuracy. Then, when the exercise intensity increases, REWARD fails more frequently, and BayeSlope takes over the R peak detection. The fully adaptive design maintains a high level of accuracy (approximately 99 %) while limiting the energy consumption compared to running BayeSlope for the full excerpt, with energy savings from 31.8 % up to 58.6 %. Fig. 4.13 shows how many times BayeSlope runs in the adaptive design in terms of percentage of windows over the full excerpt for three cases analyzed. For counting the windows where an error occurs that trigger BayeSlope, the previous window also counts as triggered since BayeSlope needs an additional window for the initialization process (c.f. Section 4.3.3). From the figure, it is evident that the trend is similar to the energy reduction compared to always running BayeSlope shown in Fig. 4.12. For Subject 7, starting from excerpt 3,

Table 4.2 – Energy consumption in mJ for the three test benches and the five exercise intensities computed across the subjects

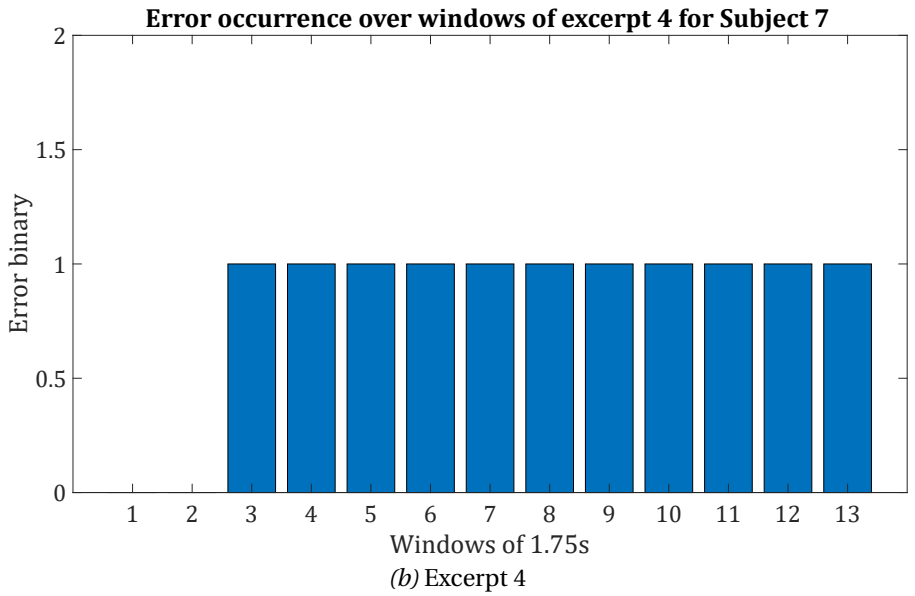
		REWARD (RW)	BayeSlope (BS)	RW + ErrDet + BS
Energy(mJ)	Before VT1	0.479±0.004	2.078±0.016	1.348±0.573
	After VT1	0.479±0.003	2.070±0.032	1.469±0.556
	Before VT2	0.476±0.004	2.071±0.037	1.84±0.299
	After VT2	0.477±0.002	2.080±0.020	1.275±0.562
	Recovery	0.476±0.003	2.075±0.032	1.823±0.412
	Total	**0.477±0.004**	**2.075±0.028**	**1.553±0.536**

the trend is slightly different than the one in Fig. 4.12a. In fact, excerpt 3 has a 6.5 % reduction in energy of the adaptive design compared to BayeSlope, while it is less than 1 % in excerpt 4. To explain this result, I drew the error detection pattern in excerpt 3 and 4 for Subject 7 shown in Fig. 4.14, where 1 indicates that an error occurred. In excerpt 3, the error does not happen on every window but alternating, although BayeSlope runs for two windows whenever the error in the previous window is 0. On the contrary, in excerpt 4 the error is triggered in consecutive windows and BayeSlope runs as well on every window. When BayeSloperuns on every window an overlapping occurs to avoid missing peaks at the border between two windows. This does not occur in the adaptive design (excerpt 3) when the algorithm runs once on two windows, avoiding this overhead with a small advantage in energy consumption. The big differences in the three subjects show how the proposed design can adapt to the subject and different exercise intensities to reduce energy consumption instead of constantly falling in the worst case scenario. This personalized and adaptive reduction in energy consumption can lead to a longer battery lifetime for WSNs and better usability.

Table 4.2 shows a summary of the average energy consumption for the five exercise intensities. As we saw for the accuracy analysis, higher exercise intensities require to run BayeSlope more often in the adaptive design. How-



(a) Excerpt 3



(b) Excerpt 4

Figure 4.14 – Error occurrence in two excerpts of Subject 7 with the same percentage of BayeSlope triggers shown in Fig. 4.13. Excerpt 3 has an alternating pattern, while in excerpt 4 the error occurs in consecutive windows, which explains the slightly bigger energy reduction in excerpt 3 compared to expert 4

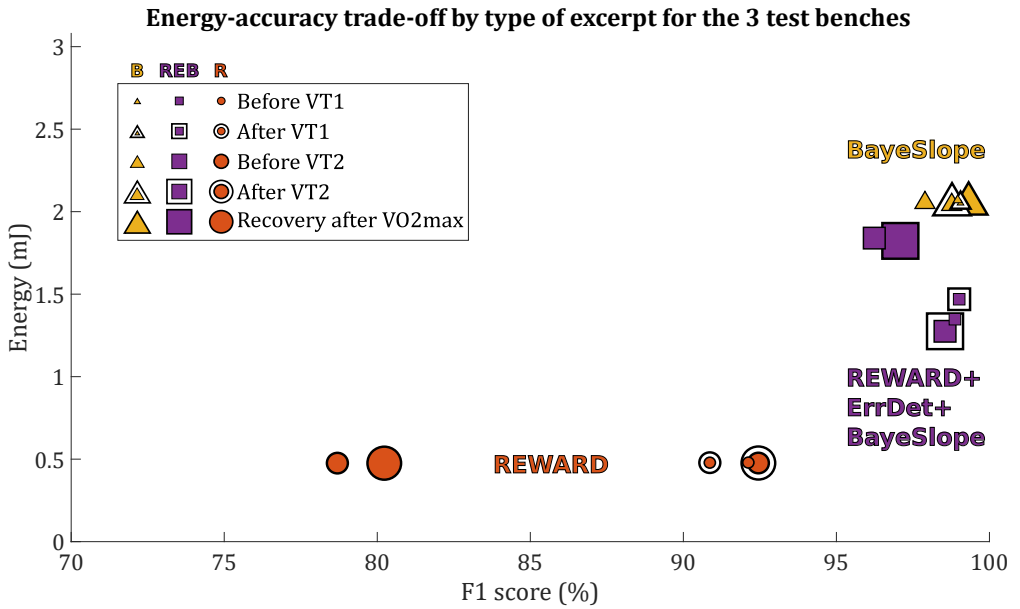


Figure 4.15 – Energy-accuracy analysis of the three test benches and the different exercise intensities

ever, the algorithm achieves significant energy savings compared to always running BayeSlope. In fact, during the fully adaptive design, the CL of cores (where only one is used in this application) is clock-gated if not used, and the power consumed accounts only for the FC and the CL leakage, which for this design is negligible. On average, for this application, the FC consumes only 20 % of the energy consumed by FC+1-coreCL within a window of analysis. Therefore, if the CL is active only for specific windows, it limits the total energy consumption over the 25-second excerpt. In fact, the adaptive design achieves energy savings up to 38.7 %, considering the average for the five exercise intensities, and up to 74.2 % for the overall dataset analyzed, compared to the always active CL running BayeSlope.

4.5.3 Energy-Accuracy Test Benches Comparison

Figure 4.15 shows the energy-accuracy comparison between the three test benches and an analysis on the different exercise intensities. I use once again F_1 score as a measure of algorithm detection accuracy. For the three excerpts before and after VT1, and after VT2, REWARD is fairly accurate, and

consuming the minimum energy for this application. These values of F_1 score are similar to the ones I showed in the paroxysmal atrial fibrillation (PAF) application in Chapter 2. As during exercise, PAF is characterized by sudden changes in the ECG morphology (e.g., ectopic beats, which are usually smaller in amplitude, rhythm irregularities and missing P waves). REWARD fails to adapt to this sudden change by design, although within the medical acceptability. However, performing the fully adaptive design (in purple) is always more advantageous in terms of accuracy, with a performance increase of up to 8.2 %. Moreover, it is comparable in F_1 score to BayeSlope although more energy-efficient, with energy savings up to 38.7 %.

However, when the exercise intensity, hence, the number of peaks within a window increases, the hysteresis thresholds of REWARD do not adapt to the smaller peaks within a window of analysis (1.75 s), as described in Section 4.3.1 and Fig. 4.4. In fact, before VT2 a non-linear increase in HRV parameters occurs [34], representing a significant increase in O_2 and CO_2 consumption and further increase in exercise intensity, which can explain the decreased accuracy of REWARD. The excerpt extracted during recovery after exhaustion (i.e., when reaching VO_{2max}) represents the highest intensity and, hence, disruption of the ECG morphology, specifically in the amplitude of the R peak and the RR intervals (HRV reaches its minimum). Therefore, it is the reason for a decreased performance in REWARD. On the contrary, the F_1 score of the fully adaptive design is only up to 1.7 % lower than BayeSlope, which is the most accurate. The energy savings for these two excerpts are lower than the other three, though still significant (up to 12.2 %).

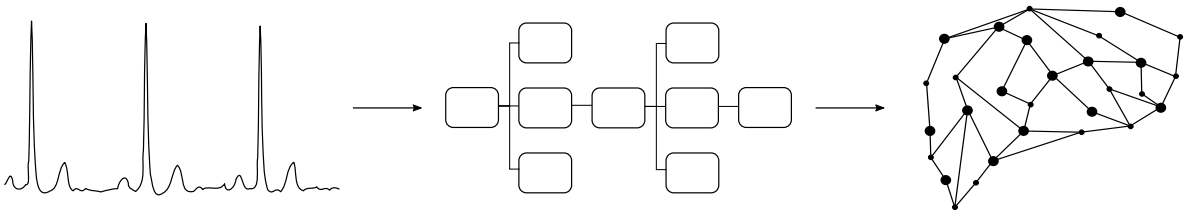
This analysis shows how the new BayeSlope algorithm is highly accurate and more robust than the lightweight REWARD when sudden changes in the ECG morphology occur. However, if we consider the design where BayeSlope is mapped on a PULP-based platform and running on CL (with the preprocessing modules running on FC), the device consumes on average 4.6 × more than the mapping of REWARD (and the preprocessing) in FC. In contrast, the adaptive design enhances the energy-accuracy trade-off, maximizing accuracy while limiting energy consumption on modern ULP platforms. This adaptive design is not limited to applications where intense physical exercise is involved but also but can also be applied to pathologies where the ECG

morphology changes, such as PAF [56] and other types of arrhythmia [132]. Moreover, if BayeSlope is parallelized in the 8-core CL (c.f. Section 5.2), more computing resources can be assigned to HRV analysis (c.f. Appendix A) and pathology detection for fully on-node processing to ensure low-rate transmission and data privacy.

4.6 Conclusion

In health and wellness monitoring, specifically cardiovascular, there exist pathologies and physical conditions where sudden changes in the measured biosignals occur. In particular, during intense physical exercise, sudden changes in the ECG heart beats amplitude and rhythm cause errors in some standard R peak detection algorithms, and therefore on any further analysis based on the HR. Moreover, more accurate algorithms often require a higher amount of computing resources leading to a need for more capable platforms with a flexible resource management.

In this chapter, I have proposed a new online design to detect R peaks in a single-lead ECG signal which adapts at run time to the changes in its morphology. Furthermore, this adaptive design exploits the core heterogeneity of modern ULP platforms, which can run efficiently more complex algorithms using different types of cores. The online adaptive design uses a standard lightweight algorithm, REWARD, and an error detection to measure the algorithm's accuracy. When REWARD fails, a more accurate though more complex algorithm, BayeSlope, is triggered and runs in a more efficient core. In the context of an incremental exercise stress test, the new online adaptive design achieves an F_1 score up to 99.0 % compared to 92.5 % when running only REWARD, across five different exercise intensities. By implementing the newly proposed method in the PULP architecture, it can reach energy savings up to 34.6 % compared to always running the more complex BayeSlope. Therefore, the newly proposed online adaptive design maximizes the accuracy while minimizing the energy consumption for an optimal energy-accuracy trade-off.



Conclusion and Future Work

5

Remote health and wellness monitoring through wearable technology has advanced from electrocardiogram (ECG) radio transmission to modern ultra-low power (ULP) smart sensors for pathology detection and real-time vital parameters assessment for noncommunicable diseases (NCDs) prevention. This trend towards increased complexity and broader functionality will face new challenges as population aging and incidence of NCDs is projected to grow at a faster pace. Therefore, there is a need for new approaches to improve the performance of wearable sensors in terms of maximizing the accuracy of their applications for better and personalized healthcare while minimizing the energy consumption for continuous use.

In this thesis, I have presented novel approaches for remote health and wellness monitoring, tailored to the subject and tackling the energy-accuracy trade-off problem from both algorithmic and platform perspectives.

5.1 Adaptivity is the Key

Throughout Chapters 2–4, the common factor to the proposed solutions for enhancing the energy-accuracy trade-off in modern wearable technologies is adaptivity. The thesis tackled this key factor from two main aspects.

First, in Chapter 2, I have presented a series of strategies that adapt to the subject, and exploit knowledge from multiple biosignals at run time, in tradi-

tional single-core platforms. Specifically, in the context of remote health and wellness monitoring, I proposed:

- A novel lightweight method, called REWARD. It uses adaptive thresholds on the ECG amplitude to detect enhanced QRS complexes.
- A highly accurate and ULP algorithm for heart rate (HR) estimation from a photoplethysmography (PPG) signal. This algorithm exploits the frequency-domain knowledge acquired from motion sensors to remove artifacts caused by intense physical exercise and past information to update the current HR value.
- A personalized real-time paroxysmal atrial fibrillation (PAF) prediction method that is trained with a model based on the specific characteristics of the patient and their condition, and, consequently, scales the energy consumption on a real-life ECG-based device, for better and personalized usability.

Second, in Chapter 3, I have exploited the capabilities of modern multi-core heterogeneous platforms through modular and personalized strategies. Specifically, I proposed software (SW) and hardware (HW) optimizations applied to the typical modules of biomedical applications, including a top-down approach of parallelization techniques to maximize the attainable speed-up, memory scaling and management at acquisition time, and HW acceleration of computationally intensive kernels. Then, in the context of the previously mentioned PAF prediction method, I focused on the impact of patient-specific assignment of platform resources (i.e., different number of cores based on the patient-specific training parameters), and memory scaling. This analysis unveiled the adaptivity and scalability of computing and memory resources in modern ULP wearable sensors.

By combining the findings from these two perspectives, in Chapter 4, I proposed a final adaptive design that takes into account run-time adaptivity based on accuracy performance and platform heterogeneity for the ultimate enhanced energy-accuracy trade-off. First, I explored the flaws of REWARD during specific conditions of sudden changes in the ECG signal, which can occur during intense physical exercise or in certain pathologies. Next, I designed

an error detection algorithm that can detect when REWARD fails. Then, I proposed a novel and more robust slope-based R peak detection algorithm, called BayeSlope, that is triggered when an error in REWARD's output is detected. Moreover, the more complex BayeSlope is offloaded on a different type of core, with additional capabilities.

In conclusion, the designer of new wearable sensors for remote health and wellness monitoring should focus on adaptivity, by means of personalization, online multibiosignal-based knowledge acquisition, modularity and scalability, as the key to enhance the energy-accuracy trade-off.

5.2 Future Work

The next step for future work related to this thesis can be divided into short-term and long-term. The first is based on the preliminary work that has been done, but not yet implemented or presented as a main contribution, and the latter is a collection of ideas for future research or device development. For both long-term and short-term, I will refer to the specific contributions of this thesis that can be expanded beyond it.

5.2.1 Short-Term

- The algorithm for HR estimation from PPG presented in Section 2.3 can be integrated into a newly presented multi-modal device for online cognitive workload monitoring [150], as it was tested in the same microcontroller used in this work. Moreover, the method can benefit from the modular parallelization techniques explored in Section 3.4, as it includes most of the modules presented in Section 3.2. Since there are instances of the same module applied to the 3-axis accelerometer and the PPG signal, it is suitable for a lead parallelization. Additionally, there are also modules where window and data-level parallelization can be applied. Furthermore, there are instances of computationally expensive kernels that can be accelerated, such as the fixed-point fast Fourier transform (FFT). Finally, considering the application duty cycle of approximately 23 %, the approach can benefit from the three orthog-

onal optimizations discussed in Section 3.4, and it can be thought for a multi-modal device, such as the one presented in [151].

- In the context of the adaptive design presented in Chapter 4, an approximate version of BayeSlope can be integrated and tested in PULP. I have already done preliminary work on the comparison of the piecewise linear approximation of computationally intensive functions (gaussian and logistic function) and the original ones. Specifically, I estimated the reduction in active cycles of the approximated version of circa 15 %. These functions are not data-dependent though more testing is needed to estimate the impact on the energy consumption.

- The adaptive design is also suitable for parallelization, specifically the module BayeSlope, following the window parallelization technique presented in Section 3.4. This can significantly reduce the energy consumption to leave room for additional inference in pathology detection, such as PAF prediction, or heart rate variability (HRV) analysis (e.g., for ventilatory thresholds detection, c.f. Appendix A).

- Finally, for a more adaptive approach, once BayeSlope is triggered, in the next window the physiological parameters of REWARD can be updated based on BayeSlope's output. I have tested this strategy on one incremental exercise stress test recording, and the original REWARD without parameters update showed 38 % of RR outliers compared to BayeSlope. In contrast, the parameters update reduces the RR outliers to 8 %. This can lead to less triggering of BayeSlope as REWARD should fail less. More testing on the full dataset needs to be performed, as well as an analysis on the effect of this adaptivity on each subject in terms of accuracy and energy consumption. Additionally, as mentioned in Section 4.5.1, the RR ratio distribution used for the error detection can be more adaptive to the different exercise intensities. First, separated distributions can be computed for different intensities. Second, the distribution can be trained and adapted online via machine learning algorithms to detect the exercise intensity and adapt the tail thresholds.

5.2.2 Long-Term

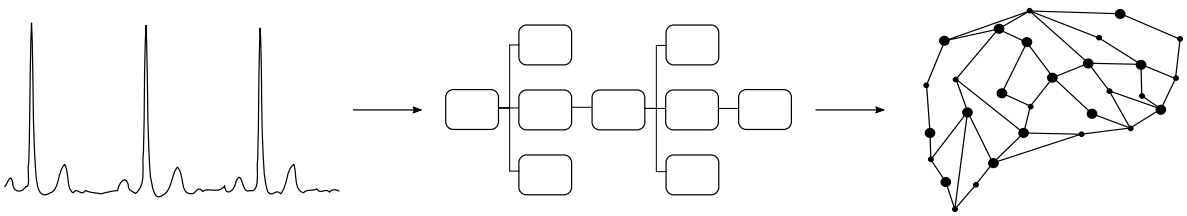
- Considering personalized biomedical applications, such as the one proposed in Section 2.4 and Section 3.5, it could be relevant to explore how to dynamically change the operating frequency and voltage to lower the energy consumption based on the characteristics of the patient. An energy-accuracy evaluation must be performed. Additionally, for the online PAF prediction method and other modular biomedical applications analyzed, it would be very promising to explore the effect on energy consumption of dynamic memory management at run time in modern platforms like PULP. This is useful to optimize further the power consumption of the system. As memory management at acquisition time has been proven to reduce energy consumption (c.f. Chapter 3), there could be benefits by parallelizing the buffering and the processing [45].

- The Relative-Energy (Rel-En) approach could be applied to a PPG signal [50] paired to the frequency analysis of the HR estimation from PPG (c.f. Section 2.3) to explore pulse rate variability as a substitute to ECG for more comfortable and low-cost sensors. Additionally, the frequency analysis can be used to remove motion artifacts and reconstruct the filtered PPG in modern ULP platforms that can afford more computationally expensive modules.

- The adaptive design presented in Chapter 4 can be applied to ventilatory threshold detection algorithms, such as the preliminary work presented in Appendix A. This work paired with the adaptive design can be extended to a real-time and subject-specific detection based on HRV patterns [34]. The final method following the combination of algorithmic and platform optimizations can tailor the design of a new ULP wearable device to substitute the gas analysis during incremental exercise stress tests to be used in sport medicine.

- Finally, this thesis can open a new and exciting field of exploration into the design of targeted platforms by considering biomedical application requirements. Key factors to consider for platform specialization are the memory and computing resources organization based on the needs

of the application, as discussed in Chapter 3 and Chapter 4; and the personalization described in Chapter 2 and Chapter 3.



Appendix

THIS appendix describes a preliminary work for an optimal exercise training system using the ventilatory thresholds obtained by the medical excerpts in the dataset presented in Chapter 4.

A Sub and Superoptimal Training Detection Using Ventilatory Thresholds

Referring to Section 4.2, when VT1 is reached, hyperpnea occurs. Physiologists refer to the phase before VT1 as suboptimal training and after VT1 as superoptimal training (80% and 120% of VT1). The goal of this preliminary work was to classify in real-time ECG and PPG segments as suboptimal or superoptimal exercises based on a training model that considers the full dataset. The classification was implemented in a system, which includes two devices, one running the ECG and another one running the PPG, based on the platform used in [119]. In fact, another goal of the work was to observe if using ECG and PPG data together could achieve more accurate results compared to using only ECG. The devices connect via the Bluetooth Low-Energy (BLE) protocol to a tablet that runs an Android application. Here is a list of the components of the system:

- STM32L151xD, which is an Ultra-low-power 32-bit microcontroller unit (MCU) Arm Cortex®-M3 with a 384KB Flash, a 48KB SRAM and different power saving modes;
- an ECG sensor ADS1291 sampling at 250 Hz;
- a PPG sensor sampling at 125 Hz;
- a 3-axis accelerometer sampling at 250 Hz;
- nRF8001 BLE module;
- UARTs to send the signals from the dataset for more reliable testing;
- an Android application.

The first step consists of training a classification model to identify a suboptimal (before VT1) or superoptimal exercise. This was performed offline by considering the data from the incremental stress test phase of the dataset for 19 out of 22 subjects (3 of them did not perform the second phase of the experiment or one of the recordings was not usable). I separated the ECG and PPG segments belonging to the two classes considering the position of VT1 that was identified by several medical experts. Moreover, I used also some segments of the second phase of the experiment in which the subject had to cycle at 80 % and 120 % of VT1 (constant exercise). However, I chose only 13 out of 19 subjects from the second phase and used the remaining (segments from 6 subjects) for testing.

I evaluated two classification models, one that uses both ECG and PPG (two-device model) and one that uses only ECG (one-device model). The features extracted from the ECG for this application are the mean and rmssd parameters extracted from the RR intervals of 20-second windows (HRV features). From the PPG, I considered the mean and standard deviation of the HR computed using the algorithm presented in Section 2.3 on 20-second windows. I trained the features with a linear support vector machine (SVM) for the one-device model and a random forest for the two-device model as they performed better at training time. In fact, the cross-validation results applied to the training data achieved a G-mean of 79 % for the one-device model

and 83 % for the two device model. I tested the classification models on the segments acquired from the constant exercise performed by the 6 subjects left out of the training. At testing time, the classification inference achieved a G-mean of 89 % for the one-device model and 86 % for the two-device model. These results are very promising considering that they use a generalized model on different subjects. In future works, the classification can be subject-specific for better performance. Moreover, the ventilatory thresholds detection could be performed in real-time considering the HRV parameters of the ECG as done in [35].

The classification model of the one-device model is deployed on the ECG-based device since it needs to transmit only the final output. On the contrary, the classification model of the two-device model is deployed on the tablet that receives the features of the two signals coming from two different devices. Fig. A.1 shows the system overview starting with the tablet on the right and the devices on the left. In the Android application, the user can choose to perform a suboptimal or superoptimal training. This information will later be used to send feedback to the user if they are performing the constant exercise correctly or not, that is, maintaining the suboptimal or the superoptimal training or deviating from it. Moreover, the user can choose if performing the exercise with one or two sensors. The application will then connect via BLE to one or two devices depending on the choice. If the user chooses only one sensor, the application sends a flag to the devices that specifies which model they should use. This applies only to the ECG-based device since the PPG is used only in combination with the PPG. Fig. A.2 shows the power management strategy derived by the two models and for both devices when they receive the flag. In the ECG-based device, if the flag is two the device will only perform the feature extraction and send the features as output to the tablet via BLE. If the flag received is one, the ECG-based device will perform the feature extraction and the classification model using a linear SVM. If the user requested two sensors, then the PPG-based device is included in the system and will perform only the feature extraction and send only the features via BLE to the tablet. The PPG-based device comprises a 3-axis accelerometer, as it is needed for the motion artifacts removal in the HR estimation.

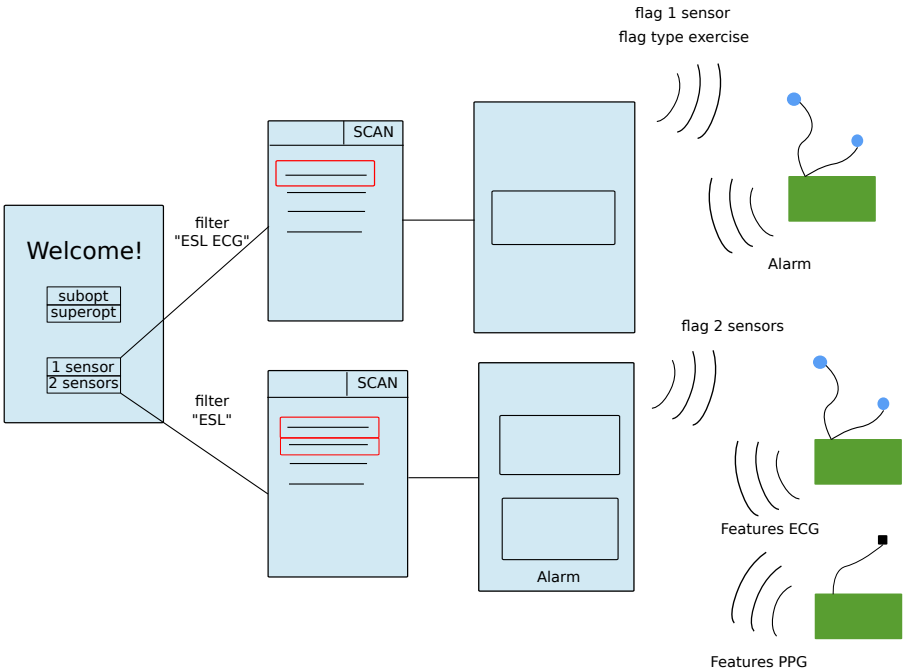


Figure A.1 – System overview of the application for optimal training

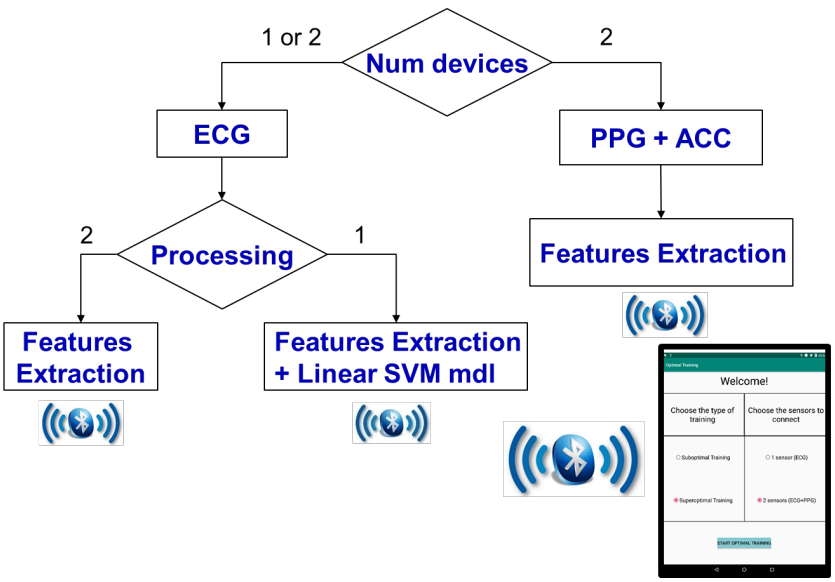
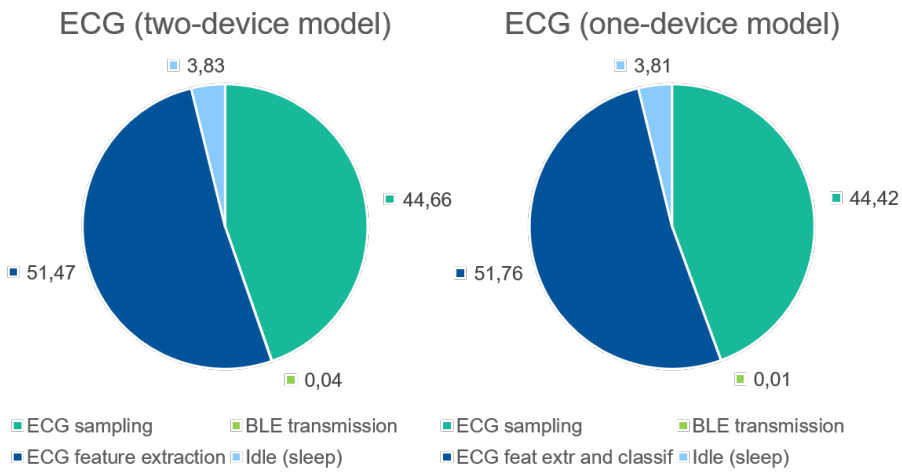


Figure A.2 – Power management and transmission strategy in both devices

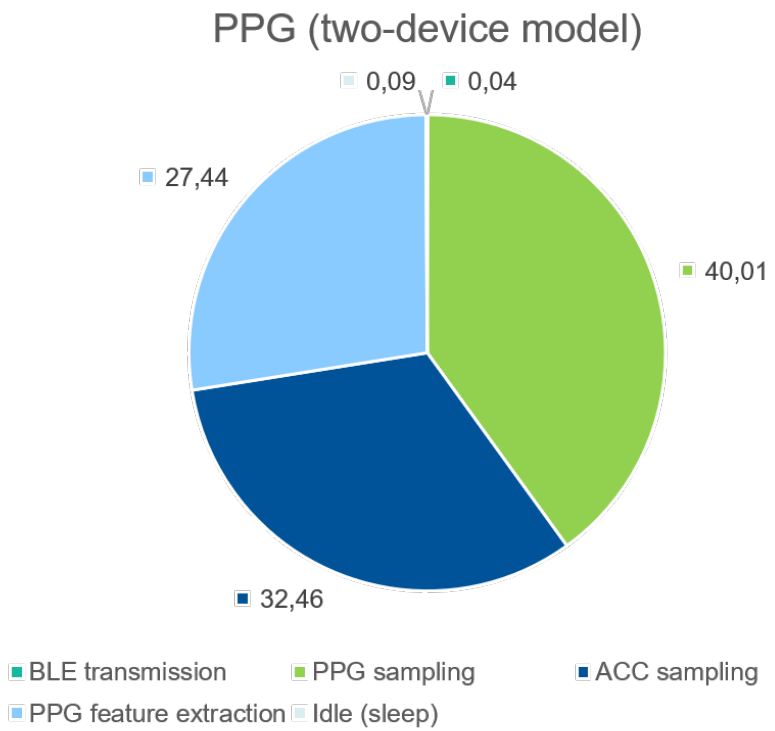


Battery rating (mAh)	**710**	**Battery rating (mAh)**	**710**
Battery lifetime (d)	**64.45**	**Battery lifetime (d)**	**64.10**

Figure A.3 – Energy consumption divided by the different components of the ECG-based device, running the two-device model (on the left) and the one-device model (on the right). On the bottom, the battery lifetime for the two models considering the total average current measured of 0.459 mA and 0.462 mA, respectively

To evaluate the performance of the two models (one-device and two-device), I profiled the energy consumption using the Simplicity Studio SW energy profiler on the Cortex-M3 based EFM32LG-STK3600, since it is within the same family of microcontrollers as the STM32L151xD. I also consider the consumption of all the components in the ECG-based device running the two models and in the PPG-based device. Finally, I estimated the battery lifetime of both devices. Fig. A.3 and Fig. A.4 show the consumption divided in the different components of the two devices and how it is affected by the two models. I also report the battery lifetime at the bottom of the figures.

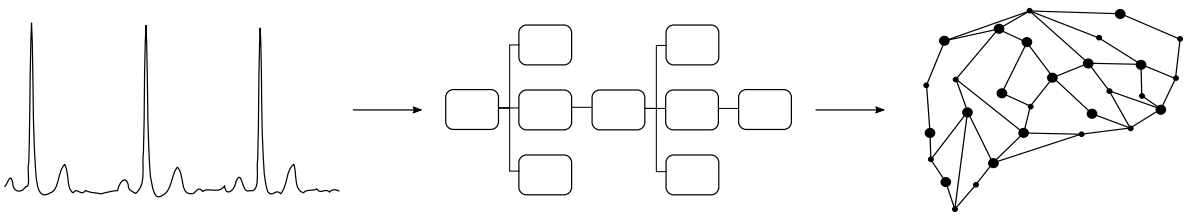
Considering the results of Fig. A.3, running the two-device model or the one-device model has a small difference in battery lifetime. This is due to the fact that the linear SVM classification model is very lightweight and the



Battery rating (mAh)	**710**
Battery lifetime (d)	**2.23**

Figure A.4 – Energy consumption divided by the different components of the PPG-based device for the two-device model. On the bottom, the battery lifetime considering the total average current measured of 13.246 mA

difference in processing is small (dark blue section). Moreover, the transmission accounts for a small percentage of the total energy consumption since the output sent to the tablet consists of one or two values for both cases. If more features were extracted and a more complex model was used, the energy consumption would have a bigger difference in the two models. Nonetheless, using the two-device model is better for the battery lifetime in this case, although it is more accurate to use one-device model, according to the test results. Furthermore, in the two-device model the second PPG-based platform need to be accounted for. In fact, the second device consumes a lot more compared to the first device with a battery lifetime of only 2.23 days. Therefore, using the two-device model is worse overall. However, the low battery lifetime is due to the sampling of the PPG and accelerometer, which accounts for almost 70 % of the total energy consumption. More studies need to be performed on optimizing the acquisition process as were proposed already for the ECG [43].



Bibliography

[1] Ke Xu, Agnes Soucat, Joseph Kutzin, Callum Brindley, Nathalie Vande Maele, Hapsatou Toure, Maria Aranguren Garcia, Dongxue Li, Hélène Barroy, Gabriela Flores Saint-Germain, Tomáš Roubal, Chandika Indikadahena, and Veneta Cherilova. Public spending on health: A closer look at global trends. *WHO*, 2018.

[2] Mohammed Al-khafajiy, Thar Baker, Carl Chalmers, Muhammad Asim, Hoshang Kolivand, Muhammad Fahim, and Atif Waraich. Remote health monitoring of elderly through wearable sensors. *Multimedia Tools and Applications*, January 2019.

[3] World Health Organization. Cardiovascular diseases (CVDs). *WHO*, 2018.

[4] Ashish Kumar, Rama Komaragiri, and Manjeet Kumar. From pacemaker to wearable: Techniques for ecg detection systems. *Journal of Medical Systems*, 42:34, Feb. 2018.

[5] Vuori Ilkka, Andersen Lars Bo, Cavill Nick, Breda João, Whiting Stephen, Mendes Romeu, Løgstrup Susanne, and Kestens Marleen. New report reveals the role of physical activity in preventing and treating cardiovascular diseases. *European Heart Network*, Jan. 2020.

[6] World Health Organization. Telemedicine - Opportunities and devel-

opments in Member States. Report on the second global survey on eHealth; Global Observatory for eHealth series - Volume 2. Technical report, WHO, 2010.

[7] Mary Boudreau Conover. *Understanding Electrocardiography*. Mosby, affiliate of Elsevier Health Sciences, 8 edition, 2002.

[8] Rashid Bashshur and Gary William Shannon. *History of Telemedicine: Evolution, Context, and Transformation*. Mary Ann Liebert, Inc., 2009.

[9] Norman J. HOLTER and J. A. GENERELLI. Remote recording of physiological data by radio. *Rocky Mountain medical journal*, 46(9):747–751, 1949.

[10] Norman J. Holter and Wilford R. Glasscock. Electrocardiographic means, Jul. 1965.

[11] Schiller. Schiller - The Art of Diagnostics. https://www.schiller.ch/, 2021.

[12] Javier Andreu-Perez, Daniel R. Leff, H. M. D. Ip, and Guang-Zhong Yang. From Wearable Sensors to Smart Implants--Toward Pervasive and Personalized Healthcare. *IEEE Transactions on Biomedical Engineering*, 62(12):2750–2762, Dec. 2015.

[13] Ali K. Yetisen, Juan Leonardo Martinez-Hurtado, Barış Ünal, Ali Khademhosseini, and Haider Butt. Wearables in Medicine. *Advanced Materials*, 30(33):1706910, Aug. 2018.

[14] Lakmini P. Malasinghe, Naeem Ramzan, and Keshav Dahal. Remote patient monitoring: a comprehensive study. *Journal of Ambient Intelligence and Humanized Computing*, 10(1):57–76, Jan. 2019.

[15] Matteo Stoppa and Alessandro Chiolerio. Wearable Electronics and Smart Textiles: A Critical Review. *Sensors*, 14(7):11957–11992, Jul. 2014.

[16] Morteza Amjadi, Ki-Uk Kyung, Inkyu Park, and Metin Sitti. Stretchable, Skin-Mountable, and Wearable Strain Sensors and Their Potential Ap-

plications: A Review. *Advanced Functional Materials*, 26(11):1678–1698, Mar. 2016.

[17] Shirley Musich, Shaohung Wang, Kevin Hawkins, and Andrea Klemes. The impact of personalized preventive care on health care quality, utilization, and expenditures. *Population Health Management*, 19(6):389–397, Dec. 2016.

[18] Kyeonghye Guk, Gaon Han, Jaewoo Lim, Keunwon Jeong, Taejoon Kang, Eun Kyung Lim, and Juyeon Jung. Evolution of wearable devices with real-time disease monitoring for personalized healthcare. *Nanomaterials*, 9(6), Jun. 2019.

[19] Ya-Li Zheng, Xiao-Rong Ding, Carmen Chung Yan Poon, Benny Ping Lai Lo, Heye Zhang, Xiao-Lin Zhou, Guang-Zhong Yang, Ni Zhao, and Yuan-Ting Zhang. Unobtrusive Sensing and Wearable Devices for Health Informatics. *IEEE Transactions on Biomedical Engineering*, 61(5):1538–1554, May 2014.

[20] Dionisije Sopic, Srinivasan Murali, Francisco Javier Rincon Vallejos, and David Atienza. Touch-based system for beat-to-beat impedance cardiogram acquisition and hemodynamic parameters estimation. In *Proc. of DATE*, volume 1, pages 6. 150–155. IEEE/ACM Press, Mar. 2017.

[21] Dionisije Sopic, Elisabetta De Giovanni, Amir Aminifar, and David Atienza. Hierarchical cardiac-rhythm classification based on electrocardiogram morphology. In *Proc. of CinC*, pages 1–4, Sep. 2017.

[22] Christopher C. Cheung, Andrew D. Krahn, and Jason G. Andrade. The emerging role of wearable technologies in detection of arrhythmia. *Can. J. Cardiol.*, 34(8):1083 – 1087, May 2018.

[23] Tuan Nguyen Gia, Imed Ben Dhaou, Mai Ali, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease. *Future Generation Computer Systems*, 93:198 – 211, Apr. 2019.

[24] Gregoire Surrel, Amir Aminifar, Francisco Rincon, Srinivasan Murali, and David Atienza. Online obstructive sleep apnea detection on medical wearable sensors. *IEEE Trans. on Biomedical Circuits and Systems*, pages 1–12, Aug. 2018.

[25] Farnaz Forooghifar, Amir Aminifar, Leila Cammoun, Ilona Wisniewski, Carolina Ciumas, Philippe Ryvlin, and David Atienza. A self-aware epilepsy monitoring system for real-time epileptic seizure detection. *Mobile Networks and Apps*, pages 1–14, Aug. 2019.

[26] Dionisije Sopic, Amir Aminifar, and David Atienza. Real-time classification technique for early detection and prevention of myocardial infarction on wearable devices. In *Proc. of BioCAS*, volume 1, pages 1–4. IEEE, Oct. 2017.

[27] Victor Kartsch, Simone Benatti, Marco Guermandi, Fabio Montagna, and Luca Benini. Ultra Low-Power Drowsiness Detection System with BioWolf. In *Int. IEEE/EMBS Conf. on Neural Engineering (NER)*, pages 1187–1190. IEEE, Mar. 2019.

[28] Michele Magno, Giovanni A. Salvatore, Petar Jokic, and Luca Benini. Self-sustainable smart ring for long-term monitoring of blood oxygenation. *IEEE Access*, 7:115400–115408, Jul. 2019.

[29] Eftim Zdravevski, Petre Lameski, Vladimir Trajkovik, Nuno Pombo, and Nuno Garcia. Importance of personalized health-care models: A case study in activity recognition. In *Studies in Health Technology and Informatics*, volume 249, pages 185–188. IOS Press, 2018.

[30] Juul Achten and Asker E. Jeukendrup. Heart rate monitoring: Applications and limitations. *Sports Medicine*, 33(7):517–538, 2003.

[31] Jan M. Mühlen, Julie Stang, Esben Lykke Skovgaard, Pedro B. Judice, Pablo Molina-Garcia, William Johnston, Luís B. Sardinha, Francisco B. Ortega, Brian Caulfield, Wilhelm Bloch, Sulin Cheng, Ulf Ekelund, Jan Christian Brønd, Anders Grøntved, and Moritz Schumann. Recommendations for determining the validity of consumer wearable heart

rate devices: Expert statement and checklist of the INTERLIVE Network. *British Journal of Sports Medicine*, 0:1–13, Jan. 2021.

[32] Kathryn E. Speer, Stuart Semple, Nenad Naumovski, and Andrew J. McKune. Measuring Heart Rate Variability Using Commercially Available Devices in Healthy Children: A Validity and Reliability Study. *European Journal of Investigation in Health, Psychology and Education*, 10(1):390–404, Jan. 2020.

[33] Laurent Schmitt, Jacques Regnard, Anne-Laure Parmentier, Frédéric. Mauny, Laurent Mourot, Nicolas Coulmy, and Gregoire Millet. Typology of "fatigue" by heart rate variability analysis in elite nordic-skiers. *International Journal of Sports Medicine*, 36:999–1007, Aug. 2015.

[34] François Cottin, P. M. Leprêtre, P. Lopes, Y. Papelier, C. Médigue, and V. Billat. Assessment of ventilatory thresholds from heart rate variability in well-trained subjects during cycling. *International Journal of Sports Medicine*, 27:959–967, Dec. 2006.

[35] François Cottin, C. Médigue, P. Lopes, P. M. Leprêtre, R. Heubert, and V. Billat. Ventilatory thresholds assessment from heart rate variability during an incremental exhaustive running test. *International Journal of Sports Medicine*, 28:287–294, Apr. 2007.

[36] Domingo Jesús Ramos-Campo, Jacobo A. Rubio-Arias, Vicente Ávila Gandía, Cristian Marín-Pagán, Antonio Luque, and Pedro E. Alcaraz. Heart rate variability to assess ventilatory thresholds in professional basketball players. *Journal of Sport and Health Science*, 6:468–473, Dec. 2017.

[37] Ravi Kondama Reddy, Rubin Pooni, Dessi P. Zaharieva, Brian Senf, Joseph El Youssef, Eyal Dassau, Francis J. Doyle, Mark A. Clements, Michael R. Rickels, Susana R. Patton, Jessica R. Castle, Michael C. Riddell, and Peter G. Jacobs. Accuracy of wrist-worn activity monitors during common daily physical activities and types of structured exercise: Evaluation study. *JMIR mHealth and uHealth*, 6(12), Dec. 2018.

[38] Francisco Rincón, Joaquin Recas, Nadia Khaled, and David Atienza. Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes. *IEEE Trans. Inf. Technol. Biomed.*, 15(6):854–863, Nov. 2011.

[39] Morghan Hartmann, Umair Sajid Hashmi, and Ali Imran. Edge computing in smart health care systems: Review, challenges, and research directions. *Transactions on Emerging Telecommunications Technologies*, page e3710, Aug. 2019.

[40] Ju Ren, Yi Pan, Andrzej Goscinski, and Raheem A. Beyah. Edge Computing for the Internet of Things. *IEEE Network*, 32(1):6–7, Jan. 2018.

[41] Lara Orlandic, Elisabetta De Giovanni, Adriana Arza Valdes, Sasan Yazdani, Jean-Marc Vesin, and David Atienza. REWARD: Design, optimization, and evaluation of a real-time relative-energy wearable R-peak detection algorithm. In *Proc. of Engineering in Medicine and Biology Conference (EMBC)*. IEEE, Jul. 2019.

[42] Francisco Rincón, Paolo R. Grassi, Nadia Khaled, David Atienza, and Donatella Sciuto. Automated real-time atrial fibrillation detection on a wearable wireless sensor platform. In *Engineering in Medicine and Biology Society*, pages 2472–2475. IEEE, Aug. 2012.

[43] Gregoire Surrel, Tomas Teijeiro, Amir Aminifar, David Atienza, and Matthieu Chevrier. Event-triggered sensing for high-quality and low-power cardiovascular monitoring systems. *IEEE Design and Test*, 37(5):85–93, Oct. 2020.

[44] Mario Konijnenburg, Roland van Wegberg, Shuang Song, Hyunsoo Ha, Wim Sijbers, Jiawei Xu, Stefano Stanzione, Chris van Liempd, Dwaipayan Biswas, Arjan Breeschoten, Peter Vis, Chris Van Hoof, and Nick Van Helleputte. A 769μW battery-powered single-chip SoC with BLE for multi-modal vital sign health patches. In *Int. Solid-State Circuits Conference (ISSCC)*, pages 360–362. IEEE, February 2019.

[45] Francesco Conti, Davide Rossi, Antonio Pullini, Igor Loi, and Luca

Benini. PULP: A ultra-low power parallel accelerator for energy-efficient and flexible embedded vision. *Journal of Signal Processing Systems*, 84(3), Sep. 2016.

[46] Loris Duch, Soumya Basu, Ruben Braojos, Giovanni Ansaloni, Laura Pozzi, and David Atienza. HEAL-WEAR: An ultra-low power heterogeneous system for bio-signal analysis. *IEEE TCAS-I*, 64(9):2448–2461, Sep. 2017.

[47] Simone Benatti, Giovanni Rovere, Jonathan Bosser, Fabio Montagna, Elisabetta Farella, Horian Glaser, Philipp Schonle, Thomas Burger, Schekeb Fateh, Qiuting Huang, and Luca Benini. A sub-10mW real-Time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC. In *Proc. of 7th International Workshop on Advances in Sensors and Interfaces, IWASI*, pages 139–144. IEEE, Jul. 2017.

[48] Eric Flamand, Davide Rossi, Francesco Conti, Igor Loi, Antonio Pullini, Florent Rotenberg, and Luca Benini. GAP-8: A RISC-V SoC for AI at the edge of the IoT. In *Int. Conf. on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, Jul. 2018.

[49] Antonio Pullini, Davide Rossi, Igor Loi, Giuseppe Tagliavini, and Luca Benini. Mr.Wolf: An energy-precision scalable parallel ultra low power SoC for IoT edge processing. *IEEE Journal of Solid-State Circuits*, 54(7):1970–1981, Jul. 2019.

[50] Sasan Yazdani, Sibylle Fallet, and Jean Marc Vesin. A Novel Short-Term Event Extraction Algorithm for Biomedical Signals. *IEEE Trans. Biomed. Eng.*, 65(4), Apr. 2018.

[51] Monalisa Singha Roy, Rajarshi Gupta, Jayanta K. Chandra, Kaushik Das Sharma, and Arunansu Talukdar. Improving photoplethysmographic measurements under motion artifacts using artificial neural network for personal healthcare. *IEEE Transactions on Instrumentation and Measurement*, 67:2820–2829, Dec. 2018.

[52] Han-wook Lee, Ju-won Lee, Won-geun Jung, and Gun-ki Lee. The Periodic Moving Average Filter for Removing Motion Artifacts from PPG Signals. *International Journal Of Control Automation And Systems*, 5(6):701–706, 2007.

[53] Toshiyo Tamura, Yuka Maeda, Masaki Sekine, and Masaki Yoshida. Wearable Photoplethysmographic Sensors—Past and Present. *Electronics*, 3(2):282–302, 2014.

[54] Zhilin Zhang, Zhouyue Pi, Senior Member, and Benyuan Liu. TROIKA : A General Framework for Heart Rate Monitoring Using Wrist-Type Photoplethysmographic (PPG) Signals During Intensive Physical Exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015.

[55] Zhilin Zhang and Senior Member. Photoplethysmography-Based Heart Rate Monitoring in Physical Activities via Joint Sparse Spectrum Reconstruction. *IEEE Transactions on Biomedical Engineering*, 62(8):1902–1910, 2015.

[56] Paulus Kirchhof et al. 2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS. *Eur. Heart J.*, 37(38):2893–2962, Oct. 2016.

[57] Elisabetta De Giovanni, Srinivasan Murali, Francisco Rincon, and David Atienza. Ultra-Low Power Estimation of Heart Rate under Physical Activity Using a Wearable Photoplethysmographic System. In *Proceedings - 19th Euromicro Conference on Digital System Design, DSD*, Oct. 2016.

[58] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Comput. Netw.*, 54(15), 2010.

[59] Arturo Martínez, Daniel Abásolo, Raúl Alcaraz, and José J. Rieta. Alteration of the P-wave non-linear dynamics near the onset of paroxysmal atrial fibrillation. *Med. Eng. Phys.*, 37(7):692–697, Jul. 2015.

[60] Malcolm S. Thaler. *The Only EKG Book You'll Ever Need*. LWW, 9 edition, 2018.

[61] Bert Uwe Köhler, Carsten Hennig, and Reinhold Orglmeister. The principles of software QRS detection, 2002.

[62] Marek Malik, J. Thomas Bigger, A. John Camm, Robert E. Kleiger, Alberto Malliani, Arthur J. Moss, and Peter J. Schwartz. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. *European Heart Journal*, 17:354–381, Mar. 1996.

[63] Robert E. Kleiger, J. Philip Miller, J. Thomas Bigger, and Arthur J. Moss. Decreased heart rate variability and its association with increased mortality after acute myocardial infarction. *The American journal of cardiology*, 59:256–62, Feb. 1987.

[64] Marta Carrara, Luca Carozzi, Travis J Moss, Marco de Pasquale, Sergio Cerutti, Manuela Ferrario, Douglas E Lake, and J Randall Moorman. Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy. *Physiological Measurement*, 36:1873–1888, Sep. 2015.

[65] Jin-Guo Dong. The role of heart rate variability in sports physiology. *Experimental and therapeutic medicine*, 11:1531–1536, May 2016.

[66] José Manuel Bote, Joaquín Recas, Francisco Rincón, David Atienza, and Román Hermida. A modular low-complexity ECG delineation algorithm for real-time embedded systems. *IEEE J. Biomed. Health Inform.*, 22(2), Mar. 2018.

[67] Jiapu Pan and Willis J Tompkins. A simple real-time QRS detection algorithm. In *EMBS*, volume 4. IEEE, 1985.

[68] Juan Pablo Martínez, Rute Almeida, Salvador Olmos, Ana Paula Rocha, and Pablo Laguna. A wavelet-based ECG delineator: evaluation on standard databases. *IEEE Trans. Biomed. Eng.*, 51(4):570–81, Apr. 2004.

[69] Nicolas Boichat, Nadia Khaled, Francisco Rincon, and David Atienza. Wavelet-based ECG delineation on a wearable embedded sensor platform. In *Proc. of WBSN*, pages 256–261. IEEE, Jun. 2009.

[70] Mohamed Elgendi, Björn Eskofier, Socrates Dokos, and Derek Abbott. Revisiting QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems. *PLoS ONE*, 9(1), 2014.

[71] Ruben Braojos, Giovanni Ansaloni, David Atienza, and Francisco J Rincon. Embedded real-time ECG delineation methods: A comparative evaluation. In *Proc. of 12th International Conference on Bioinformatics & Bioengineering (BIBE)*, pages 99–104. IEEE, Nov. 2012.

[72] Elisabetta De Giovanni, Amir Aminifar, Adrian Luca, Sasan Yazdani, Jean-Marc Vesin, and David Atienza. A Patient-Specific Methodology for Prediction of Paroxysmal Atrial Fibrillation Onset. In *Proc. of CinC*, volume 44, pages 285–191, Sep. 2017.

[73] Elisabetta De Giovanni, Adriana Arza Valdes, Miguel Peón-Quirós, Amir Aminifar, and David Atienza. Real-Time Personalized Atrial Fibrillation Prediction on Multi-Core Wearable Sensors. *IEEE Transactions on Emerging Topics in Computing*, Aug. 2020.

[74] Yan Sun, Kap Luk Chan, and Shankar Muthu Krishnan. ECG signal conditioning by morphological filtering. *Comput. Biol. Med.*, 32:465–79, Nov. 2002.

[75] Xinbo Qian and T. Hui Teo. A low-power comparator with programmable hysteresis level for blood pressure peak detection. In *TENCON 2009 - 2009 IEEE Region 10 Conference*, pages 1–4, Jan. 2009.

[76] Sasan Yazdani and Jean-Marc Vesin. Extraction of QRS fiducial points from the ECG using adaptive mathematical morphology. *Digital Signal Process.*, 56, Sep. 2016.

[77] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101:E215–220, 2000.

[78] Pablo Laguna, Roger G. Mark, Ary Goldberg, and George B. Moody.

A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In *Computers in Cardiology*, pages 673–676. IEEE, 1997.

[79] Gary M. Friesen, Thomas C. Jannett, Stanford L. Yates, Stephen R. Quint, and H. Troy Nagle. A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms. *IEEE Trans. Biomed. Eng.*, 37, 1990.

[80] AAMI. *Testing and reporting performance results of cardiac rhythm and ST-segment measurement algorithms*. The Association, 2008.

[81] EFM32™ Leopard Gecko 32-bit Microcontroller. https://www.silabs.com/products/mcu/32-bit/efm32-leopard-gecko.

[82] STMicroelectronics. STM32L151RD - Ultra-low-power ARM Cortex-M3 MCU with 384 Kbytes Flash, 32 MHz CPU, USB, 3xOp-amp - STMicroelectronics, Oct. 2017.

[83] Texas Instruments. Tm4c microcontrollers product selection guide, 2021.

[84] STMicroelectronics. STM32L4 - ARM Cortex-M4 ultra-low-power MCUs, 2021.

[85] STMicroelectronics. STM32F4 - ARM Cortex-M4 High-Performance MCUs, 2021.

[86] Georgios Karakonstantis, Aviinaash Sankaranarayanan, and Andreas Burg. Low complexity spectral analysis of heart-rate-variability through a wavelet based FFT. In *CinC*, pages 285–288, Sep. 2012.

[87] Robert C. Block, Mohammad Yavarimanesh, Keerthana Natarajan, Andrew Carek, Azin Mousavi, Anand Chandrasekhar, Chang Sei Kim, Junxi Zhu, Giovanni Schifitto, Lalit K. Mestha, Omer T. Inan, Jin Oh Hahn, and Ramakrishna Mukkamala. Conventional pulse transit times as markers of blood pressure changes in humans. *Scientific Reports*, 10, Dec. 2020.

[88] Toshiyo Tamura. Current progress of photoplethysmography and spo2 for health monitoring. *Biomedical Engineering Letters*, 9:21–36, Feb. 2019.

[89] Mio ALPHA 2 Heart Rate Activity Tracker Watch.

[90] Giulio Valenti, Klaas R Westerterp, and Human Biology. Optical Heart Rate Monitoring Module Validation Study.

[91] Jihyoung Lee, Kenta Matsumura, Ken Ichi Yamakoshi, Peter Rolfe, Shinobu Tanaka, and Takehiro Yamakoshi. Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 1724–1727, 2013.

[92] Yuka Maeda, Masaki Sekine, and Toshiyo Tamura. Relationship between measurement site and motion artifacts in wearable reflected photoplethysmography. *Journal of Medical Systems*, 35(5):969–976, 2011.

[93] K.a. Reddy, B. George, and V.J. Kumar. Use of Fourier Series Analysis for Motion Artifact Reduction and Data Compression of Photoplethysmographic Signals. *IEEE Transactions on Instrumentation and Measurement*, 58(5):1706–1711, 2009.

[94] Fast Fourier transform - MATLAB fft - MathWorks Schweiz.

[95] Matthew P. Ford, Robert C. Wagenaar, and Karl M. Newell. Arm constraint and walking in healthy adults. *Gait and Posture*, 26(1):135–141, 2007.

[96] Eishi Hirasaki, Steven T. Moore, Theodore Raphan, and Bernard Cohen. Effects of walking velocity on vertical head and body movements during locomotion. *Experimental Brain Research*, 127(2):117–130, 1999.

[97] Target Heart Rates Chart | American Heart Association. https://www.heart.org/.

[98] Fixed-point Fast Fourier Transform. https://www.jjj.de/crs4/integer_fft.c.

[99] Zhilin Zhang. Heart rate monitoring during physical exercise using wrist-type photoplethysmographic (ppg) signals, 2015.

[100] Amit Sinha, Alice Wang, and Anantha P. Chandrakasan. Algorithmic transforms for efficient energy scalable computation. In *Proc. of ISLPED*, pages 31–36. ACM Press, Jul. 2000.

[101] Wei Zong, R. Mukkamala, and R.G. Mark. A methodology for predicting paroxysmal atrial fibrillation based on ECG arrhythmia feature analysis. In *Proc. of CinC*, volume 28, pages 125–128. IEEE, 2001.

[102] G. Schreier and et al. An automatic ECG processing algorithm to identify patients prone to paroxysmal atrial fibrillation. In *Computers in Cardiology 2001. Vol.28 (Cat. No.01CH37287)*, pages 133–135. IEEE, 2001.

[103] Manab K. Das and Samit Ari. Patient-specific ECG beat classification technique. *Healthcare technology letters*, 1:98–103, 2014.

[104] Turker Ince, Serkan Kiranyaz, and Moncef Gabbouj. Automated patient-specific classification of premature ventricular contractions. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5474–5477. IEEE, 2008.

[105] Elias Ebrahimzadeh, Maede Kalantari, Mohammadamin Joulani, Reza Shahrokhi Shahraki, Farahnaz Fayaz, and Fereshteh Ahmadi. Prediction of paroxysmal Atrial Fibrillation: A machine learning based approach using combined feature vector and mixture of expert classification on HRV signal. *Comput. Methods Programs Biomed*, 165:53–67, Oct. 2018.

[106] Maryam Mohebbi and Hassan Ghassemian. Prediction of paroxysmal atrial fibrillation based on non-linear analysis and spectrum and bispectrum features of the heart rate variability signal. *Computer Methods and Programs in Biomedicine*, 105:40–49, Jan. 2012.

[107] Khang Hua Boon, Mohamed Khalil-Hani, and Balakrishnan Malarvili. Paroxysmal atrial fibrillation prediction based on HRV analysis and non-dominated sorting genetic algorithm III. *Comput. Methods Programs Biomed.*, 153:171–184, Jan. 2018.

[108] George Moody, Ary L. Goldberger, Seth McClennen, and Stephen Swiryn. Predicting the onset of paroxysmal atrial fibrillation: the Computers in Cardiology Challenge 2001. In *Proc. of CinC*, volume 28, pages 113–116. IEEE, 2001.

[109] Claudia Perlich. Learning curves in machine learning. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 577–580. Springer US, 2010.

[110] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[111] Silicon Labs. EFM32 Leopard Gecko Reference Manual, Apr. 2017.

[112] Texas Instruments. Low-Power,2-Channel,24-Bit AnalogFront-End for Biopotential Measurements, Sep. 2012.

[113] InvenSense. MPU-6000 and MPU-6050 product specification revision 3.4, Aug. 2013.

[114] Nordic Semiconductor. nRF8001 single-chip Bluetooth low energy solution, Mar. 2015.

[115] Ahmed Yasir Dogan, Jeremy Constantin, Martino Ruggiero, Andreas Burg, and David Atienza. Multi-core architecture design for ultra-low-power wearable health monitoring systems. In *IEEE DATE*, Mar. 2012.

[116] Madhuka Jayawardhana and Philip De Chazal. Enhanced detection of sleep apnoea using heart-rate, respiration effort and oxygen saturation derived from a photoplethysmography sensor. In *Proc. of the Annual International Conference of the Engineering in Medicine and Biology Society, EMBS*, pages 121–124. IEEE, Sep. 2017.

[117] Giovanna Sannino and Giuseppe De Pietro. A deep learning approach for ECG-based heartbeat classification for arrhythmia detection. *Future Generation Computer Systems*, 86:446–455, Sep. 2018.

[118] Paweł Pławiak and U. Rajendra Acharya. Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals. *Neural Computing and Applications*, pages 1–25, Jan. 2019.

[119] Fabio Isidoro Tiberio Dell'Agnola, Niloofar Momeni, Adriana Arza Valdes, and David Atienza. Cognitive workload monitoring in virtual reality based rescue missions with drones. In *12th International Conference on Virtual, Augmented and Mixed Reality in Copenhagen, Denmark*, 2020.

[120] Elisabetta De Giovanni, Fabio Montagna, Benoit W. Denkinger, Simone Machetti, Miguel Peon-Quiros, Simone Benatti, Davide Rossi, Luca Benini, and David Atienza. Modular Design and Optimization of Biomedical Applications for Ultralow Power Heterogeneous Platforms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3821–3832, Nov. 2020.

[121] John L. Semmlow and Benjamin Griffel. *Biosignal and Medical Image Processing*. CRC Press, 3 edition, 2014.

[122] Paul Kligfield, Leonard S. Gettes, James J. Bailey, Rory Childers, Barbara J. Deal, E. William Hancock, Gerard van Herpen, Jan A. Kors, Peter Macfarlane, David M. Mirvis, Olle Pahlm, Pentti Rautaharju, and Galen S. Wagner. Recommendations for the standardization and interpretation of the electrocardiogram. *Circulation*, 115(10):1306–1324, 2007.

[123] Ruben Braojos, Giovanni Ansaloni, and David Atienza. A methodology for embedded classification of heartbeats using random projections. In *IEEE DATE*, pages 899–904, New Jersey, Mar. 2013. IEEE.

[124] Rubén Braojos, Daniele Bortolotti, Andrea Bartolini, Giovanni Ansaloni, Luca Benini, and David Atienza. A synchronization-based hybrid-

memory multi-core architecture for energy-efficient biomedical signal processing. *IEEE Trans. on Computers*, 66(4):575–585, April 2017.

[125] Pasquale Davide Schiavone. zero-riscy: User Manual - PULP platform, January 2018.

[126] Pasquale Davide Schiavone, Francesco Conti, Davide Rossi, Michael Gautschi, Antonio Pullini, Eric Flamand, and Luca Benini. Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications. In *PATMOS*, September 2017.

[127] Massimo La Scala, G. Sblendorio, and R. Sbrizzai. Parallel-in-time implementation of transient stability simulations on a transputer network. *IEEE Trans. on Power Systems*, 9(2):1117–1125, May 1994.

[128] Victor Mondéjar-Guerra, Jorge Novo, José Rouco, Manuel Gonzalez Penedo, and Marcos Ortega. Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers. *Biomedical Signal Processing and Control*, 47:41–48, Jan. 2019.

[129] Hartej Singh, Ming-Hau Lee, Guangming Lu, Fadi J. Kurdahi, Nader Bagherzadeh, and Eliseu M. Chaves Filho. MorphoSys: An integrated reconfigurable system for data-parallel computation-intensive applications. *IEEE Transactions on Computers*, 49(5):465–481, May 2000.

[130] GitHub - pulp-platform/pulp-sdk, 2019.

[131] Soumya Subhra Basu. *Hardware/Software Co-Design and Reliability Analysis of Ultra-Low Power Biomedical Devices*. PhD thesis, EPFL, Lausanne, 2019.

[132] George B. Moody and Roger G. Mark. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[133] Anantha P. Chandrakasan, Naveen Verma, and Denis C. Daly. Ultralow-Power Electronics for Biomedical Applications. *Annual Review of Biomedical Engineering*, 10(1):247–274, Aug. 2008.

[134] Davide Zoni, Andrea Galimberti, and William Fornaciari. An FPU design template to optimize the accuracy-efficiency-area trade-off. *Sustainable Computing: Informatics and Systems*, page 100450, Oct. 2020.

[135] Daniele Palossi, Antonio Loquercio, Francesco Conti, Eric Flamand, Davide Scaramuzza, and Luca Benini. A 64-mw dnn-based visual navigation engine for autonomous nano-drones. *IEEE Internet of Things Journal*, 6:8357–8371, Oct. 2019.

[136] Simone Benatti, Fabio Montagna, Victor Kartsch, Abbas Rahimi, Davide Rossi, and Luca Benini. Online Learning and Classification of EMG-Based Gestures on a Parallel Ultra-Low Power Platform Using Hyperdimensional Computing. *IEEE Transactions on Biomedical Circuits and Systems*, 13(3):516–528, Jun. 2019.

[137] Maarten L. Simoons and Paul G. Hugenholtz. Gradual changes of ecg waveform during and after exercise in normal subjects. *Circulation*, 52:570–577, 1975.

[138] Jaap W. Deckers, Ruud V.H. Vinke, Jeroen R. Vos, and Maarten L. Simoons. Changes in the electrocardiographic response to exercise in healthy women. *Heart*, 64:376–380, 1990.

[139] Jonathan A. Drezner, Irfan M. Asif, David S. Owens, Jordan M. Prutkin, Jack C. Salerno, Robyn Fean, Ashwin L. Rao, Karen Stout, and Kimberly G. Harmon. Accuracy of ecg interpretation in competitive athletes: the impact of using standardised ecg criteria. *British Journal of Sports Medicine*, 46(5):335–340, 2012.

[140] Dionisije Sopic, Amin Aminifar, Amir Aminifar, and David Atienza. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE Transactions on Biomedical Circuits and Systems*, 12:982–992, Oct. 2018.

[141] Farnaz Forooghifar, Amir Aminifar, Leila Cammoun, Ilona Wisniewski, Carolina Ciumas, Philippe Ryvlin, and David Atienza. A self-aware

epilepsy monitoring system for real-time epileptic seizure detection. *Mobile Networks and Applications*, Aug. 2019.

[142] Zhixian Yan, Vigneshwaran Subbaraju, Dipanjan Chakraborty, Archan Misra, and Karl Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Proceedings - International Symposium on Wearable Computers, ISWC*, pages 17–24. IEEE, Jun. 2012.

[143] Q. Li, R. G. Mark, and G. D. Clifford. Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter. *Physiological Measurement*, 29(1):15–32, Jan. 2008.

[144] Kenneth C. Beck and Idelle M. Weisman. *Clinical Exercise Testing*, volume 32, chapter Methods for Cardiopulmonary Exercise Testing, pages 43–59. KARGER, Dec. 2002.

[145] Rik Gosselink, T. Troosters, and M. Decramer. Exercise testing: why, which and how to interpret. *Breathe*, 1:120–129, Dec. 2004.

[146] Gregory Blain, O. Meste, T. Bouchard, and S. Bermon. Assessment of ventilatory thresholds during graded and maximal exercise test using time varying analysis of respiratory sinus arrhythmia. *British Journal of Sports Medicine*, 39:448–452, Jul. 2005.

[147] Martin Buchheit, Roberto Solano, and Grégoire Paul Millet. Heart-rate deflection point and the second heart-rate variability threshold during running exercise in trained boys. *Pediatric Exercise Science*, 19:192–204, 2007.

[148] F. J. Richards. A flexible growth function for empirical use. *Journal of Experimental Botany*, 10:290–301, Jun. 1959.

[149] Research | biopac. https://www.biopac.com/research/.

[150] Fabio Isidoro and Tiberio Dell'agnola. Wearable and Self-Aware Machine Learning System for Online Cognitive Workload Monitoring and Drone Control. Technical report, EPFL, 2020.

[151] Philipp Schönle, Florian Glaser, Thomas Burger, Giovanni Rovere, Luca Benini, and Qiuting Huang. A Multi-Sensor and Parallel Processing SoC for Miniaturized Medical Instrumentation. *IEEE Journal of Solid-State Circuits*, 53(7):2076–2087, Jul. 2018.

ELISABETTA DE GIOVANNI

Route du Bois, 32 – 1024 Ecublens, Switzerland

elisabetta.degiovanni@epfl.ch

EDUCATION

Swiss Federal Institute of Technology Lausanne (EPFL) (2016 – now)

PhD Candidate on System-Level Design of Energy-Efficient Wearable Sensors for Optimal Fitness and Performance Monitoring

University of Pavia (2013/14 – 2014/15)

M. Sc. in Bioengineering – Technology for Health, Department of Electrical, Computer and Biomedical Engineering

University of Pavia (2010/11 – 2012/13)

B. Sc. in Bioengineering, Department of Electrical, Computer and Biomedical Engineering

HONORS

Merit-based Scholarship from University of Pavia (2011/12– 2012/13–2014/15)

Competition "Fondazione IBM Italia - 25x25 Talenti" (2015)

EXPERIENCE

University of Pavia (03.2013 – 09.2013)

Part-time job in organization of scientific manifestations, Department of Electronics

PROJECTS

Competition, Fondazione IBM Italia 25x25 Talenti (20.09.2015 – 30.10.2015)

Passepartout – A Platform for Sharing Information about Cultural Places to Help People with Special Needs

M.Sc. Thesis, École Polytechnique Fédérale de Lausanne (EPFL) (10.2015 – 04.2016)

Ultra-low Power Photoplethysmography System to Determine Heart Rate and SpO2 Level under physical activity

B.Sc. Thesis, University of Pavia (07.2013 – 12.2013)

Analysis of Head Movement During Walking and Running: Motion Tracking on Treadmill

PUBLICATIONS

1. E. De Giovanni, D. Atienza Alonso, S. Murali and F. J. Rincon Vallejos. **Ultra-Low Power Estimation of Heart Rate Under Physical Activity Using a Wearable Photoplethysmographic System**. 19th IEEE/Euromicro Conference On Digital System Design (DSD 2016), Limassol, Cyprus, August 31 - September 2, 2016.
2. E. De Giovanni, A. Aminifar, A. Luca, S. Yazdani, J.-M. Vesin and D. Atienza Alonso. **A Patient-Specific Methodology for Prediction of Paroxysmal Atrial Fibrillation Onset**. Computing in Cardiology, Rennes, France, September 24-27, 2017.
3. D. Sopic, E. De Giovanni, A. Aminifar and D. Atienza, **Hierarchical cardiac-rhythm classification based on electrocardiogram morphology**. 2017 Computing in Cardiology (CinC), Rennes, 2017, pp. 1-4.
4. L. Orlandic, E. De Giovanni, A. Arza Valdes, S. Yazdani, J. Vesin, and D. Atienza Alonso. (2019). **REWARD: Design, Optimization, and Evaluation of a Real-Time Relative-Energy Wearable R-Peak Detection Algorithm**. 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 3341-3347.
5. E.De Giovanni, A. Arza Valdes, M. Peón-Quirós, A. Aminifar, and D. Atienza. **Real-Time Personalized Atrial Fibrillation Prediction on Multi-Core Wearable Sensors**. IEEE Transactions on Emerging Topics in Computing, Aug. 2020
6. E. De Giovanni, F. Montagna, B. W. Denkinger, S. Machetti, M. Peón-Quirós, S. Benatti, D. Rossi, L. Benini, and D. Atienza. **Modular design and optimization of biomedical applications for ultra-low power heterogeneous platforms**. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(11):3821–3832, Nov. 2020.

TECHNICAL KNOWLEDGE

Programming skills

- Matlab
- Assembly, C, Java
- Python
- Android programming
- HTML, CSS, JavaScript, PHP, JSP
- Labview
- YAWL
- SQL, MySQL
- Machine learning

Tools

- Protégé
- TreeAge Pro
- Simi Motion 2D/3D
- Abaqus
- Wireshark

Operating Systems

- Windows, Linux, Mac OS

LANGUAGES

Italian: Native Speakers

English: Fluent

- Cambridge English: First (FCE)

French: Fluent

- Attestation EPFL Centre des langues : Level B1

Piled Higher and Deeper ***by Jorge Cham*** **www.phdcomics.com**

"The Plans" - originally published 9/19/2012