Towards Self-Sustainable Wearable Devices: Energy Harvesting, Management and Optimization Strategies

By

Yigit Tuncel

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2023

Date of final oral examination: 07/21/2023

The dissertation is approved by the following members of the Final Oral Committee:
Umit Y. Ogras, Associate Professor, Electrical and Computer Engineering, UW-Madison
Yu Hen Hu, Professor, Electrical and Computer Engineering, UW-Madison
Younghyun Kim, Assistant Professor, Electrical and Computer Engineering, UW-Madison
Joao Dorea, Assistant Professor, Animal and Dairy Sciences, UW-Madison

Dedicated to my *grandfather, Nuri*, the ever-curious engineer,
grandmother, Melahat, the best teacher I have known,
parents Meral and Uğur, the role models for the kind of person I am,
sister Melek, the dancer, the pianist, the kpop lover, my angel,
friends, who told me not to do it,
past-self, who thought I would never do it.

ACKNOWLEDGMENTS

I extend my deepest appreciation to my Ph.D. advisor, Prof. Umit Y. Ogras, for the invaluable support and guidance he provided since the first day we met. The completion of this dissertation would certainly not have been possible without the nurturing of Prof. Ogras. His technical expertise and the enthusiasm he shows for research resulted in many fruitful discussions, and ultimately enabled me to perform independently. Perhaps most importantly, he has offered endless sympathy and understanding throughout the years, which I'm extremely grateful for.

I am immensely thankful to Prof. Yu-Hen Hu, Prof. Younghyun Kim, and Prof. Joao Dorea for taking the time to serve on my Ph.D. defense committee. I had the pleasure of working with Prof. Kim and Prof. Dorea through my Ph.D. journey, and their helpful advice and practical suggestions played pivotal roles in this dissertation. Prof. Hu's valuable feedback about "application's demand" greatly improved the energy management section and resulted in a journal paper. All of their suggestions have been invaluable in improving the quality of this dissertation.

I sincerely thank Dr. Suat Gumussoy, Prof. Chaitali Chakrabarti, Prof. Bertan Bakkaloğlu, Prof. Kassem Fawaz, Prof. Şule Özev for their input and support during my research. I also wish to thank my previous advisor during my Master's studies in Turkey, Prof. Ziya Ider, for encouraging me to pursue a Ph.D.

I am also grateful for all of my friends and colleagues at eLab: Dr. Samet Arda, Prof. Ganapati Bhat, Prof. Sumit Mandal, Dr. Anish Krishnakumar, Toygun Başaklar, Sizhe An, Alper Göksoy, Conrad Holt, Vishrut Pandey, Jie Tong, Nuriye Yıldırım, Aditya Ukarande, Aishwarya Chithra, Shiva Bandyopadhyay, and Audrey Mendez. Their help and support, the conversation, and the banter made harder moments easier.

I enjoyed the lighter moments (sometimes heavier) with my friends scattered all around the world; in the US: Toygun, Ezgi, Matt, Melina, Kaan, Mehmet, Yavuz, Bengi, Hande, Daniel, Aybüke, Gökhan, Gizem, Lauren, Jordan, Emma, Yvette, Levent, Sedat, Çağatay; in Canada: Emre, Zeynep; in the Netherlands: Doğacan, Dilan, Tuğberk, Ezgi, Burkay; in the UK: Muhammet, Deniz; in Turkey: Barış, Naz,

Onur, and Oğuzhan. I am extremely happy that you are part of my life and the moments we shared together certainly helped me during these five years.

Finally, this would not have been possible without the support of my parents, Meral and Uğur, my sister Melek, and my family: Nuri, Zeliha, Melahat, Nihal, Mehmet, Zühal, Fahrettin, Fatih, Meltem, Elif, Hakan, Aycan, İlker, Taha, Nuray, Burak, Aslı, Yasemen, Metin, Yusuf, Öznur, Özge, Sertaç, Emine, Önder, Esin, Taner, Sevgi, Necla, Cem, Yulia, Itır, Özgür, Bulut, and Selen. Your love, support, and encouragement have been of immense support in my journey.

Finally, I would like to acknowledge and thank the Defense Advanced Research Projects Agency for funding the research presented in this dissertation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Wearable devices have many promising applications ranging from activity recognition to early diagnosis and prognosis of diseases through continuous monitoring of health-related signals. The emergence of new device form-factors, such as electronic patches and devices embedded into clothing, further amplifies the potential impact of wearable devices on everyday life. Despite the promising outlook, the widespread adoption of wearable devices has been hindered due to several adaptation and technology challenges owing to their limited battery, processing and storage capabilities.

On the one hand, wearable devices have severely limited battery capacity due to form-factor requirements, such as small size, light weight and flexibility. On the other hand, these devices collect significant amounts of data to enable sophisticated applications. The energy consumption soars if they transmit this data to a mobile device since wireless streaming of data is prohibitively power-consumptive. Consequently, the recharge interval is short, causing users to stop using these devices. Thus, reducing the dependency of wearable devices on batteries and achieving *self-sustainability* is necessary for unlocking the full potential of these devices and to boost their widespread adoption. To this end, this thesis presents three chapters to form a collaborative solution strategy: i) Incorporate novel, flexible, wearable energy harvesters as orthogonal sources to the battery, ii) employ energy management frameworks to make judicious use of the limited energy to maximize device lifetime, and iii) minimize energy consumption through SW and HW optimizations. In the first chapter, we present a novel analytical model and experimental validation for piezoelectric energy harvesting from joint movements . In the second chapter, we present two energy management algorithms, ECO and tinyMAN, to achieve long-term self-sustained operation. In the third chapter, we present a wearable ultra-low power IoT system for cattle health monitoring, and an open-source domain-specific system-on-chip combining a RISC-V core with accelerators targeting wearable applications. These developments reduce wearable devices' dependency on batteries, maximize their energy efficiency, and foster wider adoption.

1 INTRODUCTION

According to the annual world report on disability, around 15% of the global population lives with a disability [1]. Furthermore, between 100 to 190 million individuals encounter significant functioning difficulties. Healthcare professionals generally conduct state-of-the-art procedures for diagnosing, treating, and rehabilitating these individuals in a clinical setting [2]. However, monitoring the symptoms becomes impossible once the patient departs from the clinic due to the absence of standard protocols [2]. The use of *wearable Internet of Things* (*IoT*) *devices*, which amalgamate sensing, processing, and wireless communication, can drastically enhance this population's quality of life [3, 4, 5].

Wearable sensors and mobile health applications are rapidly becoming attractive solutions for monitoring users' daily activities continuously, enabling remote diagnostics, biofeedback, and even preventative medicine [2, 6, 7, 8]. For instance, recent studies suggest that wearable technology can be utilized for diagnosing and treating Parkinson's disease symptoms [9, 10, 11, 12] and monitoring patients in home environments [13, 14, 15, 16]. The advent of technology ranging from electronic stickers to flexible processors and ambient energy harvesting mechanisms paves the way for an even broader array of promising wearable devices [17, 18, 19, 20, 21]. However, despite this transformative potential, the practical application of wearable sensors and devices is mainly confined to research prototypes. In the following, we discuss the fundamental challenges that have hindered the adoption of wearable technology.

1.1 Challenges in Wearable Technology

User surveys and recent studies have revealed that a set of adaptation and technology challenges obstruct the widespread use of wearable devices [2, 22]. Adaptation challenges often deter new users from utilizing wearable technology and can lead existing users to abandon their devices. Such challenges encompass comfort and subtlety issues (e.g., users may feel self-conscious or awkward while wearing them), reduced compliance due to the need for frequent charging and maintenance, and privacy concerns linked with sharing sensitive data over a network [22, 23]. From the developers' perspective, technology challenges complicate the design of effective and viable solutions. These challenges typically involve the limited processing and storage capabilities of low-power designs and the constraints of battery and energy harvesting technologies [2, 24, 6]. Thus, achieving self-sustainable operation in energy harvesting wearable IoT devices is essential to address these adaptation and technology challenges. Unfortunately, these challenges are fundamentally interconnected, which requires non-trivial solutions. For example, providing enhanced processing and storage capability raises power consumption and form-factor issues, which subsequently diminish user comfort and cause compliance concerns. Consequently, novel driver applications, design methods, and run-time management algorithms must address these challenges to facilitate the broader adoption of wearable devices. The rest of this section discusses these challenges in more detail and presents our contributions to overcome them.

1.1.1 Adaptation Challenges

According to recent user surveys, the absence of *comfort* is a primary reason users abandon wearable devices [22]. Firstly, users tend not to favor devices that make them stand out or feel uncomfortable in public. These devices can potentially lead to stigmatization when used in public settings [23, 22]. Secondly, wearable devices should not disrupt daily routines or necessitate significant effort, such as always carrying a smartphone or being close to a central hub. Ideally, these devices should be stretchable or flexible, conforming to the user's skin or clothing like a patch/sleeve. This way, users can put them on and forget about them, and others around them would not even notice the device.

The small size and low-cost constraints of wearable devices drastically limit the battery capacity, rendering energy limitation a significant hurdle for wearable technology becoming ubiquitous [2, 24]. Furthermore, research indicates that the need for frequent recharging is one of the key reasons for low wearable adoption rates [22]. This aspect is particularly crucial for medical-grade devices. Consequently, reducing the maintenance effort for wearable devices is highly important. Practical solutions should ideally self-sustain by harvesting energy from the environment and optimally managing the available energy [25, 26].

Lastly, data privacy and security are highlighted as significant concerns about wearable devices used for health and activity monitoring [22]. Users may feel uncomfortable about sending sensitive personal data to the cloud, and the wireless communication links are vulnerable to adversary attacks. Therefore, it is necessary to secure personally identifiable information on the device by employing local

processing and storage, and to transmit the processed information to the cloud if necessary. As such, security and privacy concerns introduce a "*necessary overhead*" that must be accommodated when addressing the technology challenges.

1.1.2 Technology Challenges

Wearable health and activity monitoring devices accumulate significant amounts of data from multiple sensors. For instance, just an inertial measurement unit (IMU) collects data from a 3-axis accelerometer, gyroscope, and magnetometer (9 channels in total), typically sampled at 100-250 Hz. As sensor numbers rise, storage needs for just one hour of data can easily surpass tens of megabytes, while local storage capacities rarely extend beyond a few megabytes. On top of this, processing data streams from numerous sensors and making real-time decisions is a complex task. Transmitting raw data to a server for further processing is also not feasible as it could compromise security and rapidly drain the battery. Thus, the driver applications and the overhead of security-privacy mechanisms dictate the exact processing power required. This processing power could be well above what simple microcontrollers could provide, consequently reducing the battery life of the device.

The collected sensor data needs to be clinically relevant to be of value. As per the International Parkinson and Movement Disorders Society Task Force on Technology, the large volumes of data sourced from unstructured and crowd-sourced channels do not always translate into relevant clinical metrics [2]. Conversely, surplus data can diminish its direct applicability [24]. Hence, the sensor array in

a wearable device must be carefully selected according to the target applications. Moreover, filters and algorithms should efficiently extract relevant information for personalized patient treatment [24, 6]. As a result, most technological development efforts function within their respective "*islands of expertise*," resulting in limited compatibility among systems [2].

The final technological challenge is the absence of *compatible platforms* accessible to a broad developer base, which leads to two major issues: Firstly, integrating data from different devices becomes a challenge due to incompatible results. Secondly, restricted access to customized solutions inhibits a larger developer group. In contrast, open-source hardware-software platforms can offer a common framework and expedite progress [7]. For instance, application developers could utilize a common hardware platform, input-output interfaces, and libraries akin to smartphone app stores.

1.2 Contributions

Wearable devices have severely limited battery capacity due to form-factor requirements, such as small size, light weight and flexibility. For example, the Oura Ring 3 incorporates a 22 mAh – 3.7 V battery and advertises a battery life of 4 to 7 days [27]. On the other hand, these devices collect significant amounts of data to enable sophisticated health monitoring applications. The energy consumption soars if they transmit this data to a mobile device or the cloud since wireless streaming of data is prohibitively power-consumptive. Consequently, the recharge interval is short,

which causes the users to stop using these devices. Therefore, significant research effort is needed to reduce the dependency of wearable devices on batteries and achieve self-sustainability, in order to unlock the full potential of these devices and to boost their widespread adoption.

Self-sustainability is the capability of maintaining the continuous operation of a device with no external assistance. These devices ensure that the energy consumption in a given period, such as a day, is equal to the energy harvested during the period, hence achieving energy neutrality. Consequently, these devices practically have an unlimited lifetime. This is made possible by three key targets: First, the device should incorporate energy harvesting as an orthogonal energy source to the battery. Second, optimum energy management techniques should govern energy utilization to maximize device lifetime. And finally, the energy consumption of the device should be minimized to reduce the required energy budget through HW and SW optimizations. With these, achieving self-sustainability can transform existing pioneers and research prototypes into mainstream wearable technology that routinely improves the quality of life.

In light of the above discussion, this dissertation presents research and development in four main subjects: Firstly, it proposes a novel analytical model for piezoelectric energy harvesting from joint movements. The proposed model is validated with extensive experimental measurements on a 3D-printed human knee model. According to the results, an average user can obtain 0.9 mAh @ 3.6V per day with a combined light and piezoelectric wearable harvester. Next, this thesis proposes two novel runtime energy management algorithms to achieve self-sustained

operation over a long horizon (e.g., 5 years). The proposed algorithms help promote energy harvesting wearable devices as a successful alternative to battery powered ones. Thirdly, it proposes a novel wearable accelerometer-based ultra-low power IoT system that provides real-time cattle health monitoring to reduce labor and costs associated with manual animal inspection and intervention. The proposed approach can classify healthy and sick animals with 78% accuracy with less than 29 µJ energy consumption per day, outperforming 13 commonly used state-of-the-art time-series classifiers in terms of accuracy and computational complexity. Finally, it demonstrates a novel open-source domain-specific programmable system-on-chip (SoC) that combines a RISC-V core with a meticulously determined set of accelerators targeting wearable applications. Thorough experimental evaluations on an FPGA prototype show that the proposed SoC provides up to 9.1× faster execution and up to 8.9× higher energy efficiency than software implementations.

In summary, this dissertation makes the following contributions:

- Investigates the underlying mechanism of piezoelectric energy harvesting from bending motions of the knee [28, 29],
- Characterizes the energy harvesting capacity of a combined light and piezoelectric wearable energy harvester in a day [30],
- Presents ECO, an energy management framework to achieve energy-neutral operation for energy harvesting wearable devices [31],
- Presents tinyMAN, a reinforcement-learning-based energy manager to achieve self-sustained operation for energy harvesting wearable devices [32],

- Presents a novel cattle health monitoring application with ultra-low power consumption [33],
- Presents a domain-specific system-on-chip (SoC) design for energy efficient wearable edge applications [34].

The rest of the dissertation is organized as follows: The literature survey is discussed in Chapter 2. Chapter 3 presents the investigation and characterization of wearable piezoelectric energy harvesting. Chapter 4 presents two energy management algorithms, ECO and tinyMAN. Chapter 5 presents the novel ultra-low power cattle health monitoring application and the domain-specific SoC design. Finally, Chapter 6 concludes the dissertation with directions for future work.

2.1 Wearable Energy Harvesting Modalities

The concept of energy harvesting (EH), also known as energy scavenging in previous studies [35, 36], involves generating usable electrical energy from various sources within the device's surroundings. Even though the notion of energy harvesting is not new, the focus on it has surged in recent years due to the advancements in low-power electronics and device miniaturization, enabling the establishment of sensor networks and the utilization of billions of devices. Similarly, the challenge of limited battery life in wearable devices has driven research toward wearable energy harvesting.

An energy harvesting system typically comprises five key elements: (i) *Ambient energy source*, the most common ones being light, motion, radio frequency (RF) waves, and heat [37], (ii) *capture mechanism* that absorbs the ambient energy, (iii) *transduction mechanism* that converts the absorbed energy into electrical energy, (iv) *power management unit* that typically encompasses architecture-level power management circuits for impedance matching, maximum power point tracking, rectification, regulation, and energy storage, and (v) *the electrical load*. The specific setup of these stages depends on the type of modality (i.e., the energy source). The rest of this section surveys the four types of energy harvesting modalities from a wearables point of view and provides example studies from the literature, as summarized in Table 2.1.

Table 2.1: A Compilation of Wearable Energy Harvesting Studies.

Modality	Size	Source	Intensity	Voltage	Power	Reference
Light	813 mm^2	Indoor Illumination	500 lux	~2.2 V_{DC}	155 μW	[38]
	2581 mm^2		0.9 mW/cm^2	~2.0 V_{DC}	1650 μW	[39]
	487 mm^2		500 lux	–	69 μW	[40]
Motion	2964 mm^2	Knee	Walk/Jog	13.8 V_{RMS}	1.9/3.7μW	[41]
	3000 mm^2	Knee	Walk/Run	–	6.2/12 μW	[42]
	1476 mm^2	Knee	Walk	–	1.0 μW	[43]
	2380 mm^2	Robot	1.5 Hz	–	0.9 μW	[44]
RF	1836 mm^2	915 MHz	-10 dBm	~0.5 V_{DC}	10.0 μW	[45]
	20164 mm^2	1750 MHz	-15 dBm	~0.1 V_{DC}	3.9 μW	[46]
	2915 mm^2	915 MHz	-20 dBm	~0.6 V_{DC}	1.2 μW	[47]
Thermo-electric	1260 mm^2	ΔT (C)	12 C	2.8 mV_{DC}	0.14 μW	[48]
	100 mm^2	ΔT (C)	22 C	56.0 mV_{DC}	3.00 μW	[49]
	560 mm^2	ΔT (C)	16 C	~25 mV_{DC}	3.05 μW	[40]

"–": value is not available in the corresponding reference.
"~": value is extracted from a plot in the corresponding reference.

2.1.1 Light Energy Harvesting

Light energy harvesters take advantage of light energy in the environment through photovoltaic cells (PV-cell). These cells transmit photons with energy equal to or greater than the material's bandgap. The transduction process occurs in the PV-cell where the transmitted photons displace electrons from a specially prepared semiconductor material. The displaced electrons result in the direct current output from PV-cells. A voltage regulator is needed to adjust the voltage to the preferred levels, as the absence or surplus of light can lead to extremely low or high potential differences across PV-cell terminals, respectively.

Modern PV-cell technology provides low-cost, small, and flexible form factor PV-cells that can be used in wearable applications. Thus, recent studies have employed

flexible solar cells for powering various wearables, as shown in table 2.1. In one particular study, researchers developed a flexible energy source that integrates an energy harvester and an energy storage system [39]. They use a 25.8 cm^2 flexible PV-cell and report 1.65 mW harvested power under indoor lighting conditions. They further demonstrate the possibility of keeping the battery level at a constant state of charge with appropriate load management, indicating self-sustainability. The authors in [40] present a wearable device prototype that incorporates light and thermoelectric EH. This study uses eight 35×13.9 mm^2 flexible PV-cells and reports 550 μW harvested power in an indoor setting. It also showcases self-sustainable operation with the careful selection of the load duty cycle. Another recent work studies powering a smart bracelet with a flexible light energy harvester [38]. The authors use a 12.7×64 mm^2 flexible PV-cell and report 120 μW harvested power under indoor lighting conditions.

While light energy harvesting is the most commonly used EH technology, there have been fewer recent studies exploring the wearable application of this technology. This is primarily because PV-cells need unobstructed access to light, making it impossible to place them under clothing. Additionally, to achieve significant power output (e.g., 10 μW for 1 cm^2 or larger), they need to be of adequate size. Moreover, factors like shading and bending can drastically decrease the output power of PV-cells [50, 51]. Thus, they may necessitate computationally heavy maximum power point tracking (MPPT) algorithms to identify the optimum operating point at runtime [52]. For these reasons, [53] model the energy that could be harvested using flexible PV-cells based on the bending radius and inclination angle of the

energy source. The authors empirically validate the precision of these analytical models under varying shading and bending conditions. They also present an MPPT algorithm and case study for flexible PV-cells to support their use in wearable applications.

Despite the limitations outlined above, light energy harvesting remains a highly promising wearable EH method. It undoubtedly has the highest power output for outdoor lighting conditions. Numerous commercial-off-the-shelf (COTS) solutions exist for direct and straightforward integration with designs, a crucial factor for system designers. Lastly, the research field of solar cell modeling is well-developed and continues to evolve, taking into account factors such as temperature, bending, and partial shading of the PV-cell.

2.1.2 Motion Energy Harvesting

The broad field of kinetic energy harvesting encompasses natural sources of movement such as wind, underwater currents, and surface waves in oceans, along with human-made sources like industrial machinery, vehicles, and structures [54, 55]. However, in a wearable context, the energy source is the user, as energy is produced from user motion during activity. This coincides with wearable activity monitors, as the device requires power when the user is active (i.e. when there's a significant activity to track). Otherwise, the device can switch to a sleep or low-power mode where the sampling rates and duty cycle are decreased.

The human body produces kinetic energy in different forms, such as foot strike, movement of joints like the ankle, knee, hip, arm, and elbow, and upper body's

center-of-gravity motion [56]. Larger torques produced in lower limb motions, such as the ankle, knee, and hip, tend to generate more biomechanical energy. Especially during repetitive walking motions, continuous energy harvesting is feasible. This potential has driven numerous studies to concentrate on harvesting the biomechanical energy generated during walking [54, 56].

Three main mechanisms can convert motion energy into electrical energy: electrostatic transduction, electromagnetic induction (electrodynamic), and piezoelectricity [56]. Electrostatic transduction is achieved by gap closing or plate sliding of a capacitor [56]. Electromagnetic induction occurs through a coil moving across a magnetic flux density region [56]. Piezoelectric transduction requires a strain acting on the piezoelectric material [55]. In all these instances, the forces at play are due to the movement of the human body. As such, piezoelectric transduction is more suitable for low frequency human body motions (i.e., less than 10 Hz) [57]. Therefore, this thesis primarily discusses piezoelectricity-based wearable energy harvesters.

Piezoelectric energy harvesting (PEH) employs the piezoelectric effect, where an electric charge builds up in certain materials under applied mechanical stress. Piezoelectric materials comprise crystals, ceramics, and, more recently, polymers [54]. Traditionally, piezoelectric materials have been inflexible and brittle, limiting their application in wearables. Consequently, the most prevalent wearable piezoelectric application is in-sole shoe harvesters, used in children's shoes with flashing lights at the sole. While this particular use case may not be representative of modern wearable energy harvesting approaches due to its inflexibility, recent studies using

this methodology show that in-sole energy harvesting is still a pertinent method for PEH due to its high voltage output levels [58, 59, 60].

In the literature, piezoelectric energy harvesting is predominantly used for vibration energy harvesting. These vibrations are often at high frequencies, significantly higher than those associated with human movements such as walking and running. As a result, researchers have sought to excite the piezoelectric material at its mechanical resonant frequency to maximize energy generation. Typically, studies in the wearable energy harvesting literature up-convert the low-frequency mechanical forces from human motion to the resonant frequency of the piezoelectric material through complicated mechanical designs [61, 62, 63, 64]. Although these studies report encouraging energy outputs, the mechanical designs necessitate an external frame worn by the user, making the harvesters bulky, inflexible, and uncomfortable to wear, hindering these approaches from being genuinely wearable. For example, a recent study [63] uses 16 PZT-5H (483 mm^2) piezoelectric transducers placed in a circular frame mounted on the leg. It also uses 40 cube magnets (27 mm^3) in total to up-convert the walking frequency. Although the generated power is close to 6 mW, the design is bulky and lacks a flexible form factor, rendering the device impractical for wearable applications.

The advent of flexible piezoelectric materials, like polyvinylidene fluoride (PVDF) and macro fiber composite (MFC), has enabled genuinely wearable applications. Researchers have showcased the possibility of using flexible piezoelectric materials directly coupled to the body's joint movements (see table 2.1). One study placed PVDF beams at multiple body locations using a bodysuit with sewn-in pockets [41].

They measured the energy harvested at these locations during various physical activities like walking, jogging, and stair climbing. They noted higher energy outputs with higher frequency motion than lower frequency and reported 1.9 μW during walking and 3.7 μW during jogging. Similarly, another group of researchers developed a mechanical frame inspired by the human knee to investigate the dynamics of energy harvesting from human gait [44]. They examined the relationship between energy, motion frequency, and amplitude and found that harvested energy increased with both parameters. They extracted knee angle data from a motion capture dataset and used it in their framework, reporting around 1 μW generation from a knee harvester during walking.

The energy values reported in these studies may appear insignificant. However, it's important to note that these thin (thickness of less than 0.1 mm) materials can be physically stacked to increase the energy output. Furthermore, motion energy harvesters can achieve superior motion coupling to augment the harvested energy. New and novel piezoelectric materials are flexible, comfortable to wear, concealable under clothing, and can be incorporated into a sleeve, making them highly suitable for wearable applications. Compared to other EH modalities, this technology's analytical modeling challenges and higher material costs are significant drawbacks.

2.1.3 Radio Frequency Energy Harvesting

Radio frequency energy harvesting (RFEH) involves harnessing the energy from radio frequency waves disseminating through a medium. These RF waves may originate from communication links like GSM or TV towers, or specific dedicated

sources for energy transfer, as observed in wireless charging technologies. Typically, RFEH aims for the industrial, scientific, and medical (ISM) band, which ranges between 10 MHz and 25 GHz [65, 66, 67].

RFID tags and wireless charging are the two most prevalent examples of RF energy harvesting. RFID tags demonstrate far-field radiative energy transfer, wherein the energy dissipates with the inverse square of the distance $(1/d^2)$[68]. In contrast, wireless charging demonstrates near-field coupling, where power loss correlates with the inverse fourth power (or higher) of the distance[68]. While effective for short distances, typically less than a wavelength, the efficiency of this method diminishes considerably over greater distances. Therefore, far-field approaches, with power transfer efficiencies typically around 1

Both wireless charging and RFID tags represent RF energy harvesting with a dedicated source [67], meaning a known and often controllable energy source initiates the energy transfer chain. The alternative is ambient RF energy harvesting [67], where the energy source is unknown to the receiver, and the goal is to utilize whatever energy is present in the local medium. Hence, by its nature, ambient RF energy harvesting implies far-field radiative transfer, leading to specific design practices for the antenna, impedance matching network, and rectifier [67].

In the RFEH literature, designs are typically characterized and evaluated by sensitivity and power conversion efficiency [65]. Sensitivity refers to the minimum RF input power necessary to power a particular load, which depends on the specific application. Power conversion efficiency is the proportion of usable electrical output power to the power received by the RFEH circuit's antenna. It is typically reported

for dedicated source use cases, implying a known input power at the receiving antenna (>0 dBm). However, this is not likely in ambient RF energy harvesting scenarios. Thus, several spectral survey studies have assessed RF power availability at different frequency bands worldwide [47, 66, 69]. Most recently, [47] conducted a spectral survey of ambient RF power availability in various Boston locations, reporting an average power level of about -15 dBm, ranging from -25 dBm to -5 dBm. Consequently, for a practical RFEH application, studies should aim for very low power levels, typically less than -10 dBm.

Table 2.1 illustrates three recent studies in the literature that evaluate proposed energy harvesters at very low power levels. One study combines a PVDF piezoelectric energy harvester and an RF energy harvester within a single design, utilizing the PVDF's silver electrodes as a dipole antenna for RF harvesting [45]. The proposed harvester measures 153×12 mm^2, and a contribution of 7.3 μW harvested DC power from the RFEH part is reported at -10 dBm at 915 MHz. In another study, a slotted annular ring patch antenna with outer and inner diameters of 142 mm and 42 mm harvests slightly less than 4 μW of harvested DC power for -15 dBm at 1750 MHz [46]. As part of their spectral survey work, the authors in [47] utilized a 53×55 mm^2 50Ω LP0410 Log Periodic PBC antenna to harvest power, reporting 1 μW of harvested DC power for a -20 dBm input power level at 915 MHz. These reported values are contingent on the specific design considerations and optimizations implemented in each study. For instance, the chosen rectifier structure significantly influences efficiency and, consequently, the output DC power values.

While RFEH harvested power levels are lower than other energy harvesting modalities, RFEH possesses several advantages. First, ambient RF energy, emitted from many sources, provides a steady energy source with little fluctuation over time. This benefits energy management algorithms that require future harvested energy level predictions. Second, the harvester design exclusively focuses on the receiver, simplifying the energy transfer chain analysis and modeling compared to dedicated source harvesters. Nevertheless, RFEH comes with its challenges. Designing wearable antennas that deliver the same service quality as conventional rigid counterparts is a complex task due to factors like flexibility, conformity, safety, and performance requirements. Antenna parameters are difficult to predict because the antenna's bending, its placement on a flexible substrate, and proximity to human tissue influence these parameters in unpredictable ways [70]. Additionally, designing an effective flexible receiving antenna array requires a deep understanding of the field. Other challenges include designing an efficient impedance matching network for maximum power transfer between the antenna and rectifier, and rectifier design at radio frequencies. These challenges can lead to lower conversion efficiencies.

2.1.4 Heat Energy Harvesting

Thermoelectric energy harvesters (TEEH), also known as thermoelectric generators (TEG) in the literature, are solid-state devices that convert temperature differences across their two sides into electrical energy. Wearable thermoelectric energy harvesting exploits the temperature gradient between the human body and the external environment. The energy gradient is harnessed by a surface in contact with the

skin and a heatsink exposed to the air. Heat is transferred from the surface to the heatsink via thermoelectric (TE) elements, which transform the heat into electric energy.

The human body sustains a temperature around 37 C by balancing heat generation through metabolic processes and heat loss to the environment, ensuring the body's temperature remains consistent regardless of the ambient temperature. This process leads to the steady dissipation of approximately 100 W of heat from the human body on average [71]. However, TEEH devices positioned on the body can convert fraction of this power into electricity, primarily due to high thermal and contact resistances [71]. Despite the limited efficiency, wearable thermoelectric energy harvesting is of considerable interest due to the constant availability of energy throughout an individual's lifetime.

Reported power values harvested by TEEH devices display considerable variation in the literature. Usually, achieving a few tens of μW calls for large, rigid heatsinks to enhance the temperature gradient. However, flexibility and compactness are key characteristics for widespread adoption in wearable energy harvesting. To bypass the requirement for such heatsinks, recent research explores using thin nanostructured heatsinks and optimized heat spreaders [72]. Additionally, the air circulation around the harvester significantly influences the energy output of a TEEH [72]. Several studies have shown that rising air velocity leads to increased harvested energy. Therefore, substantial design effort is invested in establishing good thermal contact with air and facilitating airflow over the device.

Table 2.1 shows three recent studies focusing on flexible TEEH. One work pro-

posed a thermoelectric energy harvester that uses Bi_2Te_3 based thermoelectric material welded onto a flexible printed circuit board [48]. Their prototype consisted of eighteen p-n couples across a 42×30 mm^2 area. When worn on the wrist, this design generated 0.13 μW at an ambient temperature of 25 C, corresponding to a 12 C temperature differential. Under these conditions, the prototype generated 48 mV to power a standard COTS regulator like the LTC3107. Another study introduced a novel approach to creating a flexible thermoelectric energy harvester using lateral Y-type TE cells to improve the conventional vertical π-type cells [49]. They used unique methods to fabricate thick Bi_2Te_3 and Sb_2Te_3 films, creating a prototype measuring 1×1 cm^2. Their design showed a nearly $2.7\times$ greater temperature drop compared to the π-type structure, thus enhancing the energy harvesting performance of the device. Under a 22C temperature differential, their prototype generated 3 μW and 56 mV. Another work used COTS components to create a wrist-worn TEEH subsystem [40]. The design employed 7 TEGs in series, each connected to a rigid $14\times14\times6$ mm^3 heatsink to augment the heat gradient. With a temperature differential of 16C between the skin and environment, the design generated 3.05 μW of average power.

Thermoelectric energy-powered wearable devices are particularly appealing as they utilize body heat and are not dependent on external variables like movement, light, or electromagnetic waves. Hence, this modality can deliver more reliable electrical energy compared to other methods. However, the harvested energy values of TEEH solutions are currently too low for practical use. This limitation is primarily due to the decreased efficiency when incorporating the harvester into

a wearable design. Several factors contribute to this decrease, such as poor skin contact, the low-temperature gradient between the body and air, and performance variations due to airflow and device placement. Recent research effort is working on addressing these challenges [73, 74].

2.2 Optimal Energy Management Techniques

Energy harvesting requires algorithms that allocate the harvested energy to the device so that sufficient energy is available at all times of the day [25, 75]. For instance, light energy is typically available during the day. Thus, the device has to store sufficient energy for use during the night. Kansal et al. [25] present the general framework of energy-neutral operation for energy harvesting devices. Energy-neutral operation (ENO) ensures that the total energy consumed in a given period, such as a day, is equal to the energy harvested during the same period. To achieve this, the authors propose a linear programming approach that adapts the device's duty cycle under energy constraints. The work in [76] proposes a long-term energy management algorithm, referred to as long-term-ENO, to achieve energy neutrality for one year or more. Similarly, Geissdoerfer et al. [77] propose a feedback controller to achieve long-term-ENO. Another work proposes a low-complexity framework for runtime energy allocation for a wearable device with a known energy harvesting model [26]. A utility function expresses the quality of service (QoS), which considers the device's performance, power consumption, and network throughput. The framework maximizes the utility over a specific horizon under specific battery

energy constraints. Authors provide suboptimal closed-form solutions by relaxing one of the energy constraints, and by extensive experimental work, they show that the suboptimal results are within 5% of the optimal. Another paper proposes an energy-aware adaptive sampling algorithm for nodes with energy harvesting capability [78]. This algorithm decreases the sampling rate to decrease sensing and processing energy consumption when the battery percentage drops below a predefined level. The algorithm is evaluated using two distinct real deployment scenarios, and the results show that even with different sensors and energy sources in the two scenarios, self-sustainability can be achieved at the expense of classification accuracy due to low sampling rates. While these approaches are helpful, they do not consider the application requirements, user activity patterns, or the stochastic nature of energy harvesting.

The stochastic nature of energy harvesting has prompted researchers to investigate approaches that account for the stochasticity [79, 80, 81, 82]. These approaches typically model stochastic optimization as a Markov decision process and develop policies to perform energy management. Unfortunately, these approaches still do not consider the stochasticity arising from user activities. For example, the harvested energy changes significantly if the user moves indoors after being in the sun. User and context-aware approaches aim to address this issue by leveraging knowledge about the user's activities to reduce the power consumption of the device [83, 84]. These approaches focus only on power management and do not account for energy harvesting. Therefore, there is a need for approaches that account for variations in harvested energy due to environmental conditions and

user activities.

Reinforcement learning based approaches are suitable for energy management problems under uncertainty due to harvesting [85]. In one such study, the authors formulate an optimization problem to mitigate network interference in WBANs and increase the maximum achievable throughput with the minimum energy consumption [86]. The problem is solved by reinforcement learning, and the performance of the proposed algorithm is compared to two other previous works. The results show that the proposed approach provides savings in energy consumption per bit, and consequently, a substantial increase in network lifetime is achieved. Defining a suitable cost or reward function is an essential and challenging task in reinforcement learning literature. For this reason, the importance of the choice of reward function in reinforcement learning approaches for energy management in wearable applications is investigated [85]. After a series of simulations, the authors find out that a proper reward function should be able to relate energy consumption and performance to eliminate under- and over-utilization of energy.

Context or activity aware energy management is another subset in the EH-aware energy management literature. One such study proposes an algorithm that considers the QoS requirements of sensor nodes, the characteristics of channels between WBAN sensor nodes, and the different states and dimensions of the human body [87]. The algorithm solves an optimization problem for optimal transmission power and node distances to minimize the energy consumption of the nodes under QoS and power constraints. They do simulations for a scenario with eight sensors distributed over the human body and show that their proposed solution captures

the data rate and performance requirements while minimizing power consumption at each node. A similar work adapts a BAN node's power consumption according to the activity of the user [88]. This is achieved by a new classifier that detects the activity and sets the operating policy accordingly. In a typical scenario, the authors demonstrate a five times increase in the node's lifetime.

2.3 Energy Consumption Optimizations

Low-power design techniques have been studied due to their applications in energy-constrained devices. These methods can generally be categorized as either circuit-level or system-level optimizations [89, 90]. Circuit-level strategies scrutinize the logic in each module, inserting power/clock gating blocks to achieve power conservation. For instance, the influence of voltage distribution and DC-DC converter placement on power usage is examined in [89]. Blutman et al.[90] investigate subblock-level partitioning for stacked power domain architectures. Similarly, the method in[91] scrutinizes the high-level design description and generates detailed power domains. While circuit-level power optimizations are essential, designers often do not have control over individual sub-block specifics. Therefore, recent research has also delved into system-level techniques for low-power optimizations.

System-level techniques aim to optimize power consumption for a specific application or a domain of applications. These optimizations can occur at the software (SW) or hardware (HW) levels or consider both HW/SW co-optimizations. Cutting-edge wearable edge AI applications are typically implemented in SW on low-power

microcontroller units (MCUs). For instance, Samie et al. demonstrate an IMU-based human activity recognition (HAR) implementation on a TI CC1350 MCU (equipped with an Arm Cortex M3 core)[92], with average power and energy consumptions of 9.45 mW and 142.48 μJ per classification. Likewise, the authors in[93] implement a stress detection application that uses electrocardiography (ECG) and galvanic-skin-response data on a custom SoC with a RISC-V core. Another study introduces a prototype device that gathers and processes electroencephalography (EEG) signals to extract evoked potentials [94]. The device uses an STM32F412 MCU with an Arm Cortex M4 core, consuming approximately 18 mW of power. Although SW implementations are preferred due to their user-friendliness and fast prototyping, reducing the energy consumption of SW implementations without application-specific information can be challenging. For example, significant feature engineering and design space exploration is required to reduce data processing computations and overall SW execution time. Thus, the relatively high power consumption of SW implementations has prompted researchers to look for more energy-efficient hardware-based solutions.

Hardware techniques involve all-in-one ASIC implementations for applications or domain-specific system-on-chip (SoC) implementations. ASIC solutions offer significantly higher energy efficiency than SW running on general-purpose cores. For instance, the authors in [95] introduce a 30.6 μW dedicated HAR HW accelerator implemented with TSMC 65nm LP library. Similarly, another study showcases an ECG peak detector consuming approximately 30 μW power in ASIC [96]. Some implementations also use application-specific power gating [97, 98], but these

methods depend on manual optimization, which is impractical for large designs. Ideally, ASIC implementations consider each computation module and assign the voltage-frequency (VF) for these modules. For example, Wang et al. [99] employ an evolutionary algorithm to achieve VF partitioning of modules in a voice-over LTE application. The work in [100] suggests an iterative algorithm to perform VF partitioning for network-on-chip(NoC) architectures. However, the enhanced energy efficiency of ASIC designs compromises flexibility, restricting custom designs to a limited range of applications.

Domain-specific SoCs cater to a broad range of applications by merging general-purpose cores with a specialized set of HW accelerators. Utilizing domain-specific accelerators has emerged as a promising strategy to enhance energy efficiency and performance [101]. For example, recent studies have concentrated on accelerating the localization and perception functions in the autonomous driving domain, demonstrating up to 34$\times$ greater power efficiency than Intel i7 [102]. Other domains have demonstrated similar benefits, including speech recognition and deep learning [103]. However, despite the critical need for energy efficiency in this domain, no domain-specific HW platform exists for resource-constrained wearable edge devices.

3 WEARABLE PIEZOELECTRIC MOTION ENERGY HARVESTING

3.1 Motivation and Background

The physical flexibility of piezoelectricity-based energy harvesters (PEHs) relies on novel materials such as polyvinylidene fluoride (PVDF) and macro-fiber composite (MFC). Designs that incorporate these materials can be placed directly on human joints [41]. Very few studies investigated the potential of a wearable PEH modality despite its promising outlook [41, 44]. *As a result, a definitive exploration of this capacity is still absent from the literature.* A quantitative analysis of this topic would benefit hardware designers, system-level designers, and application developers. Furthermore, prior studies follow a purely experimental approach and *do not model the EH mechanism*. A model-based approach is critical for identifying dependencies to parameters in a given system and predicting harvested energy in real-life scenarios. These predictions are needed to enable smart energy management techniques to achieve energy-neutral operation [25, 31]. Therefore, there is also a strong need for modeling PEH directly from joint bending.

3.2 Contributions and Overall Goals

This chapter investigates the EH potential of MFC transducers placed on the knee during human gait to address the gaps in the literature, as shown in Figure 3.1. We focus on the knee joint due to three primary reasons: (1) The biomechanical energy potential at the knees is higher than other joints [104], (2) knees follow

a periodic trajectory, which allows for continuous harvesting of energy while the person wearing the device is walking, and (3) knee joint monitoring opens up a variety of possible applications, ranging from walking aid for patients to fatigue detection in athletes [105, 106]. To this end, we first present a novel and comprehensive analytical model for the EH output of a piezoelectric transducer directly placed on a joint. Then, we present an extensive experimental exploration that validates the proposed model. Our experimental results also demonstrate the EH potential of wearable knee joint-bending PEH for various gait speeds, including walking, jogging, and running. Our prototype PEH harvests 46 μW power from the knee joint while the user is walking and 161 μW while the user is jogging. These results are significantly higher than the EH capacity of other wearable PEHs in the literature [43, 42, 41, 44]. Hence, this study demonstrates the viability of a PEH device worn at the knee joint for low-power applications. For example, the produced power can sustain a variety of low-power wearable applications, ranging from human activity recognition [95] to cardiac arrhythmia detection [107].

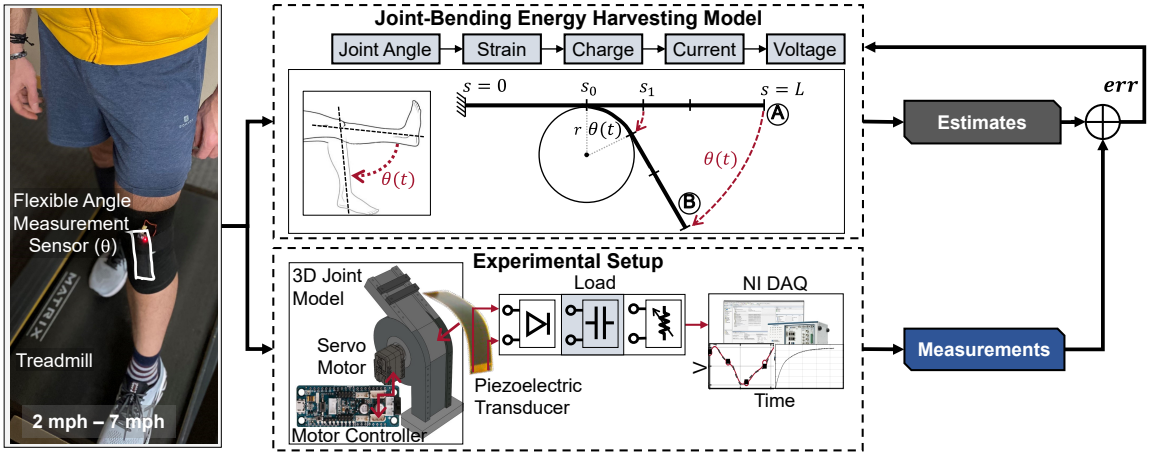


Figure 3.1: Overview of the modeling steps and experimental procedure.

In summary, the major contributions of this section are as follows:

- A closed-form expression for flexible joint-bending piezoelectric EH with less than 5% modeling error.
- A novel PEH prototype in the form of a knee sleeve capable of harvesting energy from the joint-bending movements of the knee during human gait.
- Experimental results that show up to 161 μW generation and 35-μA sustained constant current load at 1.8V at a 5 mph gait speed.

3.3 Joint-Bending Energy Harvesting Model

This section describes the proposed analytical model for EH from large-bending joint movements. Section 3.3.1 introduces key considerations for our proposed large-bending joint model and highlights the differential equation that governs the voltage generated by the piezoelectric transducer from our previous study [29]. Then, Section 3.3.2 extends the differential equation to a closed-form solution for the RMS voltage and power generated by a flexible PEH with a step-by-step derivation.

3.3.1 Joint Bending PEH Model Overview

We consider a flexible piezoelectric transducer placed directly on a joint, as shown in Figure 3.2. The mechanical forces due to joint bending acting on the piezoelectric material are much larger than the electrical forces induced by the piezoelectric

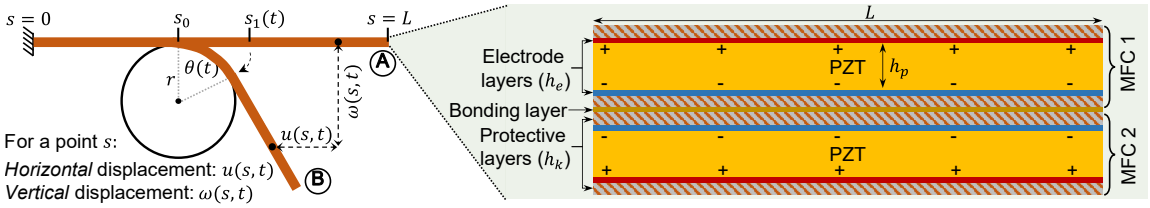


Figure 3.2: The beam-bending model and the bimorph piezoelectric transducer structure. Left) Large joint bending representation using a cantilever beam over a circular support. Right) The different layers in the bimorph piezoelectric transducer.

material. This observation allows us to decouple the mechanical and electrical domains and analyze them separately. First, in the mechanical domain, the physical displacement is modeled under *large-bending conditions*, which captures angles larger than 10°. We use this physical displacement to obtain the strain induced on the piezoelectric transducer. Then, in the electrical domain, the strain is converted to the generated current (and voltage) between the terminals of the piezoelectric transducer.

Mechanical Domain: To formulate the displacement, we use the large-bending representation of a cantilever beam over a circular support, as shown in Figure 3.2. The joint is modeled as circular support with radius r, while the piezoelectric transducer is a beam that bends along the circular support according to joint bend angle θ. The neutral position of the transducer is shown with the flat line $\mathbb{A}$. As the joint bends, the transducer moves from position $\mathbb{A}$ to $\mathbb{B}$, taking a curved form. In this illustration, positions $s = 0$ and $s = L$ show the transducer's clamped (i.e., fixed) and free ends, respectively. The point marked by $s = s_0$ is the first point that touches the circle, while $s = s_1$ is the last point that touches the circle. Using the arc length on the circular profile, we obtain $s_1(t) = s_0 + r\theta(t)$, where $\theta(t)$ is the

bending angle. Then, using $s_1(t)$, the proposed model expresses the horizontal displacement $u(s,t)$ of an arbitrary point s for $0 \leqslant s \leqslant L$ as:

$$u(s,t) = \begin{cases} 0, & 0 \leqslant s \leqslant s_0 \\ s_0 + r\sin\left(\frac{s-s_0}{r}\right) - s, & s_0 \leqslant s \leqslant s_1(t) \\ s_0 + r\sin\theta(t) + & \\ (s - s_1(t))\cos\theta(t) - s, & s_1(t) \leqslant s \leqslant L \end{cases} \tag{3.1}$$

The strain on the piezoelectric transducer is a function of its curvature. From the geometry of the problem, the curvature of the piezoelectric transducer is equal to the curvature of the circular support:

$$\kappa(s,t) = \begin{cases} \frac{1}{r}, & s_0 \leqslant s \leqslant s_1(t) \\ 0, & otherwise \end{cases} \tag{3.2}$$

Then, the average strain in the piezoelectric transducer from the large-bending representation is calculated as a function of horizontal displacement and curvature [108] as follows:

$$S_1(s,t) = -h_{pc}\kappa(s,t) + \frac{\partial u(s,t)}{\partial s} \tag{3.3}$$

where $h_{pc} = h_k + h_e + \frac{h_p}{2}$ is the distance to the neutral axis in the thickness direction of the piezoelectric transducer [109]. The piezoelectric transducers we use (MFC8528P2) can be modeled as composite beams with five layers, as shown

in Figure 3.2. They consist of a flexible substrate/protective layer at the outsides, electrode layers after that ((-)negative on one side, (+)negative on the other), and a piezoelectric layer between them. When two of these transducers are bonded together, the neutral axis falls precisely on the interface, and the amplitudes of the average strain on the piezoelectric layers on the two sides are equal. With this formulation, we ignore other acting forces on the piezoelectric transducer, such as the friction force. To satisfy this assumption, we design the experimental setup to exert minimal extra forces on the transducer, as explained in Section 3.5. Next, we use the average strain to obtain the generated electrical power by the piezoelectric transducer.

Electrical Domain: To formulate the generated power, we first obtain the accumulated charge in the piezoelectric material by applying the Gauss Law to the electric displacement field inside the material along the thickness axis:

$$Q(t) = \int_0^W \int_0^L \left(\epsilon_{33}^S E_3^S(t) + Y_p d_{31} S_1(s,t) \right) dW ds \tag{3.4}$$

where ϵ_{33}^S is the permittivity under constant strain, d_{31} is the electromechanical coupling coefficient, Y_p is the Young's Modulus of the piezoelectric layer, $E_3^S(t)$ is the electric field within the piezoelectric layer, and W and L denote the width and length of the piezoelectric patch, respectively. Since E_3^S is constant along the area and S_1 is constant along the width of the patch, the integrals simplify to:

$$Q(t) = \epsilon_{33}^S WL E_3^S(t) + Y_p d_{31} W \int_0^L S_1(s,t) ds \tag{3.5}$$

$E_3^S(t)$ is equal to the voltage divided by the thickness of the piezoelectric layer: $E_3^S(t) = -\frac{V(t)}{h_p}$. Substituting the electric field and strain expressions, the electric charge is found as:

$$Q(t) = -\frac{\epsilon_{33}^S WL}{h_p} V(t) + Y_p d_{31} W \int_0^L \left(-h_{pc} \kappa(s,t) + \frac{\partial u(s,t)}{\partial s} \right) ds \tag{3.6}$$

In Equation 3.6, the term $\frac{\epsilon_{33}^S WL}{h_p}$ is referred to as the capacitance of the patch [109], and is denoted by C from here on. To evaluate the integral in Equation 3.6, we consider the two terms inside the integral separately. Since $\kappa(s,t) = 0$ for $s < s_0$ and $s > s_1(t)$, the first term evaluates to:

$$\begin{aligned} \int_0^L -h_{pc} \kappa(s,t) ds &= \int_{s_0}^{s_1(t)} -h_{pc} \kappa(s,t) ds = \int_{s_0}^{s_0 + r\theta(t)} -h_{pc} \frac{1}{r} ds \\ &= -h_{pc} \theta(t) \end{aligned} \tag{3.7}$$

For the second term in Equation 3.6, since $u(s,t)$ is continuous at $s = s_0$ and $s = s_1(t)$, we can integrate it as:

$$\begin{aligned} \int_0^L \frac{\partial u(s,t)}{\partial s} ds &= u(L,t) - u(0,t) \\ &= u(L,t) \end{aligned} \tag{3.8}$$

With these, Equation 3.6 reduces to:

$$Q(t) = -CV(t) + Y_p d_{31} W \left(-h_{pc} \theta(t) + u(L,t) \right) \tag{3.9}$$

Next, we take the time derivative of $Q(t)$ to obtain the electrical current generated by the piezoelectric patch:

$$\frac{dQ(t)}{dt} = -C\frac{dV(t)}{dt} + Y_p d_{31} W \left(-h_{pc}\frac{d\theta(t)}{dt} + \frac{\partial u(L,t)}{\partial t} \right) \tag{3.10}$$

We obtain $\frac{\partial u(L,t)}{\partial t} = (s_1(t) - L)\frac{d\theta(t)}{dt} sin\theta(t)$ from Equation 3.1. Then, from $V(t) = \frac{dQ(t)}{dt} R_L$, we obtain the differential equation that governs *the voltage generated by the piezoelectric transducer* as follows:

$$\begin{aligned} \frac{dV(t)}{dt} = \frac{-V(t)}{R_L C} - \frac{Y_p d_{31} W}{C} \Bigg[& h_{pc}\frac{d\theta(t)}{dt} \\ & - (s_1(t) - L)\frac{d\theta(t)}{dt} sin\theta(t) \Bigg] \end{aligned} \tag{3.11}$$

where R_L is the load connected to the piezoelectric transducer. The parameters and variables used in these equations are summarized in Table 3.1.

3.3.2 Analytical Closed-form Solution

Equation 3.11 is an ordinary differential equation. There are three main benefits of finding an analytical closed-form solution over the differential equation:

1. Analytical solutions are better than differential equations at providing useful insights into both the dependency on parameters and the relative importance of the parameters,

2. They are more reliable than differential equations since they do not suffer from numerical instabilities,

Table 3.1: Symbols and Parameters used throughout Section 3.

Symbol	Parameter	Symbol	Parameter
V	Generated voltage	Q	Accumulated charge
S_1	Axial strain	Y_p	PZ* Young's modulus
ω	Vert. displacement	L	PZ length
u	Horz. displacement	W	PZ width
κ	Curvature	C	PZ capacitance
θ	Joint bending angle	R_L	PZ load
r	Radius of curvature	h_p	PZ thickness
E_3^S	Electric field within PZ	h_k	Protective layer thickness
ϵ_{33}^S	Permittivity under constant strain	h_e	Electrode layer thickness
d_{31}	Electromechanical coupling coeff. [108]	h_{pc}	Distance to neutral axis

*PZ: Piezoelectric Transducer

3. They require significantly lower computational effort for parameter sweeps and fitting than numerical approaches.

In addition, in the EH context, an analytical solution for the power output of the system can be used to develop novel maximum power point tracking algorithms. Thus, this section provides the derivation of an analytical closed-form solution for the RMS power generated by the piezoelectric transducer.

We solve Equation 3.11 to obtain the expression for $V(t)$ as:

$$V(t) = \frac{Y_p d_{31} W}{C} e^{\frac{-t}{RC}} \int_{\tau=0}^{t} \left(-h_{pc} \frac{d\theta(t)}{dt} + \sin\theta(t) \left(s_1(t) - L\right) \frac{d\theta(t)}{dt} \right) d\tau \tag{3.12}$$

A closed-form solution for the integral in Equation 3.12 does not exist. To overcome

this obstacle, we assume that the motion is periodic (i.e., θ is a periodic waveform). We validate this assumption in Section 3.5 by illustrating knee angle recordings at different gait speeds. Then, we use the Fourier series expansion of $\theta(t)$:

$$s_\theta(t) = a_0 + \sum_{k=1}^{K} [a_k cos kt + b_k sin kt]$$

It follows that $\frac{ds_\theta(t)}{dt} = \sum_{k=1}^{K} [b_k cos kt - a_k sin kt]$. Similarly, we use the Fourier series expansion of the second term in the integral $g(t) = sin\theta(t)\,(s_1(t) - L)\,\frac{d\theta(t)}{dt}$ in Equation 3.12:

$$s_g(t) = a_0^* + \sum_{m=1}^{M} [a_m cos mt + b_m sin mt]$$

Here, K and M denote the number of Fourier components. Next, we plug these Fourier series expansions of the two components into the integral in Equation 3.12 and evaluate the integral. After some algebra and simplifications, we obtain the following expression:

$$\begin{aligned} V(t) = Y_p d_{31} W \Bigg[& -h_{pc} \sum_{k=1}^{K} \frac{\alpha_k cos kt + \beta_k sin kt + \gamma_k e^{\frac{-t}{R_L C}}}{\frac{1}{R_L^2} + C^2 k^2} \\ & + (-a_0^* R_L)(e^{\frac{-t}{R_L C}} - 1) \\ & + \sum_{m=1}^{M} \frac{\alpha_m cos mt + \beta_m sin mt + \gamma_m e^{\frac{-t}{R_L C}}}{\frac{1}{R_L^2} + C^2 m^2} \Bigg] \end{aligned} \tag{3.13}$$

where α_k, β_k, γ_k, α_m, β_m, and γ_m are given in Table 3.2.

We can further simplify Equation 3.13 by eliminating the transient terms (i.e.,

by taking the limit $\lim_{t\to\infty} V(t)$) and obtain the steady-state voltage $V_{ss}(t)$ as:

$$V_{ss}(t) = Y_p d_{31} W \left[a_0^* R_L - h_{pc} \sum_{k=1}^{K} \frac{\alpha_k \cos kt + \beta_k \sin kt}{\frac{1}{R_L^2} + C^2 k^2} + \sum_{m=1}^{M} \frac{\alpha_m \cos mt + \beta_m \sin mt}{\frac{1}{R_L^2} + C^2 m^2} \right] \tag{3.14}$$

Equation 3.14 is a sum of sinusoids with a DC offset in the following form: $f(t) = F_0 + F_1 \cos(\omega_1 t + \Theta_1) + ... + F_n \cos(\omega_n t + \Theta_n)$. The RMS value of such a function is given by: $F_{RMS} = \sqrt{F_0^2 + \frac{F_1^2}{2} + ... + \frac{F_n^2}{2}}$. Using this expression with Equation 3.14, we calculate the RMS voltage V_{RMS} generated by the piezoelectric transducer, as follows:

$$V_{RMS} = Y_p d_{31} W \sqrt{a_0^{*2} {R_L}^2 + \frac{{h_{pc}}^2}{2} \sum_{k=1}^{K} \frac{{\alpha_k}^2 + {\beta_k}^2}{\left(\frac{1}{R_L^2} + C^2 k^2\right)^2} + \frac{1}{2} \sum_{m=1}^{M} \frac{{\alpha_m}^2 + {\beta_m}^2}{\left(\frac{1}{R_L^2} + C^2 m^2\right)^2}} \tag{3.15}$$

Finding the Generated Power: Finally, RMS power harvested by the transducer is

Table 3.2: Coefficients used in Equation 3.13 and onwards.

$\alpha_k = a_k C k^2 + \frac{b_k k}{R_L}$	$\alpha_m = \frac{a_m}{R_L} - b_m C m$
$\beta_k = b_k C k^2 - \frac{a_k k}{R_L}$	$\beta_m = \frac{b_m}{R_L} - a_m C m$
$\gamma_k = -a_k C k^2 - \frac{b_k k}{R_L}$	$\gamma_m = -\frac{a_m}{R_L} + b_m C m$

given by $P_{RMS} = \frac{{V_{RMS}}^2}{R_L}$:

$$P_{RMS} = \frac{(Y_p d_{31} W)^2}{R_L} \left({\alpha_0^*}^2 {R_L}^2 + \frac{{h_{pc}}^2}{2} \sum_{k=1}^{K} \frac{{\alpha_k}^2 + {\beta_k}^2}{\left(\frac{1}{R_L^2} + C^2 k^2\right)^2} + \frac{1}{2} \sum_{m=1}^{M} \frac{{\alpha_m}^2 + {\beta_m}^2}{\left(\frac{1}{R_L^2} + C^2 m^2\right)^2} \right) f \tag{3.16}$$

where f is a fitting parameter to adjust the model response to experimental measurements. We justify the need for it for three main reasons: (1) The neglected change in the parameter h_{pc} due to the epoxy layer for bonding, (2) a possible slight misalignment between the two transducers, and (3) the additive uncertainty in the capacitance of each piezoelectric transducer. The value for f is found for a given transducer with the help of experimental measurements. In the present study, we calculate it such that the normalized mean squared error between the experiments and model outputs are minimized:

$$\textit{Find } f \textit{ such that } \min_{f} \sum_{R_L} \frac{||P_{ref} - \hat{P}_{RMS}(f)||^2}{||P_{ref} - \frac{1}{N}\Sigma_{n=1}^{N} P_{ref}||^2} \tag{3.17}$$

where P_{ref} is the experimental RMS power, $\hat{P}_{RMS}$ is the model RMS power output given by Equation 3.16 and N is the number of samples in the experimental recording.

3.4 Experimental Evaluation of Joint Bending EH

This section first introduces the experimental setup and procedure used in our evaluations. Then, we present the generated power by the flexible piezoelectric transducers. Finally, we provide energy measurements at the input and output (load) sides of the energy harvester.

3.4.1 Experimental Setup and Procedure

Obtaining reproducible results is critical in a validation study. For this reason, we designed a mechanical robot frame that can mimic the bending motion of the knee joint, as shown in Figure 3.3(a). To use as input for the robot, we recorded the

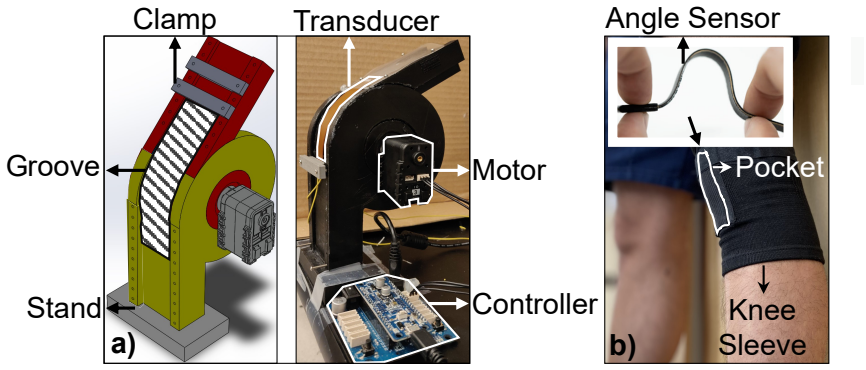


Figure 3.3: The robot frame and the flexible angle sensor. a) The 3D printed robot frame to mimic the knee bending motion. b) The flexible angle measurement sensor and the knee sleeve to collect knee bending angle during gait.

knee angle at six gait speeds, ranging from 2 mph to 7 mph with 1 mph increments. To obtain these recordings, we used a *BendLabs* flexible angle measurement sensor [110] placed inside the pocket on a knee sleeve worn by a subject, as shown in Figure 3.3(b). As an example, Figure 3.4(1a-1c) presents the data for three of

these recordings. Thus, these recordings enabled the robot frame to mimic the knee bending motion as shown in Figure 3.5(a).

To harvest the motion energy, we place the MFC8528P2 flexible piezoelectric transducers [111] on the robot frame. Figure 3.5 summarizes the steps in our experimental evaluation. In Section 3.4.2, to measure the generated power, we first connect the transducers to a variable resistance box and record the voltage on the resistor, as illustrated in Figure 3.4(2a-2c). This section also presents the modeling results. Then, we connect the transducers to an energy harvester built around the LTC3331 power-management IC, developed by Linear Technology [112]. Section 3.4.3 presents our measurement results regarding the amount of harvested energy after voltage rectification. Finally, Section 3.4.4 presents the amount of energy delivered to a resistive load after losses due to voltage rectification and regulation. Figure 3.5(c) shows the specific measurement points within the LTC3331-based

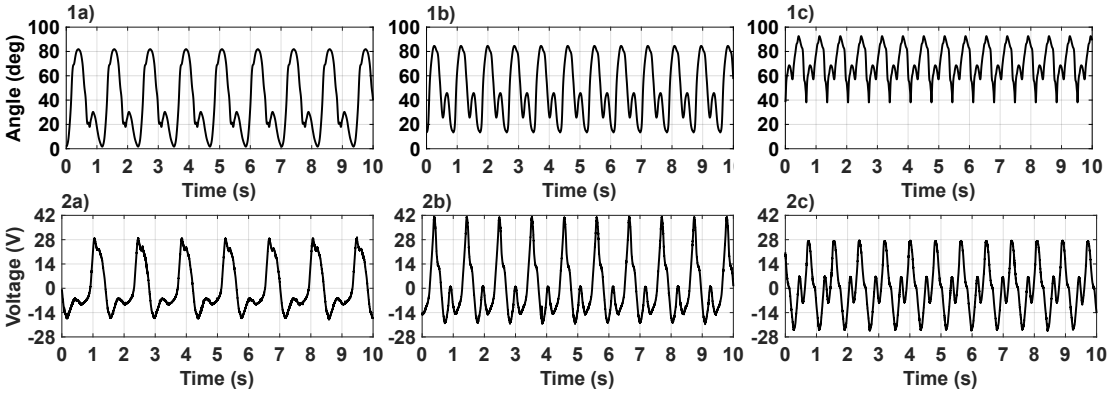


Figure 3.4: Knee angle and voltage recordings for different gait speeds. First row: Knee angle recordings collected with the *BendLabs* sensor at 1a) 3 mph, 1b) 5 mph, and 1c) 7 mph. Second row: The corresponding voltage recordings from parallel MFC8528P2s at 2a) 3 mph, 2b) 5 mph, and 2c) 7 mph.

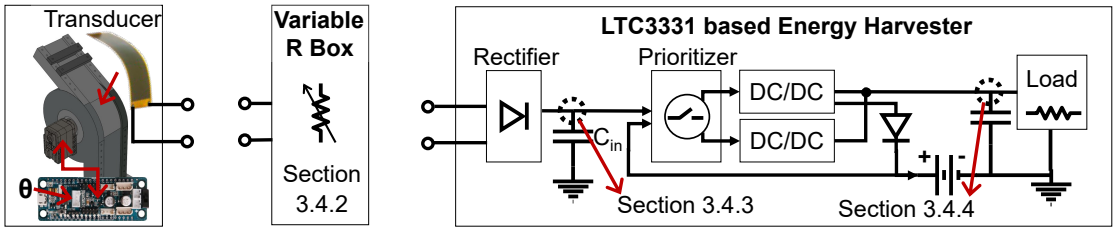


Figure 3.5: Overview of our experimental evaluation steps. a) Input actuation and transducer placement, b) the measurement point in Section 3.4.2, c) the measurement points in Sections 3.4.3 and 3.4.4

energy harvester used for the steps briefly described above for Section 3.4.3 and Section 3.4.4.

3.4.2 Generated Power by Piezoelectric Transducers

This section presents the power generated by the piezoelectric transducer when it is terminated by pure resistance (i.e., when the LTC3331-based energy harvester is not connected). To this end, we connect the terminals of the piezoelectric transducer to a variable resistor, and sweep the load resistance R_L from 50kΩ to 1.5MΩ. We connect a 1 Ω current-sensing resistor in series to R_L, and record the voltage on this sensing resistance at 1 kS/s for 25 seconds with an NI DAQ (Digital Acquisition) system. We discard the data from the first 5 seconds, and obtain the RMS current I_{RMS} delivered to the load resistor from the remaining 20-second recording. Then, we simply calculate the RMS power P_{RMS} by using the following expression: $P_{RMS} = {I_{RMS}}^2 R_L$.

To analyze the effect of connection topology on generated power, we experiment with two configurations: (1) two bonded MFC8528P2 transducers connected in *series*, and (2) two bonded MFC8528P2 transducers connected in *parallel*. The transducers are bonded into a bimorph with an epoxy layer in the middle. The layer

thicknesses and values for other experimental parameters are given in Table 3.3. For the series configuration, we connect the (+) terminal of the first unimorph to the (-) terminal of the second unimorph, and expose the (-) terminal of the first unimorph and (+) terminal of the second unimorph as the output terminals. For the parallel configuration, we connect the (-) terminals of both unimorphs to each other, and expose it as the negative output. Similarly, we do the same for the (+) terminals and expose it as the positive output. This way, we guarantee that the voltages do not add up destructively.

Table 3.3: Parameter values used in the experiments.

Symbol	**Value**	**Symbol**	**Value**
L, W (mm)	85, 28	Yp (Gpa)	30.336
h_p (μm)	180	C (nF)	264$\pm$20%
h_e (μm)	18	d31 (pC/N)	-170
h_k (μm)	40	K, M	100
h_{pc} (μm)	148	R_L (Ω)	50k-1.5M
s0 (cm)	1.3	r (cm)	4

For each gait speed, we obtain the modeling results from Equation 3.16. The value of the fitting parameter f for each case is given in the Appendix. In what follows, we illustrate the results for the 3 mph, 5 mph, and 7 mph gait speeds, which correspond to walking, jogging and running for an average person. *However, we emphasize that the reported results for modeling accuracy are from all gait speeds.*

Series Topology: Figure 3.6(a) shows the P-V curves for three gait speeds when *two bonded transducers in series* serve as the transducer. The analytical results and the

experimental measurements are in close agreement with each other. Specifically, the MAPE over all gait speeds (2–7 mph) is 2.1%.

One particular observation from the above results is that the MPP of the 5 mph curve is higher than that of the 7 mph curve, as depicted in Figure 3.6(b). Specifically, the maximum harvested power is close to 250 μW at 5 mph, while it is close to 150 μW at 7 mph. We can understand the root cause of this behavior by analyzing the knee angle waveforms in Figure 3.4(1a-1c). The range of the 5 mph waveform (15°– 85°) is wider than the range of the 7 mph waveform (40°– 90°). In addition, the 5 mph waveform has a shorter period than the 3 mph waveform. In other words, the former waveform repeats itself faster than the latter. These two effects, when combined, give rise to the sweet spot presented in Figure 3.6(b).

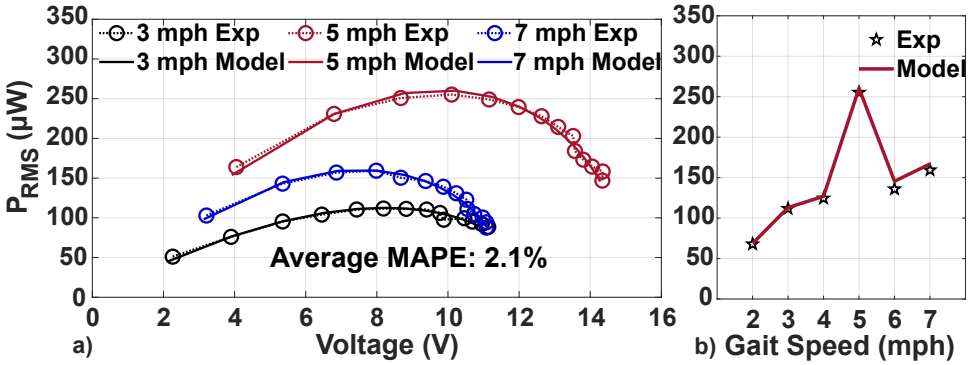


Figure 3.6: Bonded piezoelectric transducers *in series*. a) P_{RMS} vs. voltage. b) P_{RMS} vs. gait speed.

Parallel Topology: Figure 3.7(a) shows the P-V curves for three gait speeds when *two bonded transducers in parallel* serve as the transducer. The model results and the experimental measurements in this case are again in close agreement. Specifically, the average MAPE over all gait speeds is 3.2%, and again similar to the previous

case, the maximum power point of the 5 mph curve is higher than that of the 3 and 7 mph curves as shown in Figure 3.7(b).

With the parallel configuration, the peak power for all gait speeds is greater than or equal to the corresponding power with the series configuration. However, the peaks in this case are at lower voltage levels than in the series configuration. For example, the maximum power point of the 5 mph curve is at 5.5 V in the parallel configuration, whereas it is at 9 V in the series configuration. The efficiency of the buck converter is higher at low input voltages than at high input voltages [112]. *Thus, we use the parallel configuration in the rest of the experimental evaluations.*

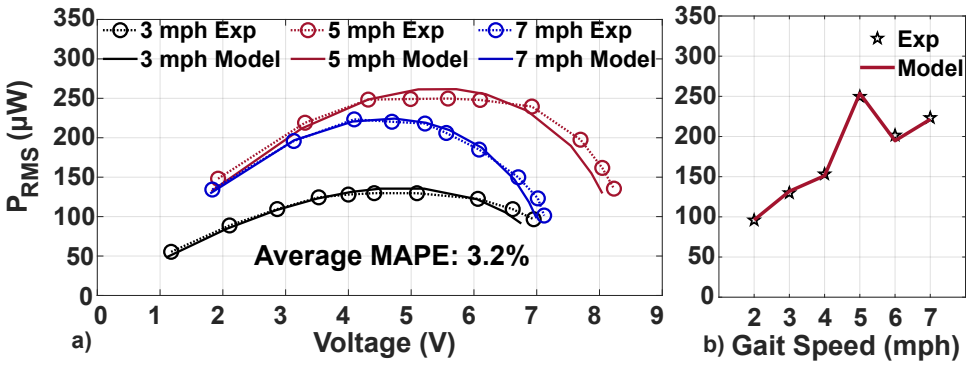


Figure 3.7: Bonded piezoelectric transducers *in parallel*. a) P_{RMS} vs. voltage. b) P_{RMS} vs. gait speed.

3.4.3 Harvested Energy on C_{in}

In this section, we discuss the energy harvested at the input side of the LTC3331-based energy harvester, as shown in Figure 3.5(c). To this end, we utilize the full-wave bridge rectifier inside the LTC3331, which has an approximately 800-mV drop, and we connect a high-dielectric 22-μF capacitor to the output of the rectifier

(i.e., pin V_{IN} of LTC3331). Then, we measure the voltage waveform on the capacitor at several gait speeds, as shown in Figure 3.8. The details of LTC3331 operation principles are presented in Section A.3 for the interested reader.

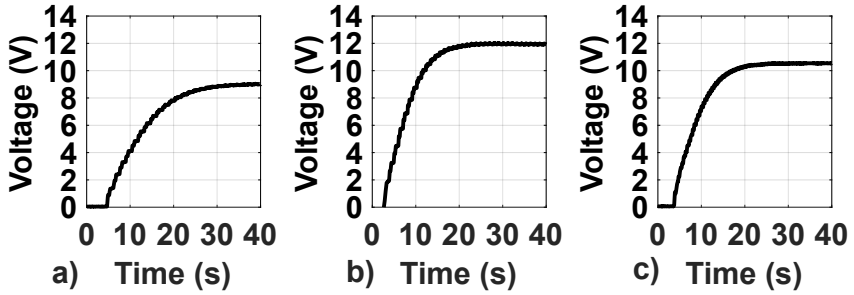


Figure 3.8: Voltage on C_{in} for a) 3 mph, b) 5 mph, and c) 7 mph

Operating at the Maximum Power Point: The harvester needs a suitable under voltage-lockout (UVLO) configuration to ideally operate at the maximum power point. Previous studies in the literature suggest setting the UVLO configuration to half of the rectified voltage level on the input capacitor so that the harvester can operate at the maximum power point [113]. Accordingly, the voltage levels in Figure 3.8 suggest that for the 3 mph condition, the optimal UVLO window configuration is 4–5 V. Similarly, for the 5 mph and 7 mph conditions, the optimal configuration is 5–6 V. To verify these settings empirically, we sweep the UVLO window configurations, as summarized in Table 3.4. For each configuration, we record the energy on the input capacitor, as shown in Figure 3.9. The behavior in this figure is stemming from the UVLO configuration of the LTC3331. In this particular case, the peaks correspond to 6V (converted to energy using $\frac{1}{2}CV^2$) and the dips correspond to 5V. The difference between a dip and a peak is the ΔE_i, and $\frac{\Delta E_i}{\Delta t_i}$ gives the average power for the i^{th} cycle. We average the power in several of

these cycles from a 30-second recording to obtain an empirical average power figure P_{avg}, as presented in Table 3.4. According to these results, the UVLO window configurations that correspond to the maximum power point are consistent with the theory. Therefore, in the rest of our evaluations, we use 4–5 V for the 3 mph condition, and 5–6 V for the 5 mph and 7 mph conditions.

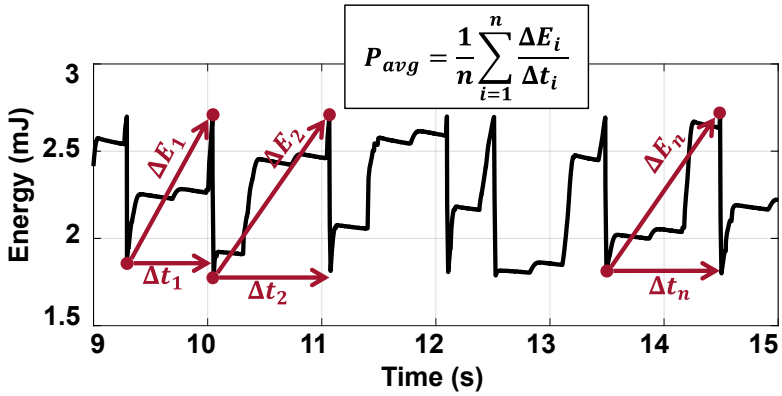


Figure 3.9: The energy stored on C_{in} when UVLO window is set to 5–6 V at 5 mph.

The harvested power obtained at 5 mph (161 µW) is approximately 35% smaller than the power measurement shown in Figure 3.7. Similarly, the values obtained for the 3 mph and 7 mph conditions are considerably smaller than the corresponding power measurements in Figure 3.7. There are three main reasons for this difference:

Table 3.4: P_{avg} with various UVLO window configurations.

	UVLO Configuration					
	3-4 V	**4-5 V**	**5-6 V**	**6-7 V**	**7-8 V**	**9-10 V**
3 mph	39 µW	**46 µW**	39 µW	33 µW	—	—
5 mph	100 µW	123 µW	**161 µW**	142 µW	151 µW	132 µW
7 mph	88 µW	84 µW	**115 µW**	94 µW	94 µW	—

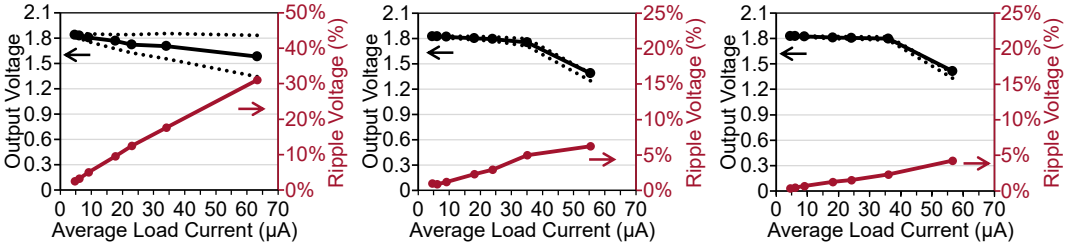
*__Bold__ typeface shows the maximum value for each row.

(1) The losses due to the integrated bridge rectifier, (2) the 22-μF high-dielectric capacitor possessing approximately 20% capacitance degradation at 5 V, and (3) the purely resistive nature of the load in Figure 3.7 versus the capacitive nature of the input impedance of the rectifier. When combined with capacitor leakage current, these three factors lead to the relatively low power values obtained by the LTC3331-based energy harvester in this section.

3.4.4 Harvested Energy on C_{out}

This section discusses the capacity of the energy harvester to power various load levels, as shown in Figure 3.5(c). We use the 5 mph condition as the input actuation and set the UVLO window to 5—6 V as noted in Table 3.4. We configure the LTC3331 to regulate the output voltage at 1.8 V with three capacitance values: 150 μF, 620 μF, and 1150 μF. The load is a variable resistor that we sweep from 25 kΩ to 400 kΩ. For each load value, the output voltage is recorded for 20 seconds. From these recordings, we calculate the average output voltage and the percent ripple around these average values, as summarized in Figure 3.10. At high load-resistance values, the current drawn by the load is small and the output is easily regulated. At lower load-resistance values, the current and the ripple in the output voltage increase. For example, the output voltage cannot be regulated at 1.8 V with the 150-μF configuration if the load draws more than 10 μA. Specifically, the output voltage goes below 1.75 V, and the percent ripple exceeds 5%.

Increasing the capacitance has two main effects on the output voltage. First, it increases the load's turn-on time (i.e., the time it takes until regulation). At



Cap. (μF)	150	620	1150
Turn-on (s)	8.84	26.37	40.56
Ripple (mV) @ 35 μA	300	86	41

Figure 3.10: Output voltage and ripple magnitudes for several output capacitance configurations.

150 μF, the output voltage is regulated in 8.84 s, whereas at 1150 μF, this time increases to 40.56 s. Second, the ripple in the output voltage decreases with higher capacitance. For an average load current of 35 μA, the 150-μF configuration has a 300-mV ripple amplitude. In comparison, the 1150-μF configuration obtains 41 mV for the same load current. Therefore, designers should choose the output capacitance values depending on the application. For instance, if the load is not too sensitive to voltage ripples and has a broad input voltage range (e.g., 1.6 V − 1.8 V), the 150-μF configuration can sustain up to 35-μA load current at 1.7 V with an approximately 20% ripple and an 8.84-s turn-on time. If the load is sensitive, the 620-μF configuration offers a ten-fold reduction in ripple voltages at the expense of a three-fold increase in turn-on times.

3.5 Exploration of the Daily EH Potential

The power measurements in the previous section do not translate directly to energy harvesting potential in a day. For this, we need to know the activity information for a user in a day. For example, if we know the amount of time a user walks, runs and sits in a day, we can give a rough estimate of the energy harvesting potential using the power measurements in the previous chapter. In addition to this, a realistic wearable harvester should employ multiple modalities to provide uninterrupted power. To this end, this section investigates the energy harvesting potential of a person using a combined motion and light energy harvester. For the motion part, we use the power measurements in the previous section. For the light energy harvester, we characterize the EH potential of a wearable light energy harvester using flexible PV-Cells.

3.5.1 Multi-Modal Energy Harvesting System

We use two distinct subsystems for light and motion EH modalities, as shown in Figure 3.11. This setup allows for having control over operating at the maximum power point for each modality separately. This section presents the operation principles of each energy harvesting system and their combination.

3.5.1.1 Motion and Light Energy Harvesting Subsystems

Figure 3.11a illustrates the main components and the operation principle of the motion EH subsystem. The charge generated by the piezoelectric element is rectified

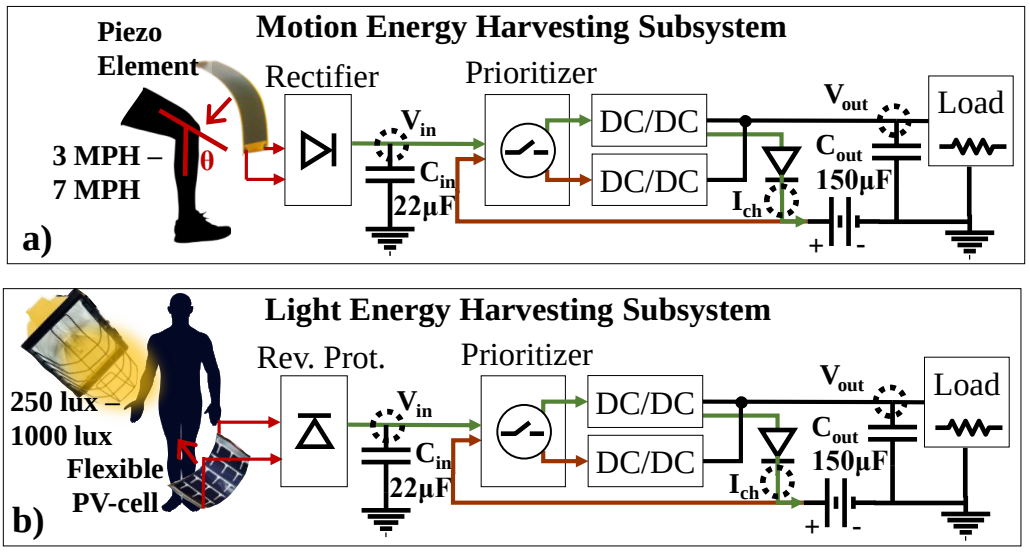


Figure 3.11: Energy harvesting subsystems. a) Motion Energy Harvesting Subsystem b) Light Energy Harvesting Subsystem.

and stored on an input capacitance. Once the voltage reaches a programmable upper threshold, a prioritizer module transfers the accumulated charge to an output capacitance through a buck converter. When the voltage on the input capacitance decreases below a lower threshold during the transfer, the buck converter is disabled. Then, the harvested energy starts accumulating again on the input capacitance. This process repeats until a regulated voltage at the output is obtained, as shown in Figure 3.12. In this example, the lower and upper thresholds are set to 5 V and 6 V respectively, as depicted in Figure 3.12a. While the output voltage is regulated, the harvested energy is transferred to the battery, as shown in Figure 3.12c. If the output regulation cannot be maintained by harvested energy, the prioritizer regulates the output using a backup battery through another buck converter.

The operation principles of the light EH subsystem shown in Figure 3.11b is the

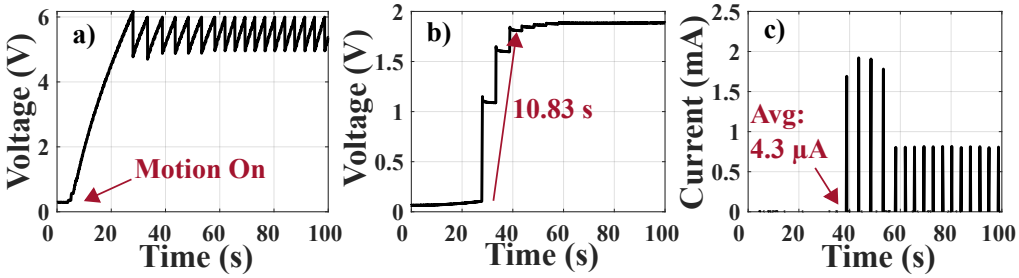


Figure 3.12: Motion subsystem operation at 7 MPH gait speed. a) V_{in} b) V_{out} c) I_{ch}

same with the motion EH subsystem. Figure 3.13 depicts the typical behavior of this subsystem under 250 lux illuminance with two PV-cells in series. In this case, the lower and upper thresholds are set to 4 V and 5 V, respectively.

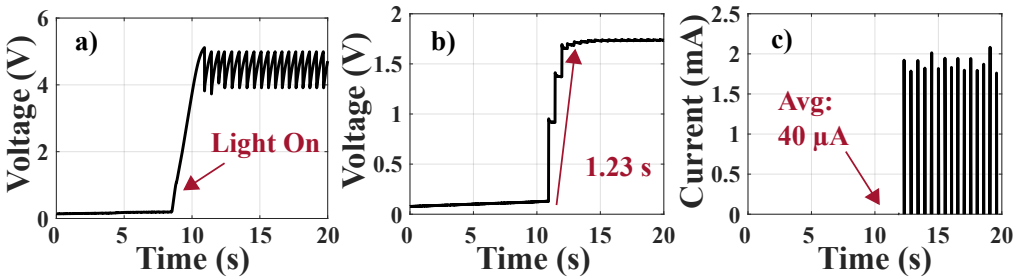


Figure 3.13: Light subsystem operation at 250 lux. a) V_{in} b) V_{out} c) I_{ch}

3.5.1.2 Combined Energy Harvesting System

Multiple transducers and EH modalities can be combined using different topologies. One extreme option is to rectify and regulate the voltage produced by each transducer separately, and then add them up. Although this is the most straightforward approach, it incurs extra losses due to additional rectifiers and regulators. At the other extreme, the harvested voltage from all transducers can be first superimposed before rectifying and regulating them altogether. This approach uses

one rectifier and one regulator, but the voltages may add up destructively due to synchronization issues.

In this work, we superimpose the AC voltages generated by the piezoelectric elements and rectify/regulate the superimposed voltage using the subsystem shown in Figure 3.11a. The same approach is applied to the superimposed DC voltage generated by the PV-cells with the subsystem shown in Figure 3.11b. Finally, we combine the two subsystems through diode OR-ing, as shown in Figure 3.14. In this configuration, the regulated output at the output of both subsystems connect to a large capacitance. This approach enables each subsystem to operate at its MPP. In addition, the subsystems are connected to the same battery. This way, both subsystems start charging the battery once the system achieves regulated output.

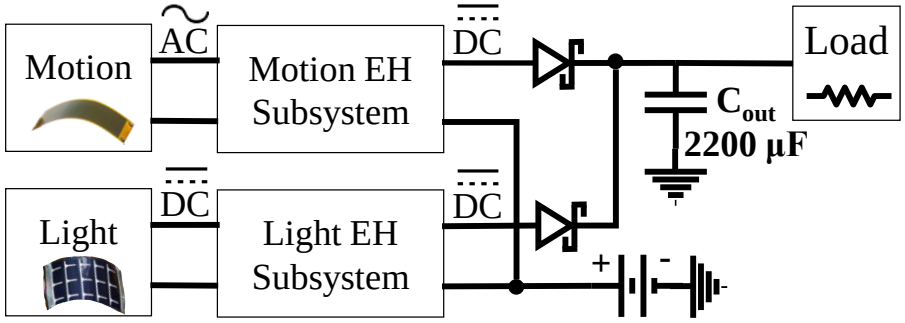


Figure 3.14: The topology for combining the motion and light energy harvesting subsystems.

We implement the motion and light energy harvesting subsystems with two LTC3331 power management ICs by Linear Technology [112], as described in Section A.3. The programmable upper and lower thresholds are critical factors for the energy harvesting performance, as they control the operation at the maximum

power point of the corresponding modality. We set these thresholds according to Table 3.5.

The subsystems connect to a common 3.6 V lithium-ion rechargeable battery, such that each LTC3331 charges the battery with its own harvested energy. For this, the "BAT_IN" pins of LTC3331s are tied to the +ve terminal of the battery. This way, each LTC3331 controls the charge on its input capacitance and sends the excess charge into the battery after the output is in regulation.

Experimental Procedure: The motion energy harvesting subsystem is thoroughly demonstrated in the previous section using two bonded piezoelectric transducers. In this section, we characterize the light energy harvesting subsystem. In the light energy harvesting subsystem, the energy source is flexible and small form-factor (2336 mm^2) PowerFilmSolar SP3-37 PV-cells [114]. The PV-cells are placed on a cylindrical frame with a 3 cm radius of curvature to imitate placement on the leg. We measure the generated current by the light EH subsystem in indoor office (250 lux) and outdoor conditions (1075 lux), as specified in [115]. For office conditions, we place the cylindrical frame on a desk that receives an illuminance of 250 lux in a typical laboratory setting. In a realistic scenario, the PV-cells do not always face the light source. To account for this, we measured the current when the PV-cells are

Table 3.5: Light and Motion EH subsystem hysteresis window configurations to operate at the MPP of each configuration.

Light EH		**Motion EH**	
250 lux	4-5 V	**3 MPH**	4-5 V
1075 lux	5-6 V	**5 MPH**	5-6 V
		7 MPH	5-6 V

turned *towards* and *away* from the light source. For outdoor conditions, we place a 500 W halogen work light with adjustable light intensity directly across the PV-cells. Then, we adjust the light intensity to read 1075 lux on a commercial lux-meter at the level of the PV-cells.

In the following, we measure the current flowing into the battery when the load is an open circuit for all different light configurations. Since the load is open, the harvested energy is used to charge the battery as soon as the output is regulated at 1.8 V. Finally, we combine the measurements with the ATUS dataset to explore energy harvesting potentials of 9593 users over one day.

3.5.2 Combined EH System Measurements

The measurements for the motion EH subsystem are summarized in Table 3.4 in the previous section. For the light subsystem, the hysteresis window of the LTC3331 is configured according to Table 3.5 depending on the illumination. When two PV-cells are used in series, the output can be regulated at 1.8 V under 250 lux, as depicted in Figure 3.13. When the PV-cells are turned away from the light source, the battery is charged with 24.5 μW average power. When they face the light source, the average power increases to 144 μW. Finally, the average power is 511 μW under 1075 lux, which is significantly more than any other configuration.

The combined system uses two PV-cells and two bonded transducers. We evaluate the performance of the systems at all six different combinations of light and gait speed conditions (i.e., 250 lux - 3 MPH to 1075 lux - 7 MPH), as shown in Table 3.7. According to our measurements, the combined system successfully

Table 3.6: Light EH subsystem measurements.

	2 PV-Cells in Series
250 lux (away)	24.5 μW
250 lux (towards)	144 μW
1075 lux	511 μW

adds up the battery charging currents of the individual subsystems. For example, under an illuminance of 1075 lux and a speed of 5 MPH, the light and motion EH subsystems harvest 511 μW and 161 μW, respectively. The combined EH system harvests 672 μW under the same conditions. This potential can be used to power a variety of sensors and low-power components, as listed in [116].

Table 3.7: Combined EH system measurements.

	Gait Speed		
Light (lux) ↓	**3 MPH**	**5 MPH**	**7 MPH**
250 (away)	73 μW	186 μW	140 μW
250 (towards)	190 μW	305 μW	259 μW
1075 (outdoor)	557 μW	672 μW	626 μW

3.5.2.1 Energy Estimation over a Day

The final step of the exploration is to estimate the total energy harvesting potential over a given time duration using the measurements throughout this work. This process is not simply the *duration* multiplied by *power*, as the harvested power varies due to the dynamic behavior of lighting and user activity. To this end, we use the ATUS dataset and obtain realistic EH potentials for various users in this dataset.

ATUS dataset consists of the distribution of different activities for various persons on a typical day in their lives [117]. We present a detailed explanation of this dataset in the next chapter. The dataset contains daily distributions of 18 different activity categories, such as sleeping, working, or exercising, for 9593 users. We use this dataset to obtain the daily energy harvesting potential of all the 9593 users. For this, we first obtain the activity and location information for each user from the dataset. The activity information is required to estimate the amount of energy harvesting from the knee bending motion. We assign each activity a gait speed from the set of {0, 3, 5, 7} MPH. For example, sleeping is assigned 0 whereas exercising is assigned 7 MPH. Similarly, the location information is necessary to estimate the energy harvesting from light as the indoor/outdoor conditions significantly affect the EH potential. We assign the average of 250 lux measurements to office conditions, half of it to home conditions and three times of it to store/shopping conditions, while 1075 lux measurements correspond to outdoor conditions. In this way, we convert the current measurements into realistic energy harvesting patterns.

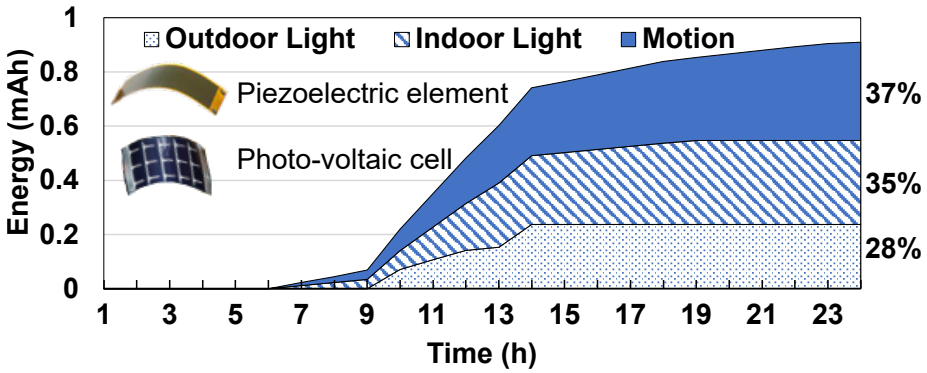


Figure 3.15: A typical user with close to 0.9 mAh cumulative harvested energy over a day. Motion EH consists 37% of the total harvested energy.

Figure 3.15 shows the EH breakdown for the median user in the dataset (i.e., a typical user). This particular user is asleep between 11PM-5AM, where no energy is harvested at all. During the day, the user spends time outdoors (6AM, 7AM and 13PM and 14PM) where the energy harvested due to outdoor light is dominant. The energy harvested by the motion EH subsystem accounts for 37% of the total energy harvested in the day. The median curve suggests that for the majority of users, close to 0.9 mAh @ 3.6 V can be harvested in a day with a typical wearable combined light and motion energy harvester such as the one used in this work.

4 OPTIMAL ENERGY MANAGEMENT

4.1 Motivation and Background

Energy harvesting solutions ideally aim for *energy neutrality*, which ensures that the energy consumption in a given period, such as a day, is equal to the energy harvested during the same period. Hence, energy-neutral operation can automate battery charging through harvested energy such that the battery level is restored at the end of each day. Consequently, the battery can be used without manual charging throughout its lifetime. Achieving energy-neutral operation is challenging due to the conflict between the uncertainty in harvested energy and quality of service (QoS) requirements of the target device. For example, consider a wearable health application where the target device must collect the vital signals and process them locally to detect abnormalities. On the one hand, the device needs a steady and sufficient amount of energy to perform its intended operation, e.g., analyzing the collected signals within a specified amount of time. On the other hand, the harvested energy may fluctuate widely or even vanish entirely during the same period. State-of-the-art research and commercial products on energy harvesting for IoT typically tackle such issues by improving the harvested power and reducing the consumed power [118, 119, 120, 121]. For example, Energiot designs energy harvesters tailored for specific use-cases to increase the harvested energy [119]. Similarly, Atmosic designs low-power wireless technologies tailored for energy harvesting devices to enable energy-neutral operation [120]. Even though those aspects are critical, the very limited and highly varying dynamics of the harvested

energy necessitate deliberate usage of the available energy. In this sense, this chapter aims to maximize the utilization of the target device (i.e., the amount of time QoS requirements are satisfied) under dynamic energy consumption requirements that guarantee energy neutrality. It presents two novel approaches to enable energy-neutral operation under varying energy harvesting conditions.

4.2 Contributions and Overall Goals

The major contributions of this section are as follows:

- ECO, an energy allocation framework that enables *energy-neutral operation* under varying EH conditions while maximizing the device utility,
- tinyMAN, a *model-free* RL based energy manager for resource-constrained wearable edge IoT devices,
- tinyMAN-MO, the *multi-objective* extension of tinyMAN for energy management under varying application demands,
- Implementations of both frameworks that demonstrate deployment on wearable devices with less than 100 KB memory footprint and less than 100 µJ energy consumption per inference.

4.3 ECO: Enabling Energy-Neutral IoT Devices through Runtime Allocation of Harvested Energy

This section proposes a runtime framework to enable energy-neutrality through constrained optimization (ECO) of the harvested energy optimally over a finite horizon divided into uniform time intervals. The main motivation behind ECO is to make judicious use of harvested energy so that the application utility is maximized. It is crucial to carefully allocate the harvested energy throughout the day because the device does not have continuous access to ambient energy. For example, light energy is available only when the user is outdoors during the day or is in a room with sufficient light. At the same time, the device needs energy to perform the application tasks throughout the day. Therefore, a portion of the harvested energy must be stored in the battery and used when there is no ambient energy available. ECO computes the amount of energy that the device can consume in each time interval to maximize its utility and guarantee an energy-neutral operation. The fundamental components of ECO are illustrated in Figure 4.1. It takes the initial battery energy at the beginning of each day and the expected amount of harvested energy during each time interval of the finite horizon (e.g., 24-hour) as inputs. We obtain the expected amount of harvested energy from a novel EH model that combines light EH and piezoelectric motion EH modalities, as described in Section 4.3.3. However, the operation of the ECO framework is not limited to this particular model. It is compatible with any other energy harvesting model. We also specify two constraints on the battery energy: i) the minimum battery energy level allowed at any point

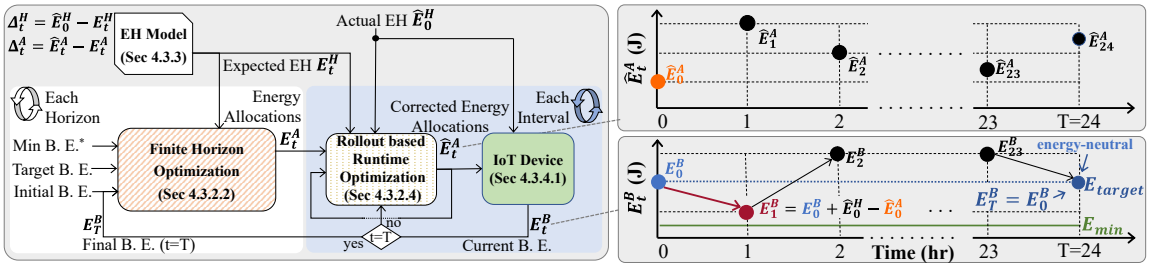


Figure 4.1: Overview of the proposed ECO framework.*B.E.: Battery Energy

in time, ii) the battery energy target at the end of the day. With the target energy constraint, the day to day battery operations of recharging and discharging are hidden from the user. Therefore, the users do not need to worry about manually recharging the battery.

The proposed framework maximizes a utility function under the battery energy constraints. The utility function can be constructed to model design metrics, such as device throughput and classification accuracy. Unlike prior work that requires a logarithmic utility function [26], ECO can handle an arbitrary function, which can be constructed by measured data. It first finds the energy that the device can consume during each time interval using the harvested energy profile. For example, suppose that the finite horizon is one whole day divided into 24 one-hour intervals. At the beginning of each day, ECO allocates the expected amount of harvested energy during a day into one-hour intervals to maximize the utility function under the battery energy constraints. This solution is not optimum due to the variations in harvested energy. Therefore, the initial allocations are corrected after each time interval using the actual harvested energy. To find the optimal amount of corrections for the remaining intervals, we propose a lightweight runtime optimization

algorithm. As a result, the proposed framework adapts to the deviations from the expected EH values with negligible runtime overhead.

This work presents a novel approach to enable energy-neutral operation under varying energy harvesting conditions. Our prior work addressed a similar problem [26]. However, it relaxes the minimum battery energy constraints and assumes an impractical logarithmic utility function to derive a closed-form solution. In contrast, the ECO framework uses a light-weight rollout technique that supports arbitrary utility functions and finds the energy allocations without relaxing any constraints. The rollout phase implicitly accounts for the harvested energy variations due to environmental conditions and user activities. Specifically, we start with an initial energy allocation using the expected EH values. Then, we use the rollout algorithm at runtime to make adjustments to the energy allocation as a function of the deviations in energy harvesting. Furthermore, the proposed technique is evaluated using ATUS data of 4772 users with light and motion EH modalities.

4.3.1 Overview

This section introduces the battery energy dynamics and the corresponding optimization constraints. It also presents the notion of the utility function for an arbitrary application.

Battery energy dynamics: The proposed allocation algorithm determines the amount of energy that can be consumed in a given interval based on the current battery level and harvested energy. Let T denote the finite time horizon of interest. We divide the time horizon into uniform time intervals t. For example, if we

consider a one-day horizon divided into 24 one-hour intervals, $T = 24$ hours and $t \in \mathcal{S}_{\mathcal{T}} = [0,1), [1,2), \ldots, [23,24)$, as illustrated in Figure 4.1. Using this notation, we denote the battery energy at the beginning of interval $[t, t+1)$ as E_t^B. The battery energy at the end of interval $t \in \mathcal{S}_{\mathcal{T}}$ depends on three factors as follows:

1. The starting battery energy E_t^B.
2. The harvested energy during the current interval $\widehat{E}_t^H$.
3. The allocated energy during the current interval $\widehat{E}_t^A$.

The battery energy level at the end of interval t equals to the level at the beginning of interval $t+1$. Similarly, the battery energy level at the end of the finite horizon is carried to the next finite horizon (e.g., next day). Hence, the battery energy dynamics can be expressed as

$$E_{t+1}^B = E_t^B + \eta \widehat{E}_t^H - \widehat{E}_t^A,\ t \in \mathcal{S}_{\mathcal{T}} \text{ and } E_T^B = E_0^B \tag{4.1}$$

Here, η accounts for the inefficiencies at the energy harvesting and battery charging interfaces. The battery level must be bounded from below to ensure that the target device has sufficient reserves (E_{min}) to execute an emergency task. In addition, the battery level can not exceed the battery capacity. To achieve these objectives, we introduce the following minimum and maximum energy constraints:

$$E_{max} \geqslant E_t^B \geqslant E_{min} \quad t \in \mathcal{S}_{\mathcal{T}} \tag{4.2}$$

To achieve energy-neutral operation, we also need to ensure that the battery energy level in the beginning of each finite horizon is greater than or equal to a specified target. For example, if the battery energy level at the beginning of each day satisfies this condition, the device can power itself without any manual charging requirements. Hence, we introduce the following condition for finite horizon boundaries:

$$E_0^B \geqslant E_{target} \tag{4.3}$$

Figure 4.1 summarizes the battery energy dynamics and the constraints presented above. We also list the parameters and notation used throughout the work in Table 4.1.

Utility function: We define the utility as the QoS that the target application provides to the user. Although the proposed framework does not depend on any particular application, we demonstrate it on health monitoring and activity tracking applications. Any application has an energy requirement to run correctly, such as guaranteeing a certain number of measurements per unit time. We denote the minimum energy required by the target application by M_E. If we fail to allocate M_E to the device, the device cannot provide useful output even if it is active. In general, the device can deliver higher utility (i.e., better QoS) with larger allocated energy. However, allocating more energy has a diminishing rate of return since excessive energy does not necessarily improve the QoS. To capture this behavior,

one can employ the following utility function:

$$u(E_t^A) = \alpha \ln \left(\frac{E_t^A}{M_E} \right)^{\gamma} \tag{4.4}$$

Table 4.1: Symbols and parameters used in the ECO framework.

Symbol	Description
T	The finite time horizon
t	The time interval $[t, t+1)$
$\mathcal{S}_{\mathcal{T}}$	Set of uniform time intervals t in the horizon T
E_t^A, E_t^H	Allocated and expected harvested energy in interval t
E_t^B	Battery energy at the beginning of interval t
E_{min}, E_{max}	Minimum and maximum battery energy constraints
E_{target}	Target battery energy constraint
$\widehat{E}_t^A$	The corrected allocation at interval t
$\widehat{E}_t^H$	The actual harvested energy at interval t
η	Parameter for harvesting and charging inefficiencies
Δ_t^A, Δ_t^H	Deviation from the expected values of allocated and harvested energy in interval t
Δ_t	Cumulative deviation from the actual values until t
$u(.)$, $\vec{u}(.)$	Utility function and vector valued utility function
M_E	Minimum energy required for positive utility
α, γ	Parameters to tweak the shape of the utility function
β	Discounting factor for utility
L	Coefficient matrix. Please see Equation 4.6
$\overline{\mathbf{E}}^{\mathbf{A}}$	$[\, E_0^A \; E_1^A \; \cdots \; E_{T-1}^A \,]^T \in \mathbb{R}^T$
$\overline{\mathbf{E}}^{\mathbf{H}}$	$[\, E_0^B + E_0^H \; E_1^H \; E_2^H \cdots \; E_{T-1}^H \,]^T \in \mathbb{R}^T$
$\overline{\mathbf{E}}^{\mathbf{B}}$	$[\, E_1^B \; E_2^B \; \cdots \; E_T^B \,]^T \in \mathbb{R}^T$
$\overline{\mathbf{B}}_{\mathbf{L}}$	$[\, E_{min} \; \cdots \; E_{min} \; E_{target} \,]^T \in \mathbb{R}^T$
$\overline{\mathbf{M}}_{\mathbf{L}}$	$[\, E_{max} \; \cdots \; E_{max} \; E_{max} \,]^T \in \mathbb{R}^T$
λ_1, λ_2	Vectors of Lagrange multipliers
$\nabla\vec{u}(.)$, $\nabla\vec{u}^{-1}(.)$	Gradient of the vector valued utility function and its inverse
N	Number of nodes in rollout
ϵ	Parameter to set the interval of coefficients in the rollout phase
$\tilde{U}(.)$	Heuristic function to compute future utility

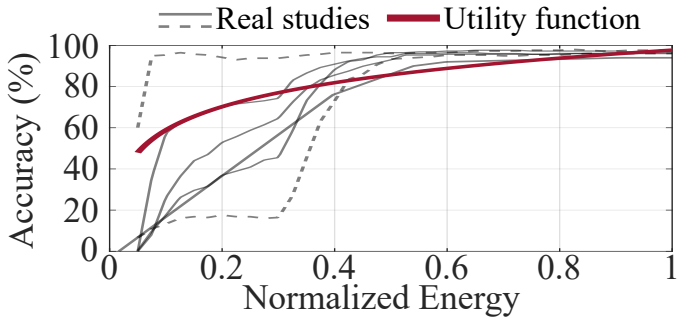


Figure 4.2: The generalized logarithmic utility function and utility curves from real studies.

where the parameters α and γ are used to tune the utility function for a specific user or application. Since this function is concave, it simplifies finding the optimal solution using the Karush-Kuhn-Tucker conditions [26]. However, such a generic utility function might not capture the behavior of the specific applications. For example, Figure 4.2 shows the utility of different applications according to the energy consumption along with the generalized utility function in Equation 4.23. Actual energy consumption and utility (measured by recognition accuracy) of real studies exhibit varying patterns [122, 123]. For example, the dashed lines in Figure 4.2 show data from two real applications, which significantly deviate from the logarithmic utility function. To address this issue, the proposed approach works with both logarithmic and arbitrary utility functions, as described in Section 4.3.2.4.

4.3.2 Optimal Energy Management

4.3.2.1 Problem Formulation

Our goal is to maximize the total utility over a finite horizon with the battery dynamics explained in Section 4.3.1 to guarantee an energy-neutral operation. The energy allocated to each interval can be expressed as a vector as: $\overline{\mathbf{E}}^{\mathbf{A}} = [\, E_0^A \; E_1^A \; \cdots \; E_{T-1}^A \,]^T \in \mathbb{R}^T$. Similarly, let $\overline{\mathbf{E}}^{\mathbf{H}} = [\, E_0^B + E_0^H \; E_1^H \; E_2^H \cdots \; E_{T-1}^H \,]^T \in \mathbb{R}^T$ be the extra energy added in each interval. Note that the first interval carries over the initial battery energy, while the rest contain only the energy harvested during that interval. Finally, we express the lower-bound and upper-bound energy levels to obtain a concise problem formulation: $\overline{\mathbf{B}}_{\mathbf{L}} = [\, E_{min} \; \cdots \; E_{min} \; E_{target} \,]^T$, $\overline{\mathbf{M}}_{\mathbf{L}} = [\, E_{max} \; \cdots \; E_{max} \; E_{max} \,]^T \in \mathbb{R}^T$. Using this notation, the optimization problem can be formulated as follows:

$$\begin{aligned} \textit{maximize} \quad & \sum_{t=0}^{T-1} \beta^t u(E_t^A) \\ \textit{subject to} \quad & L(\overline{\mathbf{E}}^{\mathbf{H}} - \overline{\mathbf{E}}^{\mathbf{A}}) - \overline{\mathbf{B}}_{\mathbf{L}} \geqslant 0 \\ & L(\overline{\mathbf{E}}^{\mathbf{H}} - \overline{\mathbf{E}}^{\mathbf{A}}) - \overline{\mathbf{M}}_{\mathbf{L}} \leqslant 0 \end{aligned} \tag{4.5}$$

The optimization objective is the total utility expressed as a function of the energy allocations over the horizon and the discount factor $0 < \beta \leqslant 1$ that enables de-emphasizing future intervals. The set of constraints given in Equation 4.16–Equation 4.3 are written in matrix form using vectors $\overline{\mathbf{E}}^{\mathbf{H}}$, $\overline{\mathbf{E}}^{\mathbf{A}}$, $\overline{\mathbf{B}}_{\mathbf{L}}$, $\overline{\mathbf{M}}_{\mathbf{L}}$, and a coefficient matrix $L \in \mathbb{R}^{T \times T}$:

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{T \times T} \tag{4.6}$$

4.3.2.2 Optimal Solution using Iterative Gradient Projection

The optimal solution to the problem given in Equation 4.5 can be obtained by writing the Lagrangian as:

$$\begin{aligned}\mathcal{L} = \vec{u}(\overline{\mathbf{E}}^{\mathbf{A}}) + \lambda_1^T(L\overline{\mathbf{E}}^{\mathbf{H}} - L\overline{\mathbf{E}}^{\mathbf{A}} - \overline{\mathbf{B}}_{\mathbf{L}}) - \\ \lambda_2^T(L\overline{\mathbf{E}}^{\mathbf{H}} - L\overline{\mathbf{E}}^{\mathbf{A}} - \overline{\mathbf{M}}_{\mathbf{L}})\end{aligned} \tag{4.7}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^T$ are the Lagrange multipliers and $\vec{u}(\overline{\mathbf{E}}^{\mathbf{A}})\colon \mathbb{R}^T \to \mathbb{R}^T$ is a vector function defined as follows: $\begin{bmatrix}\beta^0 u(E_0^A) & \beta^1 u(E_1^A) & \cdots & \beta^{T-1} u(E_{T-1}^A)\end{bmatrix}^T$. Using the Lagrangian, we can write the following Karush-Kuhn-Tucker conditions [124]:

$$\begin{aligned}&\nabla\vec{u}(\overline{\mathbf{E}}^{\mathbf{A}}) - (\lambda_1^T - \lambda_2^T)L = 0 \\ &\lambda_1^T(L\overline{\mathbf{E}}^{\mathbf{H}} - L\overline{\mathbf{E}}^{\mathbf{A}} - \overline{\mathbf{B}}_{\mathbf{L}}) = 0, \quad \lambda_1 \geqslant \overline{\mathbf{0}} \\ &\lambda_2^T(L\overline{\mathbf{E}}^{\mathbf{A}} - L\overline{\mathbf{E}}^{\mathbf{H}} + \overline{\mathbf{M}}_{\mathbf{L}}) = 0, \quad \lambda_2 \geqslant \overline{\mathbf{0}}\end{aligned} \tag{4.8}$$

where $\overline{\mathbf{0}} \in \mathbb{R}^T$ is a vector of zeros and $\nabla\vec{u}(\overline{\mathbf{E}}^{\mathbf{A}})\colon \mathbb{R}^T \to \mathbb{R}^T$ is the gradient of $\vec{u}(\overline{\mathbf{E}}^{\mathbf{A}})$ defined as follows: $\begin{bmatrix}\beta^0 \frac{\partial u(E_0^A)}{\partial E_0^A} & \beta^1 \frac{\partial u(E_1^A)}{\partial E_1^A} & \cdots & \beta^{T-1} \frac{\partial u(E_{T-1}^A)}{\partial E_{T-1}^A}\end{bmatrix}$.

Challenge: Finding the optimal solution is challenging in real application scenarios for two reasons. First, the set of equations in Equation 4.8 does not have a closed-form solution for an arbitrary utility function. Second, the solution requires the energy values that will be harvested in *future time intervals*.

Finding the optimal solution offline: Due to the challenges mentioned above, the optimal energy allocation can only be obtained using iterative algorithms and

expected values of the energy that will be harvested in future intervals. *To obtain the optimal solution as point of reference*, we adapt an iterative gradient projection algorithm used recently in network utility maximization problems [125]. It starts with random allocations and Lagrange multipliers λ_1, λ_2, as depicted in Algorithm 1 lines 3–5. At each iteration, the algorithm finds the optimal allocations $\overline{\mathbf{E}}^{\mathbf{A}}$ using the inverse of the gradient function and the latest Lagrange multipliers (line 8). Then, the Lagrange multipliers are updated following a feasible descent direction given by the optimization constraints (lines 9–10). The iteration in lines 6–11 continues until λ_1, λ_2 converge within desired tolerance bounds. The proof of convergence can be found in [125].

The need for a runtime solution: The actual energy harvested in each time interval differs from the expected values used by offline algorithms. Therefore, the energy allocations must be recomputed after each interval to account for these deviations. Iterative algorithms cannot be used repeatedly for this purpose due to their significant runtime overhead. For example, the execution time of iterative gradient projection algorithm (Algorithm 1) is 70.3 seconds on TI CC2652R microcontroller [126], as detailed in Section 5.3.4. In contrast, the light-weight energy allocation algorithm presented in [26] takes only 2 ms, resulting in over four orders of magnitude energy consumption savings, albeit with a significant loss in optimality. Since significant energy overhead is prohibitive, there is a *strong need for runtime solutions that can approach the performance of the iterative algorithms.* The following sections present our runtime solutions to the optimization problem in Equation 4.5. We start with a constant time solution for logarithmic utility function.

Then, we present a novel rollout algorithm for arbitrary utility functions.

4.3.2.3 Light-Weight Algorithm for Logarithmic Utility Functions

The energy allocation problem given in Equation 4.5 does not have a closed-form solution for two reasons: 1) Arbitrary shape of the utility function, 2) The minimum and maximum energy constraints in Equation 4.2. One can derive a closed-form expression by using a logarithmic utility function and relaxing the minimum and maximum energy constraints. When the utility function is defined as $u(E_t^A) = \ln(\frac{E_t^A}{M_E})$, the optimal energy allocations are given by:

$$\begin{aligned} \text{First interval} &: E_0^A = \frac{E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H}{1 + \beta + \beta^2 + \ldots + \beta^{T-1}} \\ \text{Subsequent intervals} &: E_{t+1}^A = \beta E_t^A \quad 0 \leqslant t \leqslant T-1 \end{aligned} \tag{4.9}$$

where E_t^H is the energy expected to be harvested in interval t and β is the discount factor [26].

Correcting the initial allocations: The initial energy allocations are found at the beginning of the finite horizon using Equation 4.9. These allocations must be corrected to account for the difference between the actual and expected energy values. Let Δ_t^H denote the difference between the expected and actual energy harvested in interval $t \in \mathcal{S}_\mathcal{T}$. Similarly, let Δ_t^A be the difference between the initial and corrected allocated energy in the same interval. Using these definitions, the

Algorithm 1: Iterative gradient projection algorithm.

Input: $\overline{\mathbf{E}}^{\mathbf{A}}$ and λ_1^*, λ_2^* ;
Output: $\overline{\mathbf{E}}^{\mathbf{A}}$ and λ_1^*, λ_2^* ;
$\overline{\mathbf{E}}^{\mathbf{A}} \leftarrow \mathtt{rand}(T, 1)$;
$\lambda_1, \lambda_2 \leftarrow \mathtt{rand}(T, 1)$;
$\lambda_1^*, \lambda_2^* \leftarrow \mathtt{rand}(T, 1)$;
while $||\lambda_1^* - \lambda_1|| > \textit{tolerance}$ ***and*** $||\lambda_2^* - \lambda_2|| > \textit{tolerance}$ **do**
  $\lambda_1 \leftarrow \lambda_1^*$ $\lambda_2 \leftarrow \lambda_2^*$;
  $\overline{\mathbf{E}}^{\mathbf{A}} \leftarrow \nabla \vec{u}^{-1}(\lambda_1^T - \lambda_2^T)L$ (follows from Equation 4.8) ;
  $\lambda_1^* \leftarrow \max\{\lambda_1 + stepSize \times (L\overline{\mathbf{E}}^{\mathbf{A}} - L\overline{\mathbf{E}}^{\mathbf{H}} + \overline{\mathbf{B}}_{\mathbf{L}}), \overline{0}\}$;
  $\lambda_2^* \leftarrow \max\{\lambda_2 + stepSize \times (L\overline{\mathbf{E}}^{\mathbf{H}} - L\overline{\mathbf{E}}^{\mathbf{A}} - \overline{\mathbf{M}}_{\mathbf{L}}, \overline{0}\}$;
  /* $stepSize = 0.001$ in our implementation */

cumulative deviation from the actual values can be expressed as:

$$\Delta_t = \sum_{\tau=0}^{t-1} (\Delta_\tau^H + \Delta_\tau^A) \tag{4.10}$$

These differences must be compensated at runtime by perturbing the initial allocations since they can lead to under- or over-utilization of harvested energy. The following lemma gives the re-allocations that maintain the optimality.

Lemma 1. *The initial allocation* $E_{t\,t\in\mathcal{S}_T}^A$ *must be perturbed according to the following equation to redistribute the energy surplus/deficit and maintain optimality.*

$$\widehat{E}_t^A = E_t^A + \Delta_t \frac{(1-\beta)}{1-\beta^{T-t}} \tag{4.11}$$

In Equation 4.11, $\widehat{E}_t^A$ denotes the corrected allocation at interval t. Note that Δ_t defined in Equation 4.10 represents the cumulative deviation until the current

interval, where positive and negative values imply a surplus or deficit, respectively. Hence, a portion of the cumulative deviation is added to the initial allocation. The derivation of Lemma 1 is given in Appendix B.

Equation 4.11 can cause the battery to drain below E_{min} or rise above E_{max} since these constraints were relaxed to find a closed-form solution. Hence, the final consideration is enforcing the minimum and maximum energy constraints at runtime. To achieve this, we project the remaining battery energy E^B_{t+1} at runtime using Equation 4.16, and compare it against E_{min} and E_{max} before committing to a solution. If there is a minimum energy constraint violation, we allocate the maximum energy that satisfies $E^B_{t+1} = E_{min}$. Similarly, if there is a maximum energy constraint violation, we allocate the excess energy such that $E^B_{t+1} = E_{max}$. That is, the allocation becomes:

$$\widehat{E}^A_t = \begin{cases} E^A_t + \Delta_t \frac{1-\beta}{1-\beta^{T-t}} & E_{max} \geqslant E^B_{t+1} \geqslant E_{min} \\ E^A_t + (E^B_{t+1} - E_{max}) & E^B_{t+1} > E_{max} \\ E^B_t + E^H_t - E_{min} & E^B_{t+1} < E_{min} \end{cases} \tag{4.12}$$

Insight: When the battery energy is not scarce (i.e. $E_{max} \geqslant E^B_t >> E_{min}$), the solution obtained by Equation 4.12 is optimal. Otherwise, the piecewise nonlinearity in Equation 4.12 is triggered and the solution becomes sub-optimal.

4.3.2.4 Runtime Rollout Algorithm for Arbitrary Utility Functions

The energy allocation technique presented in the previous section is valid only for the logarithmic utility function defined in Section 4.3.1. In real applications, the "utility" is often represented as measured metrics such as accuracy or throughput, which may not follow a logarithmic trend with the allocated energy, as shown in Figure 4.2. In fact, target devices may have a finite set of operating points with different energy consumption vs. utility trade-offs [122]. In these cases, the allocation algorithm should choose the operating points that maximize the accuracy under the overall energy harvesting budget over the horizon. Therefore, practical energy allocation algorithms must support arbitrary and discrete utility functions.

This section presents a rollout algorithm to achieve optimal runtime energy allocation under arbitrary utility functions. To achieve this objective, the proposed algorithm first finds the initial allocations at the beginning of the finite horizon, as illustrated in Figure 4.3. At the end of each interval, it determines the deviations from the actual and expected energy values (i.e., Δ_t^H and Δ_t^A). Then, it uses this information to perform a rollout phase before the start of the next time interval. Finally, it updates the energy allocations and starts the next interval. These three steps are repeated until the end of the finite horizon, as shown in Figure 4.3. The rest of this section describes each of these steps in detail.

4.3.2.5 Finding the Initial Allocations

The first step of ECO is computing the initial energy allocations in the beginning of the time horizon. This can be done using any technique, including the closed-

form solution given in Equation 4.9. In this work, we employ the iterative gradient projection algorithm presented in Section 4.3.2.2 for two reasons. First, it supports arbitrary utility functions. Second, its runtime overhead can be hidden since it runs only once per finite horizon (e.g., a day) using the expected energy values. Since the subsequent rollout phase corrects the initial allocations at runtime, we can increase the tolerance threshold in Algorithm 1 without decreasing the utility considerably. For example, the implementation on our hardware prototype takes 0.993 seconds to find the initial allocations. This execution time amounts to only 0.03% of a one-hour interval (or 1.6% of a one-minute interval). Combined with the efficient rollout phase that runs every interval, the total overhead is less than 0.1% of running Algorithm 1 in each interval. At the end of this step, we obtain the initial allocations $\overline{\mathbf{E}}^{\mathbf{A}}$.

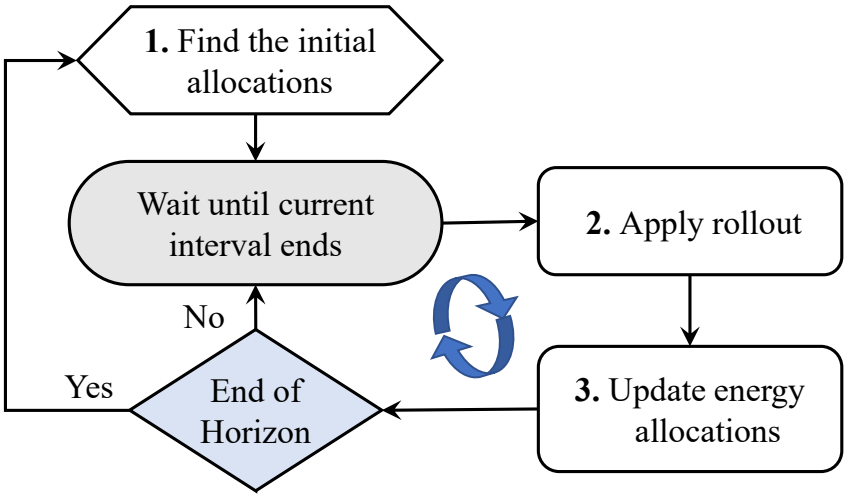


Figure 4.3: The overview of the rollout phase in the ECO framework.

4.3.2.6 The Rollout Phase

At the end of each interval, ECO adjusts the energy allocations using the cumulative difference between the expected and actual energy values, i.e., Δ_t defined in Equation 4.10. Suppose that the current time interval is $t = \tau$, as illustrated in Figure 4.4. ECO starts with the initially allocated energy and explores different re-allocation strategies represented by the branches. The first branch results in the smallest allocation, while the consecutive branches allocate a linearly increasing amount of energy.

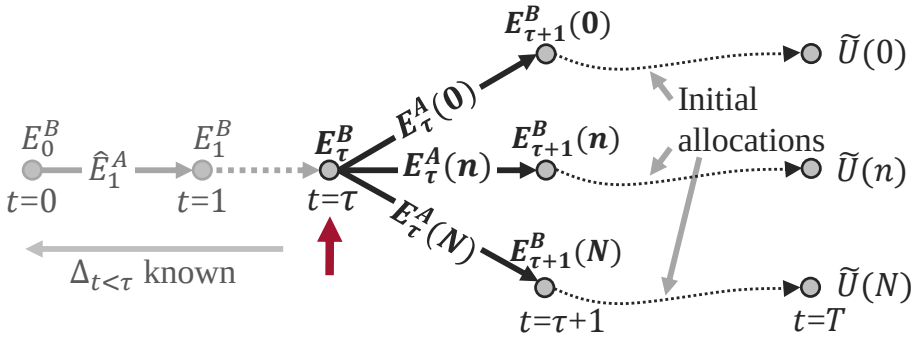


Figure 4.4: Illustration of the rollout phase.

To quantify the linearly increasing energy allocation illustrated by the branches in Figure 4.4, our rollout strategy replaces the $\frac{1-\beta}{1-\beta^{T-t}}$ term in Equation 4.12 with a coefficient as follows:

$$E_\tau^A(n) = \bar{\mathbf{E}}^{\mathbf{A}}[\tau] + \Delta_\tau(1 - \epsilon + \frac{2\epsilon}{N}n),\ 0 \leqslant n < N \tag{4.13}$$

Each different value of n leads to a different reallocation, i.e., branch, as illustrated in Figure 4.4. The parameter N can be the number of operating points or can be a

finite number (e.g., 10) that controls the exploration space. Then, the algorithm follows each branch and uses the initial allocations as the base policy for the rest of the intervals to find the expected utility. The total future utility at the end of the horizon is computed as follows:

$$\tilde{U}(n) = \begin{cases} -\infty & E^B_{t>\tau}(n) < E_{min} \\ -\infty & E^B_{t>\tau}(n) > E_{max} \\ -\infty & E^B_T(n) \leqslant E_{target} \\ \sum_{t=\tau+1}^{T} u(\overline{\mathbf{E}}^{\mathbf{A}}[t]) & otherwise \end{cases} \tag{4.14}$$

The first two conditions check if the allocation violates the minimum and maximum battery level constraints during the remaining time intervals. The third condition checks the target battery energy constraint at the end of the day. These checks eliminate aggressive allocations and hence reduce over-allocation.

4.3.2.7 Updating the Energy Allocations

The rollout phase calculates the expected utility achieved by each branch. Using this information, the revised energy allocations $\widehat{E}^A_t$ are obtained from a modified

version of Equation 4.12 as follows:

$$\widehat{E}_t^A = \begin{cases} \underset{E_t^A(n)}{\text{argmax}}\, u\big(E_t^A(n)\big) + \tilde{U}(n), & E_{t+1}^B \geqslant E_{min} \\ \underset{E_t^A(n)}{\text{argmax}}\, u\big(E_t^A(n)\big) + \tilde{U}(n) + (E_{t+1}^B - E_{max}), & E_{t+1}^B > E_{max} \\ E_t^B + E_t^H - E_{min}, & otherwise \end{cases} \tag{4.15}$$

where $u(.)$ is the utility function and $\tilde{U}(.)$ is the heuristic that gives the future utility when the base policy is applied. If the battery constraints are not violated, the algorithm follows the arc $(E_t^A(n))$ that maximizes the sum of immediate utility and future utility $u\big(E_t^A(n)\big) + \tilde{U}(n)$. If the device violates the minimum energy constraint during this period, it will go into a low-power state to conserve energy. Otherwise, the battery constraints are enforced as explained in Equation 4.12. This is done by first doing a linear descending sort of the values of sum and then checking the constraints. This sequential optimization technique guarantees an improvement over the base policy it is initially supplied with [127].

4.3.3 Energy Harvesting Model

This work employs a combination of light and motion energy as the ambient energy sources and a flexible Li-ion battery as the energy storage unit. We use the ATUS dataset [117] to extract the location and activity patterns of 4772 users. Then, we apply our light and motion EH models to the extracted data and generate 24-hour EH estimates per user. In what follows, we describe the data pre-processing steps we apply to the ATUS dataset and the EH models. Finally, we present example EH

values from different users to illustrate the final EH model.

4.3.3.1 ATUS 2018 dataset pre-processing

ATUS conducted by the US Department of Labor [117] is a comprehensive dataset that contains the amount of time people spend doing various activities. The dataset consists of daily activity patterns of 9593 users with 18 different categories of activities, such as personal care, working, or eating. 4772 users completed the survey on weekdays, whereas the remaining 4821 users completed it during weekends. We reduce the number of activity labels to 10 by grouping the infrequent activities, such as religious activities and eldercare, under the title “Other". The precise mapping of ATUS activity labels to our labels is provided in Appendix C. Figure 4.5a) shows the reduced set of activity labels and the percentage of 4772 users doing a particular activity throughout the day.

The location information is required to estimate the amount of energy harvesting from the light because the irradiance affects light EH efficiency. We assign a location label from the set of {‘home’,‘office’,‘store’,‘outdoors’} to each activity following common sense. For example, sleep activity is assigned ‘home’ as the location, ‘work’ is assigned ‘office’, ‘shopping’ is ‘store’, ‘eat’ is assigned ‘home’ or ‘office’ depending on time of day. The mapping of location labels to activities is provided in Appendix C. Figure 4.5b) shows the percentage of 4772 users being at the corresponding locations as a function of time. As a result, we obtain hourly activity and location information per user for a typical day in their lives.

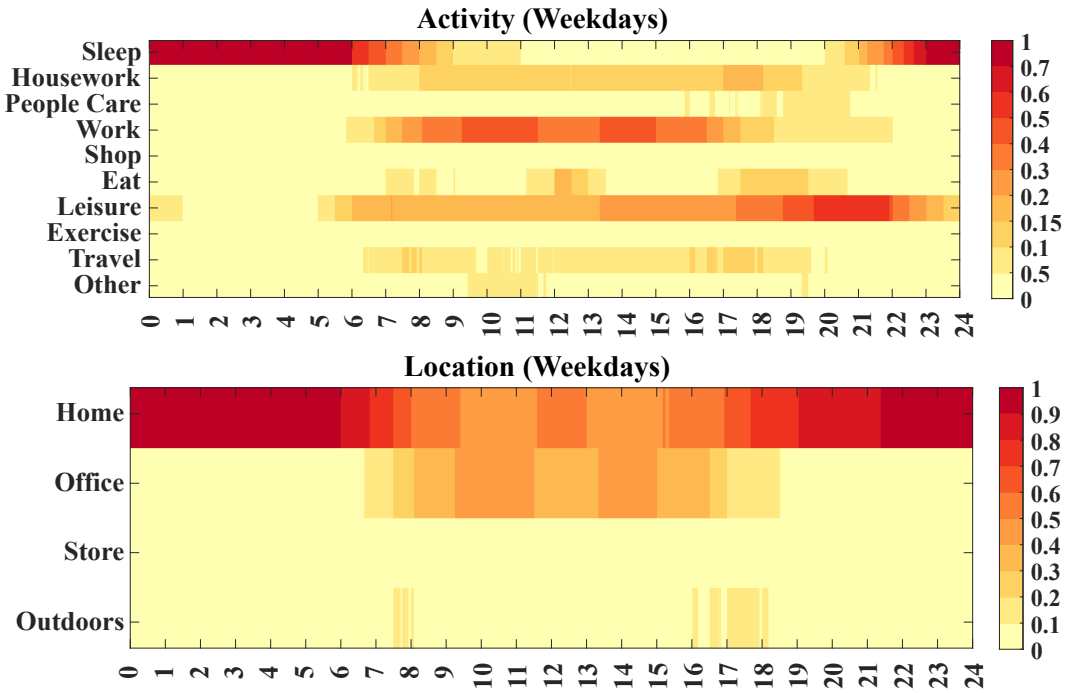


Figure 4.5: Illustration of the ATUS dataset. Percentage of 4772 users a) doing different activities b) at different locations throughout the day.

4.3.3.2 Light energy harvesting model

A light energy harvester uses a photovoltaic cell (PV-cell) to transduce the environment's light energy into usable electrical power. FlexSolarCells SP3-37, a lightweight, flexible, and small form-factor PV-cell, is used as the transducer [128]. We identified the I-V characteristics of SP3-37 in a previous study through a set of controlled experiments with a halogen lamp [53, 129]. This information is used to compute the harvested light energy when the irradiance is known.

PV-cells generate relatively high power in outdoor lighting conditions (over 1 mW/cm^2) where the irradiance is high [58]. To obtain the outdoors irradiance, we first estimate the sun's position at a given date and time using Sandia's Ephemeris

model [130]. Then, we convert the position information to radiation using Ineichen's model [131]. The output corresponds to the irradiance around the user when the user is outdoors.

The energy generated by a PV-cell quickly reduces for indoor lighting conditions where the irradiance is lower [129]. Furthermore, previous studies showed that indoor lighting conditions at the home, office, and store environments significantly differ from each other. The work in [132] provides measurements of irradiance in a typical office lighting condition. Besides, the information from [133, 115] shows that on average, store lighting intensity is $3\times$ of office lighting and home lighting is $0.5\times$ of office lighting. We use this information to find the irradiance when the user is at home, at an office, or at a store. Therefore, we combine the hourly location information obtained from the ATUS dataset, the irradiance amounts, and the I-V characteristics of SP3-37 to estimate hourly light EH per user.

4.3.3.3 Motion energy harvesting model

A piezoelectric energy harvester transduces mechanical energy into usable electrical power. We use joint bending as the source of mechanical energy, which is enabled by advances in flexible piezoelectric transducers, such as Polyvinylidene Fluoride (PVDF) and Macro-Fiber Composite (MFC). A recent study analyzes piezoelectric EH from joint bending motion and shows that 7.8 μJ per step can be harvested from the knee with an MFC8528P2 piezoelectric element during walking [29]. We use this knowledge to compute the harvested motion energy from the user's activity information.

Piezoelectric transducers generate higher power with increasing piezoelectric material [29]. Thus, we can use larger piezoelectric elements or connect multiple components in parallel to increase harvested power. Furthermore, MFC piezoelectric elements can be stacked on top of each other as long as they do not cause user discomfort. Following this, we consider ten elements per knee, where five MFC8528P2 elements lie across the knee, and another five sits on top. Low-effort walking with sporadic stepping is defined as 20-39 steps/min [134]. Using the middle value in this interval, we obtain 30 steps/min$\times$60 mins = 1800 steps/hour for activities that involve walking. We combine this with the activity information obtained from the ATUS dataset and compute 7.8 μJ/step$\times$1800 steps/hour$\times$10 = 0.14 J for activities that involve walking. For the sake of simplicity, we assume 2$\times$ the walking energy for activities that involve exercise and no harvesting for activities with little to no movement, such as working and sleeping.

4.3.3.4 Combined light and motion energy harvesting model

We combine the hourly light and motion EH values by superimposing them. For example, consider a user who is performing an exercise outdoors between 3-4 PM. The light EH from the outdoor irradiance at 3 PM and motion EH from exercising are added together to find the expected EH between 3-4 PM. This operation is performed for each hour to generate the 24-hour EH of a user. These values constitute the expected EH for a user, as shown with black markers on Figure 4.6. The user may deviate from their expected day, causing differences between the expected and actual harvested energy. In addition, there could be other variations due to partial

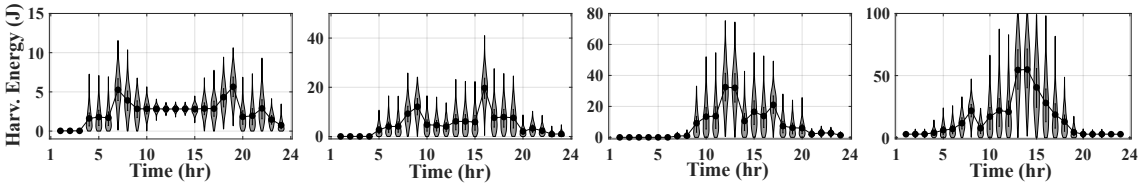


Figure 4.6: EH models of the median users from the four clusters.

shading, dirt, or scratches on the PV-cells for light energy harvesting and variations due to inappropriate placement, misalignment, and sliding of the piezoelectric material on the knee for motion energy harvesting. Such occasions can have a negative or a positive impact on the amount of harvested energy. For example, a misplaced PV-cell could receive more direct but less light due to partial shading by a piece of clothing. Thus, any other day of the same user is generated from the expected EH with random variations. We add deviations from the expected EH values to model random variations by considering each hour as a rectified Gaussian distribution. The variance of each distribution is found by a moving variance filter with a window size of 7. For example, the variance at 2 PM is computed by taking the moving variance of EH values between 11 AM to 5 PM. This approach inherently accounts for sleeping, as illustrated with the low-variance levels in the early hours of the day in Figure 4.6. We also introduce a normalization factor to the model to control the maximum standard deviation in the day. In Section 5.3.4, we sweep this factor from 0% to 25% to observe the effect of the uncertainty in EH on performance. Finally, we do this procedure for every user in the dataset and obtain 4772 EH models in total.

Clustering the ATUS dataset into user groups can help analyze the user behavior and use the proposed ECO framework for new users. To this end, we first divide

each day into three intervals: Midnight–8AM, 8AM–4PM, 4PM–midnight. Then, we added the energy harvested in each of the three intervals to reduce the data size to 4772×3. Finally, we apply k-medoids clustering over these three features to obtain four (k=4) well-separated user groups. Figure 4.6 shows the EH models of the median users in each cluster. All four clusters have typically low (0-6 J) harvested energy in the first eight and last eight hours of the day. The first cluster has the least expected energy (less than 8 J) in the middle eight hours, whereas the fourth cluster harvests more than 25 J during the same period, and the other clusters fall in between.

We envision that a new user will be placed in one of these clusters according to their energy harvesting usage pattern. When a new user starts using the device, we will log the harvested current going into the battery with the help of a coulomb counter and begin generating a coarse model. This coarse model will then be used to place the user in a cluster. Then as the user continues to use the device, their energy harvesting pattern is updated in time. We demonstrate the performance of ECO with varying degrees of model accuracy, from very accurate to very coarse, in Section 4.3.4.

4.3.4 Experimental and Simulation Results

This section evaluates the performance of the proposed ECO framework and compares it to the prior work in the literature. We demonstrate the proposed framework on a wearable hardware prototype designed for health and activity monitoring as the driver applications. However, we stress that the proposed approach does not

depend on any particular application and can be applied to any energy harvesting device.

Experiment parameters: We employ the prototype shown in Figure 4.7 to demonstrate the proposed ECO framework under realistic scenarios. It consists of an MPPT charger (TI BQ25504 [135]), a microcontroller (TI CC2652R [126]), an inertial measurement unit (ICM-20948 [136]), and various additional sensors that can be useful for wearable healthcare applications. The standby leakage currents of all components in our prototype device add up to 54.6 μA, which gives the total leakage energy in an hour as 0.64 J (3.3V×54.6μA×3600s). Similarly, we keep the microcontroller (μC) in the sensor controller state, and activate it and the non-volatile ram (NVram) only intermittently. Therefore, we calculate two different sums of active currents: The active currents of the sensors and power circuitry add up to 1.3 mA, while those of the μC and NVRam add up to 4.7 mA. Using these values, we obtain the total active energy of the sensors in an hour as 15.8 J with 100% duty cycle, and total active energy of the μC and NVRam as 55.8 J with 100% duty cycle. However, the μC and NVRam do not need to be active all the time. We consider 30 seconds of active time in an hour sufficient for them, which makes their duty cycle 30/3600 =0.8%. This scales the total active energy of the μC and NVRam down to 0.44 J in an hour. We obtain M_E as 0.64 J + 0.44 J=1.08 J, which is the minimum amount of energy required for the device without any useful output. The rest of the harvested energy is used to duty cycle the sensors. For example, if 11.08 J of energy is harvested in an hour, 10 J of it can be used to turn on the sensing circuitry with a 10/15.8=63% duty cycle, provided that battery constraints are not

violated (and will not be in the future intervals). Finally, we can claim that if a total of 24×1.08 J=25.9 J is harvested throughout the day, energy-neutrality can be achieved with our prototype device.

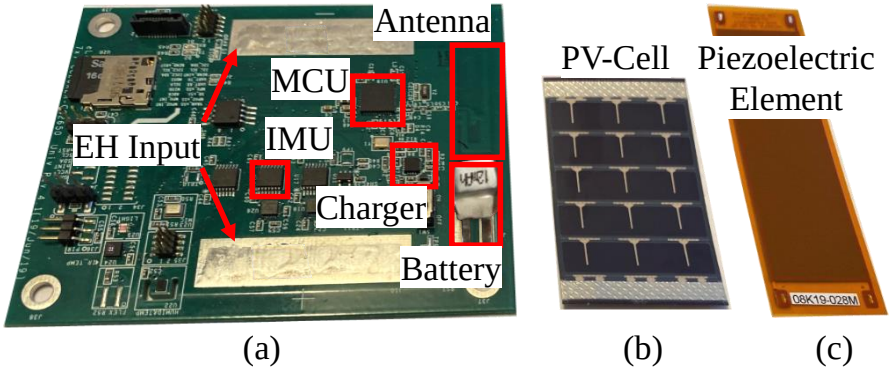


Figure 4.7: Pictures for the components used in the prototype. a) IoT Device b) SP3-37 c) MFC8528P2

Given the analysis above, Figure 4.6 shows that the amount of harvested energy is well above the required 25.9 J threshold for some users in the dataset. However, for some users it can be just above or even below the threshold. In the former case, the utility will be very near zero even though energy-neutrality is achieved. In the latter case, energy-neutrality can not be achieved. In fact, Figure 4.9-2b shows that if the initial battery energy is scarce, close to 5% of the users in the dataset cannot achieve energy-neutrality without violating the minimum energy constraint.

We use a 12 mAh Li-Po battery GMB 031009 [137] as the energy storage element. The IoT device parameters, such as E_{min} and M_E, are obtained from this prototype and listed in Table 4.2. The amount of energy stored in the battery is 160 J when it is full. We consider 10% and 90% of the battery capacity as scarce energy level and abundant energy level, respectively. The user's expected location and

Table 4.2: Parameter values used during the evaluation of ECO framework.

Parameter	Value	Parameter	Value
E_{min}	10 J	M_E	1.08 J
T	24	β	0.99
N	20	ϵ	0.2
E_{begin}, E_{target}	Scarce: 16 J Abundant: 144 J	EH standard deviation	0%, 5%, 10%, 15%, 20%, 25%

activities and the environment around them may deviate from the expected day, causing significant differences between the expected and actual harvested energy. To observe the effect of such modeling errors, we sweep the standard deviation in the generated energy harvesting patterns from 0% to 25% with 5% intervals. This amounts to a modeling mean absolute percentage error (MAPE) of 0% to 60% with 12% intervals, respectively. Therefore, at higher standard deviation settings, the expected EH pattern is only coarsely accurate.

Choice of energy storage: Supercapacitors are attractive alternatives to batteries due to their more efficient charging. In our prototype device we used a lithium-ion battery since it can retain its charge for a longer period of time and has less leakage. With choice, the ECO framework is built around a battery powered system. However, this is not a stringent requirement, ECO can also work with supercapacitors.

Cost and comfort of the wearable prototype: The MFC8528P2 piezoelectric element costs 165$ each and the SP3-37 PV-Cell costs 3$ each. In addition, the bill of materials of the prototype board is currently less than 20$. Our driver applications are in the medical context, such as falling prevention/detection in elderly care, or freezing of gait detection in Parkinson's disease patients. To this end, we place our device

on a comfortable knee sleeve, where the PV-cells are placed on the sleeve and piezoelectric elements are inserted into pockets on the sleeve. Our experience with this setup shows that the sleeve is comfortable and wearing it does not impact people's behavior in any way.

4.3.4.1 Implementation Overhead

The execution times and energy consumption of the ECO framework and the prior approach [26] are measured on the prototype. We also execute the iterative gradient projection algorithm (Algorithm 1) at every interval t to obtain a reference utility and measured its energy consumption (tolerance: 1×10^{-6}, $stepSize$=1×10^{-3}).

Table 4.3 shows the execution time, energy consumption, and utility for the three approaches for a typical user. ECO achieves nearly the same utility as the iterative algorithm while having significantly less execution time and energy consumption. Specifically, the energy overhead of ECO is only 87.5 µJ/hour, which is negligible compared to the energy harvested in an hour. The initial run of the iterative algorithm in ECO uses larger tolerance and step size (1×10^{-3}, 5×10^{-3}), since the rollout phase makes up for the loss in optimality. The execution time of the

Table 4.3: Execution time and energy measurements

		Iterative	**ECO**	**Bhat [26]**
Time (s)	once	N/A	9.93×10^{-1}	N/A
	every t	7.03×10^{1}	2.50×10^{-2}	2.00×10^{-3}
Energy (mJ)	once	N/A	3.48	N/A
	every t	2.46×10^{2}	8.75×10^{-2}	7.00×10^{-3}
Normalized Utility*		1.0	0.98	0.83

*The utility is normalized with respect to the iterative technique.

correction in each interval is only 25 ms. The approach in [26] takes virtually zero time to execute since it is a closed-form solution. However, it suffers from low utility, which renders it impractical. These results show that the ECO framework achieves the same utility as the iterative algorithm while having 1000$\times$ smaller energy overhead.

Complexity: The ECO framework runs the iterative gradient projection algorithm (Algorithm 1) once to compute initial allocations. Then, in the rollout phase, it iterates over N branches (O(N)) to try different coefficients for the current interval, as depicted in Figure 4.4. After each branch is visited and a total utility for each is computed, these utility values are sorted (NlogN) and checked for constraint violations starting from the highest. Therefore, the complexity of the proposed ECO framework is O(N) + Nlog(N).

4.3.4.2 Daily Energy Allocation Simulation Results

This section demonstrates the performance of the ECO framework in multiple steps. We first compare the daily energy allocations and evolution of battery energy obtained by ECO to those obtained by the iterative approach and prior work in the literature [26, 25]. Moreover, the proposed framework's effectiveness is demonstrated by the increase in average utility over the prior approach across all 4772 users in the dataset. We also apply ECO to a non-logarithmic utility function to demonstrate that it can support arbitrary functions.

4.3.4.3 Evaluations using logarithmic utility function

We first use the logarithmic utility function defined in Section 4.3.1 to demonstrate the performance of ECO. To this end, we first use Algorithm 1 at each interval to recompute the optimal energy allocations as the day progresses. Second, we use two prior approaches in the literature [26, 25]. Finally, we run ECO and compare the obtained energy allocation and battery energy curves with the other solutions.

Energy allocations and battery energy throughout the day: Figure 4.8 compiles the results for a typical user, who is chosen randomly among the 4772 users in the dataset. The columns in the figure show the hourly energy allocations, battery energy and EH values throughout the day, respectively. To highlight the shortcomings and advantages of either approach, we focus on three days with different battery energy and EH conditions. The rows in Figure 4.8 correspond to these different days. Specifically, the first row corresponds to the case with abundant initial battery energy and high standard deviation in energy harvesting as shown in Figure 4.8-1c. In this case, the relaxed $\mathtt{E}_{min}$ constraint is not triggered from Figure 4.8-1b, due to the abundant starting energy at the beginning of the day. However, the relaxed $\mathtt{E}_{max}$ constraint is triggered around 4PM which causes the energy allocations in the later half of the day deviate from the optimal, as shown in Figure 4.8-1a. The deviation from the optimal is much smaller for the ECO framework. Moreover, prior approaches violate the $\mathtt{E}_{target}$ constraint at the end of the day, whereas ECO satisfies it.

The second row in Figure 4.8 corresponds to scarce initial battery energy and low standard deviation in EH. In this case, the approach in [26] over-allocates energy

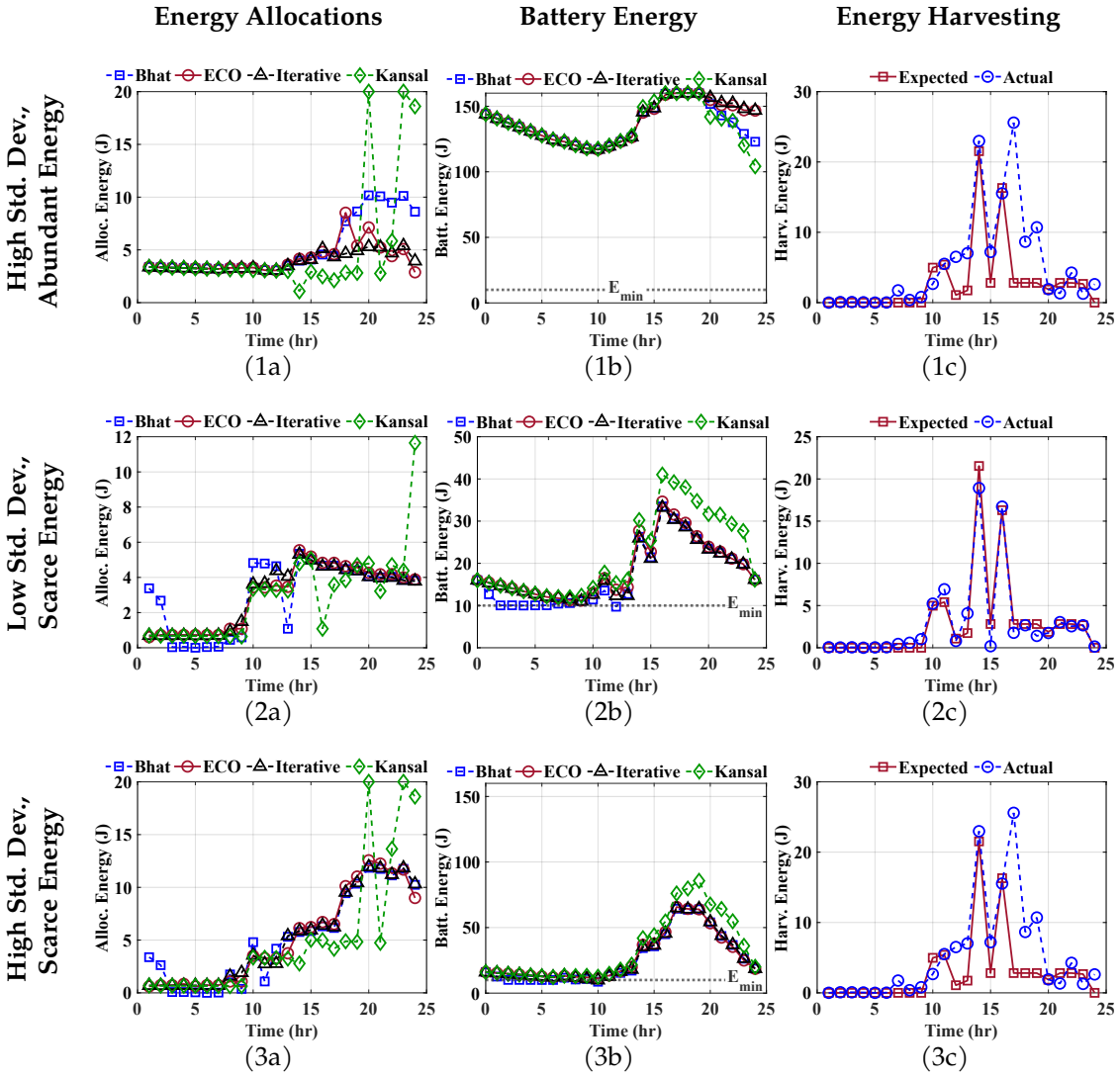


Figure 4.8: Comparison of energy allocations and battery energy (b.e.) evolution under different conditions for the same user: 1st row) 25% standard deviation in EH and 144 J initial b.e., 2nd row) 5% standard deviation in EH and 16 J initial b.e., 3rd row) 25% standard deviation in EH and 16 J initial b.e.
Kansal: [25], Bhat: [26]

at the beginning of the day and triggers the minimum energy constraint only a couple of hours into the day due to the amount of energy expected to be harvested

later. The work in [25] under-allocates energy later in the day and deviates from the optimal battery level curve as can be seen in Figure 4.8-2b. In contrast, ECO behaves more conservatively and follows the energy allocation and battery energy curves obtained by the iterative approach much closer, as depicted in Figures 4.8-2a and 2b, respectively. As a result, the ECO framework obtains higher utility than the prior approaches.

The third row in Figure 4.8 corresponds to the scenario with scarce initial battery energy and high standard deviation in EH. Similar to the previous case, the prior approaches under- or over-allocate the energy, causing battery energy to deviate from the optimal. The proposed ECO framework behaves much more conservatively and, as a result, closely follows the iterative solution, as shown in Figures 4.8-3a and 3b.

In general, ECO behaves much closer to the optimal than prior work regardless of the battery level and uncertainty in expected harvested energy. This capability of ECO is enabled by the rollout phase, which avoids the piecewise nonlinearity by being less aggressive. In contrast, the prior approaches under- or over-allocate the energy and trigger the nonlinearity that is enforced at runtime.

Utility and E_{min} violations throughout the day: Next, we compare the proposed ECO framework's performance to the prior work in the literature across all 4772 users. For this, we analyze the average utility and the E_{min} and the E_{target} violation occurrences for the different approaches. First, the EH model generates seven separate days per user for each standard deviation value. For each day, we obtain the results from the ECO framework and the prior work as well as the iterative

solution. Then, the average utility of each approach is found across each day and user. For example, for the day shown in the 3rd row in Figure 4.8, ECO achieves 0.91 normalized utility, whereas the prior work in [26] achieves 0.31, and the one in [25] achieves 0.59. This behavior occurs because the prior approaches under-allocate the energy which is punishing in the logarithmic utility function. To obtain the percentage of minimum energy violations, we count the number of intervals the lower threshold in piecewise nonlinearity in Equations 4.12 and 4.15 are triggered and divide them to total number of intervals (i.e. 24×7 intervals per user). Similarly, we count the number of target energy violations and divide it to the total number of intervals to obtain the percentage of target energy violations. These steps are done twice, for scarce and abundant battery energy conditions respectively.

Figure 4.9 compares the utility figures in the first column, the $\mathtt{E}_{min}$ violation occurrences in the second column and the $\mathtt{E}_{target}$ violation occurrences in the third column for all three approaches, as a function of mean absolute percentage error (MAPE) in the expected harvested energy. The rows correspond to abundant and scarce battery energy conditions, respectively. In both cases, the proposed ECO framework outperforms the prior approaches as depicted in Figure 4.9-1a and 2a. When the battery energy is abundant, all approaches have very small number of $\mathtt{E}_{min}$ violations as shown in Figure 4.9-1b, as expected. In addition, ECO causes close to zero $\mathtt{E}_{target}$ violations, whereas the previous approaches cause significantly more, as depicted in Figure 4.9-1c. Under scarce battery energy conditions, the proposed ECO framework again outperforms the prior approaches as illustrated in Figure 4.9-2a. Specifically, it obtains between 10% – 35% more utility on the average.

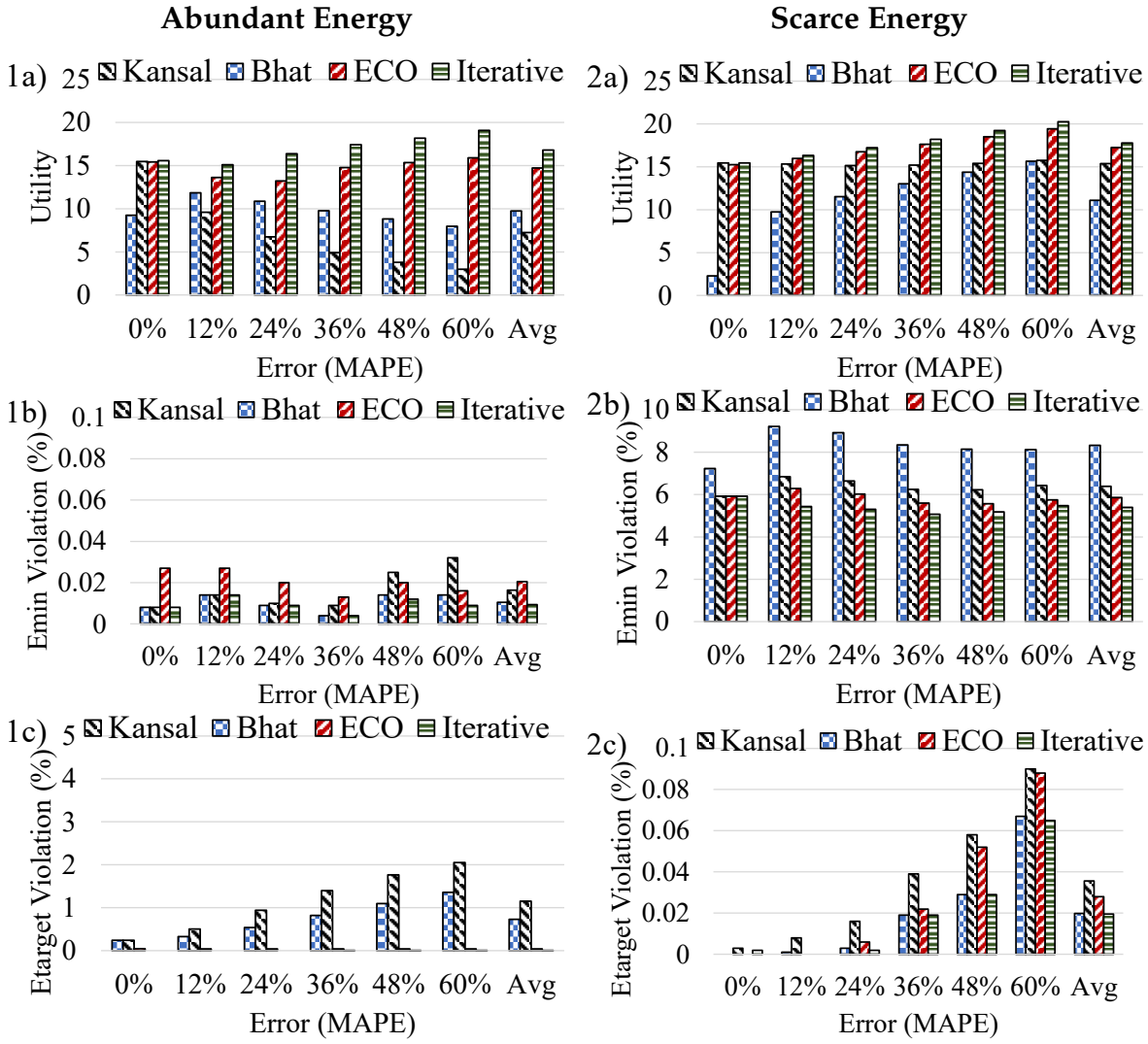


Figure 4.9: Comparison of Utility, E_{min} violation and E_{target} violation occurrences. 1st column) 144 J initial battery energy, 2nd column) 16 J initial battery energy. Kansal: [25], Bhat: [26]

Moreover, Figure 4.9-2b shows that E_{min} violations are inevitable for about 5% of the entire dataset when the battery is scarce even for the iterative algorithm. When a minimum energy violation happens, the device goes to a sleep state. The cost of

this action is zero utility during sleep periods, which is reflected in the reported utility. The prior approaches lead to significantly higher E_{min} violations than ECO. Specifically, ECO achieves close to 3% lower E_{min} violations than the prior work. Finally, all approaches have very small number of E_{target} violations under scarce battery energy as shown in Figure 4.9-2c. It can be seen that the utility obtained by the optimal iterative algorithm and the other approaches increase as the MAPE increases. This observation is due to the rectified gaussian distribution forcing the harvested energy to be positive. It results in higher mean harvested energy over a day as the MAPE increases. As a result of higher harvested energy, all approaches obtain higher average utility.

The effect of M_E and E_{min}: We selected the values for parameters M_E and E_{min} from our prototype as explained in the beginning of Section 4.3.4. To observe their effects on ECO and the other approaches, we swept the values of E_{min} and M_E independently of each other, as depicted in Figure 4.10. According to these results, increasing M_E has a negative effect on the utility, as shown in Figure 4.10-a. This behavior is expected because M_E is the minimum energy required for obtaining positive utility. Increasing M_E does not affect the percentage of E_{min} and E_{target} constraint violation occurrences. Similarly, increasing E_{min} also degrades the utility, as depicted in Figure 4.10-b. This is because the allocation algorithms are forced to allocate lower energy to prevent violating the E_{min} constraint, which decreases the utility. In addition, Figure 4.10-c shows that increasing E_{min} significantly increases the percentage of E_{min} constraint violation occurrences, as expected. Increasing E_{min} does not affect the percentage of E_{target} constraint violation occurrences.

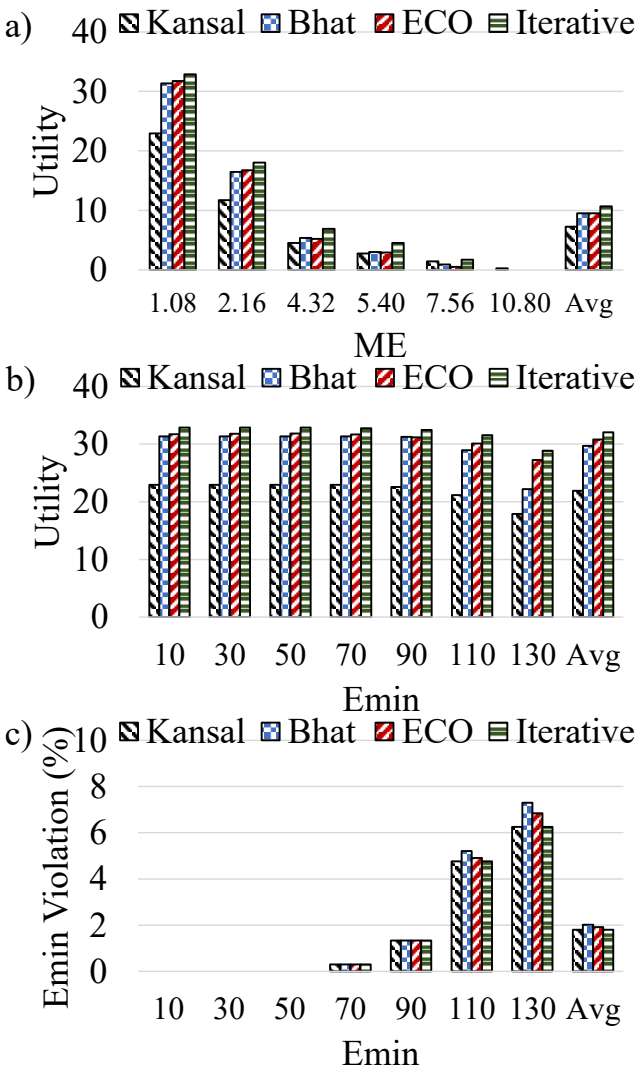


Figure 4.10: The effect of a) increasing ME on utility, b) increasing E_{min} on utility, c) increasing E_{min} on E_{min} Violation Percentage.
Kansal: [25], Bhat: [26]

4.3.4.4 Evaluations using an arbitrary utility function

One of the key advantages of the proposed ECO framework is its general applicability to arbitrary utility functions. Next, we use the energy vs. accuracy relationship

of a gesture recognition application from the literature to demonstrate the performance of the ECO framework with an arbitrary utility function.

Figure 4.11 shows 12 design points for a gesture recognition application from the literature [123]. A function with the form $y = ax^b + c$ fits much better to these data points than the logarithmic function in Section 4.3.1 (i.e. R^2 value for the two functions are 0.96 and 0.82, respectively). Therefore, one can use the discrete points or the fit shown with the blue curve in Figure 4.11 as the utility function (i.e., the utility is defined as the classification accuracy).

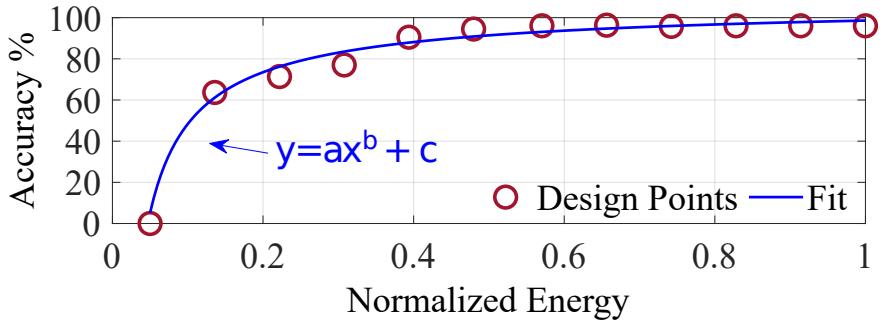


Figure 4.11: Design points for a gesture recognition application and the fitted energy vs. accuracy curve.

Accuracy and E_{min} violations throughout the day: We follow the same procedure used in generating Figure 4.9 to demonstrate the performance of the proposed ECO framework with the utility function depicted in Figure 4.11. Figure 4.12 shows the accuracy in the first column, E_{min} violations, and E_{target} violations across all users in the second and third columns, respectively. The rows correspond to abundant and scarce battery energy conditions.

ECO obtains similar accuracy to the iterative solution when the battery energy is abundant and outperforms the approach in [25], as shown in Figure 4.12-1a.

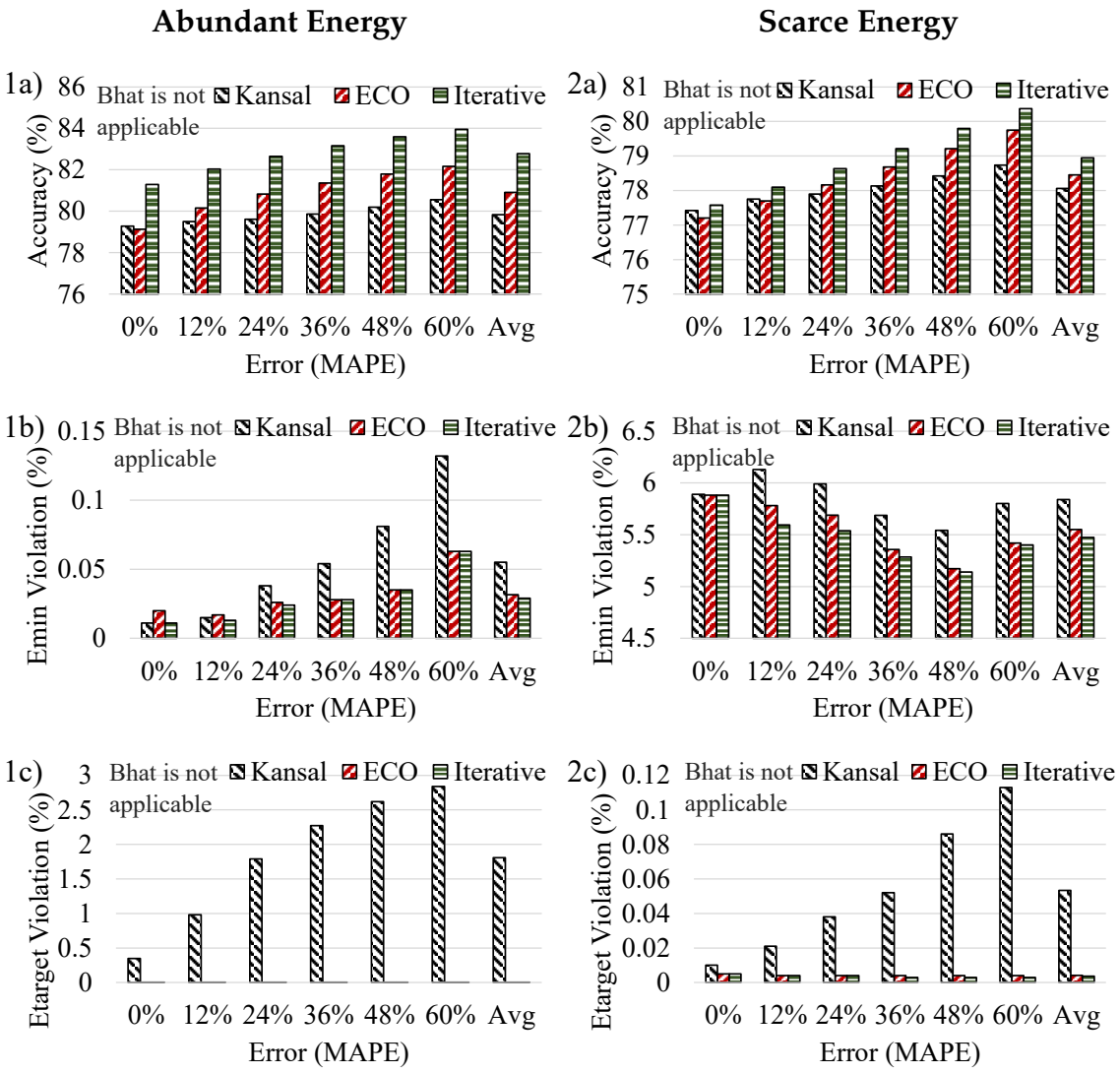


Figure 4.12: Accuracy, E_{min} violation and E_{target} violation occurrences. 1st column) 144 J initial battery energy, 2nd column) 16 J initial battery energy.
Kansal: [25], Bhat: [26]

More specifically, ECO obtains close to 2% higher accuracy on average. In addition, ECO causes E_{min} violations in less than 0.03% of the total 4772×7×24 intervals as shown in Figure 4.12-1b. In contrast, the prior work causes two times that.

Moreover, Figure 4.12-1c illustrates that ECO significantly outperforms the prior work in E_{target} violations. ECO causes essentially no violations, whereas the prior work has E_{target} violations in 2% of the total number of days. Under scarce battery energy conditions, the performance of all three approaches deteriorates, as expected. However, ECO can still achieve more accuracy compared to the prior work. For example, Figure 4.12-2a shows that at 60% MAPE in EH, the iterative solution and the ECO framework achieve 80.37% and 79.75% accuracy, respectively. Moreover, Figure 4.12-2b shows that E_{min} violation occurrences are within less than 1% of the optimal when battery energy is scarce. In contrast, the prior work causes more E_{min} violations for all error levels in harvested energy. Finally, all approaches have a very small number of E_{target} violations under scarce battery energy as expected, as shown in Figure 4.12-2c.

4.3.4.5 Three-year Energy Allocation Simulation Results

This section demonstrates the performance of the ECO framework in continuous use throughout three years. For this, we have generated an annual energy harvesting profile for the median users of the four clusters which are shown in Figure 4.6. We used the highest model error (i.e., 25% standard deviation - 60% MAPE) when generating each day. In addition, our model also incorporates the seasonal solar energy increase during the summer months. Note that this only has an impact on the harvested energy if the user spends time outdoors.

Battery degradation, leakage, and inefficiency: Recent studies show 5% to 10% annual capacity degradation for lithium-ion batteries [138]. Following this, our

three-year analysis includes a daily linear degradation that corresponds to 5% annual degradation. Similarly, the work in [138] claims that a battery can complete its lifetime in as short as 2 years if it is operated near its full capacity most of the time. If operated at 50% capacity, its lifetime can be as long as 6 years. We argue that the ECO framework can be configured to achieve this long lifetime by setting the E_{begin} and E_{target} values accordingly, such that the battery is kept in the middle region most of the time. We do not model the battery leakage separately as it is already considered in the calculation of 1.08 J as M_E.

We assume that the device starts with a 90% battery capacity (i.e., we assume an initial charging of the battery). Then, the battery level at the end of the first day is assigned as the E_{begin} for the next day. The E_{target} for this day is also set to the same value. The time interval is 1 hour, and the simulation continues for 3 years. Therefore, in total there are $365 \times 3 \times 24$ intervals. Figure 4.13 shows the average utility and E_{target} violations for each cluster over these intervals. According to these results, ECO achieves higher utility than the previous approaches while doing significantly less E_{target} violations. Specifically, ECO achieves 91% of the utility obtained by the iterative algorithm, while causing E_{target} violations in less than 10% of the total number of days. In contrast, the approach in [26] and [25] achieve 78% and 51% of the optimal utility while causing E_{target} violations in about 30% of the days. This suggests that ECO deviates less from the optimal battery level and is more likely to guarantee energy-neutral operation.

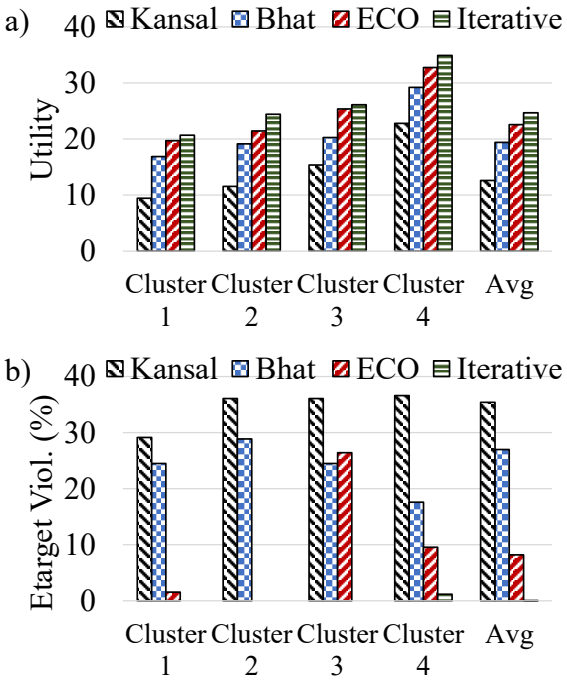


Figure 4.13: 3-year simulation results. a) Utility b) E_{target} violations (%). Kansal: [25], Bhat: [26]

4.4 tinyMAN: Lightweight Energy Manager using Reinforcement Learning for Energy Harvesting Wearable IoT Devices

Energy management techniques are required to ensure ENO (i.e., the device maintains a certain battery level and minimum performance level) despite the variations in the harvested energy. These techniques use the available energy judiciously to maximize the application performance while minimizing manual recharge interventions to tackle this challenge [31]. They must satisfy the following conditions to

be deployed on a resource-constrained device:

1. Incur low execution time and power consumption overheads,
2. Have a small memory footprint
3. Be responsive to the changes in the environment
4. Learn how to adapt to environment and workload changes.

Although most current energy management algorithms can satisfy (1) and (2), they fail to meet the other two requirements since their performance critically depends on fixed, non-adaptive predictive models of the future harvested energy. To the best of our knowledge, no energy management framework in the literature fulfills these four criteria.

This section presents a lightweight energy manager that enables ENO while maximizing the utilization of the device under dynamic battery energy constraints and EH conditions. One can formulate this as a constrained optimization problem and solve it with modern solvers like CPLEX [139] and Gurobi [140]. However, these approaches are impractical for online deployment since they require long execution times [141] and significant computational power beyond what simple wearable devices can provide. As a powerful and practical alternative, this section presents a reinforcement learning (RL) based energy management framework, tinyMAN, for resource-constrained wearable edge devices. tinyMAN is suitable for online deployment thanks to its negligible energy overhead, and it is *model-free* as RL implicitly learns the variations in harvested energy. tinyMAN takes the battery

level and the previous harvested energy values as inputs (states) and maximizes the device's utility by judiciously allocating the harvested energy throughout the day (action). The utility is defined by an arbitrary concave function of allocated energy (e.g., $u = \log(x)$ or $u = \sqrt{x}$). To train tinyMAN, we employ the Twin Delayed Deep Deterministic Policy Gradient (TD3) Algorithm, a state-of-the-art RL algorithm for continuous action spaces [142]. Hence, the energy allocation values that tinyMAN yields can take continuous values according to the current energy availability. To this end, we first develop an environment for the RL agent to interact with. This environment uses the light and motion EH modalities and *American Time Use Survey* [117] data from 4772 different users to model the dynamic changes in the harvested energy and battery. Over time, by interacting with the environment, the agent learns to manage the harvested energy on the device according to the battery energy level and the harvested energy. According to our evaluations, tinyMAN achieves up to 17% higher device utility than prior approaches while making 80% less battery constraint violations, such as not satisfying ENO or depleting battery prematurely.

4.4.1 Overview

The problem formulation is very similar to ECO. This section briefly refreshes the battery energy dynamics and constraints to formulate the optimization problem. Then, it describes the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm used to train the tinyMAN RL agent.

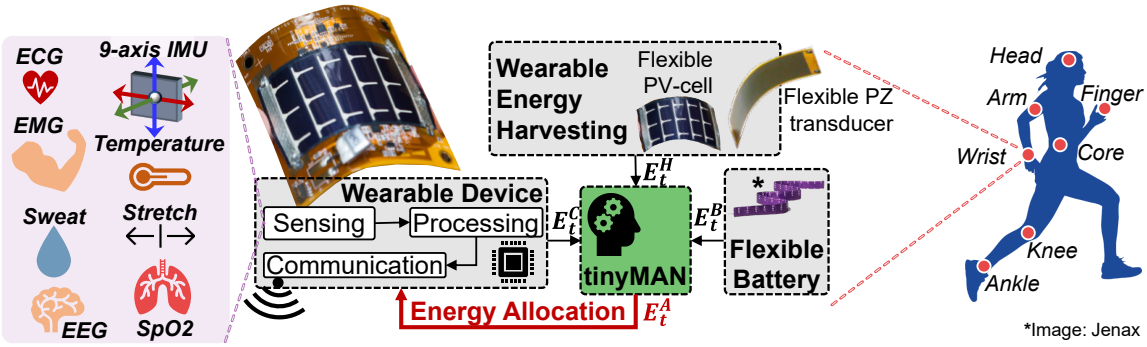


Figure 4.14: Illustration of the wearable device environment the tinyMAN agent interacts with.

4.4.1.1 Problem Formulation

The proposed tinyMAN framework is deployed in an environment that consists of a wearable device, a flexible energy harvester, and a flexible battery, as depicted in Figure 4.14. In the following, we define the battery energy dynamics, the relevant constraints, and the utility function of the device and explain the EH source model.

Battery dynamics and constraints: tinyMAN finds the optimum energy allocations that maximize the utilization of a target device under ENO and battery constraints. This work uses a prototype wearable device as the target platform to deploy tinyMAN. The device houses a flexible, small form-factor LiPo battery with a capacity of 12 mAh and can charge the battery through energy harvesting. Therefore, the battery energy dynamics in the environment is a function of:

1. E_t^B the battery energy level at the start of time interval t
2. E_t^A the allocated energy at the start of time interval t

3. E_t^H the harvested energy in time interval t

Our energy management framework uses an episodic setting where each episode corresponds to a single day ($T = 24$ hours), and each step t in an episode lasts an hour. Using these definitions, we write the battery energy dynamics as follows:

$$E_{t+1}^B = E_t^B + \beta E_t^H - \alpha E_t^A,\ t \in T \tag{4.16}$$

where β corresponds to the efficiency of the harvester and α corresponds to the percent utilization of the allocated energy (i.e., $E_t^C = \alpha E_t^A$ is the actual consumed energy).

There are two physical constraints on the battery level. It is bounded from below at zero and from the top at the battery capacity (E_{cap}^B). Furthermore, we want the device to have an emergency reservoir at all times to serve as backup energy:

$$E_{cap}^B \geqslant E_t^B \geqslant E_{min}^B,\ \ t \in T \tag{4.17}$$

To achieve ENO, tinyMAN ensures that the battery energy level at the end of an episode is equal to a specified target:

$$E_T^B \approx E_{target} \tag{4.18}$$

For achieving ENO, we set $E_{target} = E_0^B$ such that the battery energy level at the end of the episode is equal to the battery energy level at the beginning of the same episode. We enforce these constraints using the reward function as explained in

Table 4.4: Components used in the prototype wearable device.

Component	**VDD**	$\mathbf{I}_{idle}$	$\mathbf{I}_{active}$	**Part #**
Microcontroller	1.8-3.8V	0.9 µA	Sensor Cont.: 30 µA, Active: 3.4 mA	CC2652R
IMU	1.7-3.6V	8 µA	Acc only: 450 µA, Gyro only: 3.2 mA	MPU9250
Nonvolatile Ram	1.6-3.6V	10 µA	Rewrite: 1.3 mA, Read-out: 0.2 mA	MB85AS4MT
Humid. & Temp. Sensor	2.7-5.5V	0.1 µA	1 Hz: 1.2 µA	HDC1000
Ambient Light Sensor	1.6-3.6V	0.3 µA	1.8 µA	OPT3001
Boost Converter for EH	2.5-5.2V	0.3 µA	-	BQ25504
LDO linear regulator	2.0-5.5V	35 µA	-	TLV702

Section 4.4.2.1.

Utility: The utility is a metric that represents the useful output produced by the device, such as accuracy or throughput, depending on the target application running on the device. For example, for a state-of-the-art heart rate monitoring application, the utility can be defined by the sampling frequency. The proposed tinyMAN framework supports any arbitrary utility function.

For the current work, we define the utility according to the minimum energy consumption of the device in an hour, which is calculated using the components listed in Table 4.4. The sum of the idle currents of these components amounts to 54.6 µA, resulting in an idle energy consumption of $E^A_{min} = 0.64$ J when the device operates at a VDD of 3.3V. Thus, if the allocated energy falls below this minimum threshold ($E^A_t < E^A_{min}$), the utility is zero (or negative), indicating that the device does not produce any useful output. However, it is important to note that the utility function can take on any shape based on the specific requirements of the application. In our study, we employed a logarithmic utility function with a diminishing return rate, as elaborated in Section 4.4.2.1.

EH Source: The EH source uses the dataset presented in [30] to generate EH scenarios according to different user patterns. This dataset uses light and motion energy modalities as ambient energy sources. It combines power consumption measurements with the activity and location information of 4772 users from the American Time Use Survey dataset [117] to generate varying 24-hour EH patterns per user. We divide the EH dataset [30] into four clusters according to the users' EH patterns throughout the day. The hourly distributions of these four clusters are illustrated in Figure 4.15. These distributions are based on the mean and the

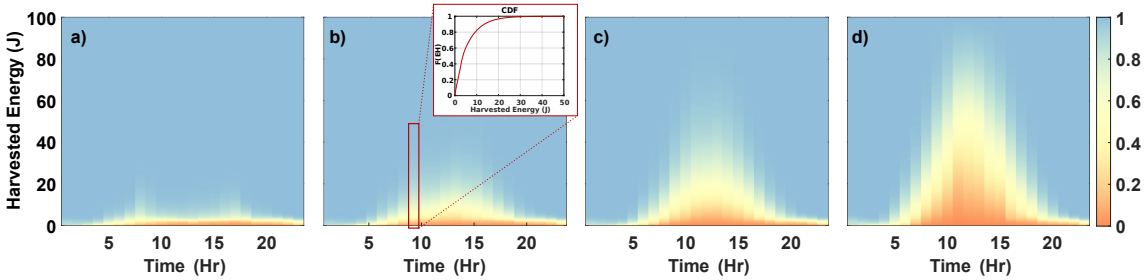


Figure 4.15: Cumulative distribution function of the harvested energy for a) Cluster 1, b) Cluster 2, c) Cluster 3, and d) Cluster 4

standard deviation of EH patterns in the same cluster. Therefore, the EH source generates a harvested energy value at every hour according to the distributions in the dataset as the day progresses.

4.4.1.2 Twin Delayed Deep Deterministic Policy Gradient (TD3) Algorithm

The objective of an RL agent is to maximize the cumulative reward by interacting with the environment. According to the state s of the environment and the current policy π, the agent chooses an action a. Based on this action, the environment

returns the next state s' and reward r. In this study, we adopt the Twin Delayed Deep Deterministic Policy Gradient [142] (TD3) algorithm, which extends the deep deterministic policy gradient (DDPG) [143] method due to its superior performance over state-of-the-art reinforcement learning (RL) approaches and its reduced sensitivity to hyperparameter tuning. The DDPG method consists of an actor that employs policy gradient methods to obtain the optimal policy and a critic that uses a deep Q network (DQN) to assess the actor's actions. In DDPG, the actor is trained to learn the correlation between state and action, whereas the critic is trained to establish the relationship between state-action pairs and the expected cumulative returns (Q-values). However, one limitation of DDPG is that the critic is prone to overestimating the target Q-value. This can lead to challenges with policy stability and convergence to local optima due to the approximation errors in the Q-value that are used to enhance the policy [142]. To address the overestimation error, TD3 [142] offers three significant improvements over DDPG: (i) clipped double Q-learning, (ii) target policy smoothing, and (iii) delayed policy updates.

Clipped double Q-learning: TD3 adopts two critic networks instead of one, which enables it to use the lower Q-value of the two to generate the targets in Bellman's optimality equation. Both critic networks are updated by TD3 using the following loss:

$$L_{critic}(\theta_i) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(y - Q(s_t, a_t; \theta_i)\right)^2\right] \tag{4.19}$$

$$y = r + \gamma \arg_Q \min_{i=1,2} Q(s_{t+1}, \tilde{a}; \theta'_i) \tag{4.20}$$

where $\mathcal{D}$ is the experience replay buffer that stores transitions (s_t, a_t, r_t, s_{t+1}) for every time step in the environment and y is the target value.

Target policy smoothing: When calculating target values, the target actor-network generates the action $\tilde{a}$. However, the DDPG approach can lead to target values with high variance, even for similar actions, because deterministic policies are susceptible to overfitting to sharp peaks in the value estimate. To mitigate this issue, the TD3 algorithm introduces some noise ϵ to the action instead of using the action given directly by the target actor-network, as follows:

$$\tilde{a} = \pi(s_{t+1}, \phi') + \epsilon : \epsilon \sim clip(\mathcal{N}(0, \sigma), -c, c) \tag{4.21}$$

The policy of the target actor-network for state s_{t+1} is denoted by $\pi(s_{t+1}, \phi')$, and the range of $-c$ to $+c$ denotes that any added noise is bounded to ensure that the target action stays close to the actual action. This regularization technique helps to decrease the variance in the target values.

Delayed policy updates: Deterministic policy gradient methods update the parameters of the actor-network by maximizing the Q-values, which are obtained utilizing the actions created by the actor-network as follows:

$$L_{actor}(\phi) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})\sim\mathcal{D}}\Big[Q(s_t, a_t; \theta_1)|_{a_t=\pi(s_t;\phi)}\Big] \tag{4.22}$$

This update rule can result in divergence during the agent's training, particularly when a suboptimal policy is overestimated. As a result, the agent may update on states with high error, leading to instability and a shift towards suboptimal

policies. To address this issue, TD3 adopts a less frequent update schedule for the actor-network compared to the value network. This approach of infrequent policy updates generates value estimates with lower variance and therefore facilitates the development of better policies.

4.4.2 tinyMAN – Lightweight Energy Manager

This section provides the environment dynamics and introduces the RL framework, the core algorithm used in tinyMAN.

4.4.2.1 Environment Dynamics

Our goal is to maximize the utilization of the device under certain battery energy level constraints. In our framework, the environment dynamics are determined to enable the adaptability of tinyMAN by any device and application.

State Space: The state is a 6-tuple that consists of:

- **Current battery energy** ($\frac{E_t^B}{E_{max}^B} \in [0,1]$)**:** The energy level of the battery at the beginning of the current step t divided by the battery capacity.
- **Harvested energy in the previous time step** ($\frac{E_{t-1}^H}{E_{max}^B} \in [0,1]$)**:** Harvested energy during the previous step $t-1$ divided by the battery capacity.
- **Time** ($t \in \mathbb{Z}$)**:** The current time step t, corresponding to the current hour of the day.
- **Initial battery energy level** ($\frac{E_0^B}{E_{max}^B} \in [0,1]$)**:** The energy level of the battery at the beginning of the episode ($t=0$) divided by the battery capacity.

- **Cumulative E^H ($\sum_{\tau=0}^{t-1} \frac{E_\tau^H}{E_{max}^B} \in \mathbb{R}$):** Cumulative harvested energy in the previous time steps divided by the battery capacity.

- **Cumulative E^A ($\sum_{\tau=0}^{t-1} \frac{E_\tau^A}{E_{max}^B}$):** Cumulative energy allocations in the previous time steps divided by the battery capacity.

Action Space: The action is the allocated energy at every time step ($E_t^A \in [E_{min}^A, E_t^B]$). Since the application on the device needs a minimum energy level to stay in the idle state, we set a minimum level constraint on the action (E_{min}^A).

Reward function: Our objective is to maximize the utility of the device under certain constraints on the battery energy level. tinyMAN supports any arbitrary utility function, but to have a fair comparison with the literature [26, 31], we use the following logarithmic utility function in this work:

$$u(E_t^A) = \ln\left(\frac{E_t^A}{E_{min}^A}\right) \tag{4.23}$$

In an RL setting, the reward function can impose the constraints on the battery. There are two constraints that can be imposed on the reward function: (*i*) emergency reservoir energy constraint (Equation 4.17) and (*ii*) ENO constraint (Equation 4.18). Considering the objective and the constraints on the battery, the reward function

becomes:

$$r_t = \begin{cases} u(E_t^A) & E_t^B \geqslant E_{min}^B \text{ and } t \neq T \\ -|E_t^B - E_{min}^B| & E_t^B < E_{min}^B \text{ and } t \neq T \\ -|E_t^B - E_{target}| & E_t^B \leqslant 0 \text{ and } t \neq T \\ -|E_t^B - E_{target}| & t = T \end{cases} \tag{4.24}$$

Here, we impose the emergency reservoir energy constraint using the term $-|E_t^B - E_{min}^B|$ and the ENO constraint using the term $-|E_t^B - E_{target}|$. Moreover, the agent is heavily punished using the term $-|E_t^B - E_{target}|$ if it depletes the battery ($E_t^B \leqslant 0$) during the episode. An episode terminates if time T is reached or the battery is completely drained.

According to the environment dynamics explained in this section, we develop our environment in Python and register it as an OpenAI's Gym [144] environment.

4.4.2.2 The Proposed tinyMan RL Framework

Algorithm 2 summarizes the training of the tinyMAN agent for a given cluster of users. First, we initialize an empty replay buffer $\mathcal{D}$ and critic and actor networks with random weights. We also initialize the target critic and actor networks with the same parameters. The networks consist of fully connected layers with three hidden layers and 64 neurons in each layer. At the start of each episode n, we randomly choose an EH pattern and an initial battery energy level. The EH pattern is generated using the hourly distributions depicted in Figure 4.15 and is unique for each episode, allowing tinyMAN to learn the EH patterns of users in that cluster.

Algorithm 2: tinyMAN - RL based Energy Manager

Input: Number of total time steps N, Minibatch size $\mathcal{B}$, Discount factor γ, Policy update delay p_{delay},
The standard deviation for smoothing noise added to target policy σ,
The standard deviation for Gaussian exploration noise added to the policy σ',
The clipping factor of the target policy smoothing noise c.
Initialize: Replay buffer $\mathcal{D}$, Critic networks $\mathbf{Q}_{\theta_1}$,$\mathbf{Q}_{\theta_2}$ and actor-network π_ϕ with parameters θ_1, θ_2, ϕ.
Target networks $\mathbf{Q}_{\theta_1'} \leftarrow \mathbf{Q}_{\theta_1}$, $\mathbf{Q}_{\theta_2'} \leftarrow \mathbf{Q}_{\theta_2}$, $\pi_{\phi'} \leftarrow \pi_\phi$
$done \leftarrow$ True
Parameter update count, $p_n = 0$.
while *Total Number of Steps* $<$ N **do**
    if $done =$ True **then**
        # End of an episode, start a new one
        Reset the environment to a randomly chosen initial battery energy E_0^B and an EH pattern.
        $done \leftarrow$ False
    Choose an action with exploration noise $a_t \sim \pi(s_t;\phi) + \epsilon : \epsilon \sim \mathcal{N}(0,\sigma')$
    Collect samples $\{s_t, a_t, r_t, s_t', done\}$ by interacting with the environment and store them in $\mathcal{D}$.
    Sample random $\mathcal{B}$ transitions from $\mathcal{D}$.
    $\tilde{a}_t \leftarrow \pi(s_{t+1},\phi') + \epsilon : \epsilon \sim clip(\mathcal{N}(0,\sigma), -c, c)$
    $y \leftarrow r + \gamma \arg_Q \min_{i=1,2} Q_{\theta_i'}(s_{t+1}, \tilde{a}_t)$
    $Loss_{critic}(\theta_i) = \mathbb{E}\Big[\Big(y - Q_{\theta_i}(s_t,a_t)\Big)^2\Big]$
    Update critic network parameters.
    $p_n = p_n + 1$
    if $p_n \bmod p_{delay}$ **then**
        $Loss_{actor}(\phi) = \mathbb{E}\Big[\, Q_{\theta_1}(s_t,a_t)|_{a_t=\pi(s_t;\phi)}\Big]$
        Update actor parameters and target network parameters.

The agent then interacts with the environment using its actor-network (π_ϕ). The resulting transitions $(s_t, a_t, r_t, s_{t+1}, done)$ are then stored in the experience replay buffer $\mathcal{D}$. The algorithm then samples a minibatch ($\mathcal{B}$) of transitions from $\mathcal{D}$. Using this minibatch of samples, the smoothed target action values are determined based

on the noise ϵ and the clipping factor c. The target value y is computed using the target critic-networks and target action values. Finally, the actor and critic networks are updated using the loss functions described in Section 4.4.1.2. The training concludes once the total number of time steps reaches N.

The EH dataset [30] is divided into four clusters according to the users' EH patterns throughout the day. The agent is trained separately on each cluster. We implement tinyMAN in Python by utilizing PFRL [145] library for the TD3 algorithm using Adam optimizer with a learning rate of 1E-4. Since the proposed tinyMAN framework is deployed on a wearable device, the characteristics of the target device, such as battery capacity, minimum battery energy level (E^B_{min}), and minimum energy allocation (E^A_{min}) should be identified before the training. These characteristics do not change over time during the training. We set the emergency reservoir energy as $E^B_{min} = 10$ J, roughly corresponding to 5 minutes of active time for the components listed in Table 4.4. This parameter and others, as listed in Table 4.5, can be tailored according to the requirements of another device or application.

Table 4.5: Definition of the hyperparameters for tinyMAN and their values

Parameter	Description	Value	Parameter	Description	Value
N	Total time steps	10^6	σ	TP* smoothing noise std.	0.2
$\mathcal{B}$	Minibatch size	100	c	Clipping factor of TP* noise	0.5
$\mathbb{D}$	Replay buffer size	10^5	N_{Layer}	# hidden layers	3
γ	Discount factor	0.995	N_{Neuron}	# hidden neurons	64
p_{delay}	Policy update delay	2	lr	Learning rate	10^{-4}
σ'	Exploration noise std.	0.3	E^B_{min}	Reservoir energy	10

*TP: Target Policy

4.4.3 tinyMAN-MO – Addressing Varying Demand by the Application

So far, the proposed tinyMAN framework had a single objective; to maximize the utility under ENO constraints. It also considers a stationary utility function, i.e., the shape of the utility function does not change over the horizon. This assumption holds for many continuous monitoring applications, such as heart or respiration rate monitoring, where the application operates within the allocated energy budget without making explicit demands. In contrast, many other applications, such as sleep trackers, pedometers, or activity trackers, can demand energy and have utility functions that vary over time. For instance, a workout tracker demands high energy and produces utility when the user is active (i.e., working out or playing sports). Therefore, meeting the varying demand is critical for many applications. Motivated by these observations, we present tinyMAN-MO, the multi-objective extension of tinyMAN, to co-optimize meeting the application's varying demand and maintaining a non-zero battery level.

4.4.3.1 Multi-objective Problem Formulation

The introduction of meeting the application's demand as a new constraint may contradict the ENO constraints presented in Section 4.4.1.1. For example, if the total energy demand of the application exceeds the harvested energy in the horizon T, it is not possible to achieve energy neutrality. Therefore, we transform the problem into a multi-objective problem with two conflicting goals: meeting the application's demand and maintaining a non-zero battery energy level.

Energy Demand: The target application considered in this study is an activity tracker, chosen as a representative example. However, it is crucial to emphasize that the approach presented here is not limited to this specific application and can be effectively applied to a wide range of other applications. Activity tracker application demands more energy when the user is active. Thus, we generate varying demand curves based on each user's activity information in the American Time Use Survey dataset. There are 10 activity labels in the pre-processed dataset in [30]. We map these labels to three activity levels as {*Active* task, *Daily* task, *Not active* task}. For example, "exercise" falls under *Active* task, while "personal care" and "work" falls under *Daily* task, and "sleep" and "eat" falls under *Not active* task. As a result, we obtain daily activity level patterns for 4772 users with 1-minute resolution, as illustrated for 50 randomly selected users in Figure 4.16-a) using a lasagna plot. As one can expect, the early and late hours of the day and lunchtime are mostly *Not active*. Then, by using the energy consumption values of three commercial products:

i. *Not active*: 50 mJ/min – Oura Ring [27]

ii. *Daily*: 140 mJ/min – Amazon Halo [146]

iii. *Active*: 480 mJ/min – Whoop Strap [147]

we convert the activity level information into the application's hourly energy demand. As an example, Figure 4.16-b) shows the calculated hourly demand for a randomly selected user in Figure 4.16-a). This user is active around midday, which

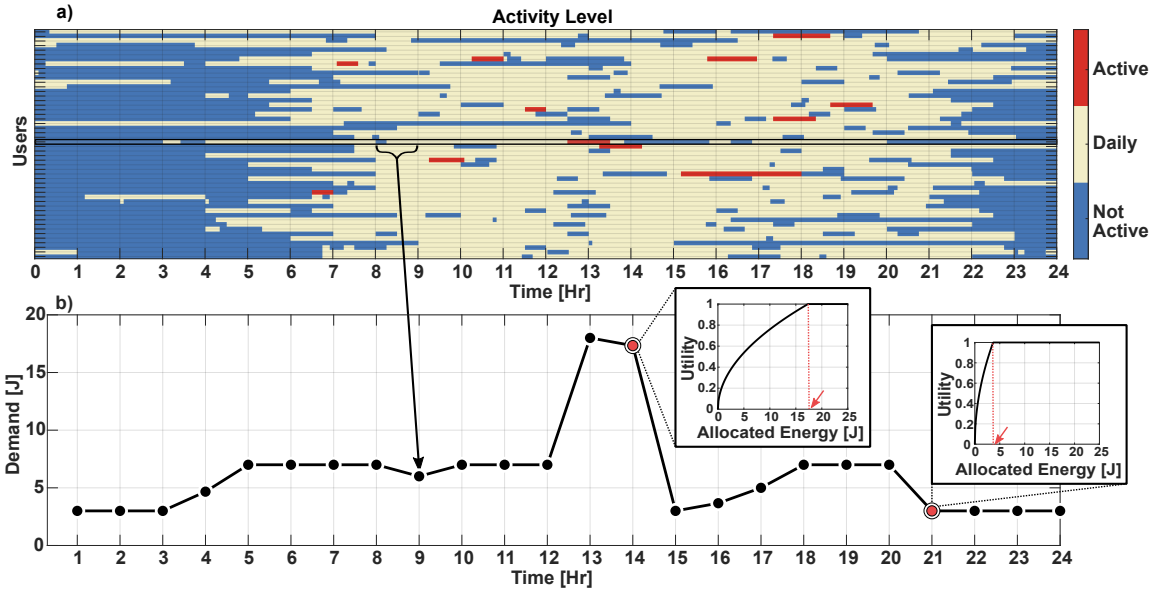


Figure 4.16: Illustration of the varying application demand for different users. a) Activity levels of randomly selected 50 users in the dataset. Each row is a user, each box in the grid corresponds to 60 minutes, and colors correspond to activity levels. b) The demand for one of the users. The insets show the different utility curves due to different demands.

results in high demand during that time. Similarly, after the active period, the resting/eating section leads to low application demand.

Utility: We revise the utility function in Section 4.4.1.1 to incorporate the demand:

$$u(E_t^A, D_t) = \begin{cases} \sqrt{\frac{E_t^A}{D_t}} & E_t^A < D_t \\ 1 & E_t^A \geqslant D_t \end{cases} \tag{4.25}$$

where D_t denotes the demand at time step t. In this case, the utility function equals 1 when $E_t^A \geqslant D_t$, which means *the demand is met*, and allocating more energy does not produce additional utility. Moreover, we use *square-root* instead of *log* to have a

slower rate of decrease in utility when $E_t^A < D_t$. We keep the function's concavity, not to dismiss the ability to solve it optimally using convex optimization solvers. If this was not a concern, the utility could be set to zero for $E_t^A < E_{min}^A$, similar to Section 4.4.1.1. We emphasize that tinyMAN-MO supports any arbitrary utility function and that the utility function can have any shape according to the needs of the application. As a result, we obtain a time-varying utility function characterized by the application's demand. For instance, the insets in Figure 4.16-b) depict two different cases for the utility function. At $t = 14$, the application's demand is 17.33 J, and the utility is 1 if that amount of energy is allocated to the application. Similarly, at $t = 21$, the application demands 3 J to yield a utility of 1. With this revised utility function, *meeting the application's demand is equivalent to maximizing the utility*.

4.4.3.2 Multi-objective Environment Dynamics

tinyMAN-MO's main objective is to learn the trade-off between meeting the application's varying demand and the battery energy level. We revise the environment dynamics explained in Section 4.4.2.1 to reflect this change.

State Space: We add the demand for the current time step to the state vector, making it a 7-tuple:

- **Demand** ($\frac{D_t}{E_{max}^B} \in [0, 1]$)**:** Demand at the current time step t divided by the battery capacity.

Action Space: The action is identical to tinyMAN and corresponds to the allocated energy at every time step ($E_t^A \in [E_{min}^A, E_t^B]$).

Reward function: The multi-objective extension incorporates two conflicting goals: meeting the application's demand and maintaining a non-zero battery energy level. Based on the revised utility function, we express meeting the application's demand as maximizing the utility. Thus, the agent aims to learn the balance between maximizing the utility function (given by Equation 4.25) and maximizing the battery energy level while adhering to specific battery-related constraints. As the revised utility function produces values between 0 and 1, we use the percent battery energy level as the second objective to ensure compatible scales between the two objectives. Consequently, the reward function now produces a vector of two elements as follows:

$$r_t = \begin{cases} [u(E_t^A, D_t), \frac{E_t^B}{E_{max}^B}] & E_t^B \geqslant E_{min}^B \\ [u(E_t^A, D_t), -\frac{E_t^B}{E_{max}^B} + 1] & E_t^B \geqslant E_{max}^B \\ [-|E_t^B - E_{min}^B|, -|E_t^B - E_{min}^B|] & E_t^B \leqslant E_{min}^B \end{cases} \tag{4.26}$$

In both objectives, the emergency reservoir energy constraint is incorporated by including the term $-|E_t^B - E_{min}^B|$. The reward vector for this condition has a scale of an order of magnitude higher than the first two conditions. This implies that if the battery level falls below a minimum level ($E_t^B \leqslant E_{min}^B$) during the episode, the agent will be heavily penalized. Additionally, a negative reward is given for the battery level objective if it exceeds the maximum battery capacity to ensure that saving more energy than the battery capacity is not advantageous. An episode terminates when either time $T = 24$ is reached or the battery is entirely depleted.

4.4.3.3 Multi-objective RL Framework

Existing multi-objective reinforcement learning (MORL) methods transform the multidimensional objective space into a single dimension using static weights for each objective and then employ standard RL algorithms to obtain a policy optimized for those weights [148]. Repetitively solving a multi-objective optimization problem by scalarizing the multidimensional objective space is not feasible, and it requires domain expertise in order to assign preferences (ω) to each objective statically. Recent MORL approaches suggest training a single policy network that covers the entire preference space [149, 150, 11]. The goal is to obtain a policy that yields a solution on the Pareto front for a given preference. The Pareto front is a set of solutions that are non-dominated by each other but are superior to the rest of the solutions in the solution space. To accomplish this, we utilize a multi-objective version of TD3 (MO-TD3) [11] to obtain a single policy network that covers the entire preference space. MO-TD3 is an extension of the TD3 algorithm [142] where the network also takes preferences (ω) of the objectives as inputs along with the states. It also employs Hindsight Experience Replay buffer [151] to efficiently explore the preference space and novel parallelization approach to increase the sample efficiency of the algorithm [11].

Since our multi-objective environment returns a reward vector, the Q-network is also vectorized to efficiently learn to model multiple objectives for a given preference vector ω. Specifically, the network takes state s and preference vector ω as inputs and outputs a vector of Q-values. Hence, the update rule of the vectorized network

is modified as follows:

$$L_{critic}(\theta_i) = \mathbb{E}_{(s_t, a_t, \mathbf{r_t}, s_{t+1}, \boldsymbol{\omega}) \sim \mathcal{D}} \left[\left(y - \mathbf{Q}(s_t, a_t, \boldsymbol{\omega}; \theta_i) \right)^2 \right] \tag{4.27}$$

$$\mathbf{y} = \mathbf{r} + \gamma \arg_{\mathbf{Q}} \min_{i=1,2} \mathbf{Q}(s_{t+1}, \tilde{a}, \boldsymbol{\omega}; \theta'_i) \tag{4.28}$$

The actor-network also takes the preference as inputs, and the parameters of this network are updated by maximizing the $\boldsymbol{\omega}^T \mathbf{Q}$ using the following loss:

$$L_{actor}(\phi) = \mathbb{E}_{(s_t, a_t, \mathbf{r_t}, s_{t+1}, \boldsymbol{\omega}) \sim \mathcal{D}} \left[\mathbf{Q}(s_t, a_t, \boldsymbol{\omega}; \theta_1)|_{a_t = \pi(s_t, \boldsymbol{\omega}; \phi)} \right] \tag{4.29}$$

Similar to tinyMAN, we first identify the characteristics of the target device: the minimum battery energy level as $E^B_{min} = 1$ J to allow the tinyMAN-MO agent to cover the corner cases where the battery is drained and the minimum energy allocation as $E^A_{min} = 0.64$ J. The agent is trained separately on each of the four clusters. Algorithm 3 outlines the training of the tinyMAN-MO agent. Different from tinyMAN, at the beginning of each episode during training, we randomly sample a preference vector ($\boldsymbol{\omega} \in \Omega : \sum_{i=0}^{L} \omega_i = 1$) from a uniform distribution. For a given preference, the agent interacts with the environment using its actor-network (π_ϕ) and obtains transitions $(s_t, a_t, \mathbf{r_t}, s_{t+1}, \boldsymbol{\omega}, done)$, which are then stored in the experience replay buffer $\mathcal{D}$. To explore the preference space, we also store N_ω transitions for each transition, where a different preference ($\boldsymbol{\omega}' \in \Omega : \sum_{i=0}^{L} \omega'_i = 1$) is sampled. These additional transitions $(s, a, \mathbf{r}, s', \boldsymbol{\omega}', done)$ are also stored in $\mathcal{D}$. The remaining steps are similar to tinyMAN, except that the critic networks yield vectorized Q-values and that the actor and critic networks are updated using the

loss functions described in this section. The training ends after the total number of time steps reaches N. The steps that differ from the tinyMAN implementation (Algorithm 2) are highlighted with blue in Algorithm 3.

We implement tinyMAN-MO in Python using Adam optimizer with a learning rate of 1E-4. The hyperparameters for tinyMAN-MO are presented in Table 4.6.

Table 4.6: Definition of the hyperparameters for tinyMAN-MO and their values

Parameter	Description	Value	Parameter	Description	Value
N	# time steps	3×10^6	σ'	Exploration noise std.	0.1
$\mathcal{B}$	Minibatch size	128	σ	TP* smoothing noise std.	0.2
$\mathbb{D}$	Replay buffer size	5×10^5	c	Clipping factor of TP* noise	0.5
γ	Discount factor	0.995	N_{Layer}	Number of hidden layers	3
N_ω	Number of prefs-HER	3	N_{Neuron}	Number of hidden neurons	64
p_{delay}	Policy update delay	10	lr	Learning Rate	10^{-4}

*TP: Target Policy

4.4.4 Experimental Evaluations

4.4.4.1 Single-objective Results

We compare the performance of tinyMAN to three prior prediction-based approaches in the literature [26, 25, 31], and also to an oracle solution. The oracle solution is obtained by an offline solver (e.g., CVX) using the actual harvested energy during the day. We emphasize that this oracle solution is unfair and unrealistic, but it serves as a reference point since it provides the maximum theoretical utility achievable by the given harvested energy.

For each cluster, we first sort all users in the cluster according to their cumulative daily harvested energy. Then, the users corresponding to the 25%, 50%, 75% of this sorted list are excluded from the training set for illustration purposes. Additionally,

Algorithm 3: tinyMAN-MO - Multi-objective RL based Energy Manager

Input: Number of total time steps N, Minibatch size $\mathcal{B}$, Discount factor γ, Policy update delay p_{delay},
The standard deviation for smoothing noise added to target policy σ,
The standard deviation for Gaussian exploration noise added to the policy σ',
The clipping factor of the target policy smoothing noise c, Number of preferences sampled for HER N_ω.
Initialize: Replay buffer $\mathcal{D}$, Critic networks $\mathbf{Q}_{\theta_1}$,$\mathbf{Q}_{\theta_2}$ and actor-network π_ϕ with parameters θ_1, θ_2, ϕ.
Target networks $\mathbf{Q}_{\theta_1'} \leftarrow \mathbf{Q}_{\theta_1}$, $\mathbf{Q}_{\theta_2'} \leftarrow \mathbf{Q}_{\theta_2}$, $\pi_{\phi'} \leftarrow \pi_\phi$
$done \leftarrow$ True
Parameter update count, $p_n = 0$.
while *Total Number of Steps* $< N$ **do**
  if $done =$ True **then**
    # End of an episode, start a new one
    Reset the environment to a randomly chosen initial battery energy E_0^B and an EH pattern.
    Sample a preference vector $\boldsymbol{\omega}$.
    $done \leftarrow$ False
  Choose an action with exploration noise $a_t \sim \pi(s_t, \boldsymbol{\omega}; \phi) + \epsilon : \epsilon \sim \mathcal{N}(0, \sigma')$
  Collect samples $\{s_t, a_t, \mathbf{r}_t, s_t', \boldsymbol{\omega}, done\}$ by interacting with the environment and store them in $\mathcal{D}$.
  for $j = 1$: N_ω **do**
    Store transition $(s_t, a_t, \mathbf{r}_t, s', \boldsymbol{\omega}_j', done)$ in $\mathcal{D}$
  Sample random $\mathcal{B}$ transitions from $\mathcal{D}$.
  $\tilde{a}_t \leftarrow \pi(s_{t+1}, \boldsymbol{\omega}; \phi') + \epsilon : \epsilon \sim clip(\mathcal{N}(0, \sigma), -c, c)$
  $\mathbf{y} \leftarrow \mathbf{r} + \gamma \arg_{\mathbf{Q}} \min_{i=1,2} \mathbf{Q}_{\theta_i'}(s_{t+1}, \tilde{a}_t, \boldsymbol{\omega})$
  $Loss_{critic}(\theta_i) = \mathbb{E}\left[\left(\mathbf{y} - \mathbf{Q}_{\theta_i}(s_t, a_t, \boldsymbol{\omega})\right)^2\right]$
  Update critic network parameters.
  $p_n = p_n + 1$
  if $p_n \bmod p_{delay}$ **then**
    $Loss_{actor}(\phi) = \mathbb{E}\left[\mathbf{Q}_{\theta_1}(s_t, a_t, \boldsymbol{\omega})|_{a_t = \pi(s_t, \boldsymbol{\omega}; \phi)}\right]$
    Update actor parameters and target network parameters.

we randomly select 10% of the remaining users as test users and also removed them from the training data for evaluation purposes. Furthermore, for each cluster, we

evaluate the performance of our approach at four different initial battery energy levels: $E_0^B = \{16, 48, 112, 144\}$J.

Figure 4.17 summarizes the average daily utility and the total number of constraint violations obtained for all approaches using the test users for each cluster. In

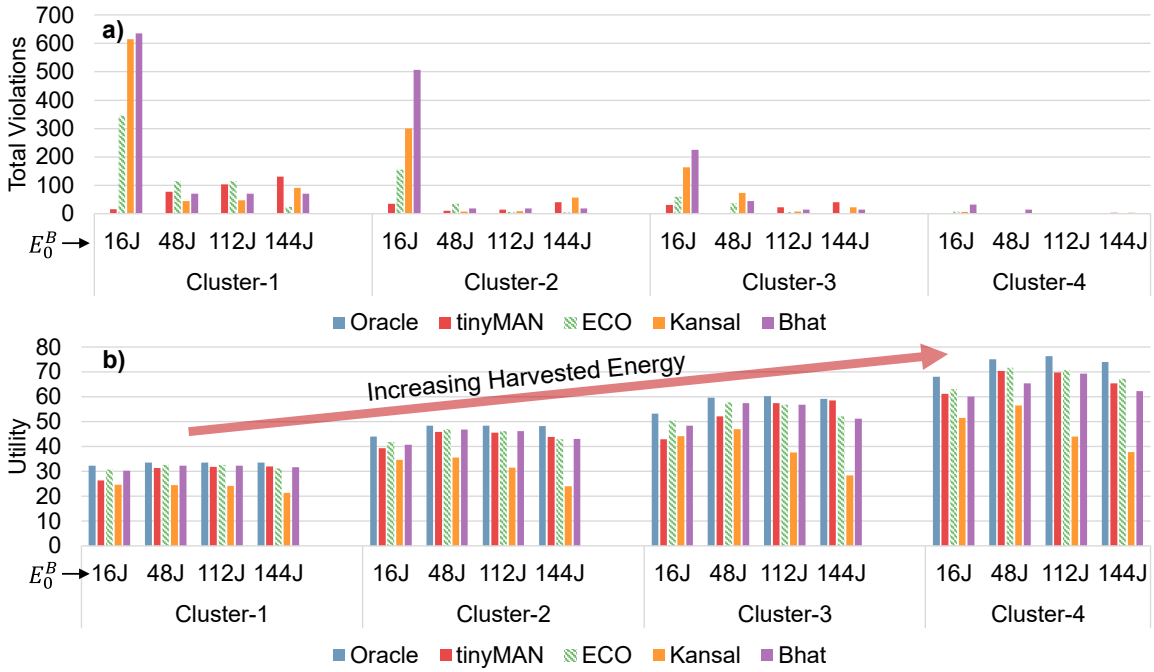


Figure 4.17: Summary of the performance of all approaches. a) Total constraint violations b) Average utility.
Kansal: [25], Bhat: [26]

general, we observe that the total utility increases as the harvested energy increases from cluster 1 to cluster 4 since the available energy to allocate on the device increases. *tinyMAN always stays within 10% of the optimal utility. Moreover, tinyMAN causes at least 20% fewer constraint violations than prior approaches*, as depicted in Figure 4.17-b). Specifically, it outperforms prior approaches in scenarios with low battery energy levels (16J) by achieving 80% fewer violations. We also note that

tinyMAN violates the ENO constraint by only a small margin in scenarios with high battery energy levels. Finally, we emphasize that tinyMAN is trained for various battery energy levels and EH patterns and does not require EH predictions during run-time. In contrast, the performance of prior approaches heavily relies on accurate predictions.

Figure 4.18-(1a, 1b, 1c) illustrates the allocations made by tinyMAN for the initial battery level of $E_0^B = 16$J for three users (25%, 50%, 75%) in cluster 4. In addition to the behavior of the tinyMAN agent, Figure 4.18 also illustrates the energy allocations computed by three prior prediction-based approaches and the offline oracle solution. As the prior approaches are prediction-based, they use the specific expected EH pattern for a user, depicted with the blue line in Figure 4.18-(3a, 3b, 3c). On the contrary, tinyMAN implicitly learns the actual EH patterns during training, making it a prediction-free approach.

Finally, the oracle solution uses the actual harvested energy during the day, depicted with the red line in Figure 4.18-(3a, 3b, 3c). It can be seen that tinyMAN's allocations (actions) are around the optimal values shown with the red line in Figure 4.18-(1a, 1b, 1c). In contrast, prior approaches tend to deplete the battery and severely under allocate to avoid depletion, especially in the early hours of the day (before 10 AM). As a result of this, they often violate minimum and target battery energy level constraints. Moreover, this behavior causes their total daily utility, $U = \sum_{t=0}^{T} u(E_t^A)$, to decrease. If the deviation between the actual and expected EH values were greater in Figure 4.18-(3a, 3b, 3c), the performance of prior approaches would deteriorate significantly. Indeed, tinyMAN achieves up to

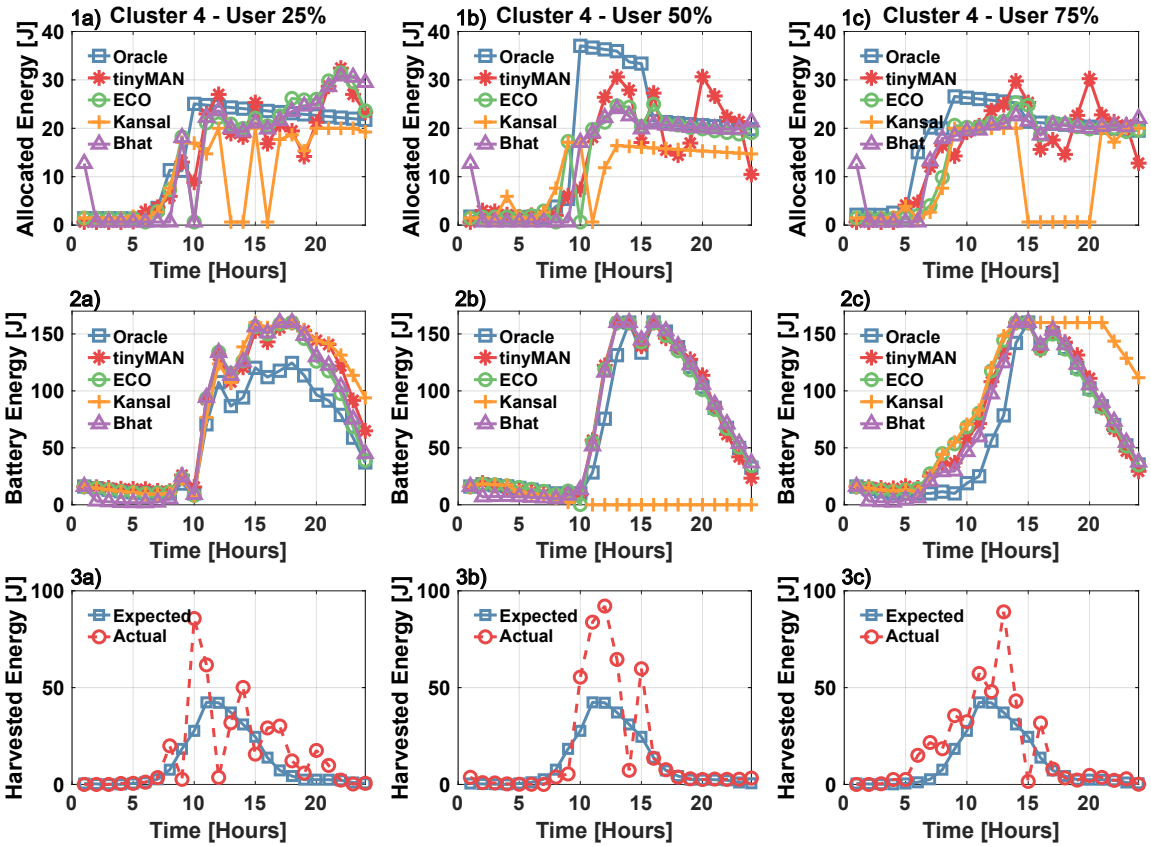


Figure 4.18: Comparison of tinyMAN with the prior approaches and the offline oracle solution for Cluster 4.
Kansal: [25], Bhat: [26]

17% higher utility than the prior approaches and within 8% of the oracle solution *while not causing any violations* for the three example cases depicted in Figure 4.18. In contrast, [31] has 6, [26] has 23, and [25] has 24 total violations. Thus, we argue that tinyMAN is the preferred energy management solution for wearable edge devices operating under conditions of uncertainty in harvested energy.

4.4.4.2 Multi-objective Results

We compare the performance of tinyMAN-MO to an oracle solution obtained by an offline solver (e.g., CVX) using the actual harvested energy during the day. The offline solver transforms the objective vector into a scalar using the preferences for each objective. Similar to the single objective case, this oracle solution is unfair and unrealistic but serves as a reference point.

Similar to tinyMAN, we first sort all users in a cluster according to their cumulative daily harvested energy. Then, the users that correspond to the {10%, 25%, 50%, 75%, 90%} of this sorted list are excluded from the training set for evaluation purposes. For each user, we evaluate the performance of tinyMAN-MO at four different initial battery energy levels: $E_0^B = \{16, 48, 112, 144\}$J. Furthermore, we divide the preference space into twenty equally spaced vectors {[0,1], [0.05, 0.95], ..., [1,0]}. Using this set of preferences, Figure 4.19 illustrates the Pareto-front solutions obtained by tinyMAN-MO and the Pareto-optimal curve of the oracle for three users (25%, 50%, 75%) for all clusters. In this plot, the x-axis shows the percentage of the total daily demand met (objective-1), and the y-axis shows the average battery level during the day (objective-2). Consequently, the left-most data point in a given curve corresponds to the preference vector of [0,1] (i.e., only care about the battery level), and the right-most data point corresponds to [1,0] (i.e., only care about meeting the demand). Therefore, for a preference of [0,1], the agent ignores the demand and always allocates $E_t^A = E_{min}^A$, unless the battery is fully charged by the harvested energy. If the battery is at capacity, the agent allocates the harvested energy directly to the application, inadvertently increasing objective-1.

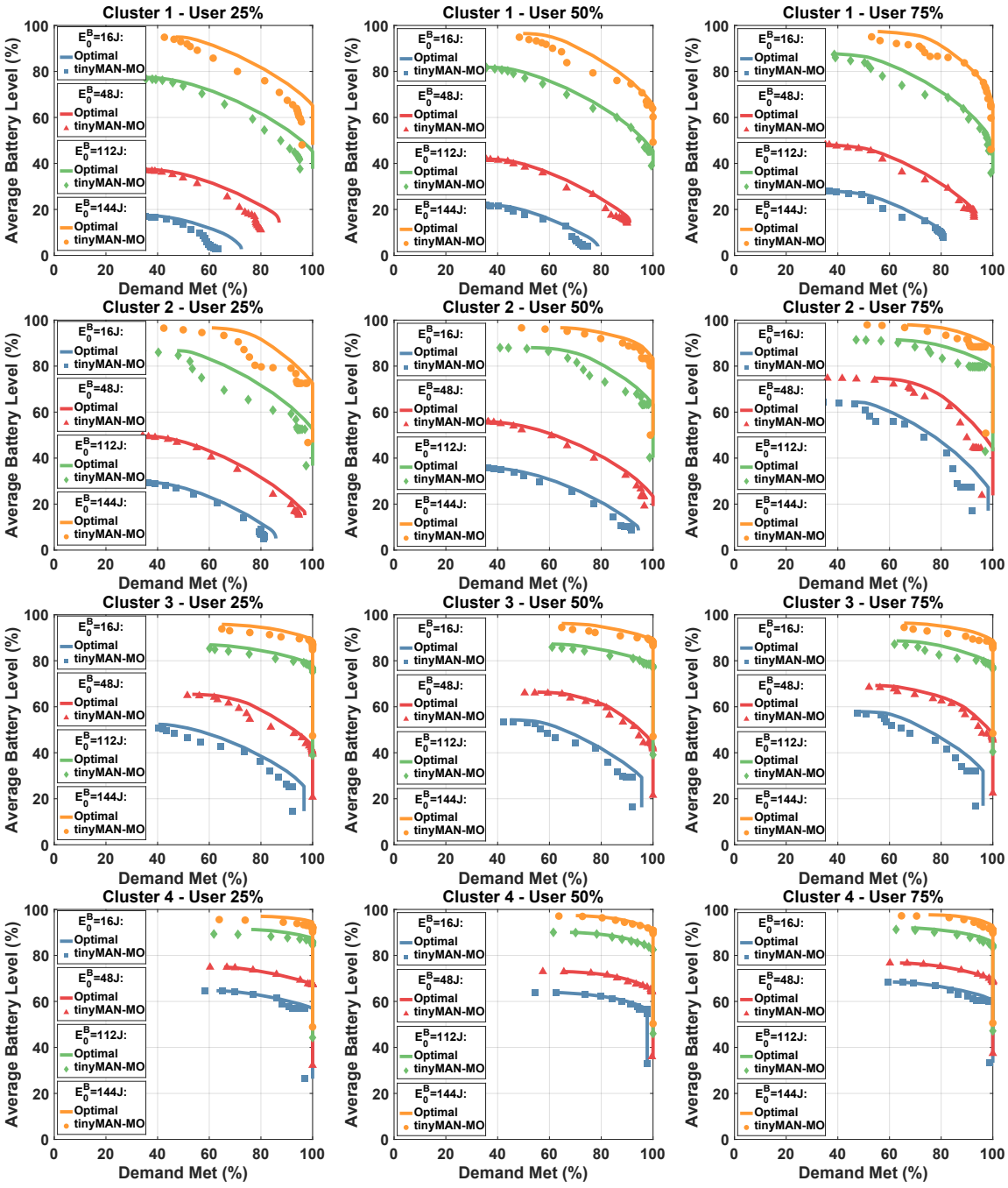


Figure 4.19: The Pareto front solutions obtained by tinyMAN-MO and the optimal solution.

For a preference of [1,0], the agent aims to meet the demand only and may hit the minimum battery level constraint, as shown in 1a). Also, when energy is abundant, the demand can be met at a preference of as early as [0.5,0.5], which appears as vertical "falls" on the right side of the plots. Finally, both objectives increase in value as the harvested energy increases from Cluster 1 to Cluster 4, as expected. In general, the Pareto-front solutions obtained by tinyMAN-MO are very close to the Pareto-optimal solutions obtained by the oracle, as summarized in Figure 4.20. Specifically, the mean absolute percentage error between the Pareto curves is always less than 10%. Moreover, the MAPE decreases as harvested energy increases.

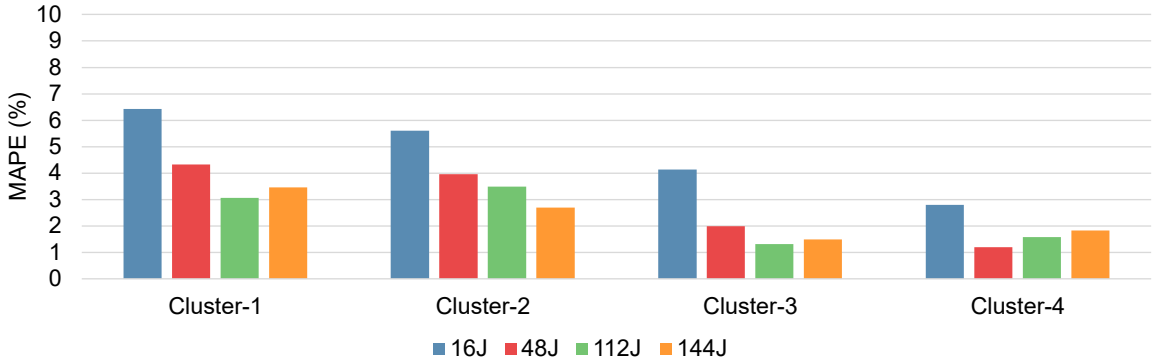


Figure 4.20: Summary of the mean absolute percentage error from five test users.

4.4.5 Deployment on a Wearable Device

The TI CC2652R microcontroller used on our prototype device incorporates an ARM Cortex M4F running at 48 MHz and has 352KB of flash memory and 80KB of SRAM. These scarce resources highlight the importance of evaluating the trained RL agent regarding its deployability on the target platform. As the network architecture used by tinyMAN and tinyMAN-MO is the same, their memory footprint,

energy consumption, and execution time overhead are nearly identical. Thus, in this section, we collectively refer to both frameworks as tinyMAN. Therefore, we evaluate the deployability of tinyMAN from three aspects: (i) The execution time per inference, (ii) the energy consumption per inference, and (iii) the memory utilization of the target hardware platform. To do this analysis, we follow the Tensorflow Lite Micro (TFLM) flow to convert and deploy the trained models on the target device [152]. Then, we measure the current consumption of the TI microcontroller. Using these measurements, the execution time and energy consumption overhead of a single policy network call of the proposed tinyMAN are measured as 1.98 ms and 23.17 μJ, respectively. As a result, the total time and energy consumption of the tinyMAN framework for a single day is 47.52 ms and 0.556 mJ, respectively. These values are insignificant compared to the 24-hour period and 160 J battery capacity. Additionally, we use the "Memory Allocation" report of TI Code Composer Studio to obtain the memory utilization of the device. The memory footprint of tinyMAN becomes 118 KB, which easily fits into the onboard memory of the target device. We note that the reported memory footprint is for the entire application, including necessary drivers and I/Os for debugging, such as UART and timers. These results suggest that tinyMAN is easily deployable on a resource-constrained wearable IoT device.

5 SW/HW OPTIMIZATIONS TO MINIMIZE ENERGY CONSUMPTION

5.1 Motivation and Background

Wearable devices collect data from various sensors, such as accelerometers and biopotentials, to enable sophisticated health monitoring applications, as illustrated in Figure 5.1. The decadal plan from the Semiconductor Research Corporation highlights that the collective data acquisition rate of sensors will be as high as 10^{20} bits/sec by 2032 [153]. Offloading the sensor data to another device or cloud is prohibitive since wireless data transfer is power-hungry [153, 92]. Therefore, wearable devices must be capable of implementing complex artificial intelligence (AI) and signal processing algorithms at the edge, which requires powerful microprocessors. At the same time, wearable devices also have severely limited battery and energy harvesting capacity due to small form factor requirements, as discussed in the earlier chapters. For example, the Fitbit Inspire 2 consumes ~1.5 mW on average, while average wearable energy harvesting potential hardly reaches 100 μW, leading to a substantial (>15×) gap between the required and available energy budget [154, 30]. Therefore, there is a strong need for approaches that enable practical wearable devices that satisfy both computational power and energy requirements.

The energy gap can be bridged by two complementary efforts: reducing the energy consumption requirements and optimizing the harvested energy. Chapters 3 and 4 have focused on the latter. Instead, the current chapter focuses on the former; it presents two studies for reducing the energy consumption of wearable applications.

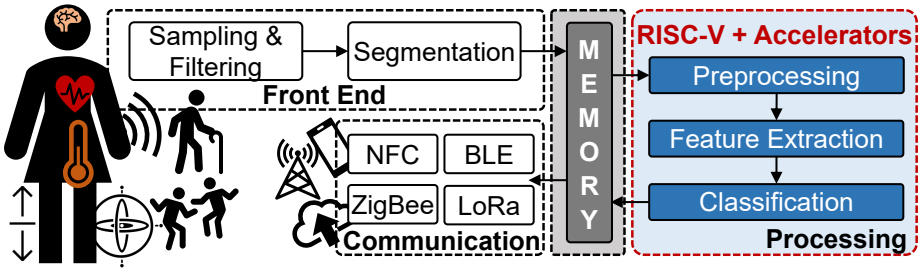


Figure 5.1: Main building blocks in a typical wearable edge AI app.

5.2 Contributions and Overall Goals

Historically, general-purpose microprocessors have been favorable for implementing wearable applications since they provide the flexibility to implement a wide range of embedded applications. Indeed, most commercial hardware platforms employ Arm or RISC-V cores as the primary processor [92, 93, 94]. However, SW programmability comes at the expense of low energy efficiency. Achieving low energy consumption with SW implementations is challenging and requires significant engineering effort. As a representative study, this section presents a cattle health monitoring application that depicts ultra-low power consumption with a 36× reduction in memory footprint compared to a raw-data based SW implementation thanks to detailed data and feature engineering.

As an alternative to pure SW solutions, custom HW accelerators address this problem and provide 2–3 orders of magnitude higher energy efficiency by achieving faster operation and lower power consumption [101, 96, 95]. However, their benefits diminish if programming the SoC requires significant human effort to support rapidly evolving applications [155, 153]. Hence, there is a strong need for SoCs that can approach the energy efficiency of custom designs while preserving

programmability. To this end, this chapter presents an open-source domain-specific programmable system-on-chip (SoC) that combines a RISC-V core with a meticulously determined set of accelerators targeting wearable applications. The proposed design is deployed on an FPGA and it executed six real-life use cases to demonstrate the efficacy of the proposed SoC. Thorough experimental evaluations show that the proposed SoC provides up to 9.1× faster execution and up to 8.9× higher energy efficiency than software implementations in FPGA *while maintaining programmability*.

In summary, the major contributions of this section are as follows:

- A heavily optimized cattle monitoring application on TI CC2652R microcontroller with only 29 µJ daily energy consumption,
- A domain-specific system-on-chip (SoC) design for wearable edge AI-based health monitoring devices that satisfy two competing requirements: programmability and high energy efficiency.

5.3 Towards Smart Cattle Farms: Automated Inspection of Cattle Health with Real-Life Data

This section presents a cattle health monitoring application that depicts ultra-low power consumption with a 36× reduction in memory footprint compared to a raw-data based SW implementation thanks to detailed data and feature engineering. To achieve such savings, we thoroughly analyze accelerometer data collected from 54 cows to identify patterns that relate to their daily behavior. Then, we systematically

construct 33 features that enable us to classify healthy and sick animals using eleven shallow decision tree (DT) classifiers and majority voting. Our approach outperforms *thirteen state-of-the-art* (*SOTA*) *time series classifiers* in the literature, with 78% accuracy in differentiating healthy and sick cows. Moreover, the proposed approach uses only 1 KB on-chip SRAM and consumes 29 µJ in a day on a TI CC2652R microcontroller, enabling a low-power implementation at the edge.

5.3.1 Background

Cattle diseases have a significant negative impact not only on the animals' welfare but also on the economic performance of the cattle industry [156, 157, 158]. For example, Bovine Respiratory Disease (BRD) is responsible for approximately 75% of the morbidity and 57% of the mortality in US feedlots, which is estimated to cost the agriculture industry about $1B annually [158, 156]. Traditionally, feedlot personnel evaluate cattle health subjectively based on cattle behavioral and appearance factors such as abnormal eye, nasal discharge, head tilt and others. This often leads to low diagnostic accuracy as cattle tend to mask the symptoms and signs of weakness in the presence of a potential predator [156]. To address this, DART (Depression, Appetite, Respiration, and Temperature) scoring is used as a unified method of indicating abnormal cattle [159].

Although DART is a more objective indicator, it still requires a significant amount of manual effort to inspect every animal individually. Moreover, it relies on skilled personnel, which is a limiting factor due to labor-shortage in several industry sectors, including agriculture [160]. As the herd size per farm increases [161],

this labor-intensive diagnosis is becoming increasingly infeasible. In contrast, an automated internet-of-things (IoT) solution can monitor cattle continuously without human presence and flag abnormal behavior. It can also increase the probability of identifying subtle changes due to disease that could otherwise be masked by cattle or are undetected by traditional indicators. Therefore, a continuous and automated IoT solution to predict the health state of a cow is a critical tool for the cattle industry. *To this end, this study illustrates how to use sensor data* (*in our case 3-axis accelerometer*) *and DART scores to differentiate healthy and sick animals.*

Existing wearable cattle health systems use a plethora of indicators for detecting diseases, such as animal activity and behavior, respiration rate, and coughing, among others [162, 163]. In particular, animal activity and behavior have been shown to change with disease progression. For example, changes in cattle feeding and resting time behavior are observed as a function of disease status [164]. Moreover, location monitoring and accelerometer data have shown that sick animals exhibit lower step counts and decreased standing time compared to healthy animals [156].

In order to track animal activity and behavior, prior studies in the literature collect data from various sensors, including accelerometers, gyroscopes, magnetometers, temperature and humidity sensors, and heart rate sensors [157, 165, 166, 167, 168, 169]. Some of these solutions also require multiple devices to be deployed on animals [167, 168]. Using multiple devices with multiple sensors might cause synchronization issues and incur energy consumption overhead to address such problems. Furthermore, *the prior approaches transmit the data to a central*

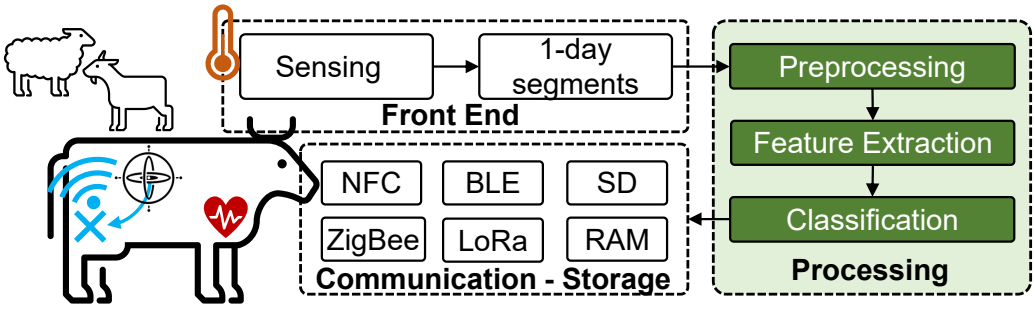


Figure 5.2: Building blocks of a smart animal monitoring system.

hub (e.g. a server machine) and continue the processing there. This approach has two major drawbacks: i) It relies on the presence of a costly, computationally capable server and line-of-sight communication to it, and ii) data transmission is power-consumptive and leads to faster battery consumption, which results in frequent recharge interventions. In contrast, a smart cattle health system, illustrated in Figure 5.2, can collect and process data at the edge. Local, real-time processing enables timely diagnosis of abnormal animals, and transmitting only the predicted outcome saves significant communication energy, one of the major sources of energy consumption. As such, the amount of transmitted data is much smaller than traditional approaches, significantly improving battery life. Furthermore, fewer human interventions to collect data and replace batteries improve the reliability and robustness of the measurements and the monitoring mechanisms.

5.3.2 Dataset Overview

This section first explains the procedure used for data collection. Then, it presents the preprocessing steps.

5.3.2.1 Dataset Collection

This experiment was conducted with 54 heifers (young cow who has not given birth) at a Livestock Research Center over 25 days following a protocol approved by Institutional Animal Care and Use Committee. A 4-point scale technique based on DART as defined by [159] was used by trained personnel to track animal health every day throughout the trial. In addition, HOBO accelerometers (HOBO Pendant G acceleration data logger) were fitted to all 54 cows (regardless of the DART score) on the same day of feedlot arrival. They were mounted to the right rear leg, an inch above the fetlock joint, and were programmed to record g-force on the x, y, z-axes at 1-minute intervals after internal processing.

For DART assessment, cows were momentarily removed from their pens and evaluated to see if they needed treatment. Healthy animals that show no clinical symptoms and had a rectal temperature below 40 °C were assigned to DART 0. Cows with morbidity symptoms were given a DART score based on symptom severity. Also, all animals that are assigned DART 1-3 had rectal temperatures greater than 40 °C. The animals were pulled every five days for reevaluation and new DART scores were given if the cow improved or regressed in health status. After 15 days, the HOBO data loggers were removed and data is downloaded using the Onset HOBO software.

5.3.2.2 Dataset Preprocessing

Data Segmentation: Since the DART evaluation is performed during specific times, the dataset contains information about when the scores are assigned, as shown in

Figure 5.3(a). For example, for the animal in Figure 5.3, DART score is assigned on three days: 19-Feb, 25-Feb, and 2-Mar. We can see that the "Moderate" DART score on 19-Feb changed to "Healthy" on 25-Feb. Since the accelerometer measurements closest to the DART examination carry the most relevant information, we extract the data segments according to the following rule:

- If the DART score is same for two consecutive assignments, we extract all days that fall between;
- If the DART score is different between two consecutive assignments, we extract ONLY the next day after the first assignment.

Hence, we extract only the data from 20-Feb on the illustrative example in Figure 5.3(a). In contrast, we extract the data from all four days since the DART score remains the same between 25-Feb and 2-Mar.

After pre-processing data from all 54 cows, we obtain 350 segments. Each segment is a $(3, 1440)$ matrix, where 3 is the number of accelerometer channels and 1440 is the minutes in a day. Overall, there are 180, 30, 89 and 51 segments for the four DART levels from "Healthy" to "Severe", respectively.

Converting DART labels to binary classes: Collected data contains four levels for DART score: Healthy, Mild, Moderate, Severe. As our task in this paper is to classify healthy and sick cows, we map these four labels to two labels: Healthy $\rightarrow$ Healthy and {Mild, Moderate, Severe} $\rightarrow$ Sick. Thus, we end up with 180 and 170 segments for "Healthy" and "Sick" classes, respectively.

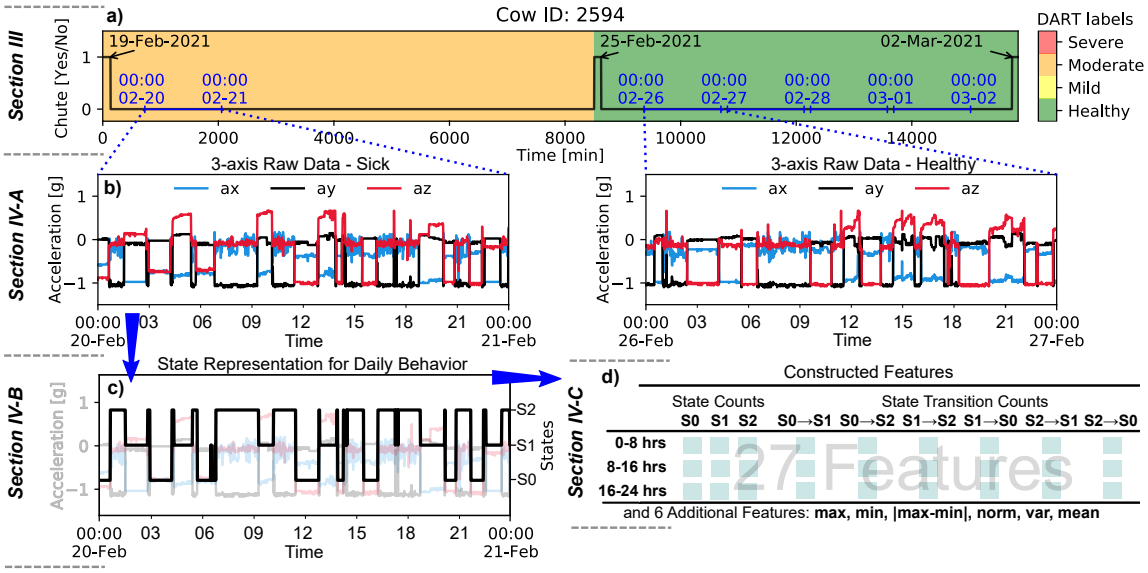


Figure 5.3: The main four steps in our data processing pipeline: a) Data processing and segmentation, b) the 3-axis raw accelerometer data, c) the 1-dimensional state representation obtained from the raw data, d) the constructed feature set.

5.3.3 Cattle Health Classification Methodology

This section presents the possible approaches to predicting the DART score using accelerometer data and describes the proposed method.

5.3.3.1 Classification Using Raw Accelerometer Data

The most intuitive approach is using the raw 3-axis accelerometer data shown in Figure 5.3(b) to design a classifier. The primary advantage of this black-box approach is the potential to obtain a fully autonomous technique with minimum design effort. However, it is very compute-intensive since the raw data size can be enormous. Low computational complexity is critical to implementing these techniques on edge devices that can be attached to cattle. Furthermore, black-

box approaches are *not interpretable* since the ML techniques internally transform the data and extract abstract features in the transformed space. For example, the "TSFresh" technique used in our comparisons extracts 789 features without a clear physical meaning.

To establish a baseline, we used thirteen popular SOTA techniques to classify the raw accelerometer data despite the above-mentioned drawbacks. Due to the indirect relation of the accelerometer data to cattle behavior, most of these techniques achieve less than 70% accuracy, as summarized in Section 5.3.4.2. In contrast, our systematic analysis enables both higher accuracies and lower complexity.

5.3.3.2 Extracting Daily Behavior from Accelerometer Data

The 3-axis accelerometer data provides a clean and consistent observation due to its internal filtering mechanism, as illustrated in Figure 5.3(b). We draw three conclusions based on the consistency of the data:

- ***State 0* (S0):** When the acceleration along the z-axis (a_z) is small (close to -g), a_x is around -0.5g and a_y is consistently close to zero,
- ***State 1* (S1):** When a_x is small (close to -g), channels a_y and a_z are consistently around zero,
- ***State 2* (S2):** When a_y is small (close to -g), a_x and a_z is consistently around zero, while a_x is noisy.

Hence, the data indicates three distinct activities, which are also evident in Figure 5.4 that shows the scatter plot of the *data from all 54 cows*. Indeed, this plot shows these

clearly separated states generalize to all cows in the dataset.

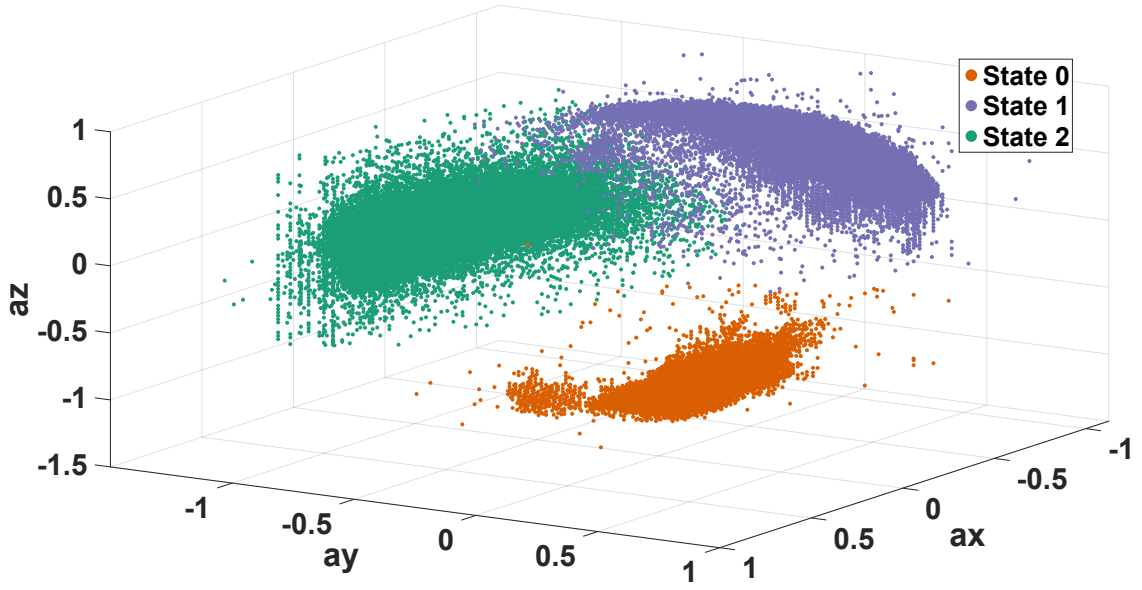


Figure 5.4: Scatter plot of all samples shows three distinct clusters.

The three clusters in raw data shown in Figure 5.4 corresponds to a physical state of the animal. Since the collected data does not label the activities, the state description is not available. An exact description is not crucial, but we can infer that states S0, S1, and S2 correspond to different activities, such as, lying down, standing, and eating or a similar activity. We use a density-based clustering algorithm (`HDBSCAN` [170]) to map each sample to one of three clusters. It generates a one-dimensional tri-state time series data, as shown in Figure 5.3(c). This data is valuable since it correlates with the daily cattle behavior, e.g., how long they lie down, feed, and how often they change their state. Thus, we transform the raw 3-axis accelerometer data to a one-dimensional time series to reduce redundancy and improve correlation with daily behavior that is representative of cattle health.

This time series data is also used with the same thirteen SOTA techniques to design classifiers. These classifiers are compared against the baseline and our proposed technique in Section 5.3.4.2.

5.3.3.3 Behavior-Related Feature Construct

Inspecting the one-dimensional state data extracted from accelerometer measurements reveals further insights about the cattle activity. For example, most cows spend most of their time between 6 AM–12 PM in state S2. Similarly, most cows have hardly any direct transitions between S0 and S1. These consistent observations motivate constructing a set of features that encode the daily behavior using the extracted state information.

We first divided each day into three 8-hour segments since the cow behavior show variations during night, morning, and later in the day. For each segment, we count the number of minutes spent in each state and the number of state transitions. As a result, we end up with 27 features that summarize the daily behavior of the animal, as summarized in Figure 5.3(d). In this way, we obtain more interpretable features than the accelerometer and time series data used before. In addition, this approach uses ~50× fewer data points: *A 1440 element time series array is reduced to just 27 features*. This massive reduction in data size allows for a significantly lower computational complexity, as demonstrated in Section 5.3.4.3.

To understand if the extracted features are representative of healthy and sick animals, we compare the feature vectors for the same animal (cow 2594) in Figure 5.5. The two health states, "Healthy" and "Sick", are clearly distinguishable in

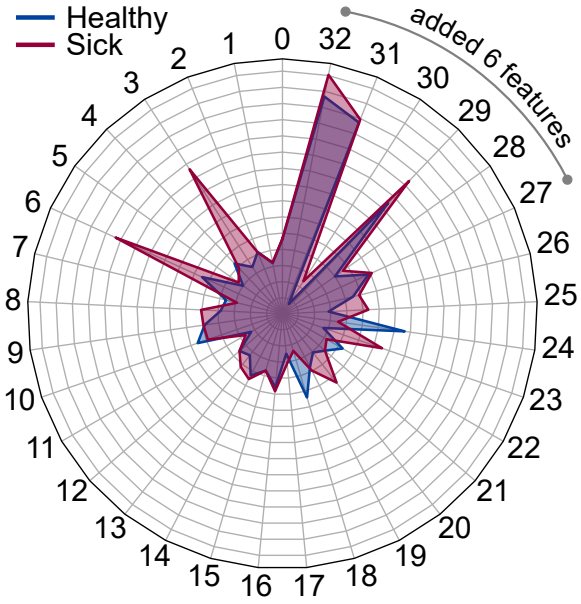


Figure 5.5: The constructed feature vectors that correspond to healthy and sick labels for cow 2594.

the plot. In addition, the maximum values in the "sick" feature vector are higher than the "healthy" one. Similarly, the variance between the features is smaller for healthy cows. In light of these observations, we add six more features that improves the distinguishability between the two classes: The maximum, minimum, $|max\text{-}min|$, norm, variance and the mean of the feature vector. Then, we train a Random Forest classifier (RFC) that consists of eleven shallow decision trees with a maximum depth of 10. The output of the RFC is determined by majority voting, i.e., the class label with higher count of classifications made by the Decision Trees wins. Our proposed approach yields higher accuracy than other approaches, as presented in Section 5.3.4-B. Furthermore, we also present the extremely low memory requirement and energy consumption of our approach in Section 5.3.4-C.

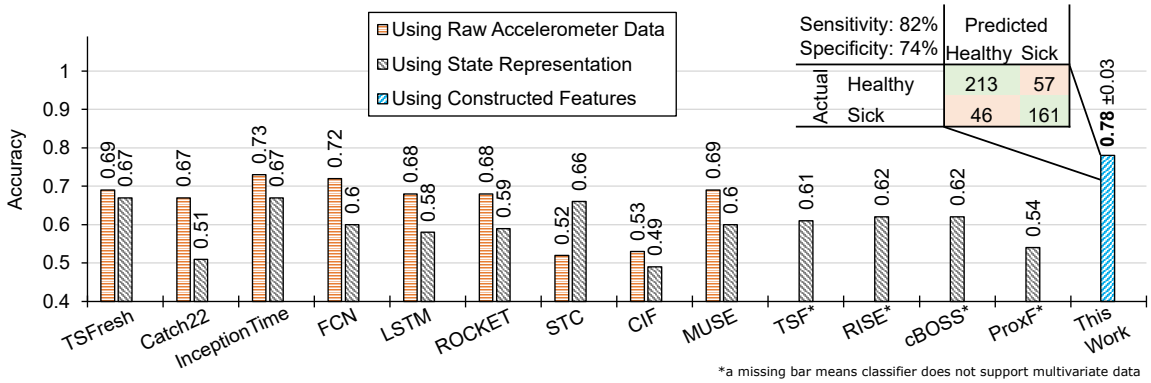


Figure 5.6: 7-fold cross-validation results of all classifiers. Consider the best performing SOTA technique, InceptionTime. Our technique not only increases the accuracy from 73% to 78%, it also uses over 200× less memory and has negligible execution time and energy consumption, as detailed in Section 5.3.4.3.

5.3.4 Experimental Results

This section presents the experimental evaluation for the proposed approach. First, we present the SOTA classifiers that we compare our approach to and explain the train-test split strategy for obtaining results. Then, we present a detailed classification accuracy comparison of our approach with SOTA classifiers. Finally, we provide energy and performance measurements on a real system.

5.3.4.1 Experimental Setup

Choice of Classifiers: According to the results presented in a recent review [171], we choose *eleven* SOTA time series classifiers that perform statistically better than the rest. Since implementations of these classifiers are beyond the scope of this work, we refer interested readers to this comprehensive review [171].

In addition, we also include LSTM and Fully Convolutional Network (FCN)

based classifiers as they are the most intuitive deep learning classifiers for a time series classification task. Hence, we use the thirteen classifiers listed in Table 5.1 for comparison.

LSTM, FCN, and InceptionTime are deep learning approaches, implemented in the `tsai`[1] python package. TSFresh and Catch22 are automatic feature extractors that extract predefined sets of features from the input data. We use the `tsfresh`[2] and `sktime`[3] python packages for the TSFresh and Catch22 implementations, respectively. TSF, RISE, ROCKET, STC, CIF, MUSE, cBOSS and ProxF are statistical methods that apply various transformations to the data and extract features from the transformed data. We use their implementations in the `sktime`[3] python package.

Train and Test Set Split: We aim for a system that generalizes to all cows rather than focusing on a single subject. Therefore, we train the classifiers on a training

[1] `tsai`. [https://github.com/timeseriesAI/tsai]
[2] `tsfresh`. [https://github.com/blue-yonder/tsfresh]
[3] `sktime`. [https://github.com/alan-turing-institute/sktime]

Table 5.1: A compilation of state-of-the-art time series classifiers.

Acronym	Name	Acronym	Name
FCN[1]	Fully Convolutional Network	**ITime**[1]	InceptionTime
LSTM[1]	Long Short Term Memory Network	**ProxF**[3]	Proximity Forest
RISE[3]	Random Interval Spectral Ensemble	**TSF**[3]	Time Series Forest
STC[3]	Shapelet Transform Classifier	**CIF**[3]	Canonical Interval
MUSE[3]	Multivariate Symbolic Extension		Forest
ROCKET[3]	RandOm Convolutional KErnel Transform		
cBOSS[3]	Contractable Bag of Symbolic Fourier Approximation Symbols		
Catch22[3]	Canonical Time-series Characteristics		
TSFresh[2]	Time Series Feature Extraction based on Scalable Hypothesis Tests		

set that includes data from multiple cows. We construe this training set with a leave-10-cow-out approach. Specifically, we first randomly sample 10 cows out of the 54 cows for the test set. Then, data from the remaining 44 cows are merged into a training set. Furthermore, we repeat the procedure 7 times for each classifier to minimize the effects of random sampling, i.e., we apply 7-fold cross validation. Each cross validation run has a maximum of 60%-40% class distribution to have a balanced classifier.

5.3.4.2 Classification Accuracy Results

This section presents the experimental results for classifying healthy and sick animals using i) the raw accelerometer data, ii) the extracted state representation for daily behavior, and iii) our constructed set of features.

Using the 3-axis raw accelerometer data: The raw accelerometer data has three channels, i.e., it is a multivariate data. Nine of the thirteen classifiers support multivariate signals: {TSFresh, Catch22, InceptionTime, FCN, LSTM-FCN, ROCKET, STC, CIF and MUSE}. Their average accuracy (over 7-fold CV) are shown by orange striped bars in Figure 5.6. Among them, InceptionTime obtains the highest accuracy with 73% and FCN follows closely with 72%. CIF and STC perform very poorly with 53% and 52% classification accuracy, respectively.

Although InceptionTime and FCN can obtain relatively good accuracy, using these classifiers directly on the raw data has several drawbacks. First, they are hard to interpret, as we do not understand the implicit representation of the data within the classifiers. Second, their computational complexities are high, rendering

them unsuitable for resource-constrained edge devices. Finally, the data processing overhead (e.g., number of multiplications, additions) is significant since they use all of the data. For example, the InceptionTime classifier has near 60000 trainable floating point parameters, which requires over 200 KB of space on a microcontroller.

Using the extracted state representation: All of the thirteen classifiers support univariate signals. Figure 5.6 shows the accuracy of all of them with diagonally striped black bars. Using the extracted state representation needs less processing effort than using the raw data, but the classification accuracy deteriorates significantly. The accuracies for this case are lower than 70% across the stack. In addition, most of the multivariate classifiers perform worse, STC being the only exception. For example, the accuracy of InceptionTime decreases from 73% to 67%.

Proposed Technique: Our proposed approach uses a Random Forest classifier from the `scikit-learn` python package with 11 decision trees, each with a maximum depth of 10. We achieve 78% classification accuracy with 82% sensitivity and 74% specificity using our constructed feature set, as illustrated in Figure 5.6. Moreover, our approach not only achieves the highest accuracy, but also has significantly smaller memory footprint and energy consumption, as described next.

5.3.4.3 Energy-Performance Evaluation on a Hardware Prototype

We implemented the proposed real-time technique on TI CC2652R Microcontroller, which has 48 MHz operating frequency, 352 KB flash memory, and 80 KB SRAM.

Operation and Memory Requirements: First, a decison tree maps the 3-axis raw accelerometer samples to one of the three states using the cluster labels obtained

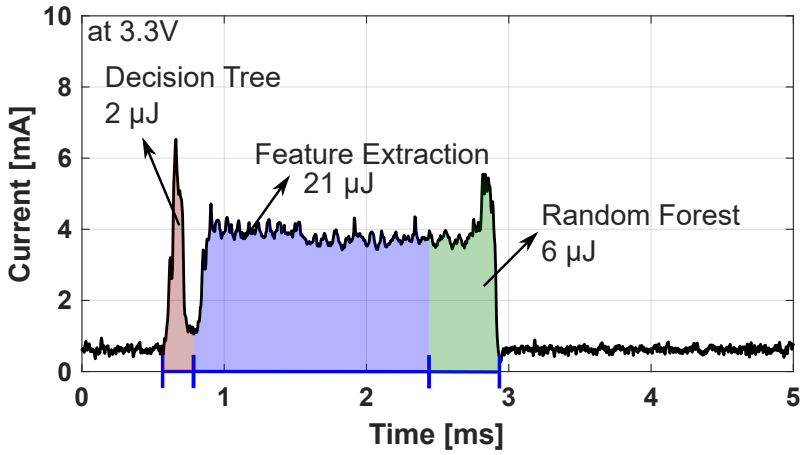


Figure 5.7: Energy consumption measurements on the TI CC2652R MCU. Each operation is repeated back to back to measure their cumulative execution over a day.

from the HDBSCAN algorithm. This mapping alone enables 3× lower memory requirements than approaches that operate on raw data. Furthermore, the states are represented with 8-bit integers instead of the 32-bit floating point numbers needed for raw data. This advantage brings up another 4× memory footprint reduction. After compiling data for an 8-hour period, we construct the features used for predicting the DART score (Section 5.3.3.3). This way, our technique executes as soon as an 8-hour long data is available, thereby reducing the memory requirements by an additional 3× compared to using data for one complete day. Finally, the Random Forest uses these features to classify the cows as "Healthy" or "Sick". With the optimizations highlighted above, our approach requires only 10 KB flash (for program) and 1 KB SRAM (for data) memory, over 36× smaller memory requirement than the approaches that operate on raw data.

Power, Execution Time, and Energy Measurements: Figure 5.7 shows the execution

time (x-axis) and power consumption (current on y-axis at 3.3V) of the proposed technique. During these measurements, all of the processing that normally happens over the course of a day are executed all at once for illustration purposes. The decision tree takes only 0.14 µs per sample (3-axis accelerometer measurement), as shown in Table 5.2. Since it is called once every minute, it is executed 1440 times per day, leading to 2 µJ daily energy consumption. In contrast, the feature generation is performed once every 8 hours with 0.6 ms execution time and 21 µJ daily energy consumption. Finally, the DART score prediction happens once a day. It takes only 0.4 ms and consumes 6 µJ energy. Overall, the proposed technique is extremely light-weight with negligible runtime overhead and only 29 µJ (2.44 nAh at 3.3V) daily energy consumption. Hence, the processing part is negligible compared to sensing and potential wireless communication energy consumption.

Table 5.2: Execution time and energy measurements for the clustering, feature extraction, and classification steps.

Operation	**Execution Time**	**Frequency**	**Energy (per Cons. day)**
Decision Tree (Clustering)	0.14 $\frac{\mu s}{sample}$	Once per minute	2 µJ
Feature Extraction	0.6 ms	Once per 8-hours	21 µJ
Random Forest (Prediction)	0.4 ms	Once per day	6 µJ

5.4 A Domain-Specific System-On-Chip Design for Energy Efficient Wearable Edge AI Applications

Artificial intelligence (AI) based wearable applications collect and process a significant amount of streaming sensor data. Transmitting the raw data to cloud processors wastes scarce energy and threatens user privacy. Therefore, wearable edge AI devices should ideally balance two competing requirements: (1) maximizing the energy efficiency using targeted hardware accelerators for local processing and (2) providing versatility using general-purpose cores to support arbitrary applications. To this end, this section presents an open-source domain-specific programmable system-on-chip (SoC) that combines a RISC-V core with a meticulously determined set of accelerators targeting wearable applications. The proposed design runs on an FPGA prototype and six real-life use cases demonstrate the efficacy of the proposed SoC. Thorough experimental evaluations show that the proposed SoC provides up to 9.1× faster execution and up to 8.9× higher energy efficiency than software implementations in FPGA *while maintaining programmability*.

5.4.1 Proposed Domain-Specific SoC Design

To overcome the shortcomings of SW implementations and custom designs, we present a domain-specific SoC for wearable devices. The proposed design addresses the two challenges mentioned above by integrating a programmable subsystem for flexibility and an extensible HW accelerator cluster for improving energy efficiency and performance, as illustrated in Figure 5.8. Since different applications in a

domain use various tasks for data processing, the proposed design features a programmable subsystem built around a general-purpose RISC-V core. In addition, we integrate an extensible HW accelerator cluster to boost the energy efficiency of frequently used compute-intensive tasks in wearable applications.

5.4.1.1 Programmable Subsystem

The programmable subsystem is based on the open-source PULPissimo microcontroller architecture [172]. The PULPissimo platform is used for three reasons. It (1) allows for end-to-end customization thanks to its open-source availability; (2) incorporates low-power components (e.g., RISC-V core and interconnects) and DMA for low-energy data movement; and (3) is silicon-proven. It utilizes a RISC-V processor, an in-order single-issue 4-pipeline stage 32-bit core. The RISC-V core can be enhanced at design time to include floating-point unit (FPU), DSP, and

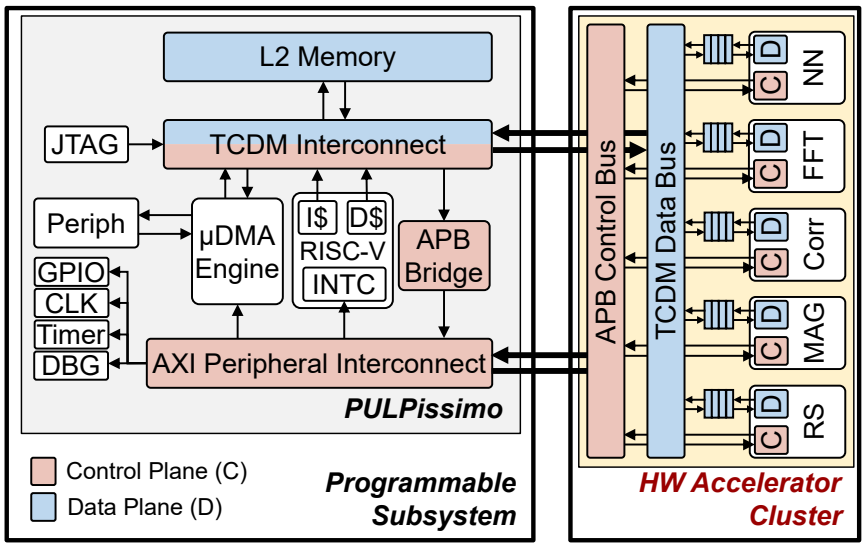


Figure 5.8: The proposed domain-specific SoC architecture.

SIMD extensions. The PULP infrastructure also provides a software development kit (SDK) with pre-built APIs and libraries to utilize the on-chip computing resources. The SoC also includes an L2-memory subsystem that allows different cores to share data, as shown in Figure 5.8. A peripheral AXI bus interconnects the RISC-V core, peripherals, and the control plane of the HW accelerator cluster. The control plane uses the simple advanced peripheral bus (APB) protocol suitable for rare and sporadic transactions. The tightly coupled data memory (TCDM) on-chip network enables high-bandwidth streaming data communication between the RISC-V core, L2 memory, and the HW accelerator cluster. The TCDM interconnect enables single-cycle transactions between the initiators and responders on-chip. This functionality is supported by providing μDMA units and interrupt controllers to issue handshake signals and move data in a streaming manner over the TCDM interconnect. The RISC-V core does not consume energy for relaying data into the accelerators since the HW accelerators read data directly from L2 memory with the help of the μDMA units. Finally, applications running in the RISC-V core easily utilize our HW accelerators using memory-mapped registers. For instance, a neural network (NN) inference call in the user program issues transactions and initiates the computation in the NN accelerator. Hence, the application developers can conveniently utilize the proposed HW accelerators instead of SW routines.

5.4.1.2 Identifying the Tasks to Accelerate

The potential benefits of domain-specific SoCs depend on identifying the tasks to accelerate. This decision is a function of both the prevalence of the task and

the energy-performance benefits of accelerating it. We analyze the full text of relevant research papers published between 2015 and 2020 on arXiv. We chose arXiv specifically as it provides open access to the full-text articles. The analysis covers 282 publications found by the following search query:

```
wearable &                                                          (5.1)
( health || healthcare || medical ) &                               (5.2)
( device || iot || technology || sensing || edge ||
system || computing || monitoring ) &                               (5.3)
( signal processing || algorithm || method || technique ||
approach || implementation || application || hardware )             (5.4)
```

We first build a custom dictionary composed of 164 technical keywords, summarizing the commonly used tasks in wearable applications, such as filtering and statistical features. Then, we find the frequency of these keywords in the selected publications. This analysis summarized in Figure 5.9 provides the following insights:

(a) Signal statistics (e.g., min, max, mean, variance) are the most common features;

(b) NN, support-vector-machines (SVM), and decision trees (DT) are commonly used for classification;

(c) Windowing (e.g., Hamming window), standardization (e.g., mean removal

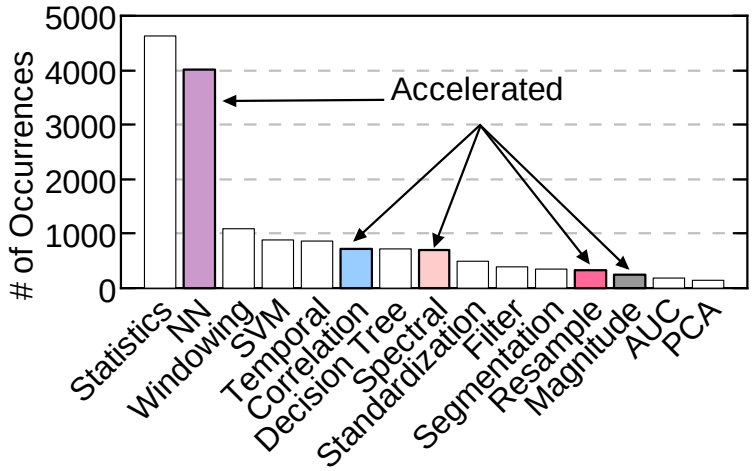


Figure 5.9: Frequency of keywords in 282 papers from relevant literature.

and normalization), filtering, segmentation, and resampling are commonly used for signal-preprocessing;

(d) Temporal (e.g., peak count), correlation (e.g., covariance), spectral analysis (e.g., fast Fourier transform (FFT)), and vector magnitude computation are frequently used for feature generation.

Using an HW accelerator for a given task is profitable only if the execution time and energy consumption savings surpass the offload overhead. Therefore, we implemented these most frequently used tasks on the proposed SoC using only the RISC-V core to profile their execution time and power consumption. As a result of this analysis (detailed in Section 5.4.3), we designed HW accelerators for NN, correlator, resampling, magnitude computation, and used an open-source FFT core [173], as discussed next.

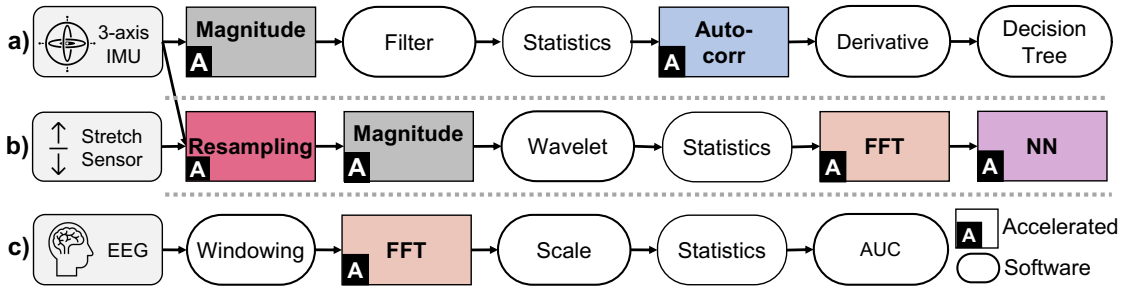


Figure 5.10: Applications used for the evaluation of the proposed SoC. a) Step counter, b) HAR, c) EEG analytics.

5.4.1.3 Design of the Proposed HW Accelerators

NN accelerator: Our design implements the inference of multi-layer perceptron-based neural networks. It has two input channels (one for the weights and another for input features) and a single output channel that emits the output probabilities. It incorporates a parameterized *neuron* submodule and multiply-accumulate engine. Hence, it can be configured to implement NNs with an arbitrary number of layers and neurons. For example, a 3-layer network with 4-8-8 neurons in its layers takes 163 cycles to execute and is pipelined to provide a throughput of 1 output every 127 cycles.

Correlation (Corr) accelerator: The correlation engine calculates the cross-correlation between any two equal-length vectors. For example, we use it to calculate the autocorrelation of a signal with time-lagged versions of itself. For efficient vector multiplication, we leverage the Multiply-Accumulate engine released by the developers of the PULPissimo system [172] in the correlation accelerator. It can output one sample per cycle when fully pipelined.

A ReSampling (RS) accelerator: Resampling changes the sampling rate of a

discrete signal either by dropping samples or by interpolating new samples. In wearable edge AI applications, reducing the number of samples is helpful to conserve memory and decrease the processing effort. Thus, we implemented a programmable, broadly usable accelerator that can be configured to downsample from an arbitrary N number of samples to M samples. It has one input channel and one output channel. The operating principle of the RS accelerator is as follows: It retains the first and last samples in an array. Then, if N/M is an integer number, it retains every N/M^{th} sample and discards the remaining samples. If N/M is not an integer number, it retains samples such that the average interval between the samples is N/M. Thus, it outputs one sample every N/M cycles on average and has a latency of $N + M + 2$ clock cycles.

A MAGnitude (MAG) accelerator: The MAG accelerator calculates the magnitudes of a vector of 2–3 tuples. The mode of operation is programmable, such that developers can use it for different purposes. For example, it can be used in the 3-tuple mode to compute the body-acceleration b_{acc} from accelerometer samples coming from a three-axis IMU (a_x, a_y, a_z), where $b_{acc} = \sqrt{a_x^2 + a_y^2 + a_z^2}$. Similarly, it can be used in the 2-tuple mode to compute the magnitudes of a vector of complex numbers, such as the output of an FFT operation. It can output one sample per cycle when fully pipelined and has a 90 clock cycle end-to-end latency.

A Fast-Fourier-Transform (FFT) accelerator: We generate an optimized 64-point FFT core with a 233 cycle latency using the Spiral IP generator [173] to accelerate the FFT computation. It has two input channels for two complex samples (16 bits each for real and imaginary), and similarly two output channels. The FFT accelerator

supports a maximum throughput of 2 complex words per cycle.

5.4.1.4 Accelerator Integration to the SoC

The RISC-V core controls the operation of HW accelerators through the APB bus. The control interface of each HW accelerator exposes memory-mapped registers in the L2 memory to the APB bus. These registers are responsible for the control operations, such as triggering the operation of an accelerator, and other runtime configuration parameters, such as the input size. We extend the PULP-SDK for each accelerator and provide APIs specific to each accelerator. At runtime, the APIs in the SDK transfer the configuration parameters from the SW code to the specific HW accelerators. For example, we configure the RS accelerator to downsample the accelerometer and stretch sensor 64 and 32 data points, respectively. Providing such handles to the developers allows for the easy runtime programmability of the SoC. Offloading the computation from SW to an HW accelerator comprises the following steps:

i. transfer the input data to a chosen buffer space in L2 memory,

ii. program the accelerator's parameters and address space,

iii. initiate accelerator computation,

iv. wait for accelerator completion, and

v. read the output data from L2 memory buffer space.

Table 5.3 presents the speedup of tasks in HW accelerators compared to computation in the RISC-V core. We emphasize that the presented speedups include all the additional programming and data transfer overheads, and hence are realistic speedups that users will observe when the respective tasks are accelerated in HW.

Table 5.3: Sample execution times of accelerated tasks (at 10 MHz).

	NN (4-8-8)	**Corr (80×80)**	**RS (205→64)**	**MAG (64 3-tuples)**	**FFT (64-point)**
SW–RISC-V (ms)	10.80	15.50	5.10	1.80	29.82
Accelerated (ms)	0.63	2.11	0.31	0.06	1.53
Speedup (×)	17.25	7.36	16.71	29.51	19.55

5.4.2 Driver Application Use Cases

We implemented three popular applications to demonstrate the proposed SoC: (1) Step counter, (2) HAR, and (3) EEG analytics. They are representative examples since 11 out of 13 tasks in these applications are among the most commonly used kernels listed in Figure 5.9. Furthermore, we analyze the concurrent execution of these three applications as additional real-life use cases.

Step counter: This application is implemented in most wearable devices due to the strong correlation between daily walking activity and cardiovascular health [174]. A recent study claims that step counter-based interventions that aim to increase daily walking activity lead to significantly fewer new cardiovascular events in the long term [174]. The authors further state that step counter is an essential tool to set realistic daily goals for each individual. In our study, we use an auto-correlation-based step counting algorithm presented in [175], which is outlined in Figure 5.10a).

We adapt an open-sourced implementation of the algorithm[1].

Human Activity Recognition: HAR is a critical enabler for a broad range of healthcare and fitness applications [176]. It can help Parkinson's disease patients by enabling auditory cues when symptoms like tremor or freezing of gait occur. Due to its importance, there are numerous HAR implementations and datasets in the literature. We use the w-HAR dataset that contains data collected from an inertial measurement unit at the ankle and a stretch sensor on the knee for 22 users and seven activities [176]. The authors provide the algorithm summarized in Figure 5.10b).

EEG analytics: This application computes the power spectral density of input EEG signals and divides them into distinct frequency bands, such as the delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz), and gamma (30–100 Hz) bands. Sleep and attention studies usually calculate the average power in these bands relative to each other. Increased activity in the delta band is associated with deep sleep, whereas an attenuated delta band activity and increased high-frequency activity is related to wakefulness [177]. In this work, we use Welch's method to calculate the power spectral density (PSD) estimate of the EEG recording and calculate the area under the PSD for the frequency band of interest. Figure 5.10c) shows the steps of the algorithm. We use the publicly available EEG data provided by the Center for Human Sleep Science at UC Berkeley [2].

[1]Neraj Bobra. https://github.com/nerajbobra/embedded_pedometer
[2]Raphael Vallat. https://raphaelvallat.com/images/tutorials/bandpower/data.txt

5.4.3 Experimental Evaluation

This section evaluates an FPGA implementation of the proposed SoC in terms of execution time and energy efficiency benefits. To this end, we implement the driver applications on the proposed SoC with and without HW acceleration. Before moving to costly tapeout, an FPGA prototype helps us validate the functionality and obtain estimates of execution time gains. As an extra step, we also measure the power consumption on the FPGA and analyze the energy savings enabled by the proposed design. Although the absolute numbers are not representative of a taped-out design, the obtained energy savings compared to the SW implementation on the FPGA gives a conservative estimate of the actual benefits. We expect significantly higher energy efficiency in the taped-out design since the FPGA hardware building blocks (lookup tables and routing matrices) consume higher power than taped-out implementations (logic gates and wires). In the following, we first introduce the experimental setup and procedure. Then, we measure the execution times and power consumption of the applications. Finally, we present the measurement results and our analysis.

5.4.3.1 Experimental Setup and Procedure

Setup: The proposed SoC is implemented and evaluated on the Xilinx ZCU102 FPGA board at four different frequencies: 10 MHz, 16 MHz, 25 MHz, and 40 MHz. The entire design is mapped to the programmable logic (PL) on the ZCU102. Since the platform does not allow configuring the clock dividers at runtime, we synthesize the design using Vivado v2019.2 for each frequency and verify that FPGA

utilization is similar in all synthesis results. We emphasize that *all four synthesized designs include all five accelerators*, irrespective of whether they are utilized or not, in order to make fair comparisons within each application and between applications.

Application use cases: The three applications presented in Section 5.4.2 are used for evaluation. In addition to running them individually, we also study three additional concurrent real-world use cases by combining (1) Step counter and HAR, (2) Step counter and EEG analytics, and (3) all three applications. We collect measurements for the pure SW implementations and the corresponding accelerated versions to evaluate all use cases at four frequencies.

Procedure: We collect (1) execution time and (2) power consumption measurements from the FPGA implementation: (1) Execution time is measured using timer functions provided in the PULP-SDK that has 30µs resolution. These measurements are used to demonstrate the speedup enabled by the HW accelerators integrated into the proposed SoC. (2) For power consumption measurements, we measure the load current on the primary rails that power the programmable logic on the ZCU102 board using a National Instruments DAQ at 250 kS/s. The primary rails are the internal supply voltage (V_{ccint}) and the BRAM voltage (V_{ccbram}) rails.

Each application use case runs 1000 times to allow for averaging of the power measurements. After averaging 1000 measurements, we apply a moving average filter with a window of 50 samples to reduce the high-frequency noise. A 50-sample window is chosen to have a negligible time-shift in order not to distort the signal. The obtained waveform allows us to clearly observe the power consumption, as depicted in Figure 5.11. Then, we use the power measurements to find the energy

consumption of the SoC.

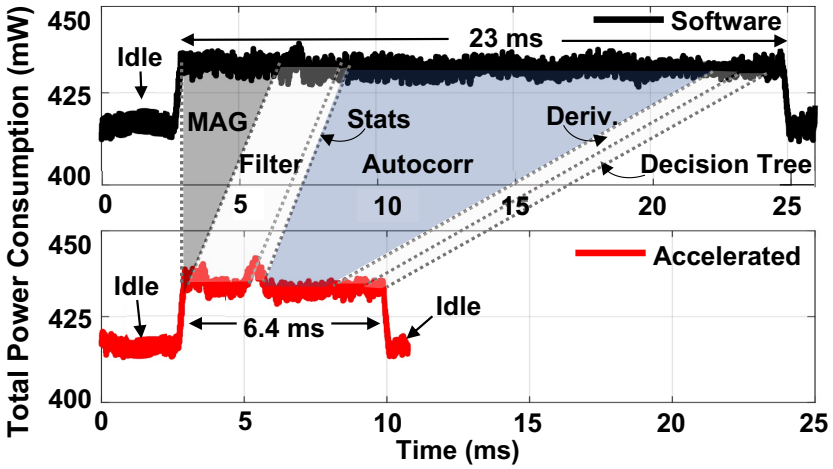


Figure 5.11: Power consumption of the SW and accelerated implementations of the step counter application at 10 MHz.

5.4.3.2 Execution Time Analysis

This section analyzes the individual applications in terms of their execution time for SW and accelerated implementations.

Profiling SW implementations: Figure 5.12 presents the execution time breakdown of the applications when implemented in SW on the RISC-V core. The execution time analysis shows that:

(a) the magnitude and auto-correlation computations constitute 83% of the total execution time of the step counter application;

(b) resampling, magnitude, FFT computation, and neural network inference constitute 94% of the HAR application execution time;

(c) FFT computation constitutes 91% of the EEG analytics application execution

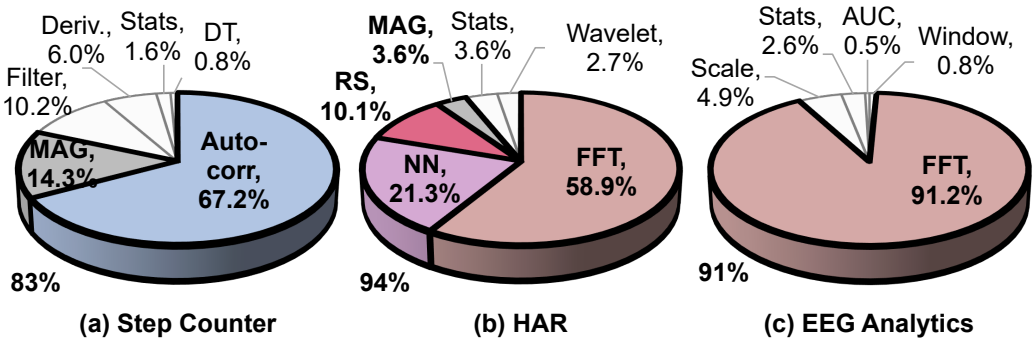


Figure 5.12: Breakdown of execution times of the 3 applications.

time.

These results justify our choice of accelerated tasks in the SoC.

Execution time: Figure 5.13-Top illustrates the execution time of the SW and accelerated implementations of the three applications. Utilizing the HW accelerators yields 3.6× speedup in execution time for the step counter application at the same frequency. Specifically, at 10 MHz, the SW implementation takes 23.0 ms to run, whereas the accelerated implementation takes 6.4 ms. Similarly, the accelerated HAR and EEG analytics applications are executed up to 9.1× and 8.8× faster than their SW counterparts, respectively. In addition, execution times scale down linearly with increasing frequency. Furthermore, the execution time speedups remain consistent across different frequencies.

5.4.3.3 Power and Energy Analysis

This section presents the power and energy consumption measurements for the SW and accelerated implementations of the six application use cases.

Power measurements: We collect current measurements from the FPGA as outlined

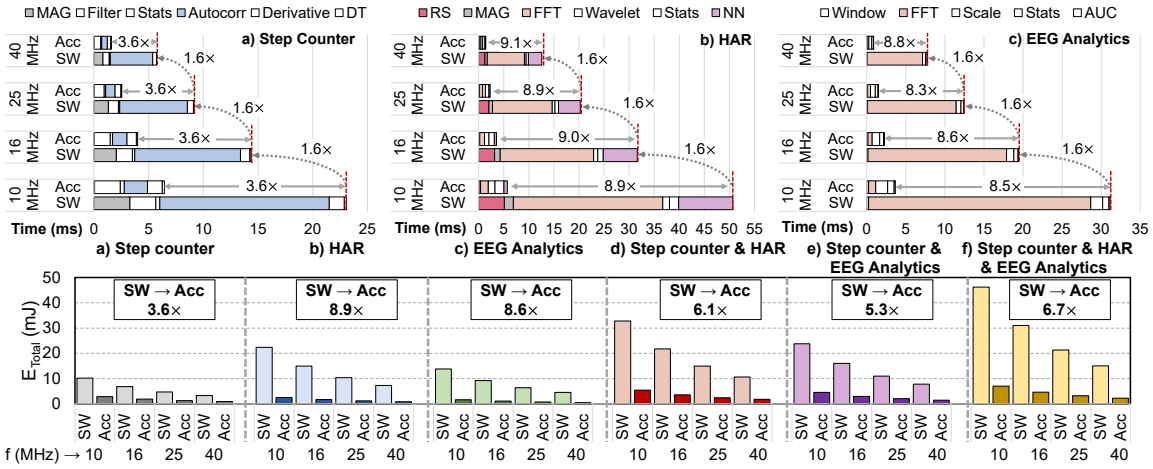


Figure 5.13: Execution times and energy consumption results. Top) Execution times of the three applications. Bottom) Energy consumption of the six application use cases.

in Section 5.4.3.1 to analyze the power and energy consumption of the design. The power consumption of the V_{ccint} and V_{ccbram} power rails are proportional to their current consumption since both have a fixed voltage level of 0.85 V.

The core is set to idle before and after program execution while taking the measurements. Figure 5.11 illustrates the power consumption measurement for the step counter application at 10 MHz. The lower power regions in this figure correspond to the idle state, and the higher power regions correspond to the active state. According to the measurements, the utilization of HW accelerators does not cause a significant change in active state power consumption. In addition, we note

Table 5.4: Average power consumption of the three applications.

Power in mW	**Step counter**		**HAR**		**EEG Analytics**	
	P_{idle}	P_{act}	P_{idle}	P_{act}	P_{idle}	P_{act}
10 MHz	416	440	417	441	419	442
16 MHz	438	475	438	473	438	473
25 MHz	465	510	460	510	460	510
40 MHz	509	577	503	579	503	579

that the power consumption measured from the V_{ccbram} rail is a constant 75 mW for all cases. For the illustrated case (at 10 MHz), the average idle and active power consumptions are 416 mW and 440 mW, respectively, as shown in Table 5.4. Similarly, for HAR and EEG analytics applications, the idle and active power consumptions are {417, 441} mW and {419, 442} mW, respectively, which suggest that power consumption *on the FPGA* is not a strong function of the executed application. According to the power measurements at different frequencies in Table 5.4, the average idle and active state power consumptions increase with frequency.

Energy consumption and savings: Energy consumption is computed by the area under the total power consumption curve when a program is executed (i.e., active region in Figure 5.11). The energy consumption scales with a similar factor as the execution time since the SW and accelerated implementations have similar power consumption, as summarized in Figure 5.13-Bottom. According to our results, the proposed SoC achieves a 3.6× reduction in energy consumption for the step counter application at all frequencies. For example, at 10 MHz, the energy consumption for the step counter application reduces from 10.14 mJ to 2.83 mJ. Similarly, 8.9× and 8.6× gains are achieved for the HAR and EEG analytics applications, respectively. Moreover, the proposed SoC achieves between 5.3× to 6.7× reduction in energy consumption for the concurrent application use cases. The absolute power values collected from the FPGA are not indicative of a chip design implemented with a low-power process technology. However, power and energy consumption analysis on the FPGA is helpful to obtain early conservative estimates and a lower bound of the gains that the proposed SoC can achieve.

6 CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusion

Wearable devices have the potential to revolutionize health and activity monitoring applications. However, their adoption by the general population has been fairly limited. To this end, this dissertation covers the major technical and adaptation challenges to their adoption, and presents three fundamental goals as the solution: i) Incorporate novel, flexible, wearable energy harvesters as an orthogonal source to the battery, ii) employ smart energy management frameworks to make judicious use of the limited energy to maximize device lifetime, and iii) minimize energy consumption through SW and HW optimizations. The rest of this thesis presents five studies towards these goals. First, we investigate the feasibility of piezoelectric energy harvesting from large joint bending motions and present an analytical model for the harvested energy with extensive experiments using a 3D-printed knee robot for validation. Then, we present two energy management frameworks for wearable devices, ECO and tinyMAN. ECO calculates initial hourly energy budget at the beginning of the day using forecasts for the harvested energy, and applies perturbations to the initial values as the day progresses. Alternatively, tinyMAN uses reinforcement learning to learn the usage and energy harvesting patterns for a cluster of users and calculates the hourly energy budget using this learned model. Next, we present a highly optimized cattle health monitoring application with ultra-low power consumption thanks. Finally, we present a domain-specific SoC design for wearable applications for energy-efficient implementations of wearable

applications. We expect the continuous and combined efforts on wearable energy harvesting modalities, energy management algorithms and low-power design techniques will enable self-sustainable wearable devices and be the next big computing revolution.

In summary, this dissertation addressed several critical gaps in the design and development of wearable devices by making the following contributions:

- Investigate the underlying mechanism of piezoelectric energy harvesting from bending motions of the knee [28, 29],
- Characterize the energy harvesting capacity of a combined light and piezoelectric wearable energy harvester in a day [30],
- Present ECO, an energy management framework to achieve energy-neutral operation for energy harvesting wearable devices [31],
- Present tinyMAN, a reinforcement-learning-based energy manager to achieve self-sustained operation for energy harvesting wearable devices [32],
- Present a novel cattle health monitoring application with ultra-low power consumption [33],
- Present a domain-specific system-on-chip (SoC) design for energy efficient wearable edge applications [34],

6.2 Future Work and Research Directions

A crucial part that was outside the scope of this dissertation is the energy-consumption due to sampling and wireless communication. Similar studies that investigate the feasibility of using particular sensors and communication modalities on wearable devices are critical. For example, we have observed that the angle measurement sensor used during the energy harvesting studies has significantly higher power consumption than a 3-axis accelerometer. Instead of using the angle sensor directly, a system design may benefit more by using the accelerometer and additional processing steps to predict the angle of the knee. Despite the extra computation, this approach would yield a lower energy consumption than using the angle sensor. Similarly, a definitive exploration and comparison of different wireless communication modalities for wearable devices is of utmost importance. Do we really need constant connection to our wearable devices? Why do we not just transmit data once a day? Such questions shall be investigated from a system-design point of view.

Finally, the aim of this thesis is to present the multiple aspects for making wearable devices self-sustainable. The methodical and detailed analysis of wearable energy harvesting modalities, energy management algorithms, and energy consumption of various wearable applications show that a collective effort involving all of these aspects is necessary to achieve self-sustainability. Therefore, a wearable device platform prototype with all of the material discussed in this thesis would pave the way for future studies. *Specifically, the following items are critical:* i) Taping out the domain-specific SoC in a recent technology node, ii) designing a prototype

PCB that houses the SoC together with the necessary components for the energy harvesting and battery management subsystems, such that we can monitor battery and the harvested energy during usage, iii) deploying the energy management frameworks discussed in this thesis on this prototype device, iv) showcasing the prototype by implementing various impactful applications, such as freezing of gait detection for Parkinson's Disease patients, gait analysis, or joint health tracking.

A APPENDIX A: DISCUSSIONS ON THE EXPERIMENTAL PROCEDURE

This section describes the details of the experimental procedure to enable reproducibility of the results reported in Chapter 3. In addition, we present valuable insights into the use of multiple piezoelectric transducers.

A.1 Design, Fabrication & Actuation of the Knee Frame

The mechanical knee joint frame in Figure 3.3(a) is designed in SolidWorks. It is 3D printed using ABS plastic with 60% density for rapid prototyping. The design for the frame consists of three parts: the upper leg, kneecap, and lower leg. The kneecap has a 4-cm radius of curvature and is hollow in the middle. The upper leg is composed of an axle and the leg part. The axle is placed inside the hollow space in the kneecap. The bottom part of the kneecap is screwed to the top part of the lower leg, and the bottom part of the lower leg is screwed to a flat base to keep the frame upright. There is a 4-cm wide groove along all three sections in which the piezoelectric transducer is placed. The groove on the upper leg is covered for guided movement of the piezoelectric transducer to ensure good coupling to the bending motion. Also, several screw holes are placed along with the frame for mounting clampers to fix the piezoelectric transducer. Finally, the frame is actuated through a Dynamixel AX-12A servo motor 0.29° resolution, which is screwed to the outer side of the kneecap while the horn is screwed to the axle of the upper leg. The motor is controlled through a dedicated OpenCM9.04 control board.

To actuate the robot frame, we collect knee angle waveforms using a one-axis Bendlabs flexible angle sensor [110]. This sensor measures the angle between the tangent lines drawn from its two ends. Therefore, we place the sensor on the knee using a compression sleeve, such that one end is on the thigh and the other end is

on the shin. Using this setup, we capture the knee bending angle at various speeds on a treadmill for 125 seconds and discard the first 20 and last 5 seconds. With the remaining 100-second data, we generate ten 10-second segments and average them. Figure 3.4(1a-1c) shows the averaged segments for 3, 5, and 7 mph gait speed. Then, the averaged segment is fed to the motor controller to actuate the robot frame. This way, our robot frame mimics knee bending motion at various gait speeds, ranging from walking (2 mph) to running (7 mph).

A.2 Piezoelectric Transducers and Their Placement

PEHs use piezoelectric transducers to convert mechanical energy into electrical energy. In the current study, we use MFC8528P2 flexible piezoelectric transducers [111]. These transducers nicely fit inside the groove on our robot frame. We clamp the piezoelectric transducer from its wire-end to the lower leg. The other end of the piezoelectric transducer slides freely inside the groove in the knee frame to prevent any additional strain caused by unwanted stretching (i.e., the transducer is not clamped from the top).

Stacking multiple transducers is a promising solution to increase the generated current. *Stacking does not result in any undue space overhead and poses no problems as long as the stacked transducers do not cause discomfort to the user*.

There are two ways to stack multiple transducers: (1) Stacking them on top of each other and fixing them from only one side, and (2) bonding two transducers with an epoxy layer in between them (e.g. a bimorph). The latter approach generates a higher current as a result of the higher strain in the piezoelectric layer due to the increased distance to the neutral axis. Our experience with the bonded structure shows that it does not cause any difficulties to the user during the knee bending motion.

The output of multiple piezoelectric transducers can be combined mainly by two different topologies. One option is to rectify and regulate the voltage produced by each transducer separately and then add the resulting DC voltages up. This approach incurs extra losses and design effort because each additional transducer

requires its own rectifier and regulator. The other approach is to superimpose the harvested voltage from all transducers and then to rectify and regulate the superimposed voltage. This approach may diminish the obtained power if the superimposed voltages add up destructively owing to synchronization issues. In our case, however, the bonded structure guarantees that no synchronization issues are present. Therefore, in this study, we superimpose the AC voltages generated by the piezoelectric transducers and rectify and regulate the superimposed voltage using the LTC3331 based energy harvester.

A.3 LTC3331 Operation Principles

LTC3331 consists of a low-loss full-wave bridge rectifier and two buck DC/DC converters. The first DC/DC is connected to the input capacitor, while the other is connected to the battery, as illustrated in Figure 3.5(c). A prioritizer module with a programmable UVLO hysteresis window controls the operation of the DC/DC converters according to the voltage on the input and output capacitors. The charge generated by the piezoelectric transducer is rectified and stored on the input capacitance C_{in} connected to the V_{IN} pin of LTC3331. Once the voltage on C_{in} reaches the upper threshold of the UVLO window, the accumulated charge is transferred to the output capacitance through the first DC/DC. When the voltage on C_{in} decreases below the lower threshold of the UVLO window, the DC/DC is disabled. For example, the smallest window configuration turns on the first DC/DC when the voltage on C_{in} reaches 4 V and turns it off when the voltage goes down to 3 V. Then, the harvested energy starts accumulating again on the input capacitance. This process repeats until a regulated voltage at the output is obtained. If the voltage on the output capacitor can not be regulated, the second DC/DC is turned on to regulate the load using the battery. Thus, the UVLO configuration keeps the voltage on the input capacitor between the two values (as long as the load is regulated). For this reason, we choose the window configuration such that this voltage interval corresponds to the maximum power points presented in Figure 3.7.

A.4 Effect of Temperature Variation on Energy Harvesting

We performed all of our experiments at standard room temperature. However, the temperature may vary depending on the time of year and the location in real-life use cases. Therefore, this section discusses the effect of temperature variation on the performance of MFC transducers.

The operating temperature for the MFC transducers is between -35°C and 85°C [111]. Marek et al. [178] analyzed the effect of temperature changes on the efficiency of MFC-based energy harvesters. They swept the operating temperature from -30°C to 70°C with 10°C steps. Their results revealed that the generated current by the MFC transducer decreases with increasing temperature. Specifically, the peak voltage at -30°C is twice the peak voltage at 70°C. They attributed this decrease to the change in the dielectric constant of the piezoelectric material from which the transducer's fibers were made.

The temperature range of interest for our wearable use case is between -20°C and 40°C. The results in [178] show a 1.4× decrease from -20°C to 20°C, and a 1.25× reduction from 20°C to 40°C. Therefore, we expect the worst-case harvested energy (at 40°C) to be 80% of the results presented in this work. Similarly, in the best case (at -20°C), it will be 1.40× larger than the presented results.

B APPENDIX B: PROOF OF LEMMA 1

Suppose that we are at interval $t = \tau$. Then, from Equation 4.9 the initial allocation E_τ^A is given by:

$$\begin{aligned} E_\tau^A = \beta^\tau E_0^A &= \beta^\tau \frac{E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H}{1 + \beta + \beta^2 + \ldots + \beta^{T-1}} \\ &= \frac{E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H}{\frac{\beta^T - 1}{\beta^\tau(\beta - 1)}} \end{aligned} \tag{B.1}$$

Let $\widehat{E}_t^A$ denote the recalculated allocations and $\widehat{E}_t^H$ stand for the actual energy harvesting values. Following the same pattern in Equation 4.9, we can write $\widehat{E}_\tau^A$ as follows:

$$\begin{aligned} \widehat{E}_\tau^A &= \frac{E_\tau^B - E_{target} + \sum_{t=\tau}^{T-1} E_t^H}{1 + \beta + \beta^2 + \ldots + \beta^{T-\tau-1}} \\ &= \frac{(E_0^B + \sum_{t=0}^{\tau-1} \widehat{E}_t^H - \sum_{t=0}^{\tau-1} \widehat{E}_t^A) - E_{target} + \sum_{t=\tau}^{T-1} E_t^H}{\frac{\beta^T - \beta^\tau}{\beta^\tau(\beta - 1)}} \end{aligned} \tag{B.2}$$

Let $\overline{\Delta}_\tau^H$ correspond to the sum of differences between actual and expected energy harvesting values until $t = \tau$:

$$\overline{\Delta}_\tau^H = \sum_{t=0}^{\tau-1} \Delta_t^H = \sum_{t=0}^{\tau-1} \left(\widehat{E}_t^H - E_t^H\right) \tag{B.3}$$

Similarly, let $\overline{\Delta}_\tau^A$ be the sum of differences between initial and corrected energy allocations until $t = \tau$:

$$\overline{\Delta}_\tau^A = \sum_{t=0}^{\tau-1} \Delta_t^A = \sum_{t=0}^{\tau-1} \left(E_t^A - \widehat{E}_t^A\right) \tag{B.4}$$

With these, Equation B.2 can be written as:

$$\widehat{E}_{\tau}^{A} = \frac{(E_0^B + \sum_{t=0}^{\tau-1} E_t^H - \sum_{t=0}^{\tau-1} E_t^A + \overline{\Delta}_{\tau}^{H} + \overline{\Delta}_{\tau}^{A}) - E_{target} + \sum_{t=\tau}^{T-1} E_t^H}{\frac{\beta^T - \beta^{\tau}}{\beta^{\tau}(\beta-1)}} \tag{B.5}$$

Further combining $\sum_{t=0}^{\tau-1} E_t^H + \sum_{t=\tau}^{T-1} E_t^H$ as $\sum_{t=0}^{T-1} E_t^H$ and $\overline{\Delta}_{\tau}^{H} + \overline{\Delta}_{\tau}^{A}$ as Δ_{τ}, we obtain the following:

$$\widehat{E}_{\tau}^{A} = \frac{E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H - \sum_{t=0}^{\tau-1} E_t^A + \Delta_{\tau}}{\frac{\beta^T - \beta^{\tau}}{\beta^{\tau}(\beta-1)}} \tag{B.6}$$

We know from Equation 4.9 that $\sum_{t=0}^{\tau-1} E_t^A = E_0^A \sum_{t=0}^{\tau-1} \beta^t$. Since $0 < \beta < 1$, we have:

$$\sum_{t=0}^{\tau-1} E_t^A = \frac{\beta^{\tau} - 1}{\beta^T - 1}(E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H) \tag{B.7}$$

Next, we plug Equation B.7 into Equation B.6 to obtain:

$$\widehat{E}_{\tau}^{A} = \frac{E_0^B - E_{target} + \sum_{t=0}^{T-1} E_t^H}{\frac{\beta^T - 1}{\beta^{\tau}(\beta-1)}} + \frac{\Delta_{\tau}}{\frac{\beta^T - \beta^{\tau}}{\beta^{\tau}(\beta-1)}} \tag{B.8}$$

The first term is nothing but E_{τ}^{A} as shown in Equation B.1. Therefore, we reach the solution provided in Lemma 1. □

C APPENDIX C: ATUS DATA AND OUR PRE-PROCESSING

Table C.1 summarizes the activity categories in the ATUS dataset as shown in the ATUS coding lexicons document [117]. We have reduced the number of categories to 10 by grouping some of the ATUS categories together and assigned them location labels as shown in Table C.2.

Table C.1: ATUS Category IDs and Labels

ATUS Major Categories			
ID	**Label**	**ID**	**Label**
1	Personal care activities	11	Eating & drinking
2	Household activities	12	Socializing, relaxing & leisure
3	Caring for & helping household members	13	Sports, exercise, & recreation
4	Caring for & helping non-household members	14	Religious & spiritual activities
5	Work & work related activities	15	Volunteer activities
6	Education	16	Telephone calls
7	Consumer purchases	18	Travelling
8	Professional & personal care services	50	Other/invalid
9	Household services		
10	Government services & civic obligations		

Table C.2: Our assigned IDs, labels and locations.

Our Categories			
ID	**Label**	**ATUS IDs**	**Location**
1	Sleep	1−1	home
2	Housework	2	home
3	People Care	3 & 4	home
4	Work	5 & 6	office
5	Shop	7	store
6	Eat	11	home/office
7	Leisure	1 (except 1−1) & 12	home
8	Exercise	13	store
9	Travel	9	outdoors
10	Others	8, 9, 10, 14, 15, 16 & 50	home

BIBLIOGRAPHY

[1] World Health Organization. World Report on Disability. [Online] `http://www.who.int/disabilities/world_report/2011/report/en/.`, 2011.

[2] Alberto J Espay et al. Technology in Parkinson's Disease: Challenges and Opportunities. *Mov Disord*, 31(9):1272–1282, 2016.

[3] Sizhe An and Umit Y Ogras. Fast and scalable human pose estimation using mmwave point cloud. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 889–894, 2022.

[4] Dimiter V Dimitrov. Medical Internet of Things and Big Data in Healthcare. *Healthcare Informatics Research*, 22(3):156–163, 2016.

[5] Shivayogi Hiremath, Geng Yang, and Kunal Mankodiya. Wearable Internet of Things: Concept, Architectural Components and Promises for Person-Centered Healthcare. In *Proc. MOBIHEALTH*, pages 304–307, 2014.

[6] Jean-Francois Daneault. Could Wearable and Mobile Technology Improve the Management of Essential Tremor? *Frontiers in Neurology*, 9:257:1–257:8, 2018.

[7] Ganapati Bhat, Ranadeep Deb, and Umit Y Ogras. Openhealth: open-source platform for wearable health monitoring. *IEEE Design & Test*, 36(5):27–34, 2019.

[8] Antonio Tricoli, Noushin Nasiri, and Sayan De. Wearable and Miniaturized Sensor Technologies for Personalized and Preventive Medicine. *Advanced Functional Materials*, 27(15):1605271, 2017.

[9] Bin Zhang, Feifei Huang, Jun Liu, and Dingguo Zhang. A Novel Posture for Better Differentiation Between Parkinson's Tremor and Essential Tremor. *Frontiers in Neuroscience*, 12, 2018.

[10] Ranadeep Deb, Ganapati Bhat, Sizhe An, Umit Ogras, and Holly Shill. Trends in technology usage for parkinson's disease assessment: A systematic review. *medRxiv*, 2021.

[11] Toygun Basaklar, Suat Gumussoy, and Umit Y Ogras. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. *arXiv preprint arXiv:2208.07914*, 2022.

[12] Ranadeep Deb, Sizhe An, Ganapati Bhat, Holly Shill, and Umit Y Ogras. A systematic survey of research trends in technology usage for parkinson's disease. *Sensors*, 22(15):5491, 2022.

[13] Sunghoon Ivan Lee et al. Activity Detection in Uncontrolled Free-Living Conditions Using a Single Accelerometer. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6, 2015.

[14] Sizhe An, Yigit Tuncel, Toygun Basaklar, Gokul K Krishnakumar, Ganapati Bhat, and Umit Y Ogras. Mgait: Model-based gait analysis using wearable bend and inertial sensors. *ACM Transactions on Internet of Things*, 3(1):1–24, 2021.

[15] Sizhe An and Umit Y Ogras. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s):1–22, 2021.

[16] Sizhe An, Yin Li, and Umit Ogras. mri: Multi-modal 3d human pose estimation dataset using mmwave, rgb-d, and inertial sensors. *Advances in Neural Information Processing Systems*, 35:27414–27426, 2022.

[17] Yasser Khan et al. Flexible Hybrid Electronics: Direct Interfacing of Soft and Hard Electronics for Wearable Health Monitoring. *Adv. Funct. Mater.*, 26(47):8764–8775, 2016.

[18] Mallika Bariya, Hnin Yin Yin Nyein, and Ali Javey. Wearable sweat sensors. *Nature Electronics*, 1(3):160–171, 2018.

[19] Hyoyoung Jeong et al. Modular and reconfigurable wireless e-tattoos for personalized sensing. *Adv. Mater. Technol.*, 4(8):1900117, 2019.

[20] Md Muztoba, Rohit Voleti, Fatih Karabacak, Jaehyun Park, and Umit Y Ogras. Instinctive assistive indoor navigation using distributed intelligence. *ACM TODAES*, 23(6):1–21, 2018.

[21] Toygun Basaklar, Yigit Tuncel, Sizhe An, and Umit Ogras. Wearable devices and low-power design for smart health applications: challenges and opportunities. In *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–1. IEEE, 2021.

[22] Anneli Ozanne et al. Wearables in Epilepsy and Parkinson's disease-A Focus Group Study. *Acta Neurologica Scandinavica*, 137(2):188–194, 2018.

[23] Dongni Johansson, Kristina Malmgren, and Margit Murphy. Wearable Sensors for Clinical Applications in Epilepsy, Parkinson's Disease, and Stroke: A Mixed-Methods Systematic Review. *Jrnl. of neurology*, pages 1–13, 2018.

[24] Walter Maetzler, Jochen Klucken, and Malcolm Horne. A Clinical View on the Development of Technology-Based Tools in Managing Parkinson's Disease. *Movement Disorders*, 31(9):1263–1271, 2016.

[25] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. Embedd. Comput. Syst.*, 6(4):32–es, 2007.

[26] Ganapati Bhat, Jaehyun Park, and Umit Y Ogras. Near-Optimal Energy Allocation for Self-Powered Wearable Systems. In *Proc. Int. Conf. on Comput.-Aided Design*, pages 368–375, 2017.

[27] DEKRA Testing and Certification. FCCID - Test Report USA FCC Part 15.247, 15.209. [Online] `https://fccid.io/2AD7V-OURA2101/Test-Report/67469RRF-002-5536194`.

[28] Yigit Tuncel, Toygun Basaklar, and Umit Ogras. Wearable piezoelectric energy harvesting from human gait: Modeling and experimental validation. *IEEE Sensors Journal*, 22(16):16617–16627, 2022.

[29] Yigit Tuncel, Shiva Bandyopadhyay, Shambhavi V Kulshrestha, Audrey Mendez, and Umit Y Ogras. Towards wearable piezoelectric energy harvesting: Modeling and experimental validation. In *Proceedings of the ACM/IEEE Intl. Symposium on Low Power Electronics and Design*, pages 55–60, 2020.

[30] Yigit Tuncel, Toygun Basaklar, and Umit Ogras. How much energy can we harvest daily for wearable applications? In *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6. IEEE, 2021.

[31] Yigit Tuncel, Ganapati Bhat, Jaehyun Park, and Umit Ogras. Eco: Enabling energy-neutral iot devices through runtime allocation of harvested energy. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[32] Toygun Basaklar, Yigit Tuncel, and Umit Y Ogras. Tinyman: Lightweight energy manager using reinforcement learning for energy harvesting wearable iot devices. *arXiv preprint arXiv:2202.09297*, 2022.

[33] Yigit Tuncel, Toygun Basaklar, Mackenzie Smithyman, João Dórea, Vinícius Nunes De Gouvêa, Younghyun Kim, and Umit Ogras. Towards smart cattle farms: Automated inspection of cattle health with real-life data. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–2. IEEE, 2023.

[34] Yigit Tuncel, Anish Krishnakumar, Aishwarya Lekshmi Chithra, Younghyun Kim, and Umit Ogras. A domain-specific system-on-chip design for energy

efficient wearable edge ai applications. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 1–6, 2022.

[35] Joseph A Paradiso and Thad Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive computing*, (1):18–27, 2005.

[36] Mohammed Rahimi, Hardik Shah, Gaurav S Sukhatme, John Heideman, and Deborah Estrin. Studying the feasibility of energy harvesting in a mobile sensor network. In *IEEE Intl. Conf. on Robotics and Automation*, volume 1, pages 19–24, 2003.

[37] Sujesha Sudevalayam and Purushottam Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Commun. Surv*, 13(3):443–461, 2010.

[38] Petar Jokic and Michele Magno. Powering smart wearable systems with flexible solar energy harvesting. In *IEEE Intl. symposium on circuits and systems (ISCAS)*, pages 1–4, 2017.

[39] Aminy E Ostfeld, Abhinav M Gaikwad, Yasser Khan, and Ana C Arias. High-performance flexible energy storage and harvesting system for wearable electronics. *Scientific reports*, 6:26122, 2016.

[40] Michele Magno, Davide Brunelli, Lukas Sigrist, Renzo Andri, Lukas Cavigelli, Andres Gomez, and Luca Benini. Infinitime: Multi-sensor wearable bracelet with human body harvesting. *SUSTAIN COMPUT-INFOR*, 11:38–49, 2016.

[41] Antonino Proto, Marek Penhaker, Daniele Bibbo, David Vala, Silvia Conforto, and Maurizio Schmid. Measurements of generated energy/electrical quantities from locomotion activities using piezoelectric wearable sensors for body motion energy harvesting. *Sensors*, 16(4):524, 2016.

[42] Mustafa Beyaz. Energy harvesting from knee motion using piezoelectric patch transducers. *APJES*, 7(2):255–260, 2019.

[43] Youngsu Cha. Energy harvesting using flexible piezoelectric materials from human walking motion: Theoretical analysis. *Journal of Intelligent Material Systems and Structures*, 28(20):3006–3015, 2017.

[44] Giulia Bassani, Alessandro Filippeschi, and Emanuele Ruffaldi. Nonresonant kinetic energy harvesting using macrofiber composite patch. *IEEE Sens. J.*, 18(5):2068–2076, 2018.

[45] Son Nguyen and Rajeevan Amirtharajah. A hybrid rf and vibration energy harvester for wearable devices. In *IEEE Applied Power Electronics Conf. and Exposition (APEC)*, pages 1060–1064, 2018.

[46] Michele Dini, Matteo Filippi, Alessandra Costanzo, Aldo Romani, Marco Tartagni, Massimo Del Prete, and Diego Masotti. A fully-autonomous integrated rf energy harvesting system for wearable applications. In *2013 European Microwave Conference*, pages 987–990, 2013.

[47] Ufuk Muncuk, Kubra Alemdar, Jayesh D Sarode, and Kaushik Roy Chowdhury. Multiband ambient rf energy harvesting circuit design for enabling batteryless sensors and iot. *IEEE Internet of Things Journal*, 5(4):2700–2714, 2018.

[48] Haiyan Liu, Yancheng Wang, Deqing Mei, Yaoguang Shi, and Zichen Chen. Design of a wearable thermoelectric generator for harvesting human body energy. In *Wearable sensors and robots*, pages 55–66. Springer, 2017.

[49] Trung Nguyen Huu, Toan Nguyen Van, and Ono Takahito. Flexible thermoelectric power generator with y-type structure using electrochemical deposition process. *Applied energy*, 210:467–476, 2018.

[50] Y-J Wang and P-C Hsu. Analytical modelling of partial shading and different orientation of photovoltaic modules. *IET Renewable Power Generation*, 4(3):272–282, 2010.

[51] Pooja Sharma, Siddhartha P Duttagupta, and Vivek Agarwal. A novel and universal model for accurate prediction of pv module characteristics for power optimization under various design layouts and dynamic environmental conditions. In *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–6, 2012.

[52] Christos Konstantopoulos and Eftichios Koutroulis. Global maximum power point tracking of flexible photovoltaic modules. *IEEE Transactions on Power Electronics*, 29(6):2817–2828, 2013.

[53] Jaehyun Park, Hitesh Joshi, Hyung Gyu Lee, Sayfe Kiaei, and Umit Y. Ogras. Flexible PV-cell Modeling for Energy Harvesting in Wearable IoT Applications. *ACM Trans. Embedd. Comput. Syst.*, 16(5s):156:1–156:20, 2017.

[54] Huicong Liu, Junwen Zhong, Chengkuo Lee, Seung-Wuk Lee, and Liwei Lin. A comprehensive review on piezoelectric energy harvesting technology: Materials, mechanisms, and applications. *Applied Physics Reviews*, 5(4):041306, 2018.

[55] Alperen Toprak and Onur Tigli. Piezoelectric energy harvesting: State-of-the-art and challenges. *Applied Physics Reviews*, 1(3):031104, 2014.

[56] Young-Man Choi, Moon Gu Lee, and Yongho Jeon. Wearable biomechanical energy harvesting technologies. *Energies*, 10(10):1483, 2017.

[57] Paul D Mitcheson, Elizabeth K Reilly, T Toh, Paul K Wright, and Eric M Yeatman. Performance limits of the three mems inertial energy generator transduction types. *Journal of Micromechanics and Microengineering*, 17(9):S211, 2007.

[58] Yung-Wey Chong, Widad Ismail, Kwangman Ko, and Chen-Yi Lee. Energy harvesting for wearable devices: A review. *IEEE Sensors Journal*, 19(20):9047–9062, 2019.

[59] Woo-Suk Jung, Min-Jae Lee, Min-Gyu Kang, Hi Gyu Moon, Seok-Jin Yoon, Seung-Hyub Baek, and Chong-Yun Kang. Powerful curved piezoelectric generator for wearable applications. *Nano Energy*, 13:174–181, 2015.

[60] K Ylli, D Hoffmann, A Willmann, P Becker, B Folkmer, and Y Manoli. Energy harvesting from human motion: exploiting swing and shock excitations. *Smart Materials and Structures*, 24(2):025029, 2015.

[61] Haluk Kulah and Khalil Najafi. Energy scavenging from low-frequency vibrations by using frequency up-conversion for wireless sensor applications. *IEEE Sensors Journal*, 8(3):261–268, 2008.

[62] Michele Pozzi and Meiling Zhu. Plucked piezoelectric bimorphs for knee-joint energy harvesting: modelling and experimental validation. *Smart Materials and Structures*, 20(5):055007, 2011.

[63] Yang Kuang, Zhihao Yang, and Meiling Zhu. Design and characterisation of a piezoelectric knee-joint energy harvester with frequency up-conversion through magnetic plucking. *Smart Materials and Structures*, 25(8):085029, 2016.

[64] Honglei Zhou, Yue Zhang, Ye Qiu, Huaping Wu, Weiyang Qin, Yabin Liao, Qingmin Yu, and Huanyu Cheng. Stretchable piezoelectric energy harvesters and self-powered sensors for wearable and implantable devices. *Biosensors and Bioelectronics*, page 112569, 2020.

[65] Le-Giang Tran, Hyouk-Kyu Cha, and Woo-Tae Park. Rf power harvesting: a review on designing methodologies and applications. *Micro and Nano Systems Letters*, 5(1):14, 2017.

[66] Manuel Piñuela, Paul D Mitcheson, and Stepan Lucyszyn. Ambient rf energy harvesting in urban and semi-urban environments. *IEEE Transactions on microwave theory and techniques*, 61(7):2715–2726, 2013.

[67] Tolga Soyata, Lucian Copeland, and Wendi Heinzelman. Rf energy harvesting for embedded systems: A survey of tradeoffs and methodology. *IEEE Circuits and Systems Magazine*, 16(1):22–57, 2016.

[68] Hans Gregory Schantz. Near field propagation law & a novel fundamental limit to antenna gain versus size. In *2005 IEEE Antennas and Propagation Society International Symposium*, volume 3, pages 237–240, 2005.

[69] Andrey S Andrenko, Xianyang Lin, and Miaowang Zeng. Outdoor rf spectral survey: A roadmap for ambient rf energy harvesting. In *TENCON 2015-2015 IEEE Region 10 Conference*, pages 1–4, 2015.

[70] Sana Ullah, Pervez Khan, Niamat Ullah, Shahnaz Saleem, Henry Higgins, and Kyung Sup Kwak. A review of wireless body area networks for medical applications. *arXiv preprint arXiv:1001.0831*, 2010.

[71] Raziel Riemer and Amir Shapiro. Biomechanical energy harvesting from human motion: theory, state of the art, design guidelines, and future directions. In *Journal of NeuroEngineering and Rehabilitation*, 2011.

[72] Melissa Hyland, Haywood Hunter, Jie Liu, Elena Veety, and Daryoosh Vashaee. Wearable thermoelectric generators for human body heat harvesting. *Applied Energy*, 182:518–524, 2016.

[73] Amin Nozariasbmarz, Henry Collins, Kelvin Dsouza, Mobarak Hossain Polash, Mahshid Hosseini, Melissa Hyland, Jie Liu, Abhishek Malhotra, Francisco Matos Ortiz, Farzad Mohaddes, et al. Review of wearable thermoelectric energy harvesting: From body temperature to electronic systems. *Applied Energy*, page 114069, 2019.

[74] Je-Hyeong Bahk, Haiyu Fang, Kazuaki Yazawa, and Ali Shakouri. Flexible thermoelectric materials and device optimization for wearable energy harvesting. *Journal of Materials Chemistry C*, 3(40):10362–10374, 2015.

[75] Christopher M Vigorito, Deepak Ganesan, and Andrew G Barto. Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks. In *Proc. of IEEE Comm. Society Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks*, pages 21–30, 2007.

[76] Bernhard Buchli, Felix Sutton, Jan Beutel, and Lothar Thiele. Dynamic Power Management for Long-Term Energy Neutral Operation of Solar Energy Harvesting Systems. In *Proc. Conf. on Embedd. Network Sensor Syst.*, pages 31–45, 2014.

[77] Kai Geissdoerfer, Raja Jurdak, Brano Kusy, and Marco Zimmerling. Getting more out of energy-harvesting systems: Energy management under time-varying utility with preact. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 109–120, 2019.

[78] Bruno Srbinovski, Michele Magno, Fiona Edwards-Murphy, Vikram Pakrashi, and Emanuel Popovici. An energy aware adaptive sampling algorithm for energy harvesting wsn with energy hungry sensors. *Sensors*, 16(4):448, 2016.

[79] Anuradha Pughat and Vidushi Sharma. A review on stochastic approach for dynamic power management in wireless sensor networks. *Human-Centric Computing and Information Sciences*, 5(1):1–14, 2015.

[80] Nga Dang, Roberto Valentini, Eli Bozorgzadeh, Marco Levorato, and Nalini Venkatasubramanian. A unified stochastic model for energy management in solar-powered embedded systems. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 621–626, 2015.

[81] Imran Ahmed, Imtiaz Ahmed, and Jahangir Hossain. Optimal stochastic power allocation and relay selection for energy harvesting systems. *IEEE Wireless Communications Letters*, 6(4):546–549, 2017.

[82] Meng-Lin Ku, Yan Chen, and KJ Ray Liu. Data-driven stochastic models and policies for energy harvesting sensor communications. *IEEE J. Sel. Areas Commun.*, 33(8):1505–1520, 2015.

[83] Nathan Starliper, Farrokh Mohammadzadeh, Tanner Songkakul, Michelle Hernandez, Alper Bozkurt, and Edgar Lobaton. Activity-aware wearable system for power-efficient prediction of physiological responses. *Sensors*, 19(3):441, 2019.

[84] Nailah Saleh Alhassoun, Md Yusuf Sarwar Uddin, and Nalini Venkatasubramanian. Context-aware energy optimization for perpetual iot-based safe communities. *SUSTAIN COMPUT-INFOR*, 22:96–106, 2019.

[85] Yohann Rioual, Yannick Le Moullec, Johann Laurent, Muhidul Islam Khan, and Jean-Philippe Diguet. Reward function evaluation in a reinforcement learning approach for energy management. In *2018 16th Biennial Baltic Electronics Conference (BEC)*, pages 1–4, 2018.

[86] Ramtin Kazemi, Rein Vesilo, Eryk Dutkiewicz, and Ren Liu. Dynamic power control in wireless body area networks using reinforcement learning with approximation. In *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2203–2208, 2011.

[87] Cui Su, Panpan Wang, and Rong Chai. Joint power allocation and coordinator deployment for wireless body area network. In *2013 International Conference on Wireless Communications and Signal Processing*, pages 1–6, 2013.

[88] Filippo Casamassima, Elisabetta Farella, and Luca Benini. Context aware power management for motion-sensing body area network nodes. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6, 2014.

[89] Kyle J Goodrick and Dragan Maksimović. Systematic Optimization of Multiple Voltage Domain DC Distribution Architectures. In *Proc. Work. on Control and Modeling for Power Electron.*, pages 1–8, 2018.

[90] Kristof Blutman et al. Logic Design Partitioning for Stacked Power Domains. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 25(11):3045–3056, 2017.

[91] Abhinav Agarwal and Arvind. Leveraging Rule-Based Designs for Automatic Power Domain Partitioning. In *Proc. of ICCAD*, pages 326–333, 2013.

[92] Farzad Samie, Lars Bauer, and Jörg Henkel. Hierarchical Classification for Constrained IoT Devices: A Case Study on Human Activity Recognition. *IEEE Internet of Things Journal*, 7(9):8287–8295, 2020.

[93] Antonio Pullini et al. Mr. Wolf: An Energy-Precision Scalable Parallel Ultra Low Power SoC for IoT Edge Processing. *IEEE JSSC*, 54(7):1970–1981, 2019.

[94] Marco Guermandi et al. A Wearable Device for Minimally-invasive Behind-the-ear EEG and Evoked Potentials. In *IEEE BioCAS*, pages 1–4, 2018.

[95] Ganapati Bhat, Yigit Tuncel, Sizhe An, Hyung Gyu Lee, and Umit Y Ogras. An ultra-low energy human activity recognition accelerator for wearable health applications. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–22, 2019.

[96] Xin Liu et al. A Miniature On-Chip Multi-Functional ECG Signal Processor with 30 μW Ultra-low Power Consumption. In *Annual International Conf. of the IEEE Engineering in Medicine and Biology*, pages 2577–2580, 2010.

[97] Scott Hanson et al. A Low-Voltage Processor for Sensing Applications with Picowatt Standby Mode. *IEEE JSSC*, 44(4):1145–1155, 2009.

[98] Alicia Klinefelter et al. 21.3 A 6.45 μW Self-Powered IoT SoC with Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios. In *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. of Techn. Papers*, pages 384–385, 2015.

[99] Bo Wang et al. Exploration of Power Domain Partitioning for Application-Specific SoCs in System-Level Design. In *MBMV*, 2016.

[100] Umit Y Ogras, Radu Marculescu, Puru Choudhary, and Diana Marculescu. Voltage-Frequency Island Partitioning for GALS-Based Networks-on-chip. In *Proc. Design Autom. Conf.*, pages 110–115, 2007.

[101] Samet E Arda et al. Ds3: A system-level domain-specific system-on-chip simulation framework. *IEEE Transactions on Computers*, 69(8):1248–1262, 2020.

[102] Shaoshan Liu et al. Edge Computing for Autonomous Driving: Opportunities and Challenges. *Proceedings of the IEEE*, 107(8):1697–1716, 2019.

[103] Sam Xi et al. Smaug: End-to-end Full-stack Simulation Infrastructure for Deep Learning Workloads. *ACM TACO*, 17(4):1–26, 2020.

[104] Yuchi Liu, Hamideh Khanbareh, Miah Abdul Halim, Andrew Feeney, Xiaosheng Zhang, Hadi Heidari, and Rami Ghannam. Piezoelectric energy harvesting for self-powered wearable upper limb applications. *Nano Select*, 2021.

[105] Caitlin N Teague, J Alex Heller, Brandi N Nevius, Andrew M Carek, Samer Mabrouk, Florencia Garcia-Vicente, Omer T Inan, and Mozziyar Etemadi. A wearable, multimodal sensing system to monitor knee joint health. *IEEE Sensors Journal*, 20(18):10323–10334, 2020.

[106] Bernd J Stetter, Steffen Ringhof, Frieder C Krafft, Stefan Sell, and Thorsten Stein. Estimation of knee joint forces in sport movements using wearable sensors and machine learning. *Sensors*, 19(17):3690, 2019.

[107] Johnson Loh, Jianan Wen, and Tobias Gemmeke. Low-cost dnn hardware accelerator for wearable, high-quality cardiac arrythmia detection. In *IEEE Intl. Conf. on Application-specific Systems, Architectures and Processors (ASAP)*, pages 213–216, 2020.

[108] Niell G Elvin and Alex A Elvin. Large deflection effects in flexible energy harvesters. *Journal of Intelligent Material Systems and Structures*, 23(13):1475–1484, 2012.

[109] Alper Erturk and Daniel J. Inman. A Distributed Parameter Electromechanical Model for Cantilevered Piezoelectric Energy Harvesters. *Journal of Vibration and Acoustics*, 130(4), 2008.

[110] Bend Labs. Flexible Single Axis Bidirectional Sensor. [Online] https://www.bendlabs.com/, accessed 1 February 2022.

[111] Smart-Material. MFC P2 and P3 types. https://www.smart-material.com/MFC-product-P2.html, accessed 1 February 2022.

[112] Linear Technology. LTC3331 - Nanopower Buck-Boost DC/DC with Energy Harvesting Battery Charger. [Online] https://www.analog.com/en/products/ltc3331.html/, accessed 1 February 2022.

[113] Geffrey K Ottman, Heath F Hofmann, and George A Lesieutre. Optimized piezoelectric energy harvesting circuit using step-down converter in discontinuous conduction mode. *IEEE Trans. Power Electron.*, 18(2):696–703, 2003.

[114] PowerFilmSolar. SP3-37 Product Page. https://www.powerfilmsolar.com/products/sp3-37, accessed 1 February 2022.

[115] Engineering ToolBox. Illuminance - Recommended Light Level. [Online] https://www.engineeringtoolbox.com/light-level-rooms-d_708.html/, accessed 1 February 2022.

[116] Johan J Estrada-López, Amr Abuellil, Zizhen Zeng, and Edgar Sánchez-Sinencio. Multiple input energy harvesting systems for autonomous iot end-nodes. *J. Low Power Electron. Appl.*, 8(1):6, 2018.

[117] US Department of Labor. American Time Use Survey, 2018. https://www.bls.gov/tus/, accessed 1 February 2022.

[118] European Commission CORDIS. Smart and Flexible Energy Supply Platform for Wearable Electronics. [Online] https://cordis.europa.eu/project/id/825143, accessed 31 May 2021.

[119] Energiot. Harvesting energy for the Internet of Things. [Online] https://www.energiot.com/products/energy-harvesters/, accessed 31 May 2021.

[120] Atmosic. Lowest Power Wireless Technologies for Battery-Free IoT. [Online] https://atmosic.com/, accessed 31 May 2021.

[121] Onio. Breaking free from the battery. [Online] https://www.onio.com/, accessed 31 May 2021.

[122] Ganapati Bhat, Kunal Bagewadi, Hyung Gyu Lee, and Umit Y Ogras. Reap: Runtime energy-accuracy optimization for energy harvesting iot devices. In *56th ACM/IEEE Design Automation Conference*, pages 1–6, 2019.

[123] Jaehyun Park, Ganapati Bhat, Cemil S Geyik, Umit Y Ogras, and Hyung Gyu Lee. Energy per operation optimization for energy-harvesting wearable iot devices. *Sensors*, 20(3):764, 2020.

[124] H. W. Kuhn and A. W. Tucker. Nonlinear Programming. In *Proc. of the Second Berkeley Symp. on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 1951.

[125] Nurullah Karakoç, Anna Scaglione, Angelia Nedić, and Martin Reisslein. Multi-layer decomposition of network utility maximization problems. *IEEE/ACM Transactions on Networking*, 28(5):2077–2091, 2020.

[126] Texas Instruments Inc. CC2652R Microcontroller. [Online] https://www.ti.com/product/CC2652R, accessed 1 November 2020, 2018.

[127] D Bertsekas. Rollout algorithms for discrete optimization: A survey. *Handbook of Combinatorial Optimization, D. Zu and P. Pardalos, Eds. Springer*, 2010.

[128] FlexSolarCells. SP3-12 Datasheet. http://www.flexsolarcells.com/index_files/OEM_Components/Flex_Cells/specification_sheets/00_FlexSolarCells.com_PowerFilm_Solar_SP3-12_Specification_Sheet.pdf, accessed 1 February 2022.

[129] Yigit Tuncel, Ganapati Bhat, and Umit Y Ogras. Special session: Physically flexible devices for health and activity monitoring: Challenges from design to test. In *IEEE 38th VLSI Test Symposium*, pages 1–5, 2020.

[130] Sandia National Laboratories. Sandia's Ephemeris Model, 2017. `https://pvpmc.sandia.gov/modeling-steps/1-weather-design-inputs/sun-position/sandias-code/`, accessed 5 August 2017.

[131] Pierre Ineichen and Richard Perez. A New Airmass Independent Formulation for the Linke Turbidity Coefficient. *Solar Energy*, 73(3):151–157, 2002.

[132] Wensi S Wang, Terence O'Donnell, Ningning Wang, Michael Hayes, Brendan O'Flynn, and C O'Mathuna. Design considerations of sub-mw indoor light energy harvesting for wireless sensor systems. *ACM Journal on Emerging Technologies in Computing Systems*, 6(2):1–26, 2008.

[133] FlexSolarCells. OEM Components Info, 2020. `http://www.flexsolarcells.co.uk/index_files/OEM_Components/PowerFilm-Solar-OEM-Components-Technical-Information.php/`, accessed 1 Nov. 2020.

[134] Catrine Tudor-Locke, Ho Han, Elroy J Aguiar, Tiago V Barreira, John M Schuna Jr, Minsoo Kang, and David A Rowe. How fast is fast enough? walking cadence (steps/min) as a practical estimate of intensity in adults: a narrative review. *Br J Sports Med*, 52(12):776–788, 2018.

[135] Texas Instruments. BQ25504, 2015. `http://www.ti.com/lit/ds/symlink/bq25504.pdf`, accessed 5 August 2017.

[136] InvenSense. 9-axis MotionTracking, 2017. `https://invensense.tdk.com/products/motion-tracking/9-axis/icm-20948/`, accessed 1 Nov. 2020.

[137] GMB. 031009 datasheet, 2009. `http://www.gmbattery.com/Datasheet/LIPO/LIPO-031009-12mAh.pdf`, accessed 5 August 2017.

[138] Madeleine Ecker, Jochen B Gerschler, Jan Vogel, Stefan Käbitz, Friedrich Hust, Philipp Dechent, and Dirk Uwe Sauer. Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data. *Journal of Power Sources*, 215:248–257, 2012.

[139] IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[140] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[141] Quentin Cappart, Thierry Moisan, Louis-Martin Rousseau, Isabeau Prémont-Schwarz, and Andre A Cire. Combining reinforcement learning and constraint programming for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3677–3687, 2021.

[142] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[143] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[144] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[145] Yasuhiro Fujita, Prabhat Nagarajan, Toshiki Kataoka, and Takahiro Ishikawa. Chainerrl: A deep reinforcement learning library. *Journal of Machine Learning Research*, 22(77):1–14, 2021.

[146] Amazon Halo. Amazon Halo - Health & wellness band. [Online] `https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=GVHZZMHA9M6Y6JS5/`, accessed March 2023.

[147] DEKRA Testing and Certification. FCCID - 2AJ2X-WS40. [Online] `https://fccid.io/2AJ2X-WS40`.

[148] Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398, 2014.

[149] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International Conference on Machine Learning*, pages 10607–10616. PMLR, 2020.

[150] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in Neural Information Processing Systems*, 32:14636–14647, 2019.

[151] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[152] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, et al. Tensorflow lite micro: Embedded machine learning on tinyml systems. *arXiv preprint arXiv:2010.08678*, 2020.

[153] Semiconductor Research Corporation (SRC). Decadal Plan for Semiconductors - Full Report, 2022. `https://www.src.org/about/decadal-plan/decadal-plan-full-report.pdf`, accessed 1 Mar. 2022.

[154] Michele Magno and David Boyle. Wearable energy harvesting: From body to battery. In *12th IEEE DTIS*, pages 1–6, 2017.

[155] John L Hennessy and David A Patterson. A New Golden Age for Computer Architecture. *Commun. of the ACM*, 62(2):48–60, 2019.

[156] Gregg A Hanzlicek et al. Serial evaluation of physiologic, pathological, and behavioral changes related to disease progression of experimentally induced

mannheimia haemolytica pneumonia in postweaned calves. *American jrnl. of veterinary research*, 71(3):359–369, 2010.

[157] Bhisham Sharma and Deepika Koundal. Cattle health monitoring system using wireless sensor network: a survey from innovation perspective. *IET Wireless Sensor Systems*, 8(4):143–151, 2018.

[158] Ruth Eunice Centeno-Martinez et al. Identification of bovine respiratory disease through the nasal microbiome. *Animal microbiome*, 4(1):1–18, 2022.

[159] DL Step et al. Effects of commingling beef calves from different sources and weaning protocols during a forty-two-day receiving period on performance and bovine respiratory disease. *Journal of Animal Science*, 86(11):3146–3158, 2008.

[160] AGAmerica. The U.S. Farm Labor Shortage. `https://agamerica.com/blog/the-impact-of-the-farm-labor-shortage`, accessed Sep 2022.

[161] James M. MacDonald et al. Profits, costs, and the changing structure of dairy farming. Technical report, United States Department of Agriculture.

[162] Sarah Morrone, Corrado Dimauro, Filippo Gambella, and Maria Grazia Cappai. Industry 4.0 and precision livestock farming (plf): An up to date overview across animal productions. *Sensors*, 22(12):4319, 2022.

[163] Suresh Neethirajan. Recent advances in wearable sensors for animal health management. *Sens. Bio-Sens. Res.*, 12:15–29, 2017.

[164] Marilyn J Buhman et al. Association between changes in eating and drinking behaviors and respiratory tract disease in newly arrived calves at a feedlot. *American Jrnl. of Veterinary Research*, 61(10):1163–1168, 2000.

[165] Olgierd Unold, Maciej Nikodem, Marek Piasecki, Kamil Szyc, Henryk Maciejewski, Marek Bawiec, Paweł Dobrowolski, and Michał Zdunek. Iot-based cow health monitoring system. In *Intl. Conf. on Comput. Science*, pages 344–356, 2020.

[166] Hanwook Chung et al. Using implantable biosensors and wearable scanners to monitor dairy cattle's core body temperature in real-time. *Computers and Electronics in Agriculture*, 174:105453, 2020.

[167] Anuj Kumar and Gerhard P Hancke. A zigbee-based animal health monitoring system. *IEEE Sensors Journal*, 15(1):610–617, 2014.

[168] Enrico Casella, Melissa C Cantor, Simone Silvestri, Dave L Renaud, and Joao HC Costa. Cost-aware inference of bovine respiratory disease in calves using precision livestock technology. In *Intl. Conf. on Distributed Computing in Sensor Systems*, pages 109–116, 2022.

[169] Mohit Taneja et al. Machine learning based fog computing assisted data-driven approach for early lameness detection in dairy cattle. *Computers and Electronics in Agriculture*, 171:105286, 2020.

[170] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Jrnl. of Open Source Soft., 2(11)*, page 205, 2017.

[171] Alejandro Pasos Ruiz et al. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.*, 35(2):401–449, 2021.

[172] Pasquale Davide Schiavone et al. Quentin: An Ultra-low-power PULPissimo SoC in 22nm FDX. In *IEEE S3S*, pages 1–3, 2018.

[173] Peter Milder et al. Computer Generation of Hardware for Linear Digital Signal Processing Transforms. *ACM TODAES*, 17(2):1–33, 2012.

[174] Tess Harris et al. Effect of Pedometer-based Walking Interventions on Long-term Health Outcomes: Prospective 4-year Follow-up of Two Randomised Controlled Trials using Routine Primary Care Data. *PLoS medicine*, 16(6), 2019.

[175] Dan Morris et al. RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises. In *ACM CHI*, pages 3225–3234, 2014.

[176] Ganapati Bhat et al. w-HAR: An Activity Recognition Dataset and Framework using Low-power Wearable Devices. *Sensors*, 20(18):5356, 2020.

[177] Kristina Šušmáková. Human Sleep and Sleep EEG. *Measurement Science Review*, 4(2):59–74, 2004.

[178] Marek Płaczek and Grzegorz Kokot. Modelling and laboratory tests of the temperature influence on the efficiency of the energy harvesting system based on mfc piezoelectric transducers. *Sensors*, 19(7):1558, 2019.