Human Intranet: Connecting Wearable and Implantable Devices

Ali Moin
Jan M. Rabaey
Elad Alon

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-221
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-221.html

December 1, 2021

Human Intranet: Connecting Wearable and Implantable Devices

by

Ali Moin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jan M. Rabaey, Chair
Professor Elad Alon, Co-chair
Professor Bruno Olshausen

Fall 2019

Human Intranet: Connecting Wearable and Implantable Devices

Abstract

Human Intranet: Connecting Wearable and Implantable Devices

by

Ali Moin

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Jan M. Rabaey, Chair

Professor Elad Alon, Co-chair

With the explosive growth of the *smart* society and Internet of Things, enormous amounts of information are available in the enhanced world around us and the cyber-world beyond. Hence the traditional human input/output modalities (sight, smell, hearing, taste and touch as inputs, and motor control as output) no longer have the necessary bandwidth or expressiveness to effectively deal with the increasing pace of this *augmented world*. One possible solution is extending the range of human body's existing input/output modalities through wearable and implantable sensors and actuators. Human Intranet (HI) is an open and scalable platform enabling efficient and robust connectivity of wearable and implantable nodes surrounding the human body to each other and to the cloud.

The first half of this thesis presents two embedded systems that are developed as HI nodes: an implantable neuromodulation device featuring closed-loop sense-interpret-actuate functionality aimed at treating neuropsychiatric disorders, and a wearable electromyography-based hand gesture recognition device with real-time classification and in-sensor model training and updates for human-machine interface applications. The experimental results showing simultaneous low-noise, low-power neural recording of local field potential with high-compliance electrical stimulation in a nonhuman primate and online recognition of 21 hand gestures in human subjects are presented.

The second half focuses on the efficient and robust design of the HI data network. A design space exploration platform is developed that leverages mixed integer linear programming and discrete-event simulations to find the optimum network configuration that maximizes its lifetime under reliability constraints. In order to cope with the highly dynamic and lossy wireless channel around the human body, an adaptive network scheme is proposed that learns the network state based on body kinematics and biosignals and reconfigures its parameters accordingly to reduce energy consumption and increase robustness.

To My Beloved Parents, *Roza* and *Morteza*

For their support of my inquisition and exploration, as far back as I can remember.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to start by thanking my PhD advisers, Prof. Jan Rabaey and Prof. Elad Alon. Jan, I could not have wished for a better advisor or more pleasant experience at UC Berkeley, and feel extremely lucky to have had your unconditional support and guidance during these years. Elad, I have always admired your infinite knowledge and brilliant way of thinking. I am grateful for having such excellent role models in my life.

I was fortunate enough to take the Advanced Embedded Systems course with Prof. Alberto Sangiovanni-Vincentelli which led to a significant portion of my graduate research under his insightful supervision and Prof. Pierluigi Nuzzo's mentorhsip.

I would also like to thank Prof. Bruno Olshausen for serving on both my dissertation and qualifying exam committees and reading this thesis.

This thesis would not have been possible without a strong team effort. I would like to thank Prof. Ana Arias' group for collaboration on the printed flexible electrode arrays, Prof. Luca Benini's group for collaboration on the EMG project, Prof. Jose Carmena's group for collaboration on the NHP experiments, Prof. Rikky Muller's group for collaboration on the WAND device, and Ken Lutz for the help with IRB approval and preparing demos.

I would like to thank my awesome group mates, especially George Alexandrov, Andy Zhou, Arno Thielens, Robin Benarrouch, Guillaume Tochou, Abbas Rahimi, Matthew Anderson, Alisha Menon, and Prof. Alvaro Araujo. You made this period of my life more enjoyable, and I hope to stay in touch with you all.

I owe a huge debt of gratitude to all the Berkeley Wireless Research Center (BWRC) and EECS department staff members including Candy Corpus, Shirley Salanio, Fred Burghardt, James Dunn, and Yessica Bravo.

I would like to thank the BWRC sponsors and Semiconductor Research Corporation (SRC) for supporting my research through SONIC, TerraSwarm and CONIX centers.

Last but not least, I would like to thank my parents and family for their infinite love and support throughout my undergraduate and graduate studies.

Chapter 1

Introduction

The world around us is rapidly getting smarter with virtually every single component of it being equipped with sensors and actuators, all of which connected into a network that will soon surpass billions of nodes and trillions of sensors. While the first generation of this network, also known as Internet of Things (IoT), was rather *static* such as smart home appliances placed at fixed locations, its new generation is becoming a lot more *dynamic* with a multitude of nodes attached to robots, drones, and vehicles that are moving freely around. Moreover, the networks are evolving from being either pure sensory data collection or simple actuation entities towards complex closed-loop sense-interpret-actuate systems, enabling distributed autonomous behavior.

An important question is what the role of us, humans, will be in this new smart world surrounded by massive amounts of data available through IoT. Many concerns regarding AI dominance and data overload have already caused a pushback against technology adoption. However, a more constructive approach is to embrace the very same technologies to make us humans smarter and more capable as well. In order to do so, instead of considering the humans with their biological brain as being totally separate from the physical computing world, they should become enmeshed with it. In other words, computing should become *human-centric*. This requires extending the traditional human input/output modalities (sight, hearing, smell, taste and touch as inputs, and motor control as output) by providing enhanced sensing, actuation, and processing power through a network of wearable and implantable devices. In this thesis, we tackle this *Human Intranet (HI)*, depicted in Fig. 1.1, from various aspects: low-power embedded system design, hardware-efficient machine learning algorithms, and energy-efficient, robust networking methods.

After a brief introduction in this chapter, we describe two embedded systems that we developed as potential nodes of HI: an implantable closed-loop neuromodulation system demonstrating sense-interpret-actuate capability (Chapter 2) and a wearable gesture recognition system leveraging a hardware-efficient machine learning algorithm (Chapter 3). A typical HI is expected to accommodate tens of these nodes in the near future, each of them posing different criteria on the HI in terms of network requirements. Hence, the design space for such a network is relatively large and finding its optimum configuration manually

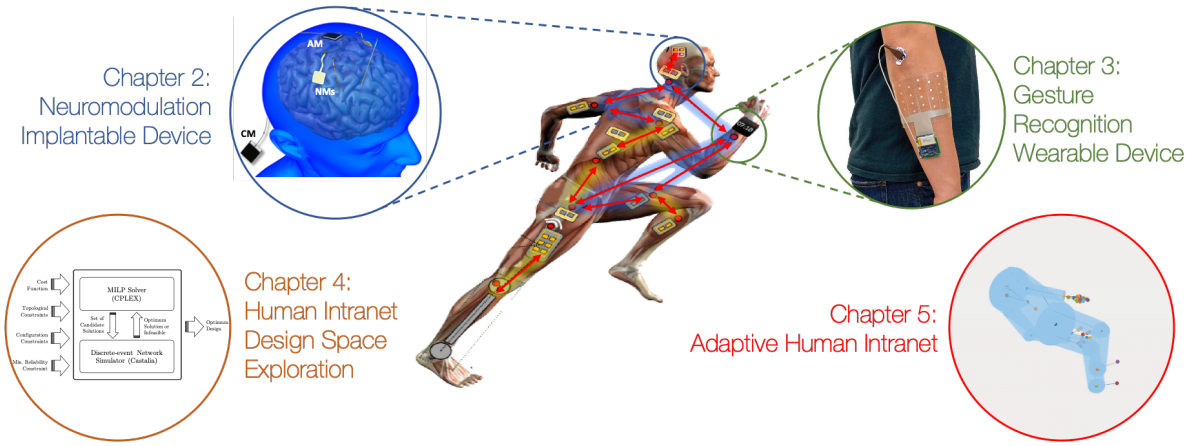


Figure 1.1: Illustration of a Human Intranet example [1] and the overall scope of this thesis.

is a daunting task. This motivates the idea of having a design space exploration (DSE) algorithm, as explained in Chapter 4, that searches for the optimum network topology and protocols in an efficient way. Finally in Chapter 5, we propose a novel adaptive network protocol that learns the status of network from body kinematics and biosignals, and reconfigures the network parameters based on them. This allows for having a more energy-efficient and robust HI by overcoming the challenges caused by lossy and highly dynamic wireless channel around the human body.

In the next chapter, a distributed, modular, intelligent, and efficient neuromodulation device, called OMNI, is presented as our first example of an HI node. Closed-loop neuromodulation systems aim to treat a variety of neurological conditions by dynamically delivering and adjusting therapeutic electrical stimulation in response to a patient's neural state, recorded in real-time. Existing systems are limited by low channel counts, lack of algorithmic flexibility, and distortion of recorded signals from large, persistent stimulation artifacts. OMNI supports closed-loop recording and stimulation on 256 channels from up to 4 physically distinct neuromodulation modules placed in any configuration around the brain, hence offering the capability of addressing neural disorders that are presented at the network level. In order to test the system *in vivo*, we introduce a non-modular version of OMNI, called WAND, that is compatible with a custom chronically implanted microelectrode array implanted in a male rhesus macaque monkey. It is capable of wirelessly recording and stimulating on 128 channels with on-board processing to fully cancel stimulation artifacts, detect neural biomarkers, and automatically adjust stimulation parameters in a closed-loop fashion. OMNI/WAND enables new research applications requiring high-throughput data streaming, low-latency biosignal processing, and truly simultaneous sensing and stimulation. It combines custom application specific integrated circuits (ASICs), an on-board FPGA, and a low power bidirectional radio. We validate wireless, long-term recordings of local field potentials (LFP) and real-time cancellation of stimulation artifacts in a behaving nonhuman

primate (NHP). Moreover, we use it to demonstrate a closed-loop stimulation paradigm to disrupt movement preparatory activity during a delayed-reach task in an NHP *in vivo*.

Chapter 3 introduces a hybrid flexible biosensing system that interfaces with the body through a screen-printed, conformal electrode array and runs an in-sensor adaptive learning algorithm. Wearable and flexible biosensing devices provide a seamless interface to the human body, allowing continuous and non-invasive access to biosignals for real-time analysis. However, most wearable devices only measure and transmit raw biosignals with little to no local processing and classification, offloading the more computationally demanding steps to an edge device or the cloud. Our large-area flexible array unobtrusively provides high channel counts without large, cumbersome cables. We implement a neuro-inspired algorithm which enables not only real-time classification but also efficient model training and updates, all performed locally on the wearable device. Using our device, we demonstrate gesture recognition with high-density surface electromyography, a high-bandwidth biosignal that varies with changing situational contexts such as different arm positions and sensor replacement on alternating days. Our system can quickly learn and classify 13 hand gestures with 98.34% accuracy, and classification accuracy can be preserved across different contexts by performing model updates without additional computation on an external device.

In Chapter 4, we address the design space exploration of HI consisting of wearable and implantable technologies, a task that is increasingly challenging as the number and variety of devices per person grow. Our method efficiently decomposes the problem into smaller subproblems by coordinating specialized analysis and optimization techniques. We leverage mixed integer linear programming to generate candidate network configurations based on coarse energy estimations. Accurate discrete-event simulation is used to check the feasibility of the proposed configurations under reliability constraints and guide the search to achieve fast convergence. Numerical results show that our application-specific approach substantially reduces the exploration time with respect to generic optimization techniques and helps provide clear identification of promising solutions.

Finally, Chapter 5 presents an adaptive HI scheme that reconfigures the network by learning from body kinematics and biosignals. The rapid advent of wearable and implantable devices necessitates efficient and robust ways of connecting them to each other and to the cloud. However, the wireless channel around the human body poses unique challenges such as high and variable path loss caused by frequent changes in node positions and surrounding environment. Our adaptive algorithm has extremely low overhead since the kinematics and biosignals are already captured by HI sensor nodes for other purposes. We exploit the periodic channel fluctuations in activities like walking by reusing accelerometer data and scheduling packet transmissions at proper times. Moreover, our algorithm predicts the network state based on changes in biosignals such as electromyography (EMG) and electrocardiography (ECG) to reconfigure its parameters in real time. In order to evaluate our method, we develop a realistic body channel emulator that generates the path loss for everyday human activities. Simulation and experimental results of our hardware implementation on a custom-made test bed show up to 19% improvement in packet delivery ratio and up to 27% reduction in power consumption.

Chapter 2

Neuromodulation Implantable Device

2.1 Introduction

Neurological disorders broadly affect a billion people worldwide and account for 6.3% of the global burden of disease [2]. Treatments for neurological disorders have traditionally been limited to pharmaceutical intervention, therapy, and counseling. For debilitating conditions, these therapies can be ineffective or even cause harmful side effects. Deep brain stimulation (DBS) is an effective solution for the symptomatic treatment of movement disorders such as Parkinson's Disease, Essential Tremor, and Dystonia. Unlike movement disorders, neuropsychiatric disorders manifest in the brain at the systems level, necessitating a distributed, network approach that acts in closed-loop and responds in real time. Recent research has demonstrated the potential of DBS and closed-loop neuromodulation in treating some neuropsychiatric disorders [3], yet a more sophisticated system than the current state-of-the-art is required in order to address disorders that manifest in complex neural pathways and affect physically distant regions of the brain.

Current research and commercial devices for the treatment of movement disorders and epilepsy such as Medtronic's Activa PC+S [4] and NeuroPace's RNS System [5] rely on specific anatomical targets and electrode placement for stimulation. More importantly, these devices are housed in a central hub with long leads connecting electrodes to recording and stimulation circuitry. This configuration limits the number of recording/stimulation channels to a small number per device, since each channel needs a corresponding wire from electrode to hub, reducing cable flexibility and increasing size and tissue displacement. There have been recent efforts to address these issues [6], but these consume too much power to be operated chronically.

In order to adequately address neurological disorders at the systems level, a device is required that has several unique features not found in current state-of-the-art neurostimulator systems:

Distributed: The device should be able to reach any region of the brain based on specific patient needs. Therefore, it should support multiple arrays, both cortical and sub-cortical,

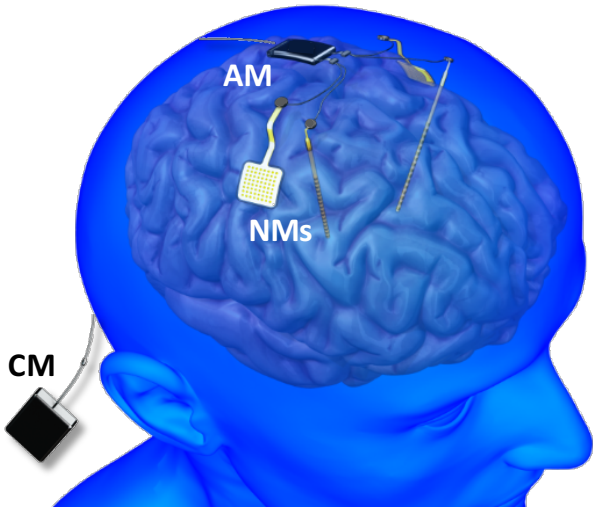


Figure 2.1: OMNI device showing multiple NMs attached to the AM, sitting just above the skull, where data and commands are multiplexed. The CM is responsible for controlling and powering all sub-modules.

with a large number of recording and stimulation channels.

Modular: The architecture and physical placement of the device should be easily reconfigurable, allowing the clinician to decide the most effective configuration of implants for each individual patient based on his or her needs.

Intelligent: The device should operate in a closed-loop fashion, leveraging advanced signal processing to determine stimulation parameters and locations in real-time, and ensuring that stimulation is only delivered when and where it is needed.

Efficient: Since the device is implanted, it should be extremely energy-efficient to have minimal need for battery recharging or wireless power transmission without compromising the performance of the system in the above aspects.

We introduce the Octopus Mimetic Neural Implant (OMNI) device (Fig. 2.1). OMNI's modular and distributed approach to neuromodulation is a key enabler for a new class of closed-loop treatment of neurological disorders. Throughout the chapter, we focus on the power and communication networks that alleviate the excessive cabling and interconnects required for high throughput recording and stimulation from hundreds of electrode sites. Finally, we present a single-module version of OMNI, called WAND, which enables *in vivo* experiments in a nonhuman primate (NHP).

This project is a collaboration between UC Berkeley, Lawrence Livermore National Laboratory (LLNL), and Cortera Neurotechnologies. The author's contribution is focused on the system integration, design of embedded modules, data link protocol, ASIC testing, integrated system bench-top testing, and *in vivo* experiments.

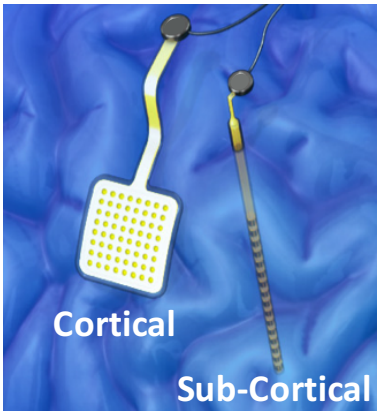


Figure 2.2: NMIC package attached to cortical and sub-cortical electrode arrays.

2.2 System Overview

OMNI consists of three main types of modules. The Neuromodulator Modules (NMs) are "smart electrodes," each of which are comprised of a high-density thin-film electrode array integrated with a single custom low-power application specific integrated circuit (ASIC). The ASIC records and digitizes the neural signals at each electrode and sends the data in packets across a cable to a central router on the skull called the Aggregator Module (AM). The AM serializes and packetizes data from 1 to 4 NMs (scalable up to 8) and sends it through a single cable to the main processor of the device called the Control Module (CM). The CM performs advanced processing on the received data and determines stimulation locations and parameters. It then sends stimulation commands to the AM, which routes them to the desired NMs. This architecture enables advanced closed-loop neuromodulation in a highly scaled and distributed manner. Each module is discussed in detail below.

2.2.1 Neuromodulator Module (NM)

NMs are OMNI's interface to the brain. They are implanted in the brain and are responsible for recording neural signals and stimulating the desired brain targets. Based on the specific symptoms of an individual patient, the clinician can decide the number and placement of NMs. Each NM consists of two main parts:

1) Micro-Electrode Arrays: Thin-film high-density micro-fabricated arrays of either 32 or 64 electrodes are placed either on the surface of the brain cortex (cortical arrays) or penetrated into the brain tissue (sub-cortical arrays) as shown in Fig. 2.2. The ability to choose between these two types of electrode arrays enables OMNI to reach multiple brain regions in a patient-specific configuration. The micro-electrode arrays are designed and fabricated by LLNL.

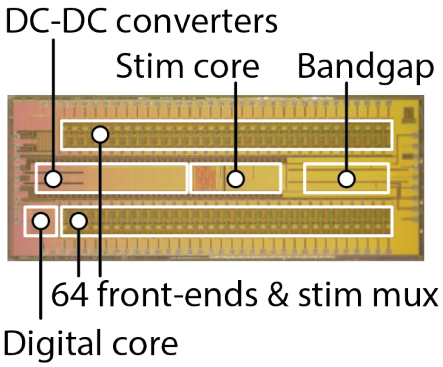


Figure 2.3: Micrograph of custom neural interface IC (NMIC) with annotated subcircuits [9].

2) Neuromodulator Module Integrated Circuit (NMIC): Each NMIC consists of 64 recording channels and 4 stimulators that can address any of the 64 channels. Neural signals are acquired, digitized, and transmitted at the electrode interface by the NMIC, which requires a single 6-wire cable bundle for power and communication. Unlike conventional neural interface ICs, the custom NMICs enable simultaneous low-noise, low-power neural recording of local field potential (LFP) with high-compliance electrical stimulation (Fig. 2.3). To address both direct and indirect artifact, the NMIC recording frontends are designed simultaneously for low noise ($1.6\,\mu V_{rms}$) recording and a large linear input range of 100 mV. The input range is over 10 times larger than conventional designs [7, 8] and avoids saturation in the presence of a large stimulation artifact. This large dynamic range is achieved with a mixed-signal architecture that integrates the ADC into the feedback loop, thereby reducing the required gain and signal swings. The architecture also resets at every sample, enabling memoryless sampling and rapid recovery from stimulation artifacts. Therefore, stimulation artifacts do not persist beyond the samples when stimulation is occurring.

The NMIC prevents large amplitude indirect artifacts by employing stimulators with highly accurate charge balancing [9]. Accurate charge balancing is achieved by reusing the same current source for each stimulation phase and a return-to-ground stimulator architecture. Stimulation current amplitudes, pulse timing, and frequencies are also rapidly reprogrammable, as quickly as every pulse. The NMIC design is done by our collaborator, Cortera Neurotechnologies.

2.2.2 Aggregator Module (AM)

The AM acts as a router for power and data, connecting up to 4 NMs (limited by connector design but scalable to 8) to the main processing unit of the system (CM). The AM is a key enabler for OMNI's modularity, allowing an arbitrary number and type of NMs to be connected to the rest of the system. It also enables two important features: (1) significant reduction in cable complexity, and (2) a safety mechanism in the event of a cable or NM

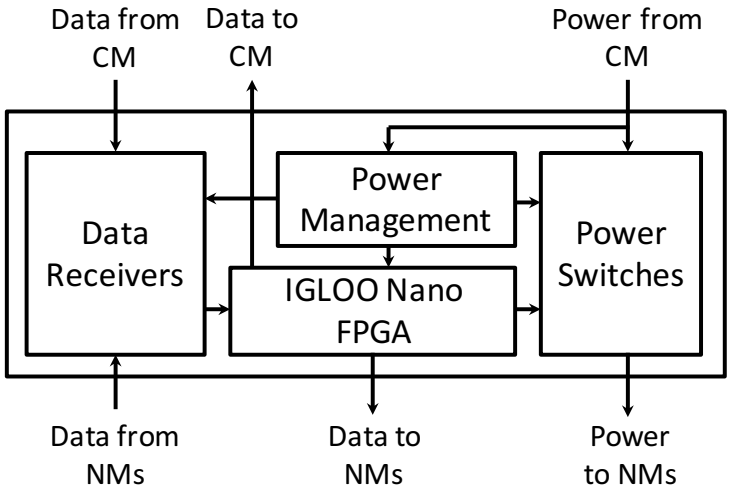


Figure 2.4: Block diagram of Aggregator Module (AM).

failure by providing safe shutdown upon a detected error. Having a separate percutaneous cable to connect each NM directly to the CM increases implant complexity, infection risk, and failure risk. Therefore, the NMs are first routed through a short 10 cm cable to the AM, which is implanted just above the skull, and from there to the CM via a longer 50 cm cable.

Fig. 2.4 shows the block diagram of the AM. An IGLOO nano (Microsemi Corp.) low-power FPGA is used for the digital protocol logic and I/O handling. The power management block generates the AM DC supply from the AC power distribution network. Also, since the data signals are single-ended and AC-coupled, there is a cross-coupled inverter pair as the receiver for each signal. These networks are discussed in detail in Section 2.3.

2.2.3 Control Module (CM)

The Control Module (CM) is an integrated and programmable data processing, storage, and telemetry module that acts as the brain of the system. A block diagram of the CM is shown in Fig. 2.5. It houses a Cortex-M3 microprocessor which is capable of running algorithms required for closed-loop treatment, allowing the device to choose where, when, and how stimulation should take place. The on-board Nordic nRF51822 2.4 GHz radio provides a 2 Mbps bidirectional wireless link for remote logging and configuration. The CM is also responsible for efficiently and reliably powering the rest of the OMNI system. It achieves this by adaptively generating differential AC power signals from a single Li-ion battery.

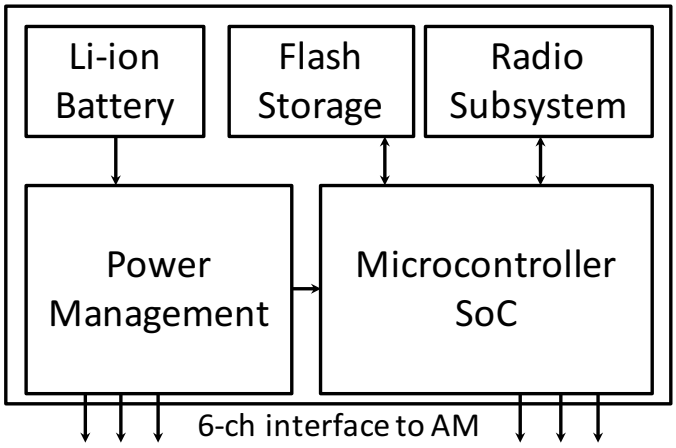


Figure 2.5: Block diagram of Control Module (CM).

2.3 Data and Power Networks

The distributed nature of OMNI poses a challenge in powering and communicating with each sub-component. In order to keep the number of bundled wires in the cable and the number of physical connections from the NM to the AM small, we designed a 6-wire interface. Two differential power wires, three data signal wires and one common wire support a custom communication and power distribution protocol.

Similar to computer networks, a complete protocol stack from the physical layer (cables) up to the application layer (control interface) is defined. The cable design minimizes cross-talk and noise while keeping cable thickness minimized by using a symmetric cable structure with a central twisted-pair for the power wires. The following subsections describe the data link layer and power distribution network, respectively.

2.3.1 Data Communication Network

A full-duplex data communication network transfers packets between the system modules. To support the modularity of the system, we have designed a custom time-multiplexed protocol that supports a variable number of up to 8 NMs. The 20 MHz bandwidth in the CM-AM link is divided into groups of 10 time slots called a *frame*, each slot with 2 MHz bandwidth and allocated to one of the 4 (up to 8) NMs. The first two time slots are used for synchronization between modules and some low-level controls such as reset. These are crucial control commands that should be at the lowest data layer and independent from the packet decoder.

There are two signals between modules in the downlink (CM-to-AM-to-NM) direction: *valid* and *data*. When there is data available for a time-slot, the *valid* signal is set to high

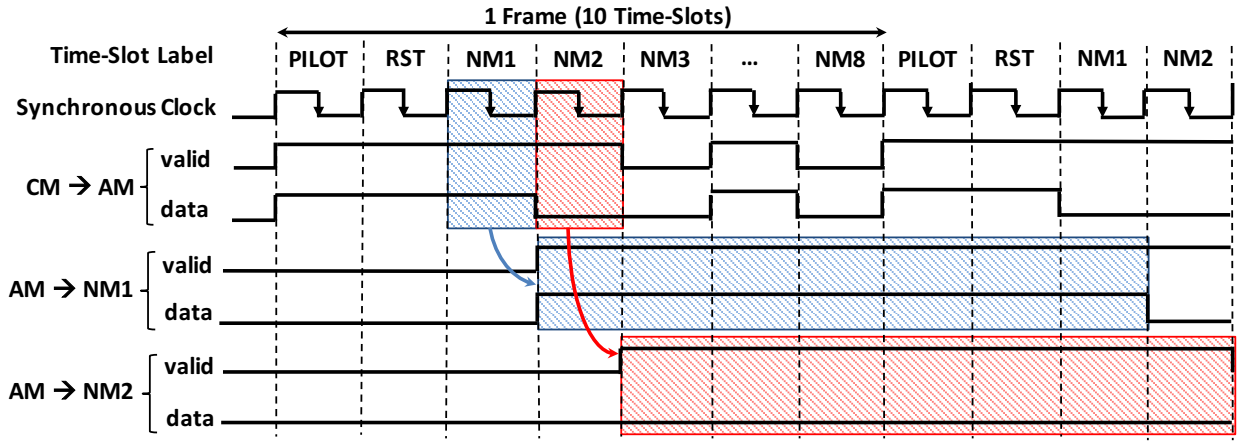


Figure 2.6: Digital protocol showing command bits from CM to NMs. The PILOT and RST bits in the CM-AM frames signify the start of a valid command to the NMs and are followed by 8 slots for command bits to 8 NMs. This example shows valid command bits being sent from the CM to two NMs. Signals are shown after being recovered from the AC-coupled link at the AM and NM receivers.

during that time-slot and the *data* signal contains the actual bit value. Therefore, the receiver picks up the data whenever the corresponding *valid* is high, allowing the CM to address any subset of NMs. The packets in this direction are CM commands. Fig. 2.6 shows how these command bits are sent from CM to AM to particular NMs.

Unlike the downlink, the uplink (NM-to-AM-to-CM) direction does not need synchronization or control signals. Instead, data is transmitted on the NM-AM *data* line with a fixed header indicating the start of the packet, followed by 64 channels of neural data. This way, we omit the *valid* line in this direction to minimize the number of wires. The packets in this direction are either digitized neural signals or register read replies.

Although a differential signaling scheme would be more robust to interference, we have chosen a single-ended approach in order to minimize power and the number of wires in the cable. Moreover, based on medical implant regulations, the implanted cables must carry zero DC component to prevent long-term tissue damage and current flow into tissue in the case of cable failure. Therefore, the signals are AC-coupled at both ends of the cable and are recovered by the receiver circuits.

2.3.2 Power Distribution Network

Unlike existing neuromodulation devices where battery, stimulators, and recording hardware are fixed and located in the same package, the OMNI system is comprised of a variable number of NMs with varying power requirements. This presents a challenge in powering the system since it must support a very broad range of load currents that are all dynamically

changing. Each NM's power dissipation can vary from a few hundred µW to a few mW between recording and stimulation. Thus, the effective load that the power driver must drive can vary by a similar amount. Any powering scheme must be able to adjust its output power so the correct amount is delivered to each module while maintaining high power efficiency. Furthermore, to ensure efficient operation of the NMs, power should be delivered with a constant voltage across all load conditions. Finally, any implanted cables should carry zero DC component, so the power distribution must be AC.

In order to address these challenges, an adaptive powering scheme is implemented to ensure the correct amount of power is delivered safely and reliably to each individual NM and AM. This powering scheme consists of two differential AC voltage signals generated via a pair of Class E power amplifiers. Each power signal's output voltage is independently set by a feedback network, allowing the power drivers to adapt to asymmetrical load variations. Without feedback, the power drivers are supplied by a constant supply voltage and generate an output voltage that varies directly with load resistance. With feedback included, the supply voltage is adjusted automatically in order to set a constant output voltage.

2.4 Integrated System Bench-Top Testing

Figure 2.7 shows the integrated system benchtop demonstration setup which shows an OMNI configuration with two NMs. A sample neural signal is recorded on NM1 and sent as digital packets to the CM through the AM using our custom communication protocol. The intermodule signals are captured on an oscilloscope and displayed in the figure. For this demonstration, the CM streams data to a PC for offline computation and configuration of stimulation patterns, and sends stimulation commands to NM2. In practice, closed-loop algorithms can run on the CM to adaptively update the stimulation parameters and locations based on subsequent recorded signals, eliminating the need for a PC.

Recent efforts [6] have improved significantly on commercial systems [4], achieving a high-channel recording and stimulation system using mainly off-the-shelf components (Table 2.1). However, this system cannot be operated chronically due to the high power dissipation of its off-the-shelf recording (Intan RHD2164) and stimulation (Cactus Semiconductor CSI021) hardware. While OMNI dissipates 2 mW for simultaneous recording on all 256 channels, a similar system using Intan ASICs would consume about 20 mW for the same number of channels. Compared to [6], OMNI supports a large number of channels at a significantly lower power consumption and with fewer implanted wires.

2.5 *In Vivo* Testing

In order to validate the capabilities of OMNI device through *in vivo* experiments, we restructured the CM board to include two NMICs on the same PCB and be compatible with a custom chronically implanted microelectrode array that was implanted in a male rhesus

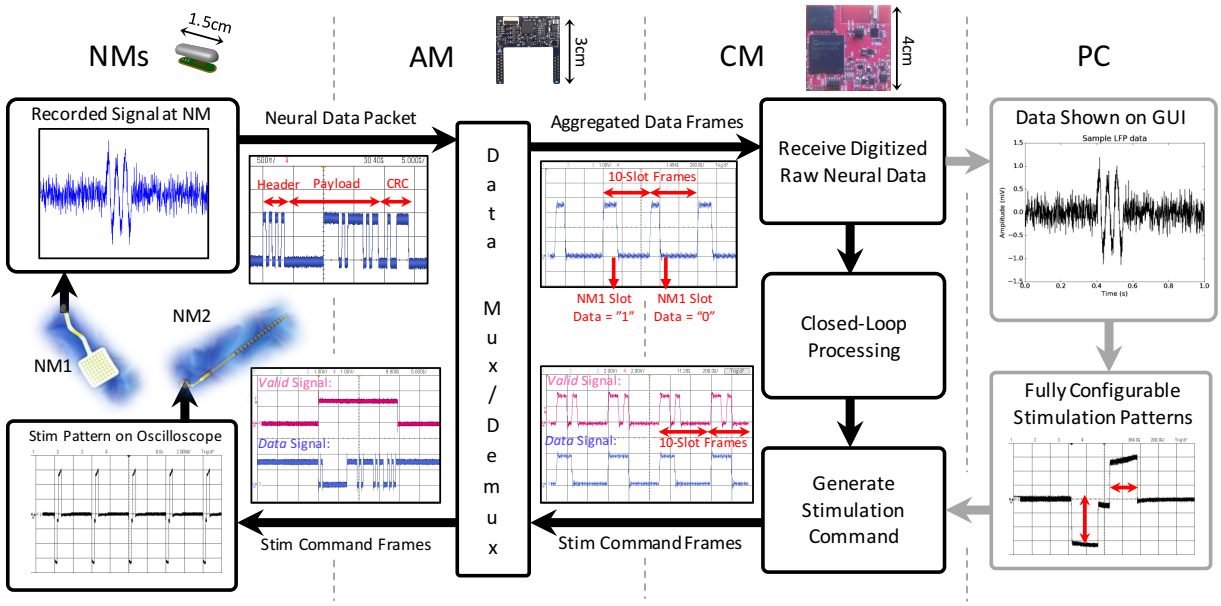


Figure 2.7: Integrated system demonstration diagram showing benchtop testing of OMNI. Generated neural data is recorded by one NM and sent to the CM through a custom digital communication protocol. The CM communicates wirelessly with the GUI running on the PC, which processes neural data and sends stimulation commands back to the CM.

Table 2.1: Comparison of OMNI with current state-of-the-art.

Specification	**This Work**	**Wheeler [6, 10]**	**Stanslaski [4]**
Electrode Arrays	**4 (scalable to 8)**	5	2
Channels	**256 (scalable to 512)**	320	8
Stimulators Per Array	**4**	2	1
Wires Per Cable	**6**	10	4

macaque monkey. We named this re-spun system WAND, which stands for Wireless Artifact-free Neuromodulation Device. The system architecture and *in vivo* experiment results are presented in this section.

2.5.1 WAND Architecture

WAND components and architecture are shown in Figs. 2.8 and 2.9. For this work, the form factor was designed to fit into the polyetherimide housing for a custom chronically implanted microelectrode array (Gray Matter Research, Bozeman, MT). The device has a board area of 10.13 cm^2 and weight of 17.95 g together with a rechargeable 500 mAh Li-ion battery pack, allowing 11.3 hours of continuous, wireless operation (Fig. 2.8(a)). The main components of

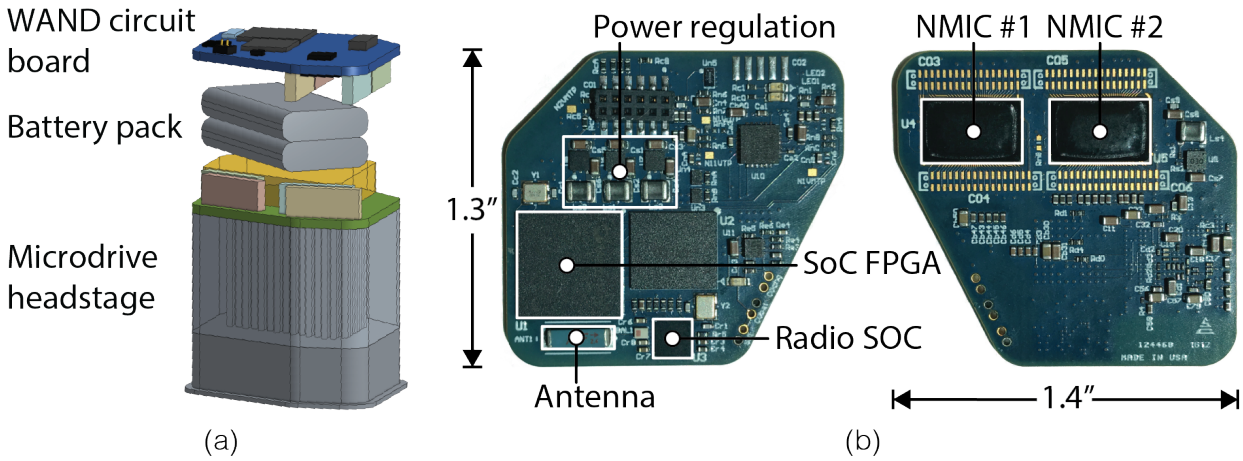


Figure 2.8: WAND system architecture. (a) 3D CAD model of WAND with the primate headstage and battery pack, shown without polyetherimide case. (b) Top- and bottom-view photographs of WAND circuit board with relevant subsystems annotated.

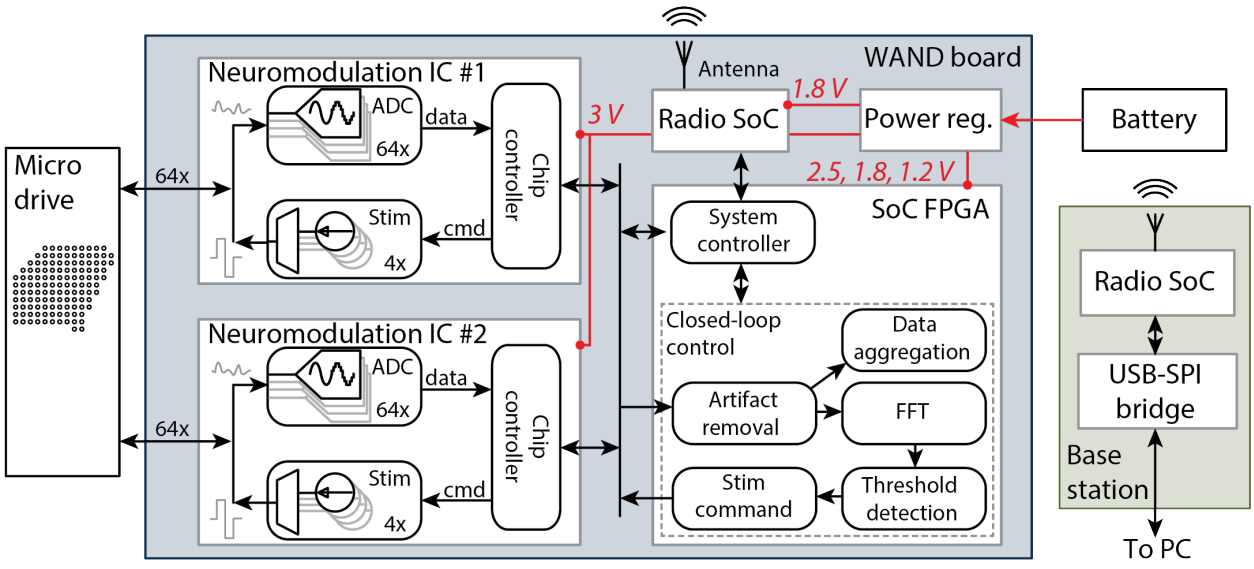


Figure 2.9: Functional diagram of the WAND system showing data and power connections on the main device board, and connections to the microdrive electrode array, battery, and a wireless base station.

WAND are the pair of custom NMICs, an SoC FPGA, a radio SoC, and support circuitry for power regulation and programming (Figs. 2.8(b) and 2.9).

All 128 channels of neural data from both NMICs are sampled, digitized (15 bits, 1 kS/s),

and transmitted, while stimulation parameters and commands are received via a bidirectional interface with the FPGA (Fig. 2.9). The on-board FPGA and microcontroller SoC interfaces with the NMICs to configure and control the circuits, aggregate and select neural and other sensor data, cancel stimulation artifact, and run closed-loop neuromodulation algorithms (Fig. 2.9). The FPGA is reprogrammable through a serial wire debug interface allowing customization for different applications. A 2.4 GHz Bluetooth Low Energy (BLE) radio SoC allows for robust bidirectional wireless communication up to 2 m from the subject. BLE offers low power telemetry, and customization of the BLE protocol enables data streaming rates close to the 2 Mbps modulation rate. WAND can stream up to 96 uncompressed LFP recording channels in real time to a PC running a custom graphical user interface (GUI) for system configuration and data visualization.

2.5.2 High-Fidelity Multi-Channel Wireless Recording

To evaluate the quality of recordings made using WAND, we recorded 96 channels of LFP activity from an NHP using a chronically implanted microdrive electrode array with access to both cortical and subcortical nuclei (Fig. 2.10(a)). We compared WAND recordings with sequentially recorded neural data from a wired, state-of-the-art, commercial neurophysiology system (Tucker-Davis Technologies, Alachua, FL). Respective recordings from each system have qualitatively similar signal properties, as assessed by computing the power spectral densities (PSD) of the recorded data (Fig. 2.10(b)). The WAND recordings exhibit lower 60 Hz interference due to the lack of long interface cables and better isolated recording references.

To demonstrate robust detection of biomarkers in WAND recordings and establish a baseline for neuromodulation experiments, we recorded LFP activity during a standard self-paced, center-out joystick task (Fig. 2.11(a,b)). In the center-out task, a trial begins with the subject holding the cursor at a center circular target for 500 ms. Following this hold period, a peripheral target appears at one of eight target locations equally distributed around the center target at a distance of 10 cm and the center target is removed from the screen, acting as a "go cue". The subject then moves the cursor (i.e. "reaches") to the peripheral target and holds at this target for another 500 ms. If successful, the subject is administered a small juice reward lasting 800 - 1000 ms. A trial was considered successful if the subject completed the two hold periods within a 10 s period. During this behavior, ongoing beta and high-gamma rhythms are inversely modulated by task-related periods of movement (Fig. 2.11(c,d)). Beta band oscillations are found to emerge during specific motor actions and notably prior to instructed reaches or movements [11, 12, 13]. In pre-motor and motor areas, this rhythm has been linked to neural activity related to motor preparation [14, 15, 16, 17]. The subject had an average reaction time (RT) of 183.3 ± 4.8 (SEM) ms across 400 trials. For LFP signals recorded from pre-motor and motor areas, we found that RT was significantly correlated with the average power of beta band activity around the Go Cue (Pearson's correlation: $r = 0.12$, $p = 0.03$).

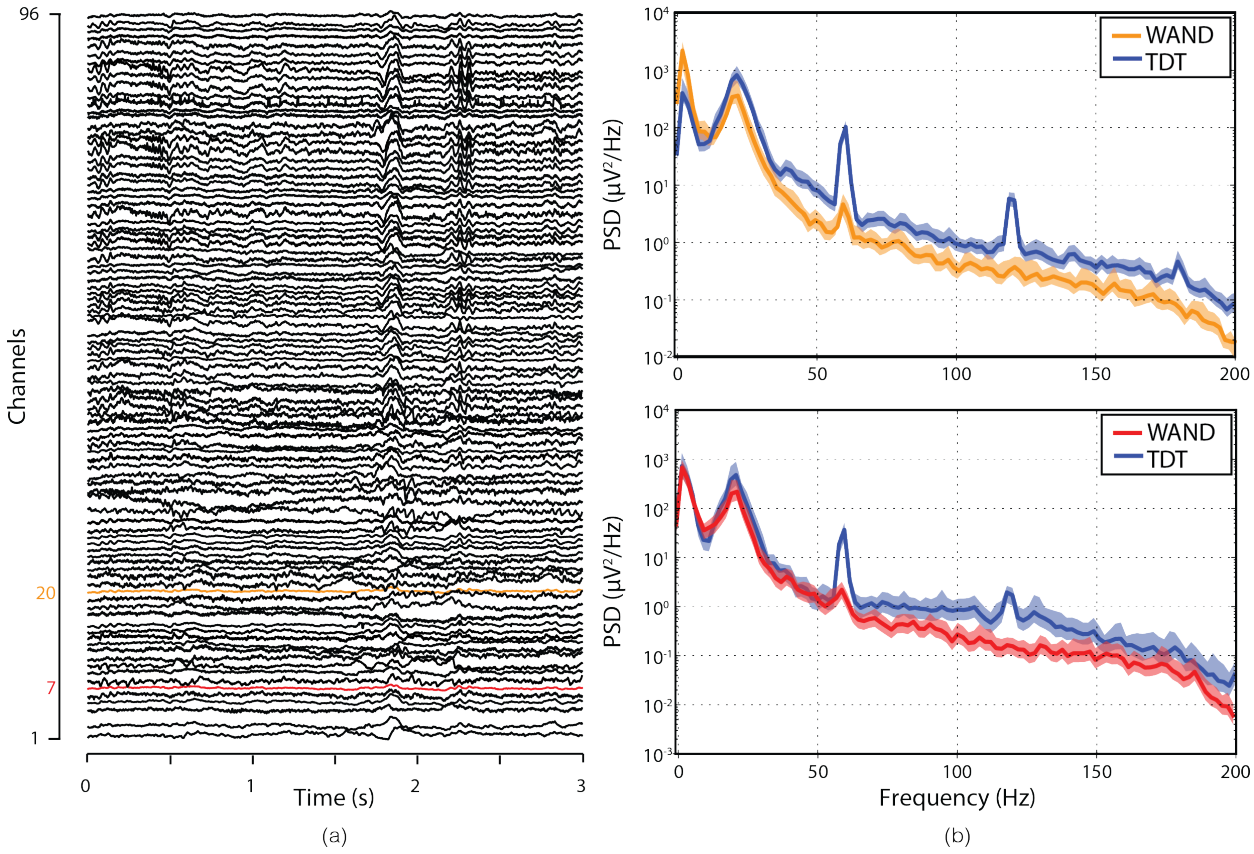


Figure 2.10: Wireless, multi-channel recording. (a) Representative 3-second segments of simultaneous LFP recordings from 96 channels taken during freely moving behavior. (b) Comparison of normalized power spectral density (PSD) from channel 20 (top) and channel 7 (bottom) for recordings taken from WAND and subsequent recordings taken from a commercial wired neurophysiology system (TDT). Error bars are the standard deviation (SD).

To validate long-term, wireless system functionality, we performed unconstrained, overnight recordings for five nights in the subject's home cage, recharging the battery between each session (Fig. 2.12(a)). We recorded on average 10.2 consecutive hours per night, with packet error rates below 0.5%. Offline analysis of the data revealed useful sleep-related biomarkers. Delta (0 - 4 Hz; Fig. 2.12(b), top row) and theta (4 - 7 Hz; Fig. 2.12(b), bottom row) power are known to have elevated power during sleep states relative to wake states [18, 19]. K-complexes are sleep-specific phasic waveforms that occur spontaneously and are observed through the obtained neural recordings during epochs of increased delta power (Fig. 2.13), consistent with classification of sleep state intervals.

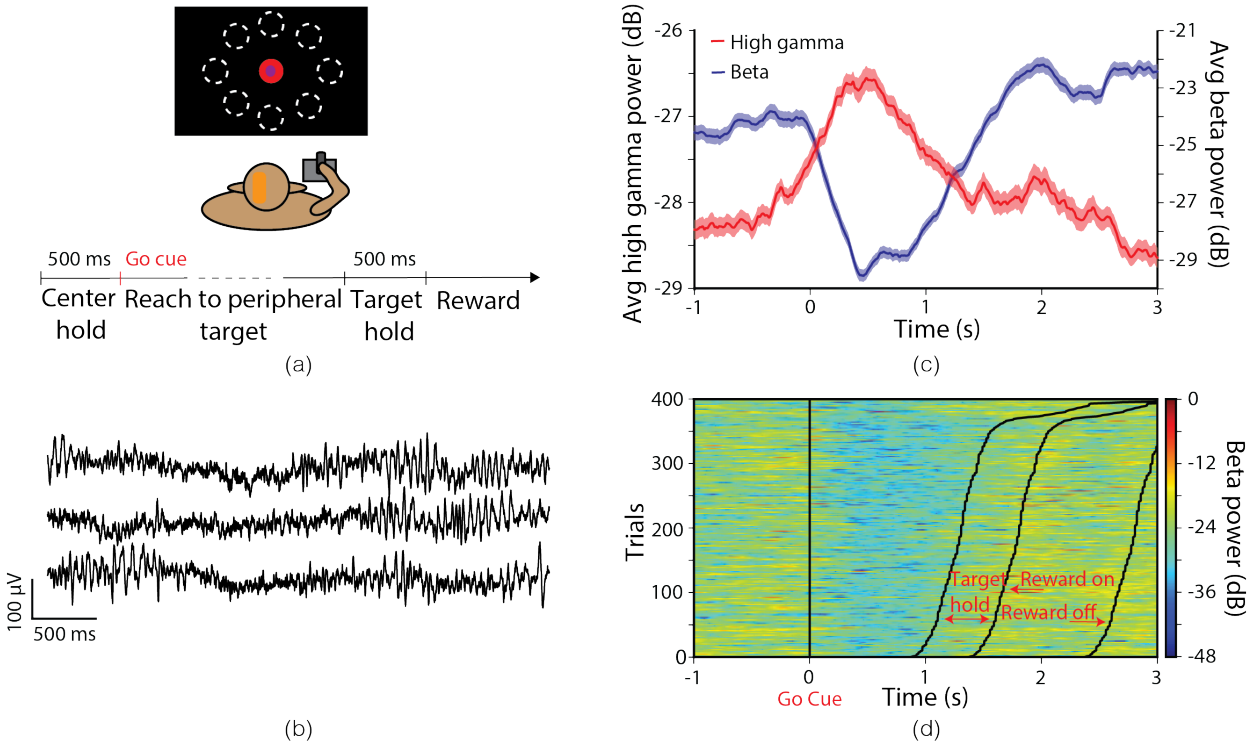


Figure 2.11: LFP recordings during joystick task. (a) Diagram of the center-out joystick task with timeline of task periods for movement and reward. (b) Representative LFP recordings from three channels during the center-out task. (c) Trial-averaged (n = 400) beta (13 - 22 Hz) and high-gamma (70 - 200 Hz) power aligned to the Go Cue during the center-out task. Error bars are the SEM. (d) Beta power aligned to the Go Cue. Each row represents activity from a single trial. Trials are organized by the time to Target Hold following the Go Cue.

2.5.3 Simultaneous Recording and Stimulation

To demonstrate WAND's ability to recover neural signals from stimulation artifacts in real-time, we performed experiments delivering open-loop stimulation while recording LFP. For each set of stimulation parameters, we recorded three consecutive segments of LFP: without stimulation, with stimulation turned on but without backend artifact cancellation, and then with artifact cancellation turned on (Fig. 2.14). During the train of identical bipolar, biphasic stimulation pulses, the segment of LFP with uncancelled artifact (Fig. 2.14, middle section) demonstrated varying artifact morphology due to the non-integer ratio between the sampling rate and the stimulation frequency (99.8482 Hz shown) (Fig. 2.15(a)). Stimulation pulses occurring completely within a single sample integration window caused only a single sample direct artifact, while pulses occurring at the boundary between two integration windows

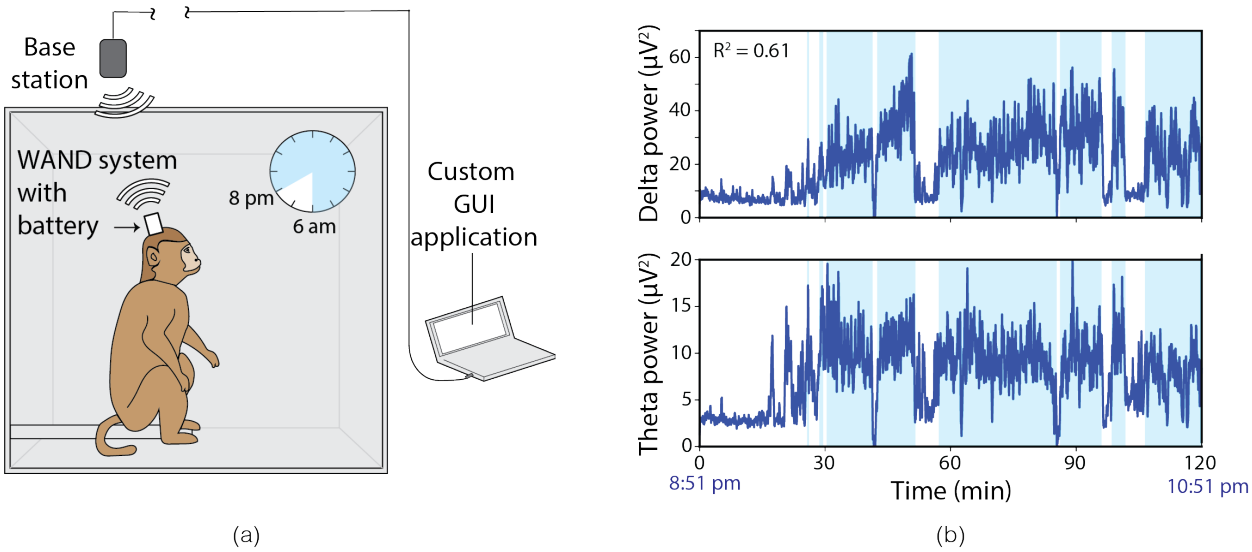


Figure 2.12: Overnight, untethered recording during sleep. (a) Cartoon description of in-cage wireless recordings. (b) Delta (0.5 - 4 Hz) and theta (4 - 7 Hz) power from two-hour segment of overnight recording beginning at 8:51 pm. The powers are significantly correlated ($R^2 - 0.61$). K-means was used to classify the activity into states of increased and decreased delta and theta activity, which are indicated by the absence and presence of the light blue background in the plots.

caused the direct artifact to last two samples. We calculated averaged templates of single- and double-sample artifacts (Fig. 2.15(b)) for varying stimulation amplitudes and pulse widths, and confirmed a linear relationship between these parameters and the artifact amplitude (Fig. 2.15(c)). For all stimulation parameters within our protocol, recorded direct artifacts on the non-stimulating electrodes remained well within the 100 mV linear input range of the frontend amplifiers, despite the high voltages (∼10 V) induced on the stimulating electrodes, thus demonstrating saturation-free recording in the presence of stimulation.

Following both single- and double-sample direct artifacts, the indirect artifacts were very small and brief, visible only in the averaged templates (Fig. 2.15(b) inset). Following a single-sample direct artifact, the indirect artifact was already suppressed to within −60 dB of the peak amplitude by the following sample, and to within the electronic noise floor of 1.65 μV_{rms} by the second sample. Indirect artifacts following double-sample direct artifacts were fully suppressed below the noise floor. These results demonstrate that the recording frontends rapidly recovered from stimulation pulses, minimizing data distortion.

The residual stimulation artifacts still caused broadband contamination of the recorded spectrum, which we quantified with the ratio, R = 32.78 dB, of signal power integrated form 1 - 200 Hz of LFP during stimulation to baseline LFP (Fig. 2.16). Since the recorded artifacts are short in duration (1-2 samples), we chose to implement a method of linear interpolation

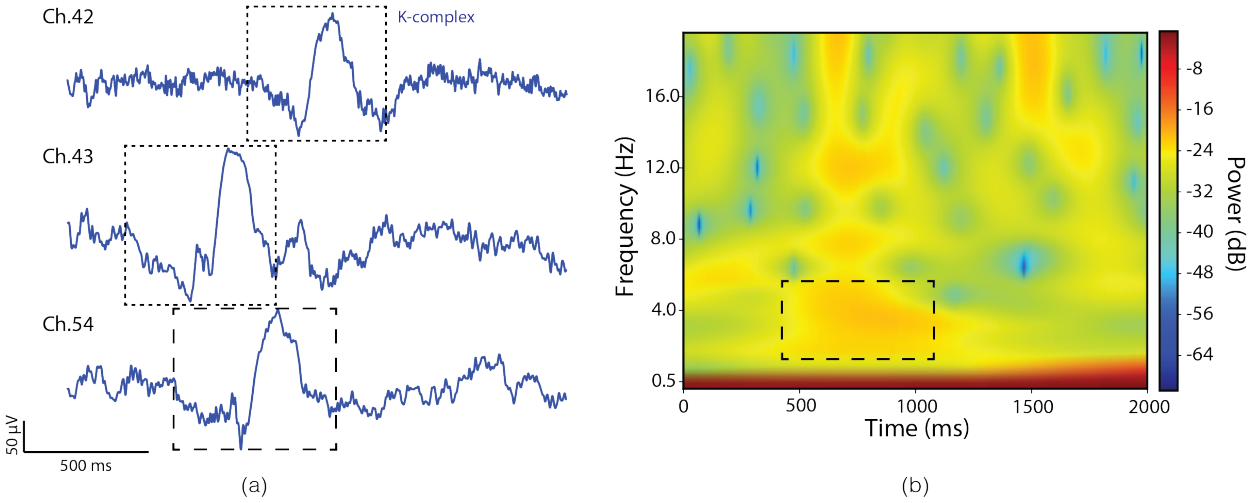


Figure 2.13: (a) Example K-complexes from the caudate (first two rows) and from the anterior cingulate cortex (ACC) (third row). (b) Spectrogram of activity from channel 54 during the same time window as the waveform shown in (a). Increased delta power occurs coincidently with the K-complex.

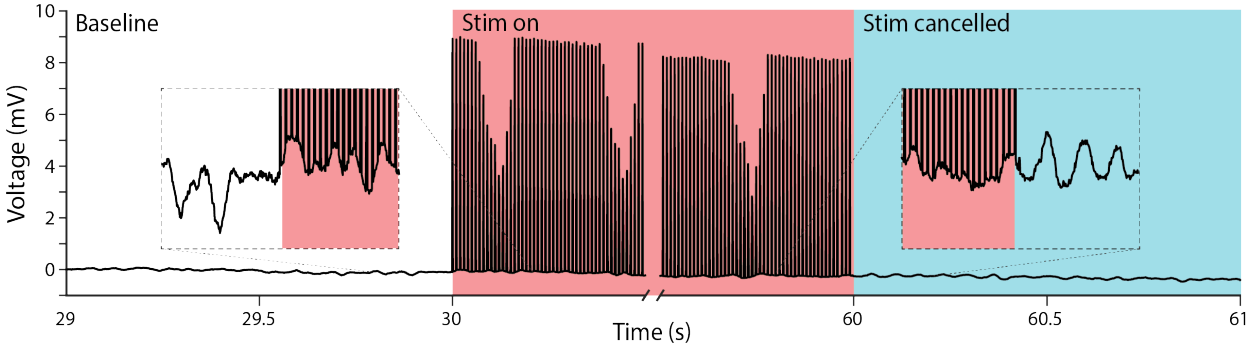


Figure 2.14: (a) 1-second segments of raw signals recorded during different epochs of the open-loop stim experiment: baseline LFP with no stim (white), stim with no artifact cancellation (red), and stim with artifact cancellation (blue).

for artifact cancellation [20]. Samples coinciding with stimulation pulses are flagged by the NMIC, ensuring accurate detection of artifacts. Samples were then buffered in the FPGA, and artifacts were removed by linearly interpolating between the pre-artifact sample and the sample following the maximum possible direct artifact duration (Fig. 2.15(a)).

While more sophisticated techniques may be employed, we found that this simple linear interpolation was sufficient to suppress the artifact power below the neural signal spectrum.

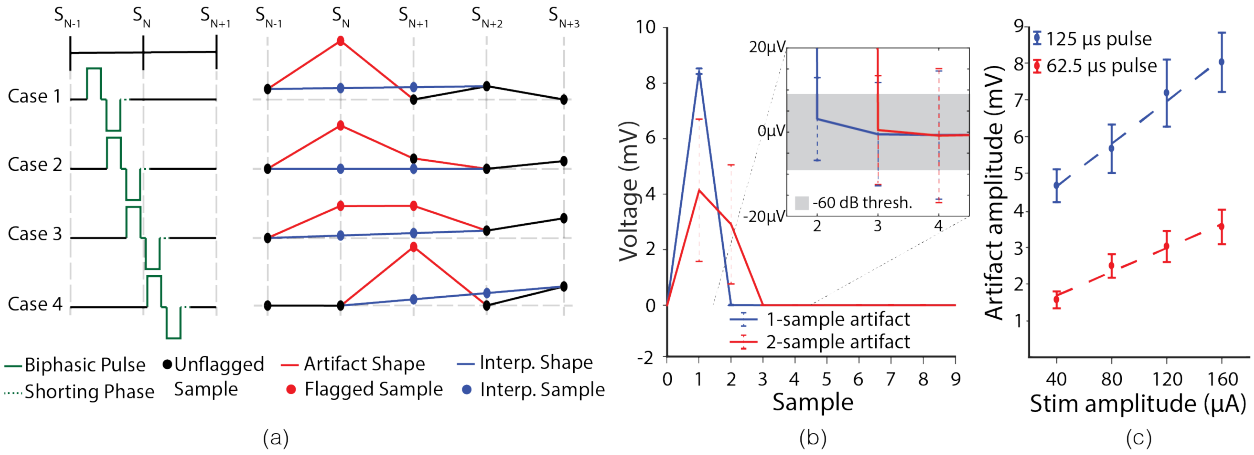


Figure 2.15: (a) Different cases of relative phase between stim pulse and sampling periods (left) with example resulting samples, artifact flags, and cancelled samples using linear interpolation (right). (b) Averaged templates of single- (blue) and double-sample (red) flagged artifacts. The inset is a zoomed portion showing decay of artifact to within −60 dB of the artifact peak (shaded gray). Error bars are the SD (n = 2106 for single-flag and n = 897 for double-flag). (c) Average amplitude of artifact from stim amplitudes between 40 µA and 160 µA and pulse widths of 125 µs (blue) and 62.5 µs (red). Error bars are the SD (n = 3003 artifacts).

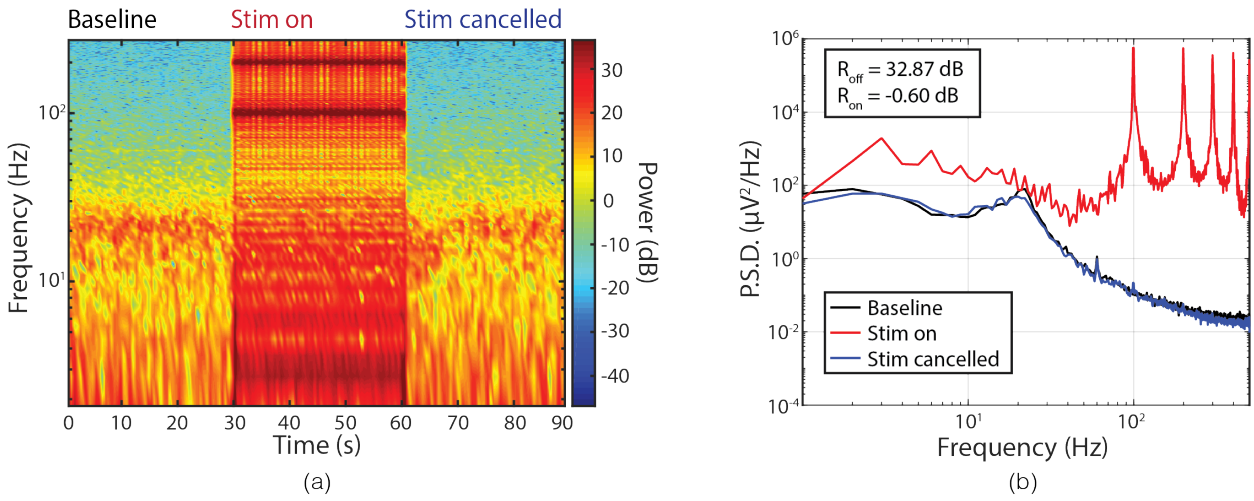


Figure 2.16: (a) Spectrogram of full 90-second recording during the different stim epochs. (b) Welch power spectral density estimate for each 30-second epoch.

Interpolation over two samples at 100 Hz in baseline LFP data without stimulation caused no significant degradation of the spectrum (R = 0.0091 dB). Furthermore, this method did not depend on the actual values of the artifacts and was not affected by the varying artifact morphology, which would have complicated and increased convergence times of template subtraction and adaptive filtering techniques. With on-board artifact cancellation enabled, we were able to recover the baseline LFP spectrum for signals recorded during the simulation pulse train with R = −0.60 dB (Figs. 2.14 and 2.16).

2.5.4 *In Vivo* Biomarker Extraction and Closed-Loop Experiment

To further demonstrate WAND's ability to mitigate stimulation artifacts in real-time, and to perform responsive stimulation using on-board computations, we designed a closed-loop stimulation experiment (Fig. 2.17) to disrupt movement preparatory activity during a delayed-reach task (Fig. 2.18). The sequence of events in the delayed-reach joystick task is similar to the center-out task, with the exception being that the peripheral target appears prior to the "go cue", which is signaled with the disappearance of the center target. The hold period for the center target lasts 400 ms before the peripheral target is shown. This initiates the "delay period" with a duration that varied randomly trial-by-trial with a range of 200 - 400 ms. After the delay period, the center target disappears from the screen signaling the "go cue" and the subject is cued to reach to the peripheral target. The range of delay durations was chosen to allow for movement preparation and to ensure that microstimulation occurred near the go cue for a nontrivial number of trials. Previous work in macaque monkeys has shown that microstimulation delivered to dorsal premotor (PMd) and primary motor (M1) cortical sites during the delay hold period of a delayed-reach task disrupts preparatory activity and causes an increase in RT [16]. Stimulation was timed synchronous to the task and was not triggered on recorded neural activity. We reproduced this result by detecting periods of preparation (holding) prior to movement using recorded neural activity in M1 and delivered stimulation to electrodes in PMd in response.

Beta band activity (13 - 30 Hz) is known to reflect movement states, with lower beta band power associated with periods of movement and higher beta band power associated with the absence of movement. Thus, we chose beta band power as the WAND control signal for closed-loop classification of hold periods prior to movement. In this way, closed-loop operation of WAND was completely agnostic to the behavior task states, and stimulation delivery relied solely on the control signal. We heuristically selected a policy of delivering a preconfigured stimulation pulse train when both the beta power and its derivative exceeded programmed thresholds during the delayed-reach task (Fig. 2.18). The pulse train parameters were selected to closely match values used in previous work (333 Hz for 57 ms) [16] within WAND specifications. To avoid stimulation multiple times within the same delay hold period, our policy also incorporated a "dead time" of three calculation periods, or 768 ms. Neural activity is recorded throughout the task, and while stimulation turn-off can be based on

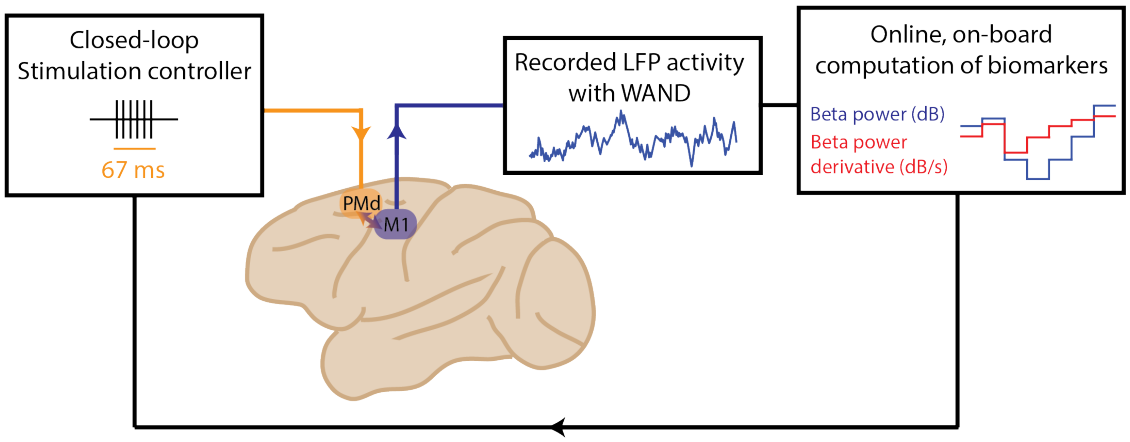


Figure 2.17: Description of closed-loop paradigm, where recorded activity in M1 is used to control stimulation in PMd.

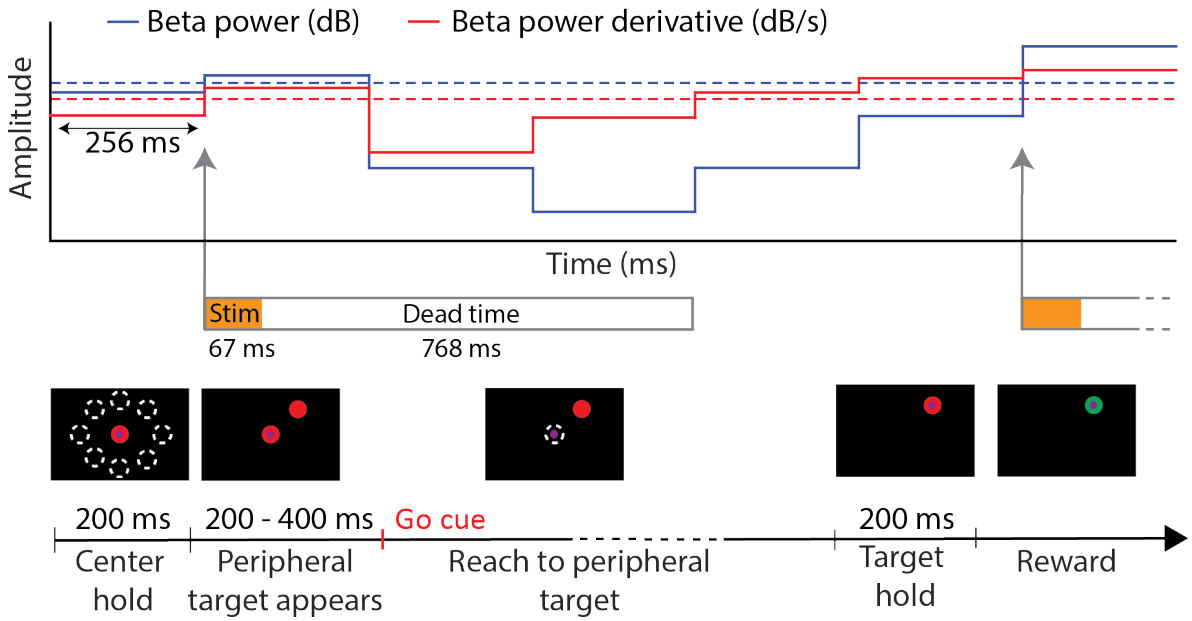


Figure 2.18: Diagram of the delayed-reach task and the closed-loop algorithm implemented during this task. Stimulation is delivered when beta power and its derivative exceed their thresholds.

neural signature, we chose to adhere to durations used in previous work to demonstrate reproducibility of an established result.

Post-hoc analysis showed that RT increased significantly in behavioral trials when stim-

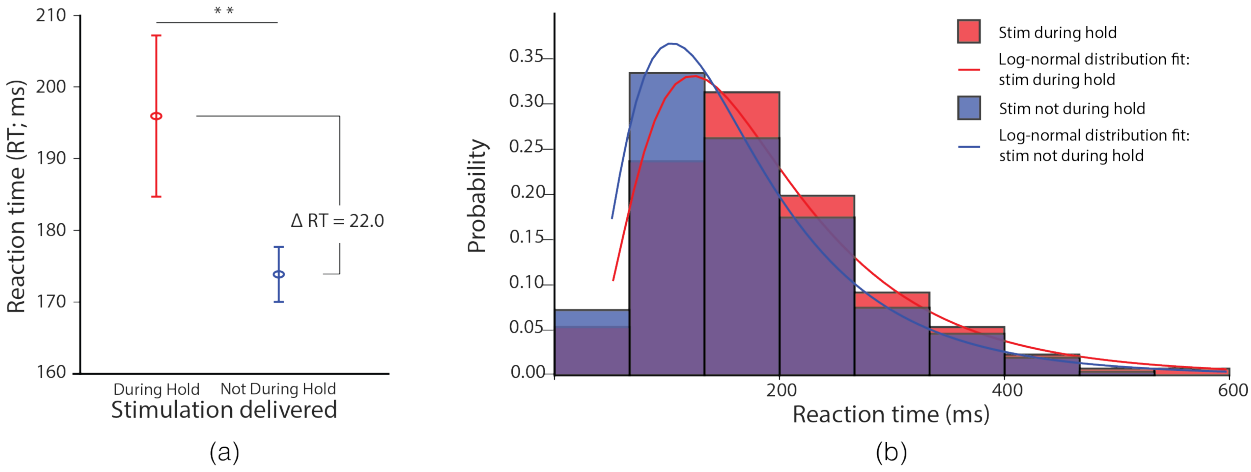


Figure 2.19: (a) The trial-averaged RTs for trials (n = 997) in which stimulation was delivered successfully during the hold period compared to when it was not. Error bars are the SEM. Significance was determined using a two-sided Mann-Whitney U-test ($** p < 0.01$). (b) Normalized RT histograms and log-normal fit to approximate the respective probability density functions.

ulation was delivered during the hold period prior to the Go Cue, relative to trials when it was not (Fig. 2.19(a)). The increase of 22.0 ms in average RT, consistent with previously reported results for microstimulation delivered in PMd, and the change in the distribution of RT (Fig. 2.19(b)) indicate that neural preparatory activity was successfully disrupted using our closed-loop neuromodulation approach.

2.6 Conclusion

OMNI advanced state-of-the-art in closed-loop neuromodulation systems by enhancing the ability to chronically record neural activity and perform closed-loop stimulation at the network level. The large number of channels, low power consumption, and reduced number of implant wires allow OMNI to cover more brain regions, simplify implantation, and operate continuously for longer periods of time. Through its modular and distributed approach to neuromodulation, OMNI is capable of addressing complex disorders that manifest in multiple brain regions at the systems level and require a dynamic approach to the therapy.

In order to test OMNI components in *in vivo* setup, we developed WAND as the only neuromodulation device to our knowledge that incorporates a large number of recording and stimulation channels, have a wireless data rate to support a large number of streaming channels, and also provide closed-loop neuromodulation capabilities. It is also, to our knowledge, the only neural interface system that actively cancels stimulation artifacts through both

hardware and software techniques for completely artifact-free recording during stimulation.

The experiments shown in this chapter are intended to outline and demonstrate the capabilities enabled by OMNI/WAND, paving a path toward use of this technology as a tool in clinical and neuroscientific research. For this work, device form factor, size, channel count, and sensor integration were designed specifically to interface with the microelectrode array implanted in the primate subject. Future research can incorporate other features of our platform, such as the inertial sensor and multi-site stimulation. Its architecture makes it amenable to function as a general-purpose research device, requiring only minor modifications to be re-optimized for new applications. Algorithmic development on the FPGA allows for extracting other neural biomarkers such as band powers or line-lengths used to detect seizure onset in epileptic patient [21], and new closed-loop classification and control algorithms may be conceived for integrating neural activity from a larger number of recording channels. Further research in electrode configuration and improved backend artifact cancellation algorithms may allow recovery of the full underlying neural signal with no lost samples, a limitation OMNI/WAND still faces.

In a clinical context, the device can be hermetically packaged and used to provide on-demand therapy in deep-brain stimulation (DBS) while continuously monitoring the neurological response during treatment. For example, in Parkinson's Disease patients treated with DBS, Swann et al. [22] discovered a spectral peak biomarker linked to dyskinesia, an adverse effect of DBS therapy. Our closed-loop paradigm can be applied to automatically adjust stimulation amplitudes in response to detection of this dyskinetic state, thereby reducing the incidence of this undesirable side effect.

Chapter 3

Gesture Recognition Wearable Device

3.1 Introduction

Wearable sensors are becoming more prevalent in health monitoring and human-machine interface (HMI) applications due to their miniaturization and enhanced comfort, mainly driven by improvements in low-power electronics and fabrication methods [23, 24, 25, 26]. Recent advances in flexible electronics have enabled the fabrication of wearable sensors that can mechanically bend and conform to non-planar and dynamic surfaces of the human body, hence allowing robust physiological measurements of low-bandwidth biosignals [27, 28, 29, 30, 31, 32]. A hybrid method interfacing soft and hard components on separate substrates is often used for larger and more complex systems [30, 32], while new fabrication advances have enabled the integration of rigid components on a single flexible substrate for small-sized sensing patches [33].

While these devices integrate the required sensing and signal conditioning circuitry components, they contain little to no intelligence or processing capabilities, relying instead on an external device to log and process their measurements. Multi-channel, high-bandwidth biosignals such as surface electromyography (sEMG) and electroencephalography (EEG), however, can significantly benefit from in-sensor intelligence. Locally processing these biosignals, as opposed to wirelessly streaming their raw data from tens of channels, reduces the communication link bandwidth utilization, lowers radio power, decreases latency for control applications, improves security, and removes the need for any external computational device. As a result, further progress in the area of flexible wearable sensors entails local, in-sensor computation, while preserving flexibility and comfort.

Typical processing steps for a biosensing system include signal conditioning, feature extraction, and implementation of a machine learning model, with the goal of real-time analysis and response. Because smart biosensors are meant to be worn and used in non-ideal, real-world conditions, they must be able to adapt to changing signal properties that result from many possible different use cases. Modern, light-weight machine learning algorithms perform well when trained on data from the same conditions that are expected during deployment.

However, when a wider set of conditions are not fully captured in the initial training of a classification model, classification accuracy degrades, resulting in sub-optimal performance or poor user experience [34, 35, 36, 37, 38, 39]. Therefore, it is particularly important to be able to train and continuously update in-sensor classification models locally and on-the-fly.

High-density sEMG is a high-data rate biosignal that can leverage in-sensor, adaptive processing, with multiple applications ranging from clinical diagnosis of muscular disorders [40, 41] to gesture recognition in HMIs [42, 43]. In this non-invasive setting, multiple closely spaced electrodes arranged in linear arrays or two-dimensional grids cover a portion of skin to measure electrical muscle activity. Compared to traditional single electrode pairs positioned over individual muscles [44, 45, 46, 47], high-density sEMG increases the amount and reliability of extracted data from muscle activity by capturing spatial variations in addition to temporal changes either at the muscle or at the motor unit level [42, 48, 49, 50]. However, high-quality recording and classification of sEMG signals from a large number of channels is a challenging task from both system design and signal processing perspectives. Existing devices are either in non-wearable form-factors [51, 52] or dependent on external devices for computation [53]. Additionally, commonly used classification models are trained offline with high-performance machines and remain static after deployment. This renders them incapable of adapting to signal variations from sweating, fatigue, varying muscle contraction effort, and electrode displacement due to changing situational contexts such as limb and body position or device doffing and donning [34, 35, 36, 37, 38, 39].

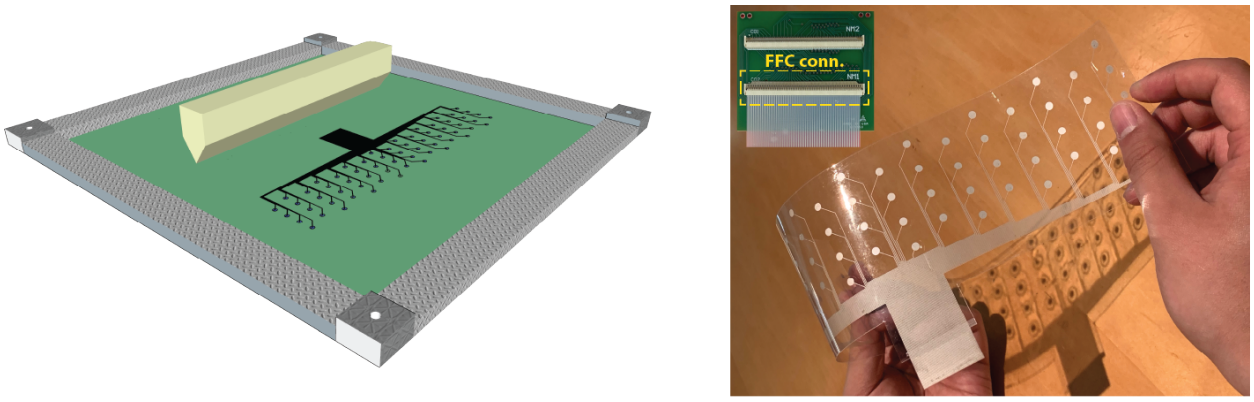
In this chapter, we introduce a wearable high-density sEMG biosensing system (Fig. 3.1) running an adaptive learning algorithm to classify data collected with a low-power custom application-specific integrated circuit (ASIC). We use a hybrid method for interfacing soft conformal sensors and hard silicon-based integrated circuits (ICs), combining a sensing electrode array printed on a flexible substrate with a miniaturized printed circuit board (PCB) including complex sensing, processing, and telemetry components. The large-area 64-channel high-density electrode array is fabricated by screen printing conductive and dielectric inks on a thin, highly flexible polyethylene terephthalate (PET) sheet that conforms well to the complex three-dimensional form of forearm muscles during contractions and relaxations. Placing a custom-designed ASIC [9] for high-channel count recording and digitizing of sEMG signals adjacent to electrode site helps with eliminating bulky cable connections between electrodes and signal conditioning circuitry, providing a low-noise signal and satisfying the small form-factor and power consumption criteria of a wearable device. We demonstrate in-sensor learning and classification using a hyperdimensional (HD) computing algorithm for hand gesture recognition [54]. The algorithm is efficiently implemented on the on-board FPGA, with less than 45K look-up tables (LUTs), capable of few-shot learning and real-time inference of 21 hand gestures. Finally, we show for the first time that HD computing can support continuous learning to learn new gestures and adapt to new contexts with very small hardware overhead. To the best of our knowledge, this is the first realization of a hybrid flexible wearable biosensing device with on-board adaptive learning and real-time inference capabilities.

Varying contraction levels of muscles is a big challenge in electromyography-based gesture

Figure 3.1: Wearable biosensing system for sEMG donned on the forearm of a subject.

recognition. Some use cases require the classifier to be robust against varying force changes, while others demand to distinguish between different effort levels of performing the same gesture. We use HD computing paradigm to build classification models that are both robust to these variations and able to recognize multiple contraction levels. Experimental results on 5 subjects performing 9 gestures with 3 effort levels show up to 39.17% accuracy drop when training and testing across different effort levels, with up to 30.35% recovery after applying our algorithm.

This work is a collaborative effort between multiple groups and researchers. The author's contributions are leading the project, design of the embedded module, firmware development, machine learning algorithm implementation, and performing dataset collection from human subjects.



(a) (b)

Figure 3.2: (a) Illustration of the screen-printing process. A squeegee and a screen are used to print sEMG electrode arrays on flexible substrates. The electrode pattern is defined by features on the screen. (b) The custom-designed, flexible 16 x 4 array of electrodes that conforms to the forearm to provide high-density, large-area sEMG recordings without individual wires. An adapter PCB with a Flat Flexible Connector (FFC) is used to interface the flexible electrode array with the rigid components.

3.2 Hybrid Flexible Biosensor Design and Fabrication

To create the flexible skin-worn interface, we screen-printed a uniform 4 x 16 array of circular electrodes (4.3 mm dia.) on a flexible PET substrate using conductive silver ink sintered by photonic curing (Fig. 3.2). Compared to high-density electrode array fabrication methods reported in literature [41, 53, 54, 55], our screen-printing solution with photonic sintering allows the use of temperature-sensitive substrate materials and requires only sub-second curing time for cheaper and faster large-scale production. A dielectric encapsulation layer was subsequently printed with via holes for exposing the electrode pads while insulating the conductive traces from the skin. Overall dimensions were chosen to wrap around the entire circumference of an above-average sized forearm, capturing activity of the extrinsic flexor and extensor muscles involved in finger movements with low inter-electrode pitch in both the proximal-distal and medial-lateral directions. Four electrodes per column were spaced 14.3 mm apart, and the columns were spaced 17.8 mm apart with vertical offsets of 7.15 mm.

The flexible electrode array was interfaced to a custom, 8-layer PCB (Fig. 3.3) using a Flat Flexible Connector (FFC) on a two-layer adapter board (Fig. 3.2(b)). All 64 channels from the electrode array were sampled and digitized at 1 kS/s using a custom neural interface IC [9] (Sec. 2.2.1) with very low power consumption (700 µW) in a small form factor. sEMG signals were processed within an on-board system-on-a-chip field-programmable gate array (SoC FPGA), where we implemented a full training and classification algorithm to make the system entirely standalone. The device can connect wirelessly to a base station computer

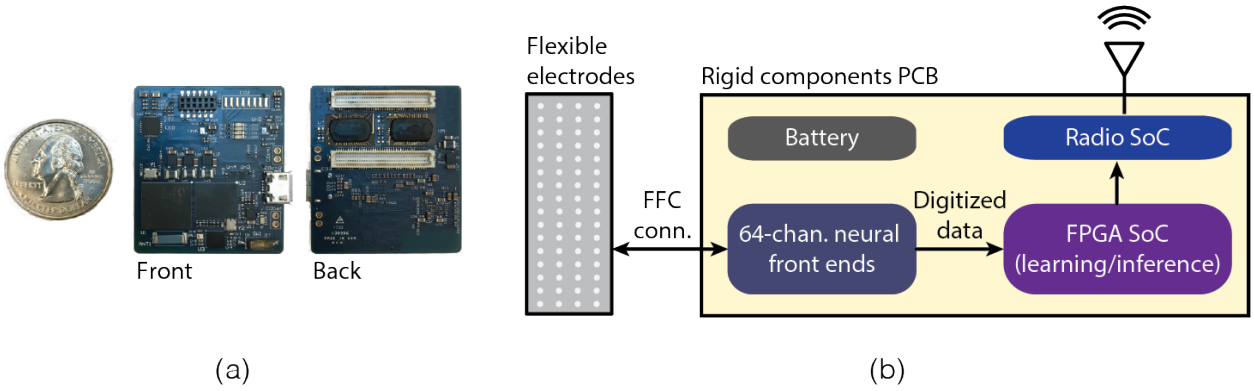


Figure 3.3: (a) The miniaturized, 8-layer printed circuit board (PCB) that accommodates the complex, rigid components responsible for sensing, processing, and telemetry. (b) Block diagram of the main components constituting the wearable system.

using a 2.4 GHz radio for reconfiguration and data logging, if needed. The board was powered using a single 3.7 V, 240 mAh lithium-ion battery. The total weight of the wearable system was 26 g, with the PCBs weighing 23 g and the electrode array weighing 3 g.

3.3 On-Body Characterization of the EMG Acquisition

To characterize the recording quality of our system, we collected and analyzed sEMG data recorded during the performance of a set of 21 common finger gestures (Fig. 3.4). These included both flexion and extension of single as well as multiple degrees of freedom (DOF). Figure 3.5 shows example waveforms recorded from all 64-channels during the flexion and extension of the middle finger DOF. Each 50 ms segment of each waveform is colored based on the mean absolute value (MAV) feature derived from that segment, indicating the local amplitude of the sEMG. Activity in the anterior flexor muscles can be seen during flexion of the DOF, while activity in the posterior extensor muscle can be seen during extension.

Signal quality as acquired using our wearable biosensing system was compared to available recordings of similar gestures using traditional commercial sEMG interfaces (Cometa used in Ninapro DB4) [56, 57, 58] as well as a benchtop high-density sEMG acquisition setup (CapgMyo) [59]. The signal-to-noise-ratio (SNR) was calculated as the total signal power during performance of a gesture divided by the total signal power measured when at rest. Figure 3.6(a) shows the best SNR (from all channels) recorded for each of the single-DOF gestures. While lower than the SNR of the commercial, 12-channel Cometa sEMG system, the SNR of our recordings were similar to that from a wired, high density sEMG acquisition setup. To compare the power spectrum of our recordings, we computed the Welch's power

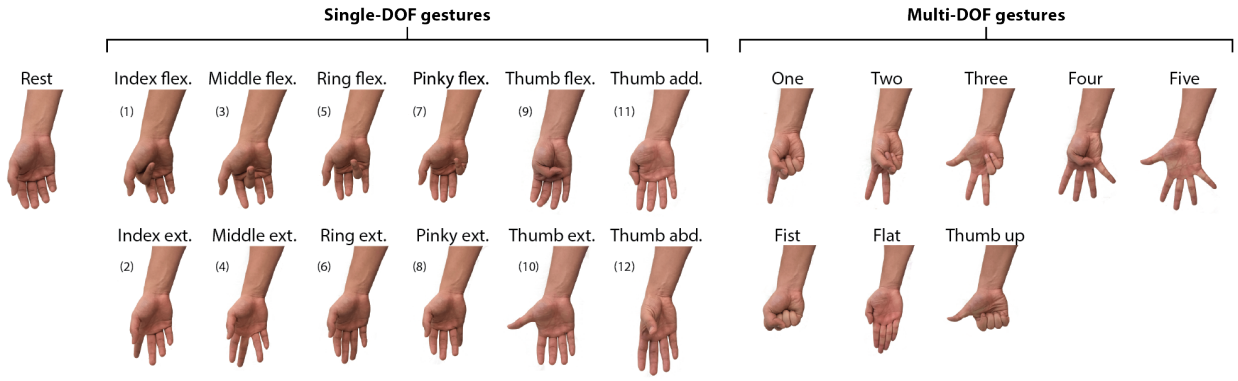


Figure 3.4: Hand gesture classes used in the study. The single degree-of-freedom (DOF) gesture subset includes individual finger flexions (flex.) and extensions (ext.). The multi-DOF gesture subset includes common, isometric hand postures involving multiple fingers.

spectral density estimate for the channel with the highest SNR while performing ring flexion (gesture 5), the gesture that produced the highest SNR across all three devices (Fig. 3.6(b)). The frequency contents of the different recordings are qualitatively similar.

3.4 Gesture Recognition Algorithm with Adaptive Learning

sEMG-based hand gesture recognition has many potential applications in HMI such as gaming [60, 61], musical expression [62], sign language recognition [63], and robotics control [64, 65]. Compared to vision-based methods, sEMG-based methods can be implemented entirely on a single, wearable device, enabling gesture recognition during more aspects of everyday life. We implemented a hyperdimensional (HD) computing algorithm [66] for training and inference of hand gestures. This approach has already shown promising results in classification tasks for biosignals such as sEMG [54], EEG [67], and electrocorticography (ECoG) [68], discriminating up to 5 classes on offline datasets.

HD computing is an emerging computing paradigm that is inherently robust against noise and supports fast, simple learning [66]. It employs random vectors with very high dimensionality (e.g., thousands of bits) to represent information, analogous to the way the human brain utilizes vast circuits of billions of neurons and synapses. A fully distributed holographic representation of data using these hyperdimensional vectors (hypervectors) makes training and classification very robust, even with minimal training data. Additionally, encoding data is fast thanks to carefully designed vector space operations. Finally, the same encoding process is used for both learning and inference, enabling sharing of hardware modules for both online training and classification [69]. This is in contrast to other neuro-inspired ap-

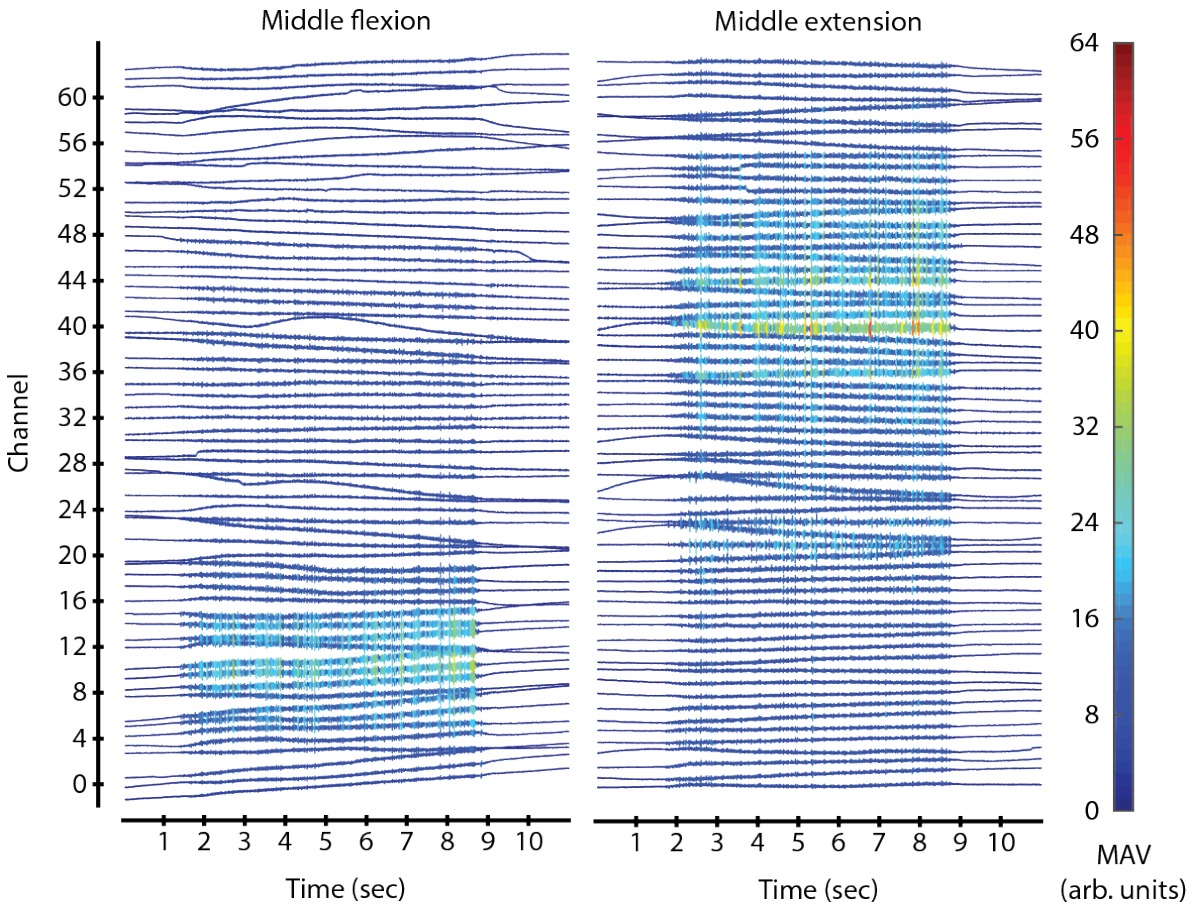


Figure 3.5: Example raw waveforms recorded from all 64-channels during middle finger flexion and extension. Channels are organized starting with channel 0 on the surface nearest the radius, with increasing channels wrapping around the anterior side of the forearm, to the surface nearest the ulna ($\sim$ channel 32), and then around the posterior side back to the radius. Each 11 s gesture trial is divided into 1.5 s rest, 2 s transition period to the gesture, 4 s hold period, 2 s transition period back to rest, and 1.5 s rest based on the instructions given to the subject. The color of the waveform indicates the local amplitude of the sEMG, as measured by mean absolute value (MAV) calculated over 50 ms segments.

proaches in which learning most often employs sophisticated frameworks and is much more computationally demanding than subsequent classification (e.g., gradient descent with back-propagation) [70].

Training and classification using HD computing consists of encoding data as hypervectors and performing comparisons between these hypervectors. A fixed symbol table, or item memory (IM), is built from an initial set of hypervectors taken randomly from a hyperdimensional space, in this case with 1,000 dimensions. Each hypervector consists of an equal

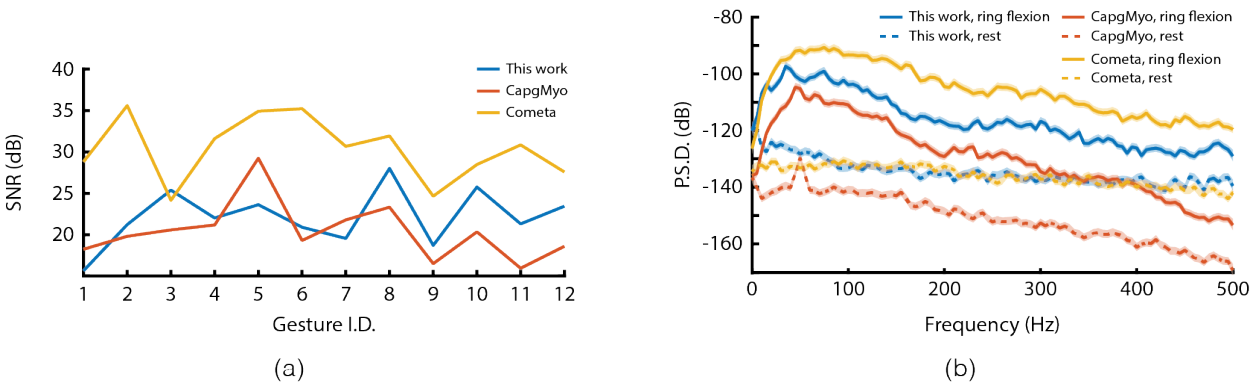


Figure 3.6: (a) Signal-to-noise-ratio (SNR) comparison of the system with available recordings of the same gestures using a commercial sEMG interface (Cometa used in Ninapro DB4) as well as a benchtop high-density sEMG acquisition setup (CapgMyo). The SNR was calculated for each electrode channel by dividing the total signal power during performance of a gesture by the total signal power measured when at rest. The highest SNR across all channels for each gesture is plotted. (b) Welch's power spectral density (PSD) comparison of the three systems for the channel with the highest SNR while at rest and while performing the gesture that produced the highest SNR across all three devices (ring flexion, gesture 5). Shaded areas are the 95% confidence interval.

number of randomly placed +1's and -1's. A fundamental property is that, with an extremely high probability, these hypervectors will all be orthogonal to each other. The hypervectors in the IM can be combined to form new composite hypervectors using well-defined vector space operations, including element-wise multiplication ($*$), element-wise addition ($+$), scalar multiplication ($\times$), and permutation (ρ). These resulting composite hypervectors can be used to robustly represent an event or class of interest [66].

Figure 3.7 describes the process of encoding sEMG data into hypervectors used for training and inference. Data is first preprocessed to extract the MAV features from 50 ms time windows of each of the 64 electrode channels to be used as inputs to the HD algorithm. We chose MAV out of six widely-used sEMG features due to its highest classification accuracy and efficient hardware implementation. The 64 features are then encoded *spatially* to form a single spatial hypervector representing that feature window. Multiple spatial hypervectors from consecutive feature windows are then encoded *temporally* into a single spatiotemporal hypervector, which is the output of the encoding process. These spatiotemporal hypervectors are calculated for all feature sequences to be used as training or inference hypervectors.

The spatial and temporal encoding steps involve various hypervector algebraic operations to encode feature values, the electrode channels they are recorded from, and sequences of features across all electrode channels [54]. Electrode channels are represented by an immutable IM composed of approximately orthogonal hypervectors E_i representing each

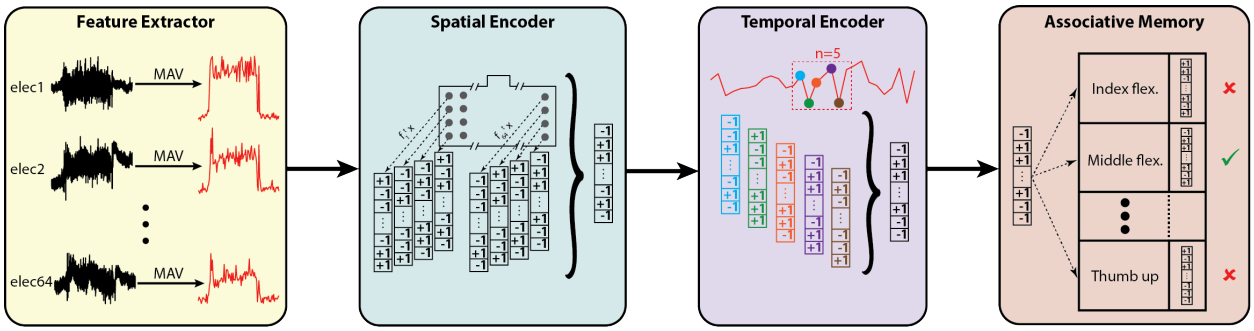


Figure 3.7: High-level flow diagram for encoding 64 electrode channels of sEMG data into bipolar ($\{-1,+1\}^{(1,000)}$) hypervectors. From left to right: MAV features are extracted from a segment of each electrode channel and encoded spatially into a spatial hypervector. Multiple spatial hypervectors are then encoded temporally into a spatiotemporal hypervector for comparison with prototype entries of an associative memory (AM).

electrode i. For each feature window t, electrode hypervectors, E_i, are modulated by the features from their respective electrode channels, f_i^t, and summed together. The spatial hypervector is then formed as $S^t = \sigma(\Sigma(E_i \cdot f_i^t))$, where σ is a bipolar thresholder that turns a positive element to +1 and a negative element to -1 (Fig. 3.8).

A sequence of n spatial hypervectors is then combined to encode relevant temporal information, i.e. the order in which they appear. A hypervector's position in the sequence, $k \in [0, n-1]$ with $k = 0$ being the newest hypervector, is encoded using permutation by rotating the hypervector over k positions. The n permuted hypervectors are bound together through element-wise multiplication to form the spatiotemporal hypervector, G (Fig. 3.9). A new spatiotemporal hypervector can be calculated after each new feature window to encode the previous n windows. We found that sequences of $n = 5$ non-overlapping feature windows resulted in the best classification accuracy while introducing a latency of 250 ms, acceptable for real-time control applications [71, 72].

The same encoding process is used to produce spatiotemporal hypervectors for both training and inference (Fig. 3.10). Training hypervectors G_t calculated using data from a single gesture class are accumulated to form a prototype hypervector representing that class. Prototype hypervectors are thresholded with σ and stored in an associative memory (AM). These accumulation and thresholding operations constitute the entire training process, enabling fast learning and updating. For classification, inference hypervectors G_i are compared to each entry of a fully trained AM. The inferred gesture is selected by finding the closest prototype hypervector in the AM using cosine similarity as the distance metric.

The training process of building an AM allows fast implementation of two types of adaptive learning: updating the model (1) with new classes and (2) for new contexts. Adding new classes to a trained classification model is a challenging task in many state-of-the-art classification algorithms, such as neural networks. Doing so often requires accessing to the

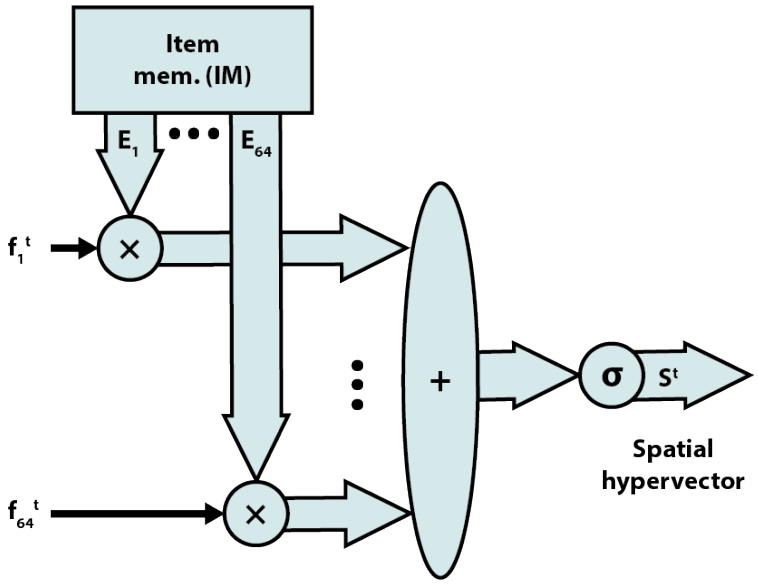


Figure 3.8: A spatial hypervector S^t is formed as a weighted sum of IM hypervectors that represent each electrode channel. The weights for each electrode channel hypervector are the calculated features. After summation, the spatial hypervector is bipolarized to +1's and -1's.

original training data or modifying the model architecture (i.e. identifying new support vectors in a support vector machine). The HD computing model, however, is inherently flexible and simple to update by appending the prototype hypervector for a new class to the AM without changing the existing prototypes. AMs containing different classes can be merged together, even if they were trained on separate occasions (Fig. 3.11).

Adapting to new contexts in HD computing is performed by merging the gesture prototype hypervector from an initial model with a prototype hypervector trained in a new context. An *updated* prototype hypervector is formed, which takes its elements from both the initial and new prototype hypervectors (Fig. 3.12). Different proportions of new and old elements can be taken to weight the contribution of each context. As more contexts are encountered, this proportion can be changed to tune the decay rate in order to avoid catastrophic forgetting of initial contexts.

3.5 Real-Time In-Sensor Implementation of Adaptive Learning and Classification

After validating the HD algorithm using a dataset collected with five subjects (Section 3.8.8), the entire processing and classification flow was implemented using hardware description lan-

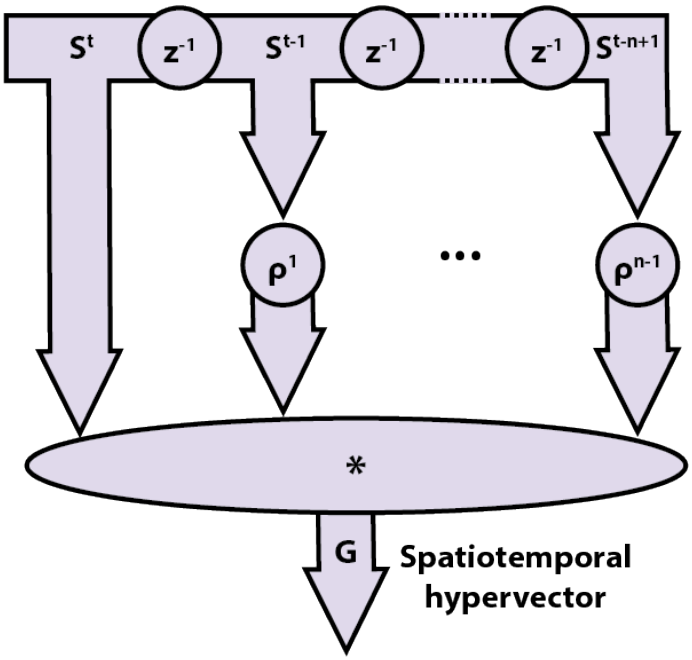


Figure 3.9: Spatial hypervectors from consecutive feature windows are bound together through permute and multiply operations, where a k-element permutation ρ^k is a k-element rotation of the hypervector. Element wise multiplication of the rotated hypervectors forms a spatiotemporal hypervector, G.

guage (HDL) for synthesis in an FPGA. Although described using bipolar ($\{-1,+1\}^{(1,000)}$) hypervectors, the algorithm can be equivalently implemented with binary ($\{0,1\}^{(1,000)}$) hypervectors by mapping from arithmetic to Boolean operations [73].

Raw 15-bit ADC codes were used as the input for feature extraction (Fig. 3.7, left-most box). Both the DC offset (using a 32-sample first-in-first-out (FIFO) buffer per channel) and the MAV feature (using 50-sample buffers) were calculated with each new sample. The implemented arithmetic operations consisted only of addition, finding of two's complement, and arithmetic right shift for division by a power of 2. Features were quantized and saturated to 6-bit integers (0-63) based on analysis of the offline dataset, optimizing for the dynamic range (range divided by step size) given the arithmetic requirements.

Thousand-dimensional IM elements were generated sequentially using a cellular automaton with a hardcoded seed [74] for optimal memory usage. The spatial hypervector was computed by iterating over the 64 channels, either adding or subtracting the channel's feature value from an accumulator value for each hypervector element, depending on the bit value of that element in the associated cellular automaton output. The resulting spatial hypervector was taken as the most significant bits (the sign bits) of the accumulators.

Subsequent processing steps were exactly as described in the previous section on HD

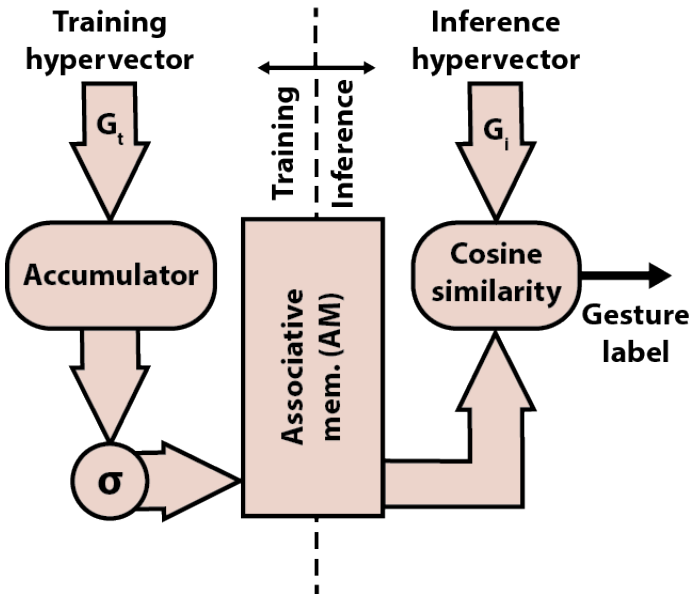


Figure 3.10: Spatiotemporal hypervectors can be used for training (G_t, left) or inference (G_i, right). All training hypervectors from the same gesture class are bundled together through element-wise accumulation, and the resulting prototype hypervector is stored in the associative memory (AM). For inference, a new spatiotemporal hypervector is compared with the entries of the AM using cosine similarity, with the closest entry being output as the inferred gesture class.

classification architecture. A shift register consisting of 21 hypervectors was used as the AM to store trained prototype hypervectors and search for the closest class during inference. A single contextual update with 50% weighting was enabled for each AM entry by merging a predetermined set of 500 bits from a newly trained prototype into the stored one.

The above described implementation utilized 84% of available resources (Section 3.8.10). Simulations verified that the algorithm (after acquiring the last sample of 50 ms feature window) had a latency of 539 cycles, or 26 µs when running at 20.48 MHz (Section 3.8.11). Simulations and source meter measurements showed that the on-board HD algorithm operates with 2.437 µJ per sEMG sample and 4.39 µJ per classification (Section 3.8.12). Model training and updating consumed negligibly more energy than classification.

We verified the online, standalone operation of our device with two subjects by conducting a set of four experiments with all training, inference, and updates performed completely on board and in real time with a single 8-second trial of each gesture. A custom-made GUI was used to instruct the user on what gesture to perform, transmit the correct gesture label and the operation mode (train/infer/update) to the device, and log data. Both raw sEMG data and the classified gesture class were streamed back to the GUI and displayed in real-time, giving the subject visual feedback during inference modes. Classification accuracy

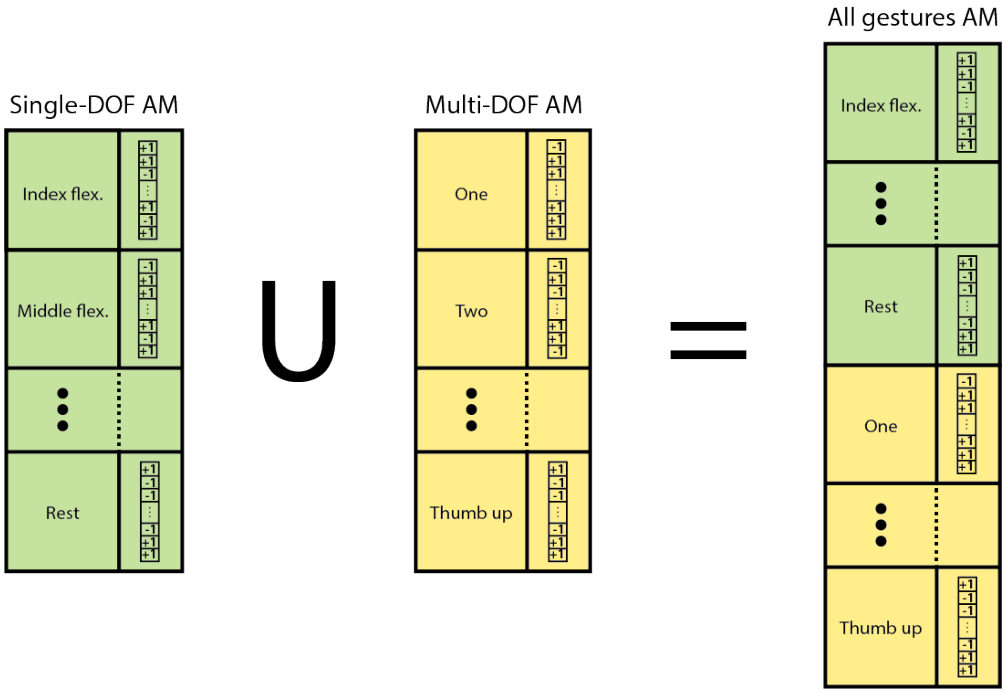


Figure 3.11: Updating the AM with new gesture classes involves appending new prototype hypervectors as new memory entries without modifying existing entries.

was measured by comparing the streamed, classification outputs with the instructed gesture classes during 4-second steady-state portions of each gesture trial (Figs. 3.13 and 3.14).

The first experiment (Fig. 3.13) demonstrated the ability to update the HD classification model with new gestures. We trained and tested an initial model with only the single-DOF gesture subset (Fig. 3.13, yellow bar) and achieved an average classification accuracy of 98.34%. We then updated the model with the multi-DOF gesture subset and tested on all 21 gestures (Fig. 3.13, green bar) with 92.87% accuracy, a 5.47% degradation. Training and testing on the multi-DOF gesture only resulted in a similar accuracy of 95.04% (Fig. 3.13, blue bar).

In the next three experiments, we demonstrated the ability to update the HD classification model with new contexts using only the single-DOF gestures. The context changes we explored were new arm position (going from relaxed position with the arm at the side to elbow rested on an armrest in an arm-wrestling position), new day with the device doffed and re-donned in between, and prolonged wear with the device worn while going about daily activities for two hours. An initial model was trained and tested in the baseline context (Fig. 3.14, solid green bar). The model was then tested in the new context, prior to updating (Fig. 3.14, solid red bar). This resulted in on average a 11.89% accuracy degradation. The model was then updated in the new context and tested again (Fig. 3.14, striped red bar). In

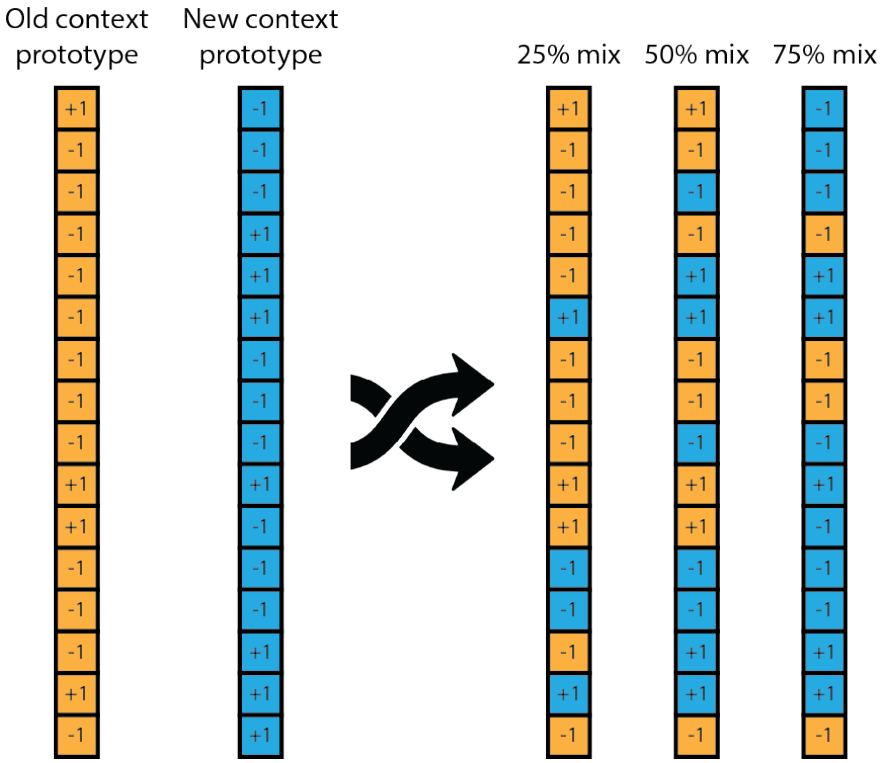


Figure 3.12: Updating the AM with new contexts for an existing class involves forming an updated prototype hypervector for that class. The updated hypervector randomly takes elements from the initial context prototype hypervector and the new context prototype hypervector. The proportion of bits taken from each hypervector determines the relative weight of each context in the updated prototype.

the case of arm position, the updated model was also tested in the initial context Fig. 3.14, striped green bar). In each case, accuracy was recovered after updating the model for the new context with an average improvement of 9.50% compared to the initial model.

3.6 Analysis of Contraction Effort Level

As mentioned in Section 3.1, EMG is highly prone to signal variations caused by factors such as changing limb position [75], electrode shift [76], and force change [77]. While the first two are undesired phenomena that the classifier has to ideally be robust against, the last could occasionally be desired in applications such as proportional control of prosthetic hands.

A subject can exert different levels of effort while performing a gesture, resulting in different EMG signal properties. Scheme and Englehart [78] have shown up to 50% error rate when the classifier was trained and tested at different force levels from 20% to 80%

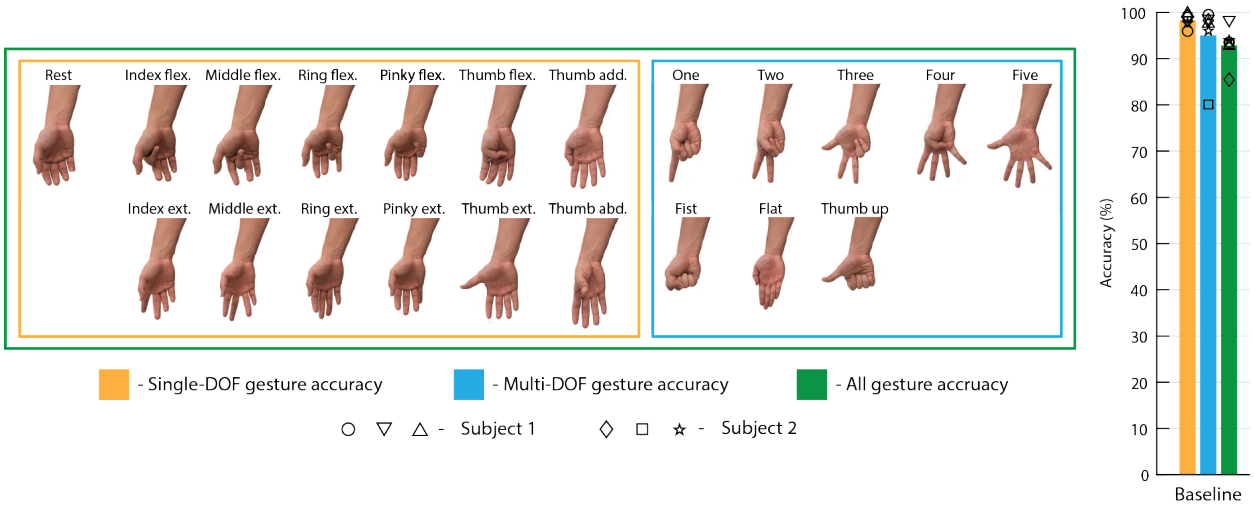


Figure 3.13: On-board, online training, update, and classification results from two subjects performing each experiment three times. Bars represent the mean accuracy across all six trials, with overlaid data points for each individual trial. The ability to update the HD classification model with new gestures is shown. An initial model was trained and tested on single-DOF gestures only (yellow bar). The model was then updated with multi-DOF gestures to cover all 21 gestures (green bar). Results for a separate model trained and tested on multi-DOF gestures only are also illustrated (blue bar).

maximal voluntary contraction (MVC), compared to moderate 7% to 19% error rate when trained and tested at the same level. Previous works have suggested to pick specific force levels that yield minimum accuracy degradation across all force levels as training dataset and to extract features that are more invariant against contraction levels as the input to the classifier [79, 80].

In this section, we propose building a general classification model based on HD computing to deal with varying muscle contraction effort levels. Using the same update mechanisms introduced in Section 3.4, we analyze the muscle contraction level variations in two different ways, depending on the application: If discrimination between different gestures is the only goal, the classifier should output the same gesture class regardless of the subject's effort level. If, on the other hand, different effort levels are relevant to the application (e.g. controlling different levels of force for gripping using a prosthetic hand), different effort levels for the same gesture must be treated as separate output classes. A classifier based on HD computing can be naturally used in both of these scenarios. In the former case, it can include minimum amount of training data from multiple effort levels for training each gesture to build an inclusive model that ignores effort level variations. In the latter, distinguishing between different effort levels of the same gesture translates to simply defining a separate class for each level of contraction.

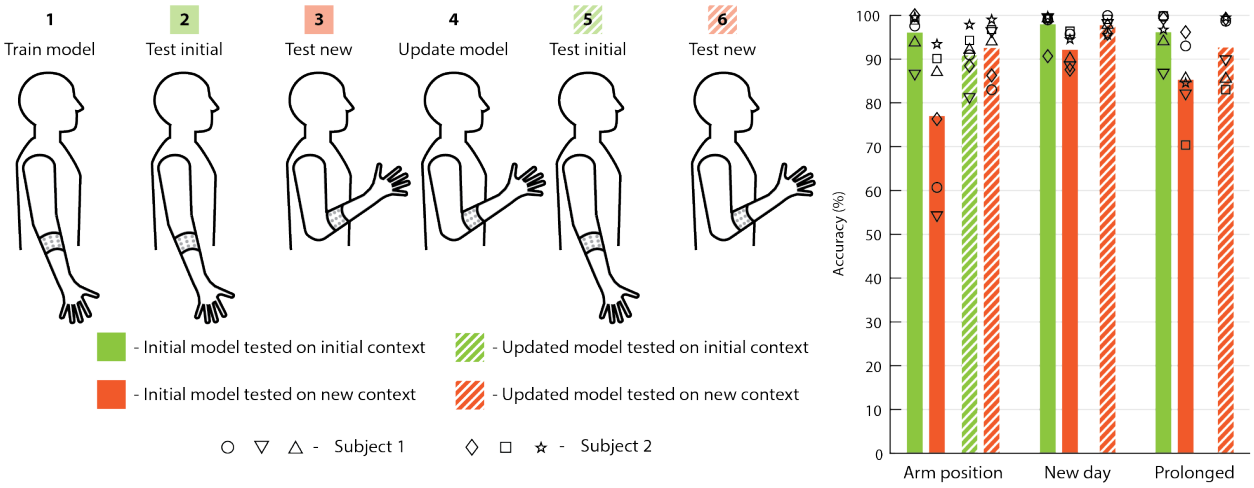


Figure 3.14: The ability to update the HD classification model with new contexts using only the single-DOF gestures. For three different context changes (arm position, new wear session, and prolonged wear), an initial model was trained on the initial context (step 1) and tested on both the initial (step 2) and the new (step 3) contexts (solid bars). The model was then updated using a single trial of each gesture in the new context (step 4), and again tested (steps 5 and 6) on both contexts (striped bars). Updated models were tested only in the new context for the new wear session and prolonged wear experiments since the old context was no longer available.

A dataset of 5 human subjects performing 9 hand gestures (Fig. 3.15) with low, medium, and high contraction effort levels was recorded using our wireless, high-channel count EMG recording system which provided visual feedback of effort level. Classification accuracy results for both gesture-only and gesture+effort cases are presented.

3.6.1 Contraction Level Experiment Setup

We used our custom, wireless 64-channel EMG signal acquisition device to record a dataset of EMG signals from five able-bodied, adult male subjects[1]. The raw recorded signals were wirelessly transmitted to a base station for offline processing. Additionally, we calculated the mean signal energy across all channels as a measure of contraction effort level, and illustrated the value as a bar graph (Fig. 3.16) in the GUI. This served as a visual feedback to the subject in real-time.

For this study, we chose a set of gestures consisting of movements of the thumb, index, and middle fingers to model simple grasping actions (Fig. 3.15): index finger flexion and extension, middle finger flexion and extension, and thumb flexion and extension as single

[1]Dataset and scripts available at `https://github.com/flexemg/flexemg_v2`

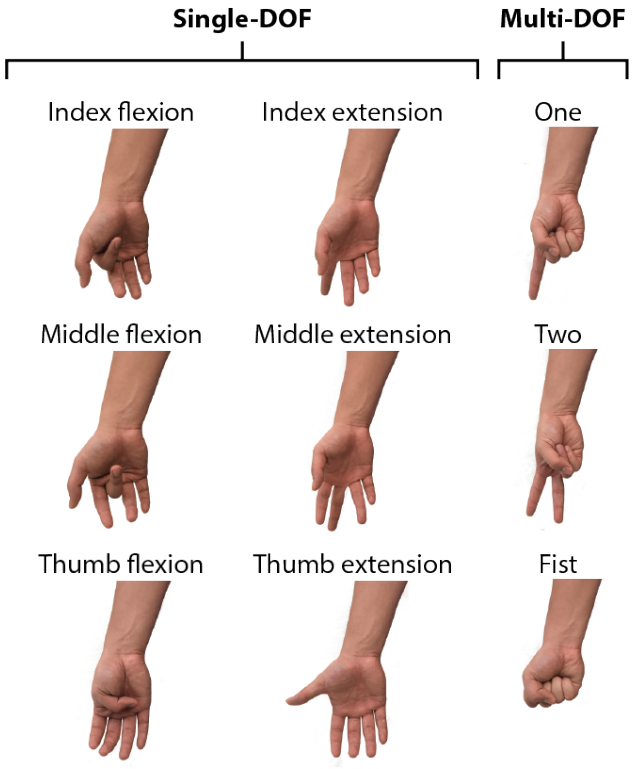


Figure 3.15: Hand gesture classes used in contraction effort level study.

degree-of-freedom (DOF) gesture subset, and one, two, and fist as multi-DOF subset. For each gesture, we started with a calibration phase during which the subject was asked to perform the gesture with the maximum contraction effort, also known as maximum voluntary contraction (MVC). This value was normalized to map to 100% in the GUI feedback bar graph (Fig. 3.16). The subject was asked to target three different effort levels (low effort at 25%, medium effort at 50%, and high effort at 75%) for each gesture, repeating each 5 times (Fig. 3.17).

3.6.2 HD Model for Contraction Effort Level Classification

In contrast to many state-of-the-art classification algorithms that often require a big training dataset, HD computing achieves high classification accuracies with small amounts of training data, i.e. only 1 out of 5 trials of each gesture in our case. Therefore, building inclusive prototype hypervectors that contain data from multiple effort levels is fast. If, on the other hand, data from individual effort levels is used to form the prototype hypervectors, HD model will distinguish the effort level in addition to the gesture itself.

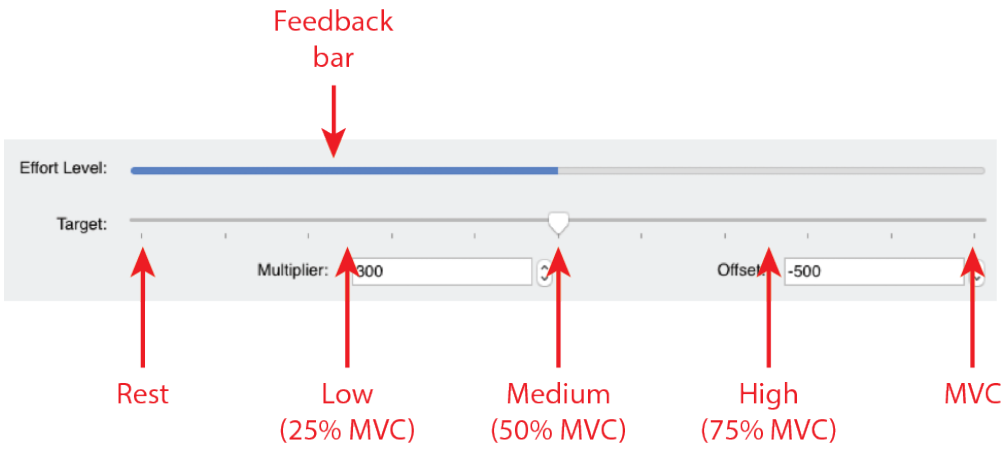


Figure 3.16: Visual feedback of real-time contraction effort level to the user. The bar represents the mean signal energy across all channels as a measure of contraction effort. The users are asked to reach 25%, 50%, and 75% of their maximum voluntary contraction (MVC). During calibration, a multiplier and an offset are determined such that the rest state and the MVC map to 0 and 100, respectively.

3.6.2.1 Gesture-Only Classification

If the only goal is to discriminate between different gestures regardless of the subject's effort level, a single gesture prototype hypervector can be formed to include information from those different effort contexts. This can be done by accumulating spatiotemporal hypervectors from multiple effort levels, and saving its bipolarized hypervector in the AM. If the prototype hypervectors of the two effort levels are already calculated and bipolarized, however, another approach is to merge them into a single prototype hypervector by randomly taking 5000 elements (half of the elements) from each prototype hypervector (Fig. 3.12).

3.6.2.2 Gesture+Effort Classification

If discriminating between different effort levels of gestures is desired, each {gesture,effort} pair must be treated as a separate output class. In this case, prototype hypervectors for each effort level can be added to the model as new entries in AM.

Note that a potential third case could involve adding new prototype hypervectors for each effort level to the AM while preserving the number of gesture classes, allowing multiple prototype entries to represent the same class. While this will improve the classification accuracy comparing to the case where prototype hypervectors were merged, it costs more memory and computation resources as three prototype hypervectors have to be generated and stored for each gesture class.

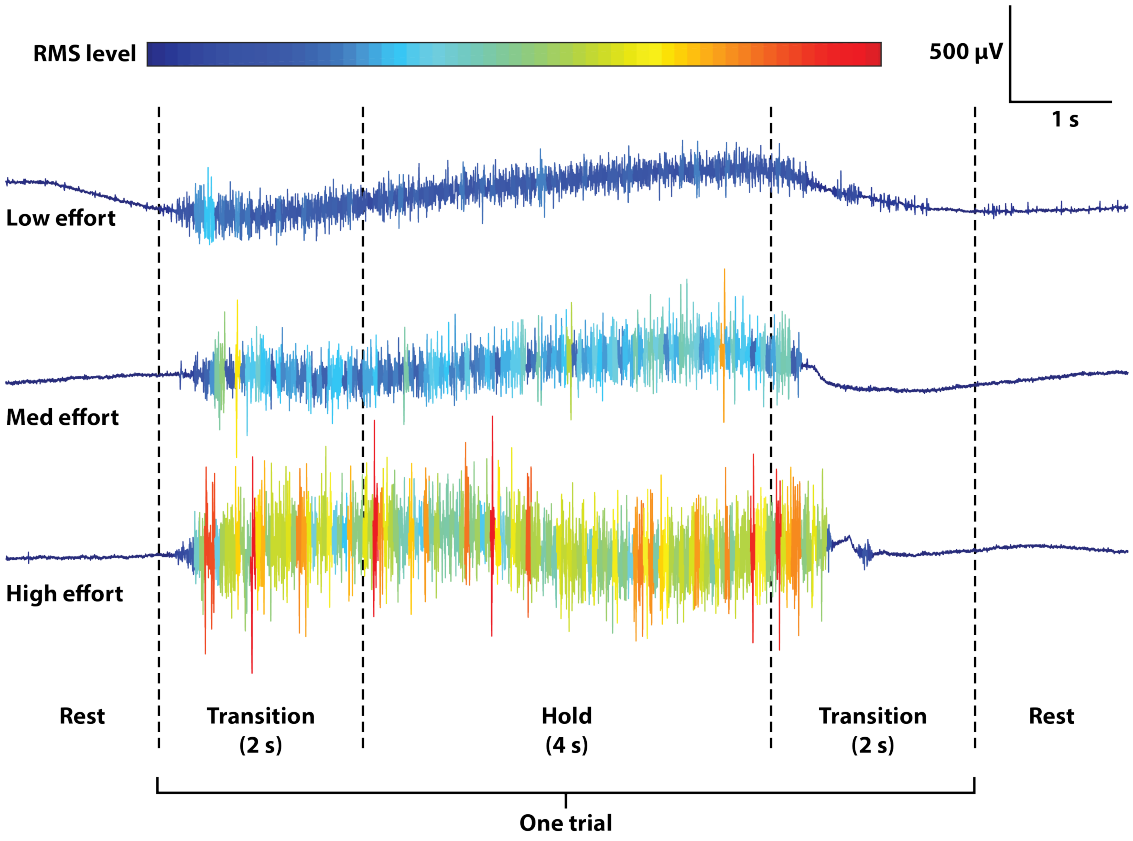


Figure 3.17: Representative EMG signals from one electrode channel recorded during a low, medium, and high effort level trial of the same gesture. The vertical dotted lines divide a single 8 s gesture trial into 2 s transition periods and a 4 s hold period based on the instructions given to the subject. The color of the waveform indicates the effort level, as measured by windowed signal power (RMS calculated over 200 ms windows with 150 ms overlap).

3.6.3 Contraction Level Results

We first treated different effort levels as different contexts of the same gesture class. An initial model was trained with gestures from one effort level context. It was then cross validated (training with one trial, inference with remaining four trials) within the same effort level and also used to classify gestures from the other level within the pair, without merging the models (Fig. 3.18(a-c), first and second pairs). When training and testing within the same effort level context, classification accuracy remained better than 93.11%. However, across different effort levels, classification accuracy dropped by between 16.57% and 39.17%, with the worst performance when the difference between effort levels was highest, i.e. low and high effort. After merging the prototype hypervectors to include both effort contexts, the classification

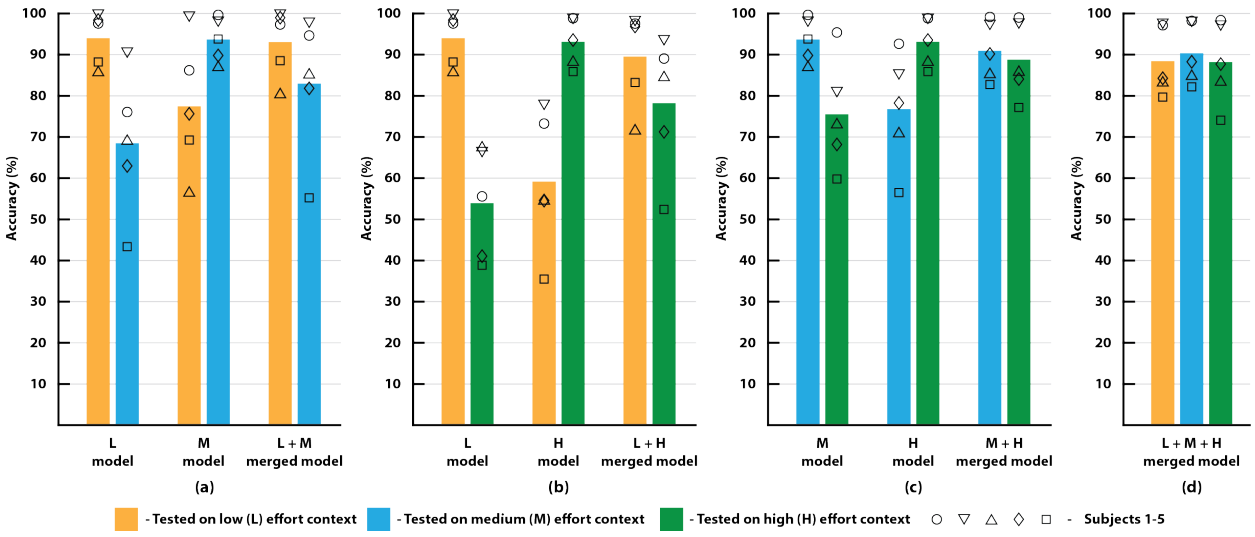


Figure 3.18: Classification accuracy measured before and after merging the models when treating effort levels as different contexts of the same gesture class. Accuracies across effort contexts before and after merging were calculated for each pair of effort levels: low (L) and medium (M) in (a), low and high (H) in (b), and medium and high in (c). Accuracy for each effort level was also calculated using a model trained with all three effort level contexts (d).

accuracy was recovered to above 78.21% (Fig. 3.18(a-c), third pairs). The poorest recovered accuracies resulted from training an initial model on medium or high effort level gestures, and then merging with low effort gestures. While prototype hypervectors for medium and high effort level gestures were more similar to each other, prototype hypervectors for low effort level gestures were more distant due to a smaller variance in the calculated feature values.

For a model trained with all three contexts by accumulating their spatiotemporal hypervectors before bipolarization, accuracy was at least 88.19% for all three effort contexts (Fig. 3.18(d)). In this case, the all-inclusive final hypervector is weighted to be more similar to medium and high effort level prototype hypervectors enabling higher accuracies in those contexts.

When treating different effort levels of a single gesture as different classes, we trained a new AM entry for each gesture and effort level, increasing the total number of classes. We calculated classification accuracy in two different ways: For the first method (Fig. 3.19, red bars), an accurate classification required matching both the gesture type and its effort level to the label. For the second (Fig. 3.19, purple bars), an accurate classification required only matching the gesture type. Notably, if we disregard the effort level classification output from this model, we achieve a better gesture-only classification accuracy than in the case where we treated different effort levels as different contexts.

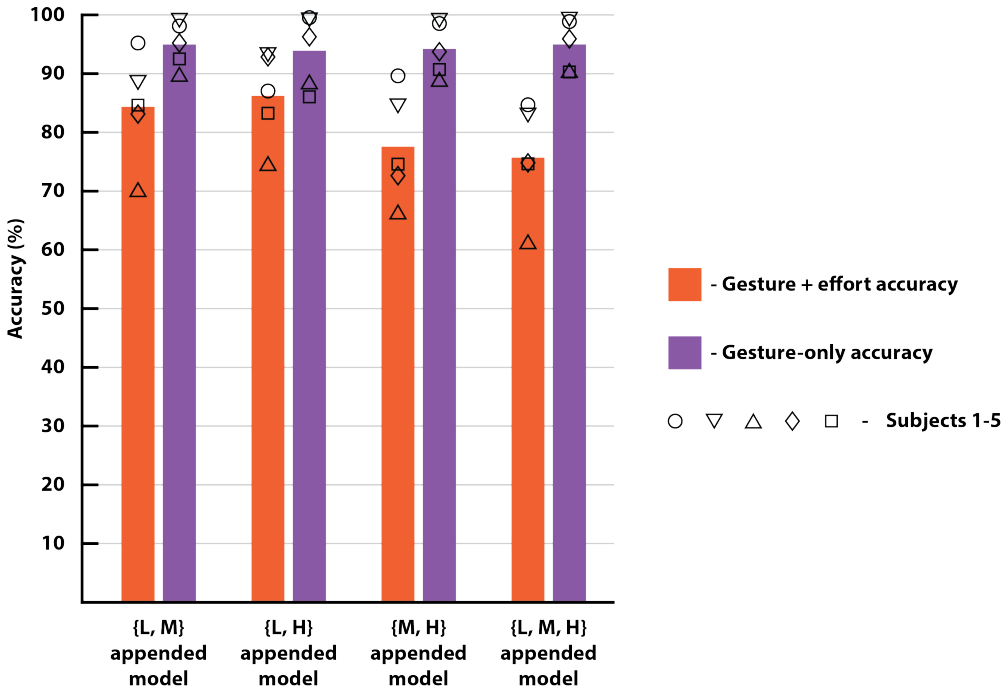


Figure 3.19: Classification accuracy measured when treating different effort levels of the same gesture as different classes. Accuracy was calculated as the success rate of matching both gesture type and effort level (red) as well as gesture type only (purple).

3.7 Conclusion

This chapter demonstrated a self-contained biosensing system that interfaces with the human body to process and classify biosignals. The entire signal chain including a printed flexible electrode interface, low-power data acquisition, as well as training and classification with a machine learning model are all incorporated into a compact device requiring no external computation. This integration preserves the wearability and comfort of such a device while enabling fast initial training and on-the-fly adaptation, which are crucial for wearable HMI applications in which biosignals vary from user to user and are not stationary.

As an example, we conducted a series of sEMG-based gesture recognition experiments, in which training, inference, and model updates were all performed on-board and in real time to adapt to changing situational contexts such as arm position and various muscle contraction levels. While our electrode array geometry was designed for forearm sEMG acquisition, the low-cost and low-complexity screen-printing process can be used to create patterns for interfacing with other parts of the human body. This could enable a wider range of applications requiring low-latency, adaptive processing of biosignals such as ECG or EEG. In general, we believe our wearable system is a big step towards smart, standalone human-machine interfaces.

3.8 Methods

3.8.1 Screen-Printed Electrode Array Fabrication

The high-density electrode array (Fig. 3.2) was printed onto a flexible PET substrate using screen-printable silver ink (NovaCentrix FG57b) and screen-printable dielectric ink (NovaCentrix DE SP1). Each layer was photonically cured with a Novacentrix PulseForge technology and had a thickness of 15 µm. The array consists of 64 circular electrodes with 4.3 mm diameter, uniformly distributed in a 16 x 4 grid covering up to a 29.3 cm x 8.2 cm area on the forearm. A dielectric encapsulation layer covers the conductive traces while exposing the electrode pads through circular openings.

An adapter PCB (Fig. 3.2(b)) with a 64-position Flat Flexible Connector (FFC, XF2W-6415-1AE, Omron Electronics) on one side and two DF12 connectors (Hirose Electric) on the other side was fabricated to interface the flexible array traces with the rigid ICs for sEMG signal acquisition and processing.

3.8.2 Rigid Components PCB Fabrication

A custom wireless neural recording and processing module (Fig. 3.3) capable of recording and processing 64 channels of electrical biosignal data as well as wirelessly streaming raw signals and classified gestures back to a base station is attached to the flexible electrode array. It records and digitizes the sEMG signals from the electrodes using a custom neuromodulation IC [9] (NMIC, Cortera Neurotechnologies, Inc.) with DC-coupled front-ends and a 1 kHz sampling rate. Data aggregation and preprocessing as well as HD model training, inference, and update is performed on an SoC FPGA (SmartFusion2 M2S060T, Microsemi). A 2.4 GHz radio (nRF51822, Nordic Semiconductor) streamed out the raw sEMG signals and the classified labels and distances. These components enable a small form factor and low power operation ($\sim$ 6 hours of streaming using a standard 240 mAh 3.7 V Li-ion battery), while eliminating the need for bulky individual cables connecting each electrode to the neural front-end, making the device comfortably wearable for extended periods with enhanced signal quality. Moreover, digitizing the signal next to its source increases the SNR.

The fabricated 8-layer PCB also contains a 6-axis accelerometer and gyroscope (MPU-6050, InvenSense), a 512 Mb low-power SDRAM (MT46H32M16LFBF-5, Micron Technology Inc.), and an extra 64-channel neuromodulation IC [9] (NMIC, Cortera Neurotechnologies, Inc.) for the purpose of providing inertial sensor data, more on-board memory, and extended number of sEMG channels to 128, respectively, although they are unused in this work.

3.8.3 Graphical User Interface

A custom graphical user interface (GUI) developed in Python language was run on the base station for receiving and logging the streamed data, configuring the sEMG signal acquisition system, and providing subjects with gesture information and timing. The GUI enabled

experimenters to select gesture subsets, update metadata for the saved files, and update gesture timing prior to each recording session. Moreover, it provided visual feedback to subjects about the current classified gesture.

3.8.4 Array Application

Before wrapping the array around the subject's forearm, a small drop of conductive hydrogel (SignaGel Electrode Gel, Parker Laboratories, Inc.) was applied to each electrode to improve the skin-electrode interface impedance. The ends of the array were tightly taped together, holding it in place on the subject's forearm. A single commercial Ag/AgCl electrode (H124SG, Covidien Kendall) was attached to the subject's elbow on the same arm as a reference potential for the voltage measurements (Fig. 3.1).

3.8.5 Gesture Recognition Experiments

For the online experiments, subjects were asked to perform a total of 21 different gestures consisting of a rest/relax position and two subsets of finger movements and positions: (1) single-degree-of-freedom (DOF) flexions and extensions of individual fingers, and (2) multi-DOF hand postures involving multiple fingers (Fig. 3.4). Each subject performed a total of 4 experiments, with each experiment repeated 3 times.

Experiment 1 was designed to test a baseline accuracy for the single- and multi-DOF gesture subsets separately, and an updated model containing all 21 gestures by adding multi-DOF gestures to the existing single-DOF trained model (Fig. 3.11). Experiment 2 then introduced an arm position context variation to the single-DOF gestures. The subject performed each gesture with their elbow rested on an armrest in an arm-wrestling position. For experiment 3, the model was first trained and tested on day 1, and then updated on day 2 after doffing and re-donning the device in approximately the same location on the arm. This introduced a new wear session context variation. In experiment 4, subjects were given a two-hour break between the initial training and subsequent update, during which they wore the sEMG acquisition device while going about their daily activities. This introduced a prolonged wear contextual variation.

Each trial lasted 8 seconds (Fig. 3.5), with 3 seconds of rest before the next trial. The subject was told to begin the gesture within a 2-second transition window which would contain the transient, non-stationary part of the sEMG signal for that gesture. After the 2-second transition window, the subject was asked to hold the gesture for 4 seconds, constituting the steady-state part of the sEMG signal. Finally, the subject was directed to return to the rest position within another 2-second transition window. These directions ensured that the steady-state portion of the gesture could easily be labeled as part of the middle 4 second segment. All experiments were performed in strict compliance with the guidelines of IRB and were approved by the Committee for Protection of Human Subjects at University of California, Berkeley (Protocol title: Flex EMG Study. Protocol number: 2017-10-10425).

3.8.6 Data Segmentation

For this study, only data from the 4-second steady-state hold period were used for classification. During training and testing, we generated a spatiotemporal vector for every 250 ms segment of data, sliding by 50 ms MAV feature windows (200 ms overlap). Thus, within a single 4-second gesture trial, 76 different vectors were encoded either for accumulating in the AM or for making 76 different inferences. Classification accuracy was calculated as the percentage of inference results that matched the labeled gesture, without any post processing or voting. Commands were sent to the device to put it in training, inference, or update mode synchronized to the experiment and instructions.

3.8.7 Algorithm Implementation

The online algorithm was implemented in Verilog hardware description language (HDL) and synthesized using Libero SoC Design Software (Microsemi Corp.). The FPGA resource utilization and algorithm latency are provided in Sections 3.8.10 and 3.8.11, respectively. The algorithm energy per classification is calculated based on the instantaneous power measurements shown in Section 3.8.12 (measured by Agilent B2902A precision source/measure unit in 20 µs intervals) and cycle counts (Section 3.8.11).

3.8.8 Offline Dataset Collection and Algorithm Validation

In order to validate the HD algorithm and tune its parameters for this application, we used our system to record an offline dataset of sEMG signals from five healthy, able-bodied, adult, male subjects who were asked to perform gestures in multiple contexts that simulated everyday use of an sEMG acquisition system. During each recording session, the subject performed 5 trials of each of the gestures. Data were automatically labeled with the gesture class and saved as .mat files. The HD computing algorithm for offline learning and classification was implemented in MATLAB (MathWorks, Inc.) with 10,000-dimensional hypervectors and floating point feature values. Scripts were run on a single node (24 cores, 64 GB RAM) of a research computing cluster. We leveraged MATLAB's Parallel Computing Toolbox to expedite running multiple experiments in parallel.

For each experiment, the relevant dataset was divided into 5 folds, with each stratified fold consisting of a single trial of each gesture. Because the HD computing model has an inherent robustness to mismatched elements [66], the AM can provide an acceptable level of accuracy for inference after bundling only a small number of spatiotemporal training hypervectors. Therefore, we implemented a leave-four-out or reverse cross validation (RCV) using a single fold for training and the remaining 4 folds for inference. This way we were able to validate few-shot learning with our model using a small, single-trial training dataset. When training on an initial context and testing on a different context without updates, cross-validation was not necessary. However, to maintain the same amount of training data used, we used a similar method of dividing the training data into 5 folds, and training with 1

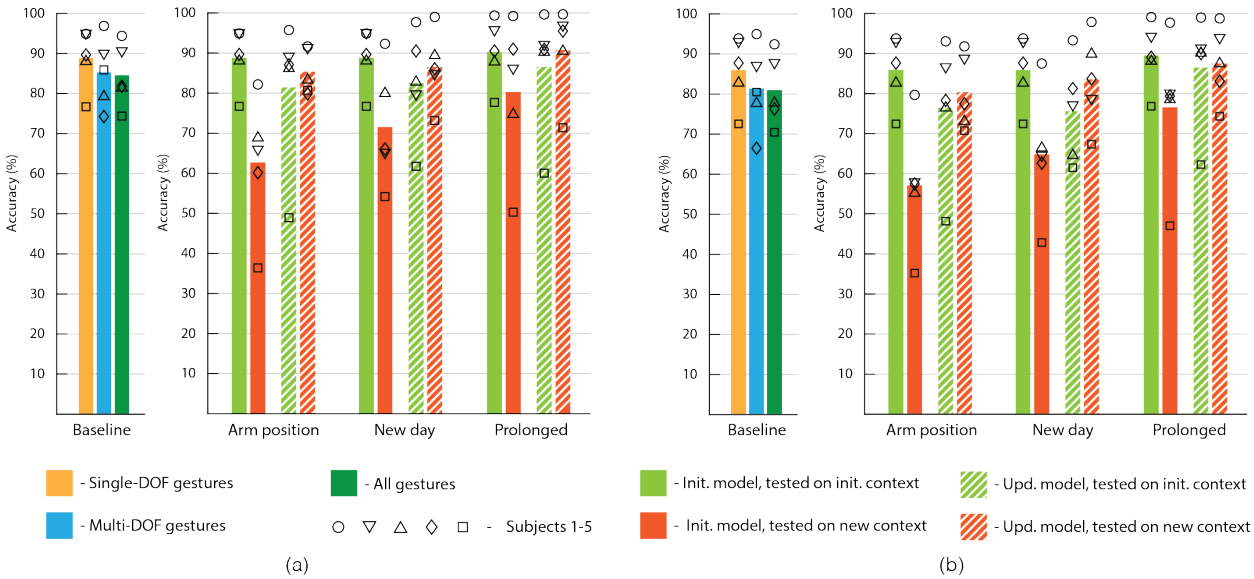


Figure 3.20: Classification accuracies using offline model and simulated hardware-implementation. Mean accuracies were calculated using reverse cross validation (RCV) on the offline dataset, with different gesture subsets within a single situational context (left side of each subfigure), as well as with single-DOF gestures only across different situational contexts (right side of each subfigure). Bars represent the mean accuracy across five subjects, with overlaid data points for each individual subject. (a) Results with 10,000-bit hypervector dimension and floating-point features. (b) Results with 1,000-bit hypervector dimension and 6-bit integer features as hardware-optimized simplifications.

of them. 5 different initial models could be trained, and cross-context classification accuracy was measured as the mean accuracy of all the models tested on the entire testing dataset. When updating a model for new contexts, we partitioned both the initial and new context datasets into 5 folds. We chose 1 fold from the initial dataset to train the initial model, 1 fold from the new dataset to update the model, and then tested the updated model on the remaining 4 folds from each dataset. Classification accuracies on the old and new context dataset were measured as the mean accuracy of the 25 possible different data partitions.

Figure 3.20(a) shows RCV classification accuracy in the baseline situational context for the single- and multi-DOF gesture subsets separately and for all 21 gestures together. The updated model for classifying all 21 gestures was created by joining the AMs previously trained separately for the single- and multi-DOF gesture subsets (Fig. 3.11). This involved no additional computation, whereas other state-of-the-art methods (Fig. 3.21) require training a new model from scratch.

To measure classification accuracy of the HD model across different situational contexts, we performed three experiments in which a model was trained using data from an initial

context and then used to classify gestures in a new dataset from a different context. Figure 3.20(a) shows the classification accuracy for the three tested contextual variations: arm position, new wear session, and prolonged wear. Although training and inference were performed on separate datasets, we still used a single trial of each gesture to train the model as in RCV. For each of the contextual variations, we saw accuracy degradations of 26.17% (arm position), 17.34% (new day/wear session), and 9.95% (prolonged wear), suggesting that updates were necessary.

Prototype hypervectors from the new contexts were also trained using only a single trial of each gesture before being merged with the initial model. We wanted both the initial and new contexts to have equal contributions to the updated model, so the merged hypervector randomly took 50% of its elements from the initial prototype hypervector and the other 50% from the new prototype hypervector (Fig. 3.12). After updating initial models for the new arm position, wear session, and prolonged wear contexts, the classification accuracy on gestures in the new context was improved significantly (22.66%, 14.96%, and 10.51%, respectively) at the expense of small degradation (7.46%, 6.37%, and 3.65%, respectively) in classifying the initial context gestures. This accuracy degradation is due to the bitwise merge operation being only an approximation of the accumulation required to fully retrain a model on data from both contexts. This approximation introduces more error with increasingly distant context prototypes.

Figure 3.20(b) shows the performance of the simplified model (implemented in MATLAB) on the offline dataset, demonstrating only a few percent of accuracy degradation when scaling down to 1,000-dimensional hypervectors and using integer features.

3.8.9 Comparison with LOOCV and Widely-Used ML Methods

Because the HD computing model has an inherent robustness to mismatched elements [66], the AM can provide an acceptable level of accuracy for inference after bundling only a small number of spatiotemporal training hypervectors. In order to validate this few-shot learning feature, we also implemented the standard leave-one-out cross validation (LOOCV) using 4 folds for training and 1 fold for inference in addition to the RCV presented in Fig. 3.20.

Figure 3.21(a) shows LOOCV classification accuracies using our HD computing model as well as three other machine learning (ML) models widely used for gesture recognition: support vector machine (SVM), linear discriminant analysis (LDA), and random forest (RF). The HD classifier achieves 92.47% accuracy for single-DOF, 91.71% for multi-DOF, and 89.85% for all gestures, which are comparable or better than the three other methods. The updated model for classifying all 21 gestures was created by joining the AMs previously trained separately for the single- and multi-DOF gesture subsets (Fig. 3.11). This involved no additional computation, whereas other methods required training a new model from scratch. Furthermore, when measuring accuracy with RCV (Fig. 3.21(b)), the HD classifier experienced a small amount of degradation compared to LOOCV accuracy, demonstrating its amenability to few-shot learning. Accuracy for single-DOF gestures was 88.87%, with the

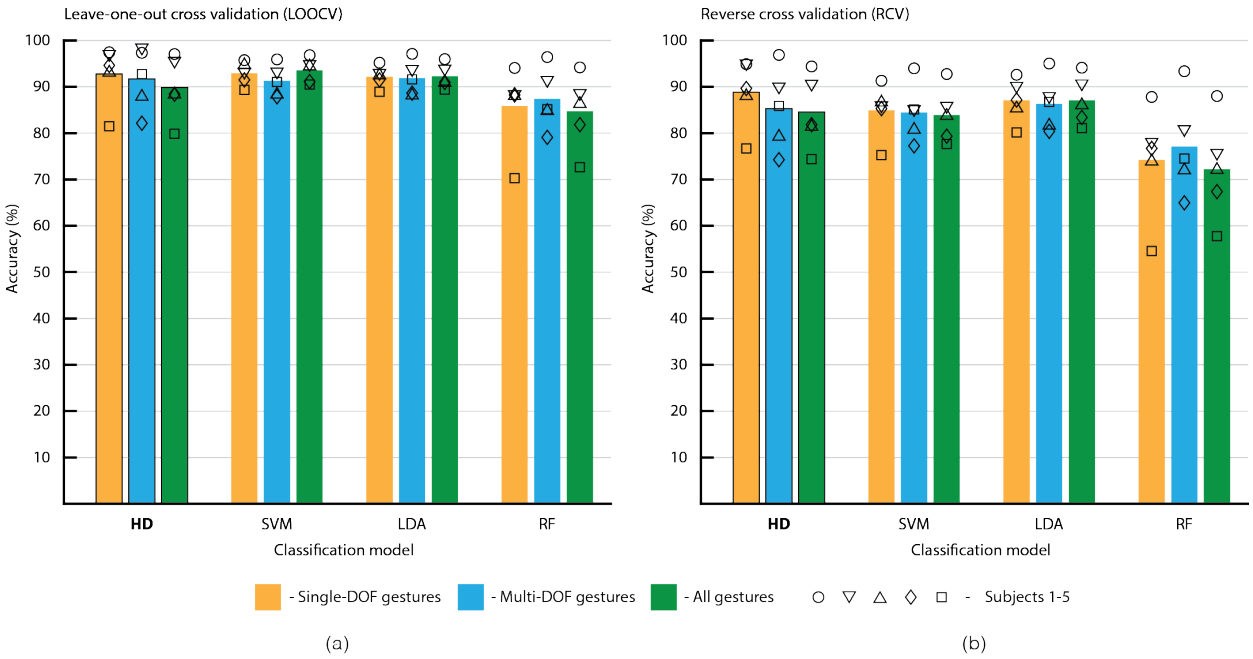


Figure 3.21: Classification accuracies on different gesture subsets with training and testing splits determined using (a) leave-one-out cross validation (LOOCV), and (b) reverse cross validation (RCV). Accuracies were compared for hyperdimensional computing (HD, bold), and three other traditional learning models (SVM, LDA and RF). Bars represent the mean accuracy across five subjects, with overlaid data points for each individual subject.

3.87% degradation resulting from the smaller training set. Thus, for the offline results in this paper, we report accuracies obtained using RCV which were slightly lower than LOOCV.

SVM, LDA, and RF algorithms were implemented in MATLAB. For SVM, we used a linear kernel and cost parameter C = 500. For LDA, we implemented the standard MATLAB function included in the Statistics and Machine Learning toolbox, with maximum regularization (gamma = 1) for the covariance matrix estimation. For the RF algorithm, we implemented the standard MATLAB function included in the same toolbox with 40 decision trees and 2 predictors selected at random for each decision split.

3.8.10 FPGA Resource Utilization

FPGA resource utilization report for the HD algorithm blocks based on the number of look-up tables (LUTs, left) and D Flip-Flops (DFFs, right) was generated using Synopsys Synplify Pro for Microsemi and illustrated in Fig. 3.22.

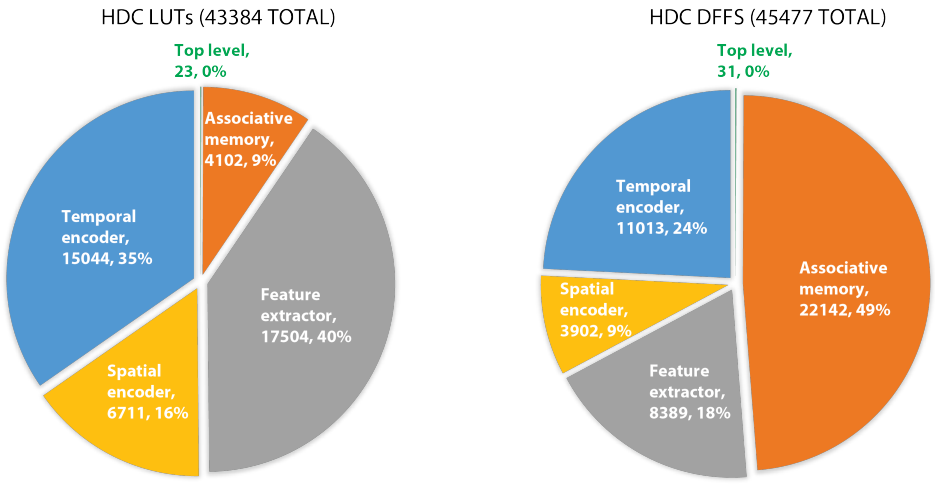


Figure 3.22: FPGA resource utilization for each submodule of HD algorithm.

Table 3.1: Inference latency.

	Clock cycles	Time (w/ 20.48 MHz clock) [µs]
Feature extractor	2	0.10
Spatial encoder	513	25.05
Temporal encoder	1	0.05
Associative memory	23	1.12
Total (predict)	**539**	**26.32**

3.8.11 Algorithm Latency

The HD algorithm clock cycles was counted through Verilog simulations in Modelsim (Mentor Graphics) and latency was calculated based on that (Tables 3.1 and 3.2).

3.8.12 Power Breakdown

The power consumption of the system was measured by an Agilent B2902A precision source/measure unit in 20 µs intervals (Fig. 3.23). The shorter peaks happening every 1 ms correspond to the accumulation of each new sample for feature extraction. The taller peaks happening every 50 ms include the additional power for the HD encoder and accessing the AM. The radio goes to sleep mode after every 3 samples and wakes up at the arrival of the next sample.

Table 3.2: Latency for training/update followed by first prediction.

	Clock cycles	Time (w/ 20.48 MHz clock) [µs]
Feature extractor	2	0.10
Spatial encoder	513	25.05
Temporal encoder	1	0.05
Total (accumulate)	516	25.20
Feature extractor	2	0.10
Spatial encoder	513	25.05
Temporal encoder	1	0.05
Associative memory (write) + Temporal encoder	1	0.05
Associative memory (predict)	23	1.12
Total (train/update)	**540**	**26.37**

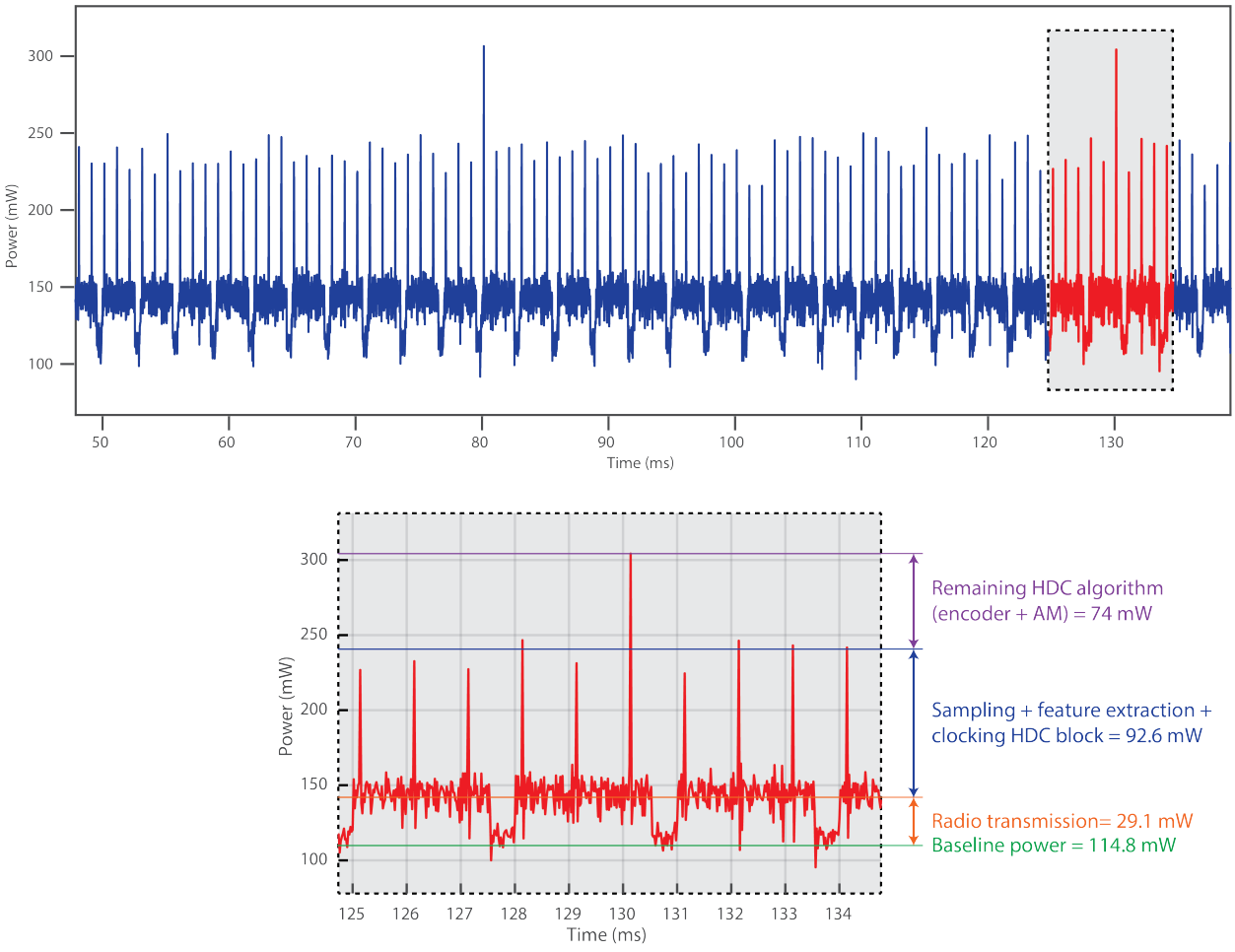


Figure 3.23: Power consumption of the system during classification.

Chapter 4

Human Intranet Design Space Exploration

4.1 Introduction

Wearable and implantable devices, such as the ones introduced in Chapters 2 and 3, are destined to become key components of the Internet of Things (IoT), with a myriad of applications, ranging from health-care and sport to entertainment and communication. Networks of these devices, often called Wireless Body Area Networks (WBANs), have been extensively investigated in the literature [81, 82]. However, most of the previous work on WBANs focused on a limited number of nodes, often targeting a specific application. With the explosive growth of the wearables market, having tens of these interconnected devices per person, performing different functions, is not unreasonable. Further, a general approach to the design of a WBAN architecture with multiple, heterogeneous nodes is needed to shorten design and deployment time. We call this architecture *Human Intranet (HI)* [1]. A Human Intranet should seamlessly integrate an ever-increasing number of sensor, actuation, computation, storage, communication, and energy nodes located on, in, or around the human body, and acting in symbiosis with the functions provided by the body itself.

As the number and variety of nodes, network protocols, and technologies available at each layer continuously increase, the design of such an open, adaptive, and scalable network platform becomes a daunting task. In a Human Intranet, design requirements often span the entire envelope of achievable performance. When a safety-critical node such as a wearable insulin delivery device is part of the network, reliability becomes of utmost importance. Conversely, for an everyday physical activity monitoring application, achieving the longest possible battery lifetime is preferred, while a few packet drops can occasionally be tolerated. Since key design concerns such as reliability and lifetime may be conflicting and depend in an inextricable way on the architecture and layers of the network, design aids that help explore a large design space and customize the network for different applications are sorely needed.

This chapter proposes a methodology for the design space exploration (DSE) of a Human

Intranet. Our contributions can be summarized as follows:

- A formulation of the DSE problem in terms of optimized mapping of system requirements into an aggregation of components from a library that encompasses all the network layers. Our goal is to select network configurations that maximize lifetime under constraints on reliability. To the best of our knowledge, this is the first DSE framework for WBANs enabling the exploration of configurations across the entire network stack.

- An efficient algorithm that decomposes the DSE problem into smaller subproblems by coordinating specialized analysis and optimization methods. We leverage mixed integer linear programming to generate candidate network configurations based on coarse energy estimations. Accurate, discrete-event simulation is used to check the feasibility of the proposed configurations under reliability constraints, and guide the search to achieve fast convergence. On our problem instances, our algorithm outperforms state-of-the-art general-purpose optimization methods, such as simulated annealing.

We demonstrate the effectiveness of our formulation and algorithm with an example. In the example, we highlight the impact of different network topologies on the system performance metrics and the substantial dependency of the selected network configuration on the target application.

Related Work. Human Intranets are special cases of Wireless Sensor Networks. DSE methods for Wireless Sensor Networks (WSNs) can be categorized in two classes, based on the approach used to evaluate the system performance metrics for different design choices [83]. Simulation-based approaches tend to be more accurate but considerably slower, since they require running lengthy simulations on complex, high-fidelity models. On the other hand, analytical approaches provide faster ways of evaluating the network performance but are less accurate due to the simplifying assumptions usually adopted to obtain tractable models.

Beretta et al. [84] explore the energy-performance tradeoffs of a WBAN using an accurate analytical model that is orders of magnitude faster than simulation. However, their focus is on a specific network configuration, a star topology with Time Division Multiplexing (TDMA) Media Access Control (MAC) layer from the IEEE 802.15.4 standard [85]. We target, instead, a larger design space for which an accurate compact model is not available. We then combine a coarse analytical model, to rapidly prune the design space and accelerate the search, with a smaller number of performance evaluations via simulation, to achieve high accuracy on a reduced set of promising candidates.

In Grassi et. al. [86], the authors encapsulate domain-specific knowledge and rules into a discrete-state Markov decision process that is used to navigate the design space. The method is applied to IEEE 802.15.4 star networks and is shown to reduce substantially the number of simulations needed to converge. Integrating domain knowledge within the decision process is desirable for efficient design space exploration. However, it is not straightforward to implement at the early stages of the design process, especially when the design space spans different topologies and MAC layer configurations, and in the presence of high temporal

variations of the WBAN channel. We leverage the approach of Nuzzo et. al. [87, 88] and Finn et. al. [89] characterized by the coordination of Mixed Integer Linear Programming (MILP) for design space optimization with simulation-based branching to incorporate Human Intranet domain-specific knowledge and steer the MILP search toward the most promising solutions. Our approach is general, since it applies to different classes of models and can be integrated with data-driven, automated model construction techniques [83].

4.2 Problem Formulation

We follow a platform-based design methodology [90] to cast the design problem as a mapping problem where system specifications are mapped to an architecture, which consists of the interconnection of components available in a library.

4.2.1 HI Architecture

As shown in Fig. 4.1, an HI consists of N heterogeneous sensing, actuation, and processing nodes that share a wireless channel around the body. We assume that nodes may be placed in M pre-determined locations. We use a binary variable n_i to indicate the presence of a node at location i, i.e. $n_i = 1$ if and only if location i is used, with $i \in \{0, \ldots, M-1\}$. We can then represent a network *topology* as a vector $\nu = (n_0, \ldots, n_{M-1})$ specifying the locations of the nodes, under the integer linear constraint $N = \sum_{i=0}^{M-1} n_i$. Mixed integer linear inequalities can also be used to specify additional topological constraints. For instance, we can require that location i be used if location j is used by writing $n_j - n_i \leq 0$.

4.2.1.1 Wireless Channel

The wireless channel model provides the instantaneous path loss (in dBm) between any pair of node locations (i, j), that is, the attenuation of the power density of an electromagnetic wave traveling between the two locations. We use a probabilistic model, based on [91, 92], in the following form:

$$PL_{i,j}(t) = \overline{PL}_{i,j} + \delta_{PL_{i,j}}(t), \tag{4.1}$$

where $\overline{PL}_{i,j}$ is the average path loss over time and $\delta_{PL_{i,j}}(t)$ captures the temporal variation. The average path loss can be inferred from measurements on human subjects for a pre-defined set of locations on the body. The temporal variation is, instead, a random variable capturing the effect of body movements and environmental changes on the channel quality. $\delta_{PL_{i,j}}(t)$ at time t is drawn out of a probability density function that depends on the observed value $\delta_{PL_{i,j}}(t - \Delta t)$ at time $t - \Delta t$ and the time Δt elapsed since the last observation. Intuitively, if little time has passed, $\delta_{PL_{i,j}}(t)$ does not significantly differ from $\delta_{PL_{i,j}}(t - \Delta t)$. The actual probability density functions can also be empirically estimated from measurement data.

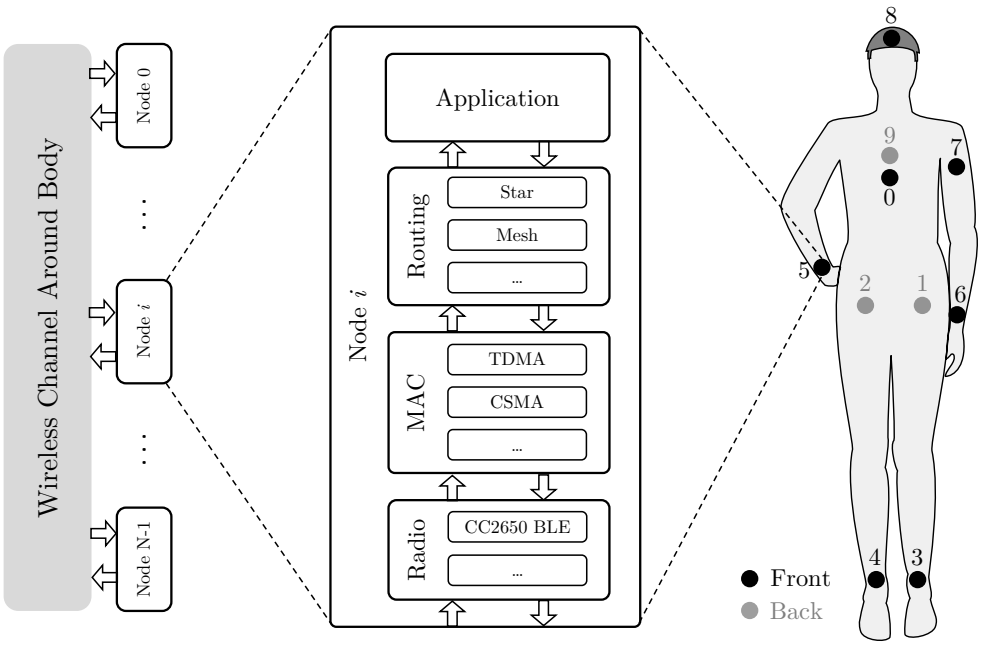


Figure 4.1: Diagram of a Human Intranet

4.2.1.2 Node

Each node supports the four standard networking layers. The radio (physical layer) is responsible for the transmission and reception of raw bit streams over the wireless medium. The MAC (data link layer) mechanism regulates how nodes gain access to the shared channel. The routing mechanism (network layer) manages addressing and routing between nodes. Finally, the application (application layer) is in charge of resource management and provides the high-level Application Programming Interface (API) to the node.

Radio. The HI radio is characterized by the following configuration parameters

$$\chi_{rd} = (f_c, BR, Tx_{dBm}, Tx_{mW}, Rx_{dBm}, Rx_{mW}), \tag{4.2}$$

where f_c is the carrier frequency, BR is the bit rate, Tx_{dBm} and Tx_{mW} are, respectively, the transmitter output power and power consumption, Rx_{dBm} and Rx_{mW} are the receiver sensitivity and power consumption.

For successful packet reception, we require that the transmitter output power (in dBm) of a node be at all times larger than the sum of the receiver sensitivity and the path loss between any pair of locations i and j occupied by nodes. Formally, $Tx_{dBm} \geq Rx_{dBm} + PL_{i,j}(t)$, $\forall\ t \geq 0, \forall\ (i,j) : n_i = n_j = 1$.

Based on the bit rate BR, we can evaluate the *packet transmission duration* as $T_{pkt} = \frac{8L}{BR}$ (in seconds), where we assume that each physical layer packet contains L bytes.

Finally, the *power consumption* used to transmit and receive packets can be estimated as follows. Most modern radios stay in sleep mode by default, and wake up either periodically or using a wake-up receiver to send or receive packets. We assume a periodic traffic with the same throughput, transmitter, and receiver power consumption for all nodes. At each round of transmissions, each node is then involved in one transmission event and $N-1$ reception events, yielding:

$$P_{rd/tx} = Tx_{mW} + (N-1)Rx_{mW}. \tag{4.3}$$

Media Access Control. The access control mechanism provides the nodes with access to the shared channel while avoiding packet collisions, by using either a contention-based or a time-multiplexed strategy. In our library, we can choose between a Carrier Sense Multiple Access (CSMA) and a Time Division Multiple Access (TDMA) protocol, which are then implemented in the discrete-event network simulator. The MAC configuration vector $\chi_{MAC} = (P_{MAC}, B_{MAC}, AM, T_{slot})$ includes the binary variable P_{MAC}, encoding the selected protocol, the integer variable AM, encoding the CSMA access mode, the buffer size B_{MAC}, and the time-slot duration T_{slot} (in seconds) for the TDMA protocol.

Remark. In CSMA, the transmitting node senses the medium to make sure that no other transmission is in progress and initiates the transmission by following a protocol given by the access mode. This causes a non-deterministic delay in communication. In TDMA, each node has exclusive access to the medium during its dedicated time slot, which makes the communication deterministic. However, scheduling the access for all nodes requires maintaining a global synchronized clock, which can be challenging, especially in an *ad hoc* wireless network with no central coordinator in charge of this task.

Routing Mechanism. We can choose between a *star* and a *mesh* network topology in our library. Therefore, the routing configuration vector $\chi_{rt} = (P_{rt}, n_{coor}, N_{hops})$ includes a binary variable P_{rt} encoding the selected protocol ($P_{rt} = 1$ if *mesh* is selected and 0 otherwise), a variable n_{coor} defining the central coordinator node in a star topology, and the maximum hop count N_{hops} for a mesh topology.

WBANs traditionally use a *star* network topology [81, 82], often with a central hub gathering bio-signals from several sensor nodes around the body, as is also included in the IEEE 802.15.6 WBAN standard [93]. However, while being attractive because of its simplicity and low energy consumption, a star topology might not be the best solution for a highly dynamic network with tight reliability constraints, because of the higher probability for the nodes to miss packets as a result of deep fading. An alternative multi-hop *mesh* topology, establishing redundant parallel links between nodes, may then be preferred. This topology would eliminate the need for a central coordinator hub, and allow for a fully distributed network.

Mesh networks generally relay messages using either flooding or point-to-point forwarding schemes [94]. We opt for a controlled flooding algorithm as a suitable choice for the highly dynamic Human Intranet, which avoids the overhead of constant routing discovery as in point-to-point forwarding schemes. In flooding, each node rebroadcasts any received packet if it is not the packet's final destination. To prevent infinite circulation of duplicate packets,

the packet payload contains a hop counter which increments every time a node is visited and blocks further retransmissions after N_{hops} is reached. Moreover, a payload includes a history of the nodes reached by the packet to avoid revisiting the same node. This causes each packet to be transmitted at most a finite number of times, denoted as N_{reTx}.

Application Layer. The application layer abstracts all the sensing, actuation, or processing functions of a node. At this layer, we keep track of the sequence numbers of sent and received packets to monitor and evaluate the network performance and power consumption. We parametrize this layer using the configuration vector $\chi_{app} = (P_{bl}, L_{pkt}, \phi)$, where P_{bl} is the node baseline power consumption due to all the node components other than the radio circuitry, L_{pkt} is the length of the generated packets, and ϕ is the data throughput (in packets per second), which is assumed equal for all nodes.

4.2.2 HI Performance Metrics

We characterize the performance of the network in terms of lifetime and packet delivery ratio.

Network Lifetime. The network lifetime is the time taken for the first node in the network to run out of energy. This is very critical to the autonomy of a BAN, where we aim to minimize the frequency of changing or recharging the battery of the wearable or implantable nodes, or maximize the effectiveness of energy harvesting.

If P_i is the power consumption of node i, and $E_{bat,i}$ is the total stored energy in that node, the network lifetime NLT is defined as

$$NLT = \min_{0 \le i \le N-1} \left\{ \frac{E_{bat,i}}{P_i} \right\}. \tag{4.4}$$

P_i can be computed as the sum of the baseline power P_{bl} and the radio power consumption P_{rd} for all nodes. While our design procedure relies on accurate computation of the overall power consumption via a discrete event network simulation, we also provide an approximate model for P_{rd} which we will use to generate promising candidate configurations to be simulated. Using (4.3), we obtain:

$$P_{rd} = \begin{cases} \phi T_{pkt}(Tx_{mW} + 2(N-1)Rx_{mW}) & P_{rt} = 0 \\ \phi T_{pkt} N_{reTx}(Tx_{mW} + (N-1)Rx_{mW}) & P_{rt} = 1 \end{cases} \tag{4.5}$$

where we consider ϕ packet transmissions per second, each lasting T_{pkt} seconds. For each transmission event, in a star topology ($P_{rt} = 0$), nodes can both receive the original packet and its retransmitted copy by the coordinator, which motivates the factor of 2 in (4.5). In a mesh topology ($P_{rt} = 1$), retransmissions are captured by the factor N_{reTx}.

Packet Delivery Ratio. The Packet Delivery Ratio (PDR) is the probability that a generated packet reaches its final destination, which is an indicator of the reliability level of the network. We use a discrete event network simulator to estimate the PDR value,

while including stochastic and second-order effects that cannot be captured by a compact analytical model.

To do so, we exploit the packet sequence numbers used by the application layer to track the statistics of the packets sent and received for each node in the network. Assume that $N^{(s)}_{i\to k}$ is the number of unique packets sent from node i to node k, during a simulation run, without counting retransmissions, and $N^{(r)}_{i\to k}$ is the number of unique packets received by node k from node i. Then the PDR of node k can be estimated as:

$$PDR_k = \frac{1}{N-1} \cdot \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \frac{N^{(r)}_{i\to k}}{N^{(s)}_{i\to k}}, \tag{4.6}$$

where the duration of a simulation run T_{sim} is selected to guarantee that the error between (4.6) and the desired probability is bounded by a positive tolerance ϵ. We can then define the overall network PDR as the average of the node PDRs:

$$PDR = \frac{1}{N} \cdot \sum_{j=0}^{N-1} PDR_j. \tag{4.7}$$

4.2.3 The Optimal Mapping Problem

Network lifetime and reliability requirements tend to conflict when higher transmission powers or redundant transmissions are used to achieve high reliability, since these result in larger power consumption. Motivated by the exploration of these trade-offs, we formulate the design problem as an optimization problem, where we search for a node configuration ν and parameter vector $\chi = (\chi_{rd}, \chi_{MAC}, \chi_{rt}, \chi_{app})$ that maximize the network lifetime subject to a lower bound PDR_{min} on the PDR. Our network is also subject to a set of *topological constraints*, specifying feasible network configurations, and *configuration constraints*, specifying feasible node configurations, e.g., lower and upper bounds on the configuration parameters for different topologies. Both the topological and configuration constraints can be lumped into the component-wise vector inequalities $r_{\mathcal{T}}(\nu, \chi) \leq 0$ and $r_{\chi}(\nu, \chi) \leq 0$, respectively. Formally,

$$\begin{aligned}
\mathcal{P} := \max_{\nu,\chi} \quad & NLT(\nu, \chi) && \text{(4.8a)} \\
\text{s.t.} \quad & r_{\mathcal{T}}(\nu, \chi) \leq 0, && \text{(4.8b)} \\
& r_{\chi}(\nu, \chi) \leq 0, && \text{(4.8c)} \\
& PDR(\nu, \chi) \geq PDR_{min} && \text{(4.8d)}
\end{aligned}$$

Based on the expressions in Sec. 4.2.1 and 4.2.2, constraints (4.8b) and (4.8c) can be posed in terms of mixed integer linear constraints. On the other hand, the objective function in (4.8a) and the PDR in (4.8d) can only be evaluated from expensive simulations. Consequently, we propose an algorithm that exploits the structure of Problem (4.8) to decompose its solution

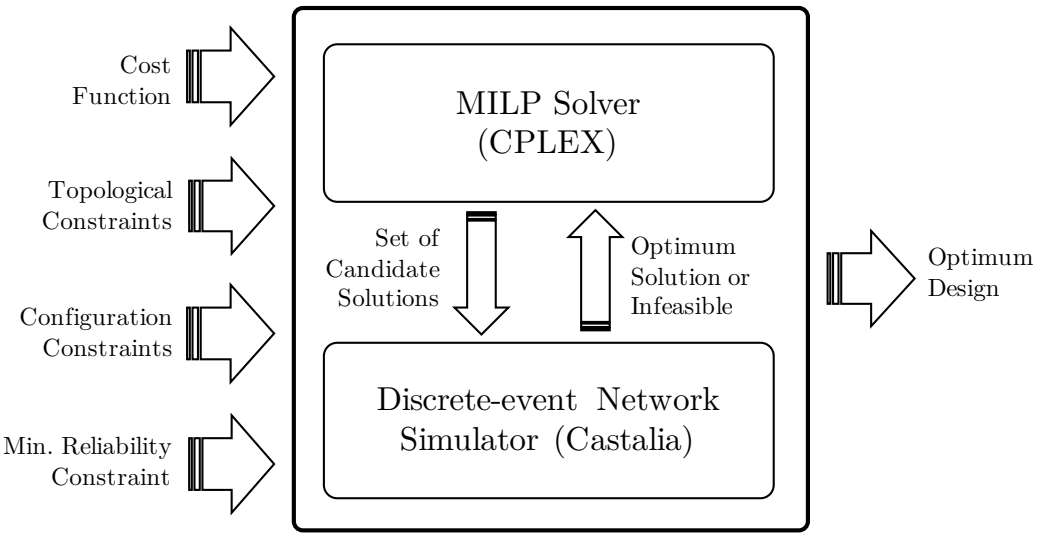


Figure 4.2: Proposed simulation-based optimization scheme.

into smaller tasks that can cooperate to rapidly navigate the design space and substantially decrease the number of simulations needed to achieve the optimum.

4.3 Proposed Algorithm

We tackle Problem (4.8) by using an iterative approach that coordinates a MILP solver and a discrete-event network simulator, as illustrated in Fig. 4.2. At each iteration, the MILP solver suggests a set of candidate solutions that satisfy all the topological and configuration constraints while maximizing an approximate expression of the NLT. A set of simulations are then run for those candidates to calculate an accurate value for the PDR, based on (4.6) and (4.7), and the objective function. If the reliability constraint is not satisfied, i.e. $PDR < PDR_{min}$, the current candidate configurations are rejected, and the MILP solver is queried to provide a new set of possible solutions at the next iteration. Our intent is to combine the efficiency of state-of-the-art MILP algorithms with the accuracy of simulation.

To do so, a key step is to derive a mixed integer linear approximation of the cost function in (4.8) that can be used by the MILP solver. We first observe that the minimum in (4.4) is achieved by a non-coordinator node, since a coordinator node (in a star topology) is usually equipped with more energy storage to perform its function. We further assume that all other nodes have the same power consumption $\bar{P}$ and stored energy $\bar{E}_{bat}$, which is a constant. Then, maximizing the network lifetime is equivalent to minimizing the power consumption $\bar{P}$ of a single non-coordinator node. Finally, a mixed integer linear expression for $\bar{P}$ can be found

Algorithm 1 Human Intranet Design Space Exploration

Input: $\mathcal{P}$
Output: $\mathcal{S}^* = (\nu^*, \chi^*)$
1: $i \leftarrow 0$, $\bar{P}_{min} \leftarrow \infty$
2: **while** true **do**
3: $(\mathcal{S}, \bar{P}^*) \leftarrow$ RunMILP$(\tilde{\mathcal{P}})$
4: **if** $\mathcal{S} = \{\}$ and $\bar{P}_{min} = \infty$ **then return** `infeasible`
5: **else if** $\frac{\bar{P}^*}{\alpha(\mathcal{S}^*, PDR_{min})} > \bar{P}_{min}$ or $\mathcal{S} = \{\}$ **then**
6: **return** $\mathcal{S}^*$
7: $(PDR_{sim}, \bar{P}_{sim}) \leftarrow$ RunSim$(\mathcal{S})$
8: $(status, \mathcal{S}^{**}, \bar{P}^{**}) \leftarrow$ Sort$(PDR_{sim}, \bar{P}_{sim})$
9: **if** $status =$ `feasible` and $\bar{P}_{min} \geq \bar{P}^{**}$ **then**
10: $\bar{P}_{min} \leftarrow \bar{P}^{**}$; $\mathcal{S}^* \leftarrow \mathcal{S}^{**}$
11: $\tilde{\mathcal{P}} \leftarrow$ Update$(\tilde{\mathcal{P}}, \bar{P} > \bar{P}^*)$; $i \leftarrow i + 1$;

based on (4.5):

$$\begin{aligned}\bar{P} = P_{bl} + \phi T_{pkt}[(1 - P_{rt})(Tx_{mW} + 2(N-1)Rx_{mW}) \\ + P_{rt} N_{reTx}(Tx_{mW} + (N-1)Rx_{mW})].\end{aligned} \tag{4.9}$$

We use (4.9) to create a relaxed version of Problem (4.8), which we call $\tilde{\mathcal{P}}$, including all the constraints except for (4.8d). $\tilde{\mathcal{P}}$ is a mixed integer linear program. We can then combine the solution of this subproblem with accurate simulation-based analysis of the performance metrics to solve the original problem $\mathcal{P}$. Our method is summarized in Algorithm 1.

At iteration i, RunMILP solves the relaxed MILP problem and returns a set of optimal solutions $\mathcal{S} = \{(\nu_j^*, \chi_j^*), 0 \leq j < S\}$ with minimum power consumption according to (4.9) (line 3 of Algorithm 1). S can be greater than zero, since multiple configurations can minimize (4.9).

A discrete-event network simulator with an accurate model of the network layers, including the time-varying probabilistic channel path loss, complements the MILP solver. RunSim takes as input the set of candidate solutions $\mathcal{S}$ and returns the vectors PDR_{sim} and $\bar{P}_{sim}$ with simulated PDR and power consumption values for all the configurations in $\mathcal{S}$ (line 7). Sort orders the feasible solutions, that is, the solutions that satisfy the reliability constraint, according to their power consumption (line 8). If no solution meets the desired reliability level, a new constraint on the lower bound of $\bar{P}$ is added (line 11) to prune the current set of solutions away from the search space.

If a feasible configuration with minimum power is found then the current solution and cost value $\bar{P}_{min}$ are updated. The algorithm terminates when either the MILP becomes infeasible, i.e. there are no further candidate solutions satisfying the topological and configuration constraints, or when the simulated power consumption of any candidate solution given by the MILP is guaranteed to be higher than the current minimum power value $\bar{P}_{min}$ (line 5). In both cases, the current optimal configuration $\mathcal{S}^*$ is returned.

Since $\bar{P}^*$ provides an estimation of the power consumption under the simplifying assumption that all the messages are correctly received and the retransmissions are successfully performed, it does not represent the lowest possible power consumption that can be obtained by simulating the network for a given PDR_{min}. Therefore, to guarantee optimality, we divide $\bar{P}^*$ by α to formulate the termination criterion in line 5. For a given PDR_{min}, α accounts for any reduction in the simulated power consumption due to the loss of packets. α can be computed as the ratio $\frac{\bar{P}}{\bar{P}_{lb}}$, where $\bar{P}_{lb}$ is the minimum power that a node must consume for the specified PDR bound. If $\frac{\bar{P}^*}{\alpha(\mathcal{S}^*, PDR_{min})} > \bar{P}_{min}$ then no further simulations need to be run since we are guaranteed that new simulations cannot achieve better power consumptions than the current best solution.

Overall, while candidate configurations are generated at each iteration based on an approximate cost function, the selection of the optimum configuration is always based on accurate evaluations obtained by simulation. The algorithm guarantees optimality, since it terminates when no more candidate configurations are available, or additional simulations cannot provide lower power consumption than the current optimum solution.

4.4 Design Example

We demonstrate the performance of the proposed algorithm on a Human Intranet scenario containing various wearable medical devices designed for monitoring human vital signs with applications in both fitness monitoring and medical diagnostics [95]. The MILP routine uses the CPLEX [96] solver via the PYTHON interface PULP [97]. CASTALIA [98], based on the OMNET++ open-source network simulator library, is used as the discrete-event simulator. The implementation of our optimization framework is available online[1]. All the optimization runs were executed on a 2.4 GHz Intel Core i7 with 8 GB of RAM. We set the duration of each simulation to $T_{sim} = 600$ s and averaged the performance metrics over 3 runs to mitigate the effect of randomness. These settings were sufficient to obtain performance estimates within 0.5% relative error.

4.4.1 Experiment Formulation

We consider the scenario in Fig. 4.1, including 10 potential node positions ($M = 10$), located at the chest, left and right hip, left and right ankle, left and right wrist, left upper arm, head, and back. $\overline{PL}_{i,j}$ for each pair of nodes is based on a two-hour measurement data set capturing the daily activity of adult subjects [99]. The probability density function of $\delta_{PL_{i,j}}$ is adapted from the set of empirical data in [98].

A set of topological constraints can be directly formulated based on the application requirements. For instance, one node must be placed on the chest ($n_0 = 1$) for respiration rate monitoring as well as the coordination (n_{coor}) in a *star* topology. At least one node

[1] https://github.com/a-moin/hi-opt.git

Table 4.1: TI CC2650 radio specifications

$\mathbf{f_c}$	2.4 GHz	**Tx Mode**	$\mathbf{Tx_{dBm}}$	$\mathbf{Tx_{mW}}$
BR	1024kbps	p_1 [1]	-20	9.55
$\mathbf{Rx_{dBm}}$	-97	p_2 [1]	-10	11.56
$\mathbf{Rx_{mW}}$	17.7	p_3	0	18.3

should be at the hip ($n_1 + n_2 \geq 1$) and one at the foot ($n_3 + n_4 \geq 1$) for gait analysis. Finally, at least one node should be placed at the wrist ($n_5 + n_6 \geq 1$) to gather several biological signals, including temperature, heart rate, pulse oxygenation, and motion signals. In addition to these four nodes, we allow up to two more nodes to be arbitrarily placed to possibly improve the mesh connectivity.

A second set of network layer configuration constraints originate from the available protocols and configuration settings, as discussed in Section 4.2.1. We use the CC2650 radio chip from Texas Instruments [100], a state-of-the-art radio chip for WSN applications, whose parameters are summarized in Table 4.1. For example, we encode the selection of one of the three transmission power levels using the binary variables p_1, p_2, and p_3. We then add the constraint $p_1+p_2+p_3 = 1$ to state that only one power value is possible in each configuration.

The MAC layer implementation in the discrete event simulator supports both the CSMA and TDMA options. For CSMA we use a *TunableMAC* implementation with non-persistence access mode [98], which reduces the collisions by backing off for a random amount of time if the medium is busy. For TDMA, we use 1 ms time slots assigned equally to all nodes in round-robin fashion. The routing constraints can also be captured by a set of mixed integer linear constraints based on (4.5) for the *star* and *mesh* options. We set $n_{coor} = n_0$ as the *star* central node at the chest, and $N_{hops} = 2$ as the maximum number of re-broadcasting hops in a *mesh* topology. For a two-hop configuration, N_{reTx} is equal to $N^2 - 4N + 5$.

For the application layer, we assume that each node generates 100-byte packets every $1/\phi = 100$ ms by burning $P_{bl} = 100\,\mu$W from a CR2032 coin cell battery. However, the coordinator node (for a *star* topology) relies on larger energy storage to perform its function. Overall, our design space contains $12,288$ potential configurations (10 node positions, 3 radio Tx power levels, 2 MAC layer options, and 2 routing schemes), which is too large to be analyzed manually.

4.4.2 Optimization Results

Figure 4.3 represents the PDR as a function of the network lifetime (NLT) for the set of feasible configurations suggested by the MILP solver. The feasible configurations span the entire range of PDRs (0 to 100%), and an NLT value from 2 days to more than a month.

The optimal solutions provided by our algorithm for different values of PDR_{min}, all above 50% (as marked by the dashed horizontal line in Fig. 4.3), are shown on the same figure. Each

[1] Not present in datasheet and based on extrapolation.

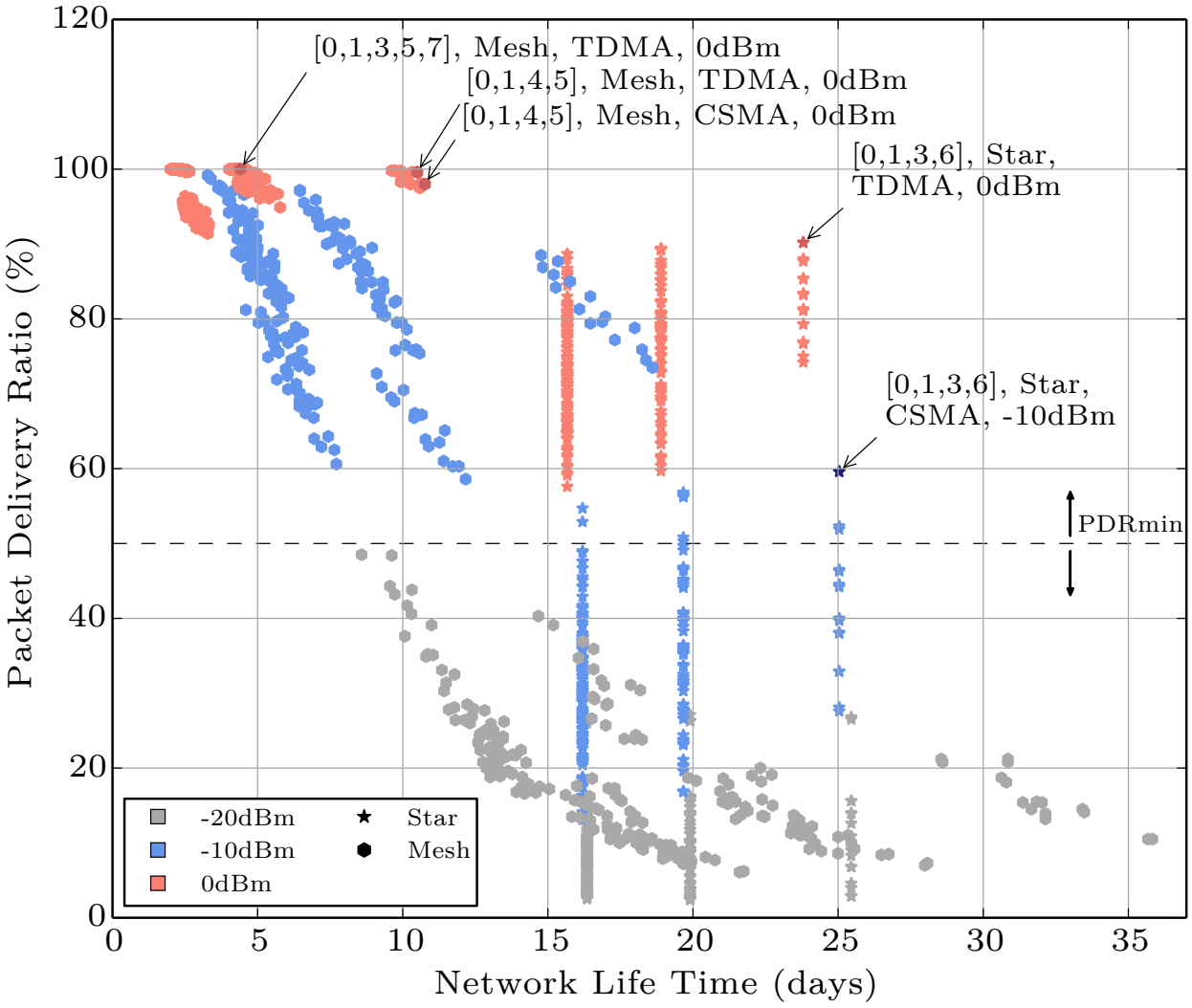


Figure 4.3: Reliability and lifetime of the feasible network configurations for the optimization problem in Sec. 4.4.1. The arrows highlight optimal configurations for different values of PDR_{min}.

optimization run took on average 52 minutes, resulting into an 87% reduction in the number of required simulations with respect to exhaustive search. In all experiments, Algorithm 1 terminated soon after the first feasible configuration satisfying the PDR constraint was found.

Our algorithm selects a star topology with a minimum number of nodes and a -10 dBm Tx power when the reliability is not a major concern (PDR_{min} below 60%). An increase in Tx power to 0 dBm is suggested for higher reliability requirements, which boosts the PDR up to about 90%. For reliability levels higher than 90%, the routing configuration switches

from *star* to *mesh*. This is in agreement with experimental studies [101] showing that a multi-hop (mesh) architecture can indeed produce very high end-to-end PDR at the expense of increased energy consumption. Finally, in a safety-critical application for which 100% reliability is required, a fifth node (n_7 on the shoulder) is added to the mesh to achieve higher redundancy. This, however, comes at the expense of a shorter network lifetime, a couple of days in this case.

The numerical results show that our algorithm enables efficient quantitative evaluation of the design trade-offs across the entire problem space of a Human Intranet and for different topologies. We also compared our approach with simulated annealing [102], a general-purpose method for optimization. On the problem instances reported here, our algorithm runs, on average, 3× faster across the whole range of PDR_{min} values of interest (from 50 to 100%).

4.5 Conclusion

Although the HI nodes such as the ones discussed in Chapters 2 and 3 are able to operate individually and on their own, an efficient and robust data network between the nodes and the cloud is required to exploit all the HI capabilities. We proposed an optimization-based design exploration approach for an HI network across the entire communication stack, capable of exploring network lifetime versus reliability trade-offs. Our algorithm combined the solution of a mixed integer linear program to select feasible candidate configurations with accurate discrete-event simulation used to evaluate the performance metrics. Our approach converged faster than general-purpose optimization frameworks on our problem instances, and was able to pinpoint the most promising configurations and network topologies while supporting a broad range of application scenarios. Our proposed DSE algorithm is useful at the design phase of the network and assuming the network status is relatively static, which is not necessarily true considering its dynamic wireless channel. Next chapter covers this dynamic behavior and proposes an adaptive method to address its challenges.

Chapter 5

Adaptive Human Intranet

5.1 Introduction

Wearable and implantable devices are becoming more prevalent in people's everyday lives with applications ranging from health monitoring and rehabilitation to human augmentation and entertainment. These devices must communicate with each other and with the cloud as an integral part of the Internet of Things (IoT), highlighting the need for a network infrastructure around the human body, traditionally named wireless body area network (WBAN) [103] and more recently the Human Intranet [104].

Energy efficiency and robustness are among the key qualifications for a well-designed WBAN. Extended battery lifetime directly affects network usability and user satisfaction, but energy resources are scarce due to the node sizes and wearability constraints. Wireless link robustness is also crucial, especially in safety-critical applications such as drug delivery [105] or control of prosthetic limbs [106]. Achieving such a reliable network while maintaining low power consumption is generally a challenging problem to address, particularly when dealing with unique characteristics of the human body.

The wireless channel around the human body is not an ideal medium for radio frequency wave transmission [107]. Lower frequencies require unacceptably large on-body antennas, whereas at higher frequencies the propagation loss quickly becomes large [108]. Additionally, the channel is highly dynamic due to fading and shadowing effects, mainly caused by changes in body posture and kinematics. In order to maintain the desired reliability, the transmission power at the transmitter must be high enough to reach the sensitivity level of the receiver at all times, including when the channel temporarily experiences high attenuation. However, a constant high transmission power sacrifices overall energy efficiency and causes interference with other nodes in the same and neighboring networks.

In order to solve this problem, transmission power control (TPC) protocols have been introduced that evaluate the channel state by monitoring received signal strength indicator (RSSI) at the receiver and adapt the transmission power level at the transmitter, accordingly [109, 110]. These methods lead to traffic overhead due to extra control packets from

receiver to transmitter. Moreover, they often fail in WBAN dynamic scenarios where the node locations are rapidly changing and the acquired link state information is outdated by the time of the next transmission. Finally in the case of periodic movements such as walking, scheduling packet transmissions at the times that channel has lower loss is a better solution than blindly increasing Tx power and consequently worsening the interference [111, 112, 113].

In this chapter, we propose an adaptive network methodology which learns from body kinematics and biosignals, predicts the wireless channel behavior based on them, and subsequently reconfigures the network to achieve both energy efficiency and robustness. Our method reuses the sensor data that already exists in a WBAN to adapt the network to different conditions with minimum overhead. These sensors include inertial measurement unit (IMU) which indicates body movements, prosthetic arm sensor which reads electrical activity of muscles, and heart rate sensor which reveals increase in physical activity. Our main contributions can be summarized as: 1) development of a body channel emulator capturing its dynamics during various movements. Use of this emulator in conjunction with our Castalia-based WBAN simulator [114, 115] enabled realistic evaluation of our adaptive network framework; 2) utilizing a custom-designed test bed containing a 2.4 GHz radio and biosignal front-ends to test our adaptive network schemes; 3) measurement and analysis of on-body channel variations and stability during main daily activities (standing, sitting, sleeping and walking); and 4) implementing computationally-light algorithms for adaptive network reconfiguration. We present a scheduling algorithm for human periodic movements and demonstrate its effectiveness with simulations. We also implement an electromyography (EMG)-controlled TPC algorithm for reliable wireless links in prostheses and an electrocardiography (ECG)-controlled TPC scheme for enhanced robustness during body movements. These algorithms are implemented on our test bed and experimental results are presented.

5.2 Related Work

A design space exploration method for WBAN is proposed in [114] to choose the best network topology and parameters for a given set of conditions before network deployment. In reality, however, there is no single best answer to this design problem due to the highly dynamic nature of the human body and its surroundings. Therefore, the network needs to be reconfigured adaptively based on its instantaneous state. Mechanisms such as TPC [109, 116, 110, 117] and link adaptation (e.g. changing modulation and data rate) [118] have been proven to be effective.

Most TPC methods are based on monitoring RSSI at the receiver and sending feedback to the transmitter [109, 116, 110]. Kim *et al.* in [117] found that RSSI is not the most accurate indicator when packet failure is mainly due to interference instead of attenuation. In these cases, increasing the transmission power would even be harmful since it worsens the interference. Moreover, all these methods suffer from the overhead of additional control packets from receiver to transmitter which their added energy consumption, traffic and

latency are not negligible [119].

The dynamic wireless channel around human body has some unique characteristics that can be leveraged. Roberts *et al.* in [112] explore the periodicity of channel path loss during certain movements and propose a new channel model, enhancing the widely used IEEE 802.15.6 model [107]. Prabh *et al.* in [113] use this periodicity in RSSI to arrange packet transmissions at proper times. However, this approach has the same overhead problem since probing packets need to be broadcast to estimate RSSI and control packets need to be sent back to the transmitter. Zang *et al.* in [111] exploit periodic body movement patterns to schedule transmissions. They run a computationally-heavy template matching algorithm based on Dynamic Time Warping (DTW) to detect the strides. The whole algorithm needs to run on the more powerful sink node and results on the next predicted transmission times are sent back to the transmitter. Hence, this method still faces the overhead problem of control packets.

In order to evaluate these methods, researchers use network simulators with human body channel models. Most of them are based on IEEE 802.15.6 standard [107] which does not capture realistic activity-dependent fluctuations in path loss due to body movements. Alam *et al.* in [120] enhance this model by finding line-of-sight (LOS) and none-line-of-sight (NLOS) distances between nodes during real movements. They multiply the NLOS portion by a constant factor to partially account for shadowing effect. Alternatively, measurement campaigns using either vector network analyzers such as in [121, 122, 123] or simple radio nodes in [111, 116] can be used to estimate the channel loss in a WBAN. However, none of the nodes used in those studies have a biosignal acquisition front-end to enable the implementation of our proposed adaptive schemes.

5.3 Body Dynamics Emulator and Test Bed

In order to leverage the advantages of both simulation-based and empirical methods in our study, we developed a realistic body channel emulator as well as a 2.4 GHz wearable node with embedded neural front-ends. The emulator was used to evaluate the performance of our proposed scheduling algorithm in realistic dynamic WBAN conditions, and the wearable nodes were used to demonstrate the efficacy of our TPC methods in real scenarios.

5.3.1 Human Body Dynamic Channel Emulator

The wireless channel around human body is highly dynamic. Most WBAN simulators either assume a static path loss between pairs of nodes or add a random variation factor to capture dynamic effects such as channel fading and shadowing [114, 115]. While such statistical models mimic the physical layer to some extent, they do not necessarily show the correlation between activity-specific kinematics of the human body and its path loss. Hence, having a wireless channel emulator that generates the path loss based on actual body movements is

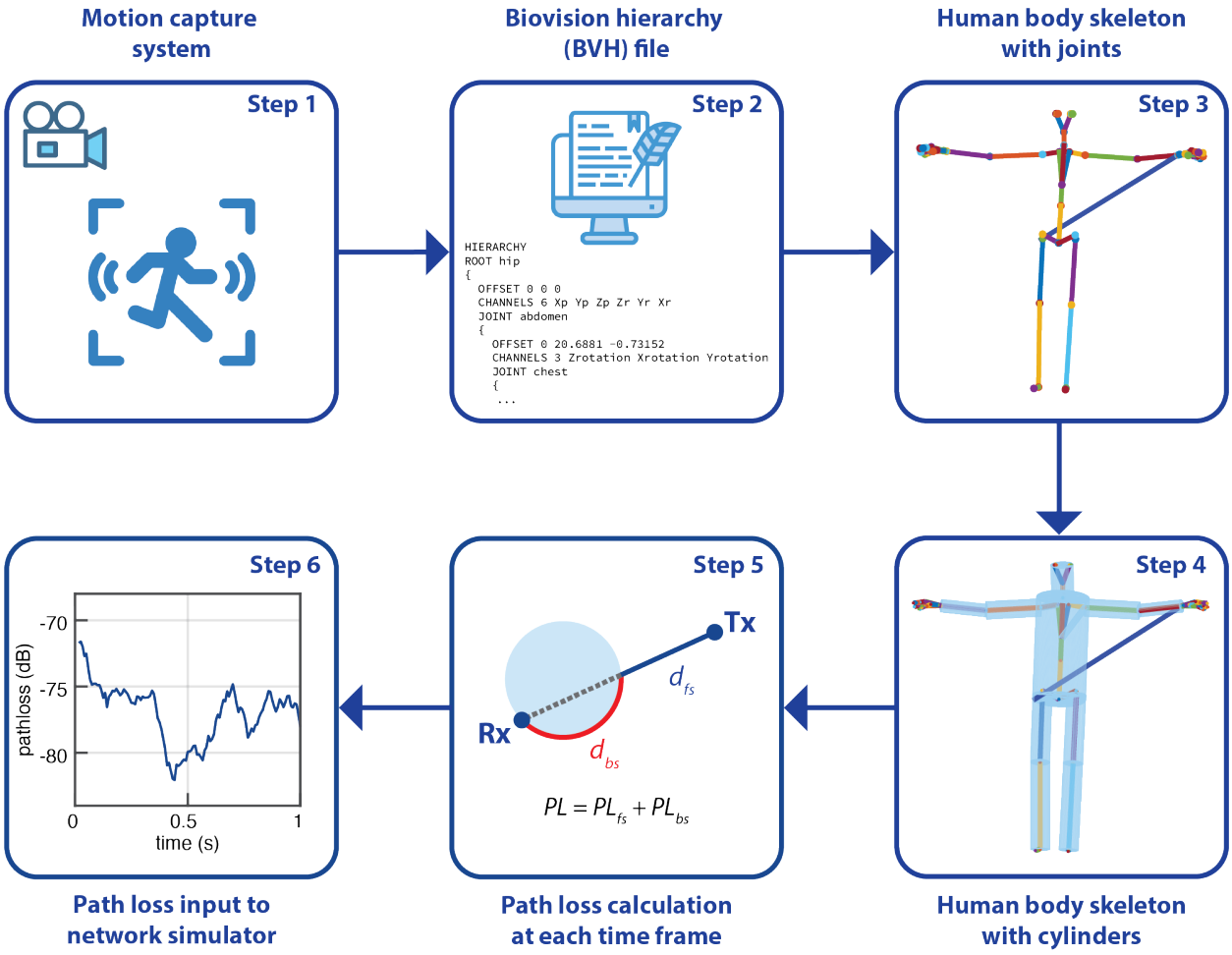


Figure 5.1: Generating a realistic human body path loss model for use in WBAN simulators.

beneficial, especially when adaptive network protocols based on those dynamics are being tested.

Figure 5.1 shows the steps that are taken to generate realistic path loss values that are used as inputs to our network simulator. The first step consists of capturing the physical extent and location of the body while holding a posture or performing a specific activity, e.g. walking. This is recorded from human subjects wearing fixed markers on their body using a dedicated motion capture system. In Step 2, these results are saved in Biovision Hierarchy (BVH) file format [124] which includes the hierarchy of body joints and their positions at each point in time. BVH files from a dataset [125] recorded by a Vicon motion capture system consisting of 12 infrared MX-40 cameras with 120 Hz sampling rate and 41 markers worn on subjects are used.

Steps 3 to 6 are performed in MATLAB to extract the path loss from BVH files, in this

case between left wrist and opposite side pants pocket. In Step 3, the skeleton hierarchy is imported and the connections between joints are drawn. A scaling matrix is then multiplied to transform the skeleton to any arbitrary height. This is repeated for each time frame in the BVH files. Step 4 involves forming cylinders to represent the body's volume in a primitive way, with cylinder dimensions based on statistical data reported in [126].

Step 5 shows the transmitter (Tx), receiver (Rx), and torso from top view. In order to calculate the path loss while considering the shadowing effect of torso, the Tx-Rx distance is divided into two portions: d_{fs} which is the free space portion without intersecting with torso, and d_{bs} which accounts for the body surface portion. The latter is calculated as the shortest curvilinear distance between the two intersecting points on the cylinder surface (red helical path in Fig. 5.1). Friis path loss formula as a function of d_{fs} (normalized to 1 m) at 2.4 GHz is then used for the free space portion:

$$PL_{fs}[dB] = 20\log_{10}(d_{fs}) + 40.0542 \tag{5.1}$$

For the body surface part, IEEE 802.15.6 CM3A model [107] is used, where N is a normal distribution with $\sigma_N = 3.8$ and d_{bs} is normalized to 1 m:

$$PL_{bs}[dB] = 6.6\log_{10}(d_{bs}) + 36.1 + N \tag{5.2}$$

Finally, the total path loss PL at each time frame is calculated as:

$$PL[dB] = PL_{fs}[dB] + PL_{bs}[dB] \tag{5.3}$$

Step 6 illustrates the full path loss after repeating these calculations for every frame. It is then saved in a file to be used as input to our full-stack network simulator [114] based on Castalia [115], which includes the estimated path loss over time for a set of different on-body links observed in a WBAN.

5.3.2 2.4 GHz Wearable Node Test Bed

We utilized a custom-designed wearable node [54] with a 2.4 GHz radio chip (nRF51822, Nordic Semiconductor) to perform empirical measurements and test our proposed adaptive methods implemented on its ARM Cortex-M0 processor. The node also features a neural front-end [9] which enables the acquisition of biosignals such as EMG and ECG.

5.4 Adaptivity for Body Dynamic Channel

In this section, we present the core idea of reusing sensor data that already exists in WBAN to learn about the network state and adaptively reconfigure its parameters. The ultimate goal is to make the network more efficient and robust with minimum added overhead. In the first part, we show that although the highly variable and often significant instantaneous path loss around the human body makes communication challenging, the channel can remain

stable for hundreds of milliseconds. We take advantage of this property by learning from the channel's behavior resulting from its kinematics and scheduling packet transmissions at appropriate moments. In the second part, we present another adaptation method based on biosignals such as EMG and ECG that are recorded in specific WBAN applications. These biosignals often correlate with body movements and postures, so the network can be adaptively reconfigured by monitoring them. Finally in the third part, we propose a state machine that integrates these adaptive schemes for an efficient and robust WBAN.

5.4.1 Human Body Kinematics

5.4.1.1 Channel Stability Analysis

Humans typically spend the majority of their daily lives in one of these states: standing, sitting, sleeping, or walking. Figure 5.2 shows the path loss variations and their moving average over 100 ms windows during these activities. We used our test bed described in Section 5.3.2 to perform these measurements with the transmitting node worn on the left wrist and the receiving node placed in the opposite side pants pocket. At first glance, a relatively large fading (variation on signal strength) is observed for all activities, even for fairly passive ones such as sitting. However, this does not necessarily imply that the channel is unstable or unpredictable. For further analysis of channel stability, we calculated two metrics for each activity: temporal autocorrelation and channel variation factor.

Temporal autocorrelation [123] of wireless channels is often used to quantify stability using the channel coherence time, i.e. the period over which this correlation remains relatively high. As Fig. 5.3(a) shows, the autocorrelation remains relatively high over several 100ms, except for walking.

Alternatively, it is possible to quantify stability using channel variation factor, defined in [127] as the ratio of the standard deviation to the square root of the mean power during a period. In this case, the period over which a channel remains stable is defined as the period during which this quantity remains low. We calculated channel variation factor for periods of 100 ms and plotted its cumulative distribution function in Fig. 5.3(b) for all four activities. This metric shows that even in the case of walking in which channel variation is periodic, channel variation factor remains relatively low (below 0.08 in 80% of our observations). This relative stability of on-body channels enables accurate channel prediction across multiple communication frames, which can help to configure the transmit power, time and duration. It is worth noting that the measurements shown in Figs. 5.2,5.3 cover brief periods of time. Over a larger time-scale, the channel predictability is potentially higher due to recurrence of certain activities (sleeping, transportation, etc.) and environments (home, office, etc.) in daily life.

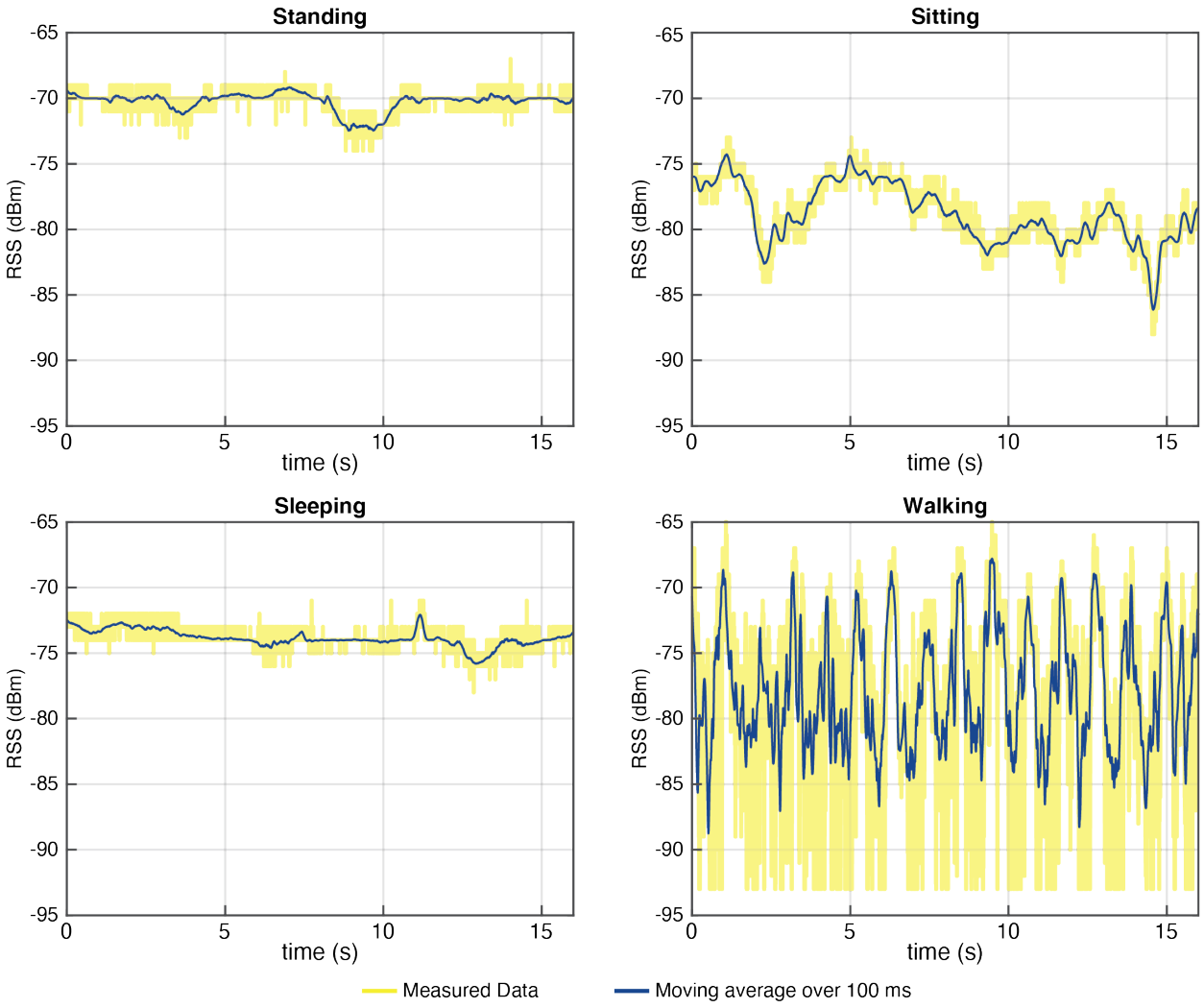


Figure 5.2: Received signal strength (RSS) measurements of the wireless channel between left wrist and right pants pocket while transmitting at 0 dBm during four common daily activities. Yellow lines show the actual measured signal strength values with sample interval of 1 ms and blue traces show running average over 100 ms windows.

5.4.1.2 Periodic Movements

Among the activities analyzed above, walking is the most challenging one with the fastest and most frequent changes in channel state. However, we can take advantage of its periodicity by predicting the peaks in received signal strength (RSS) and transmitting the packets only during those periods.

Inertial measurement units (IMUs) can be found in almost every WBAN [128], measuring

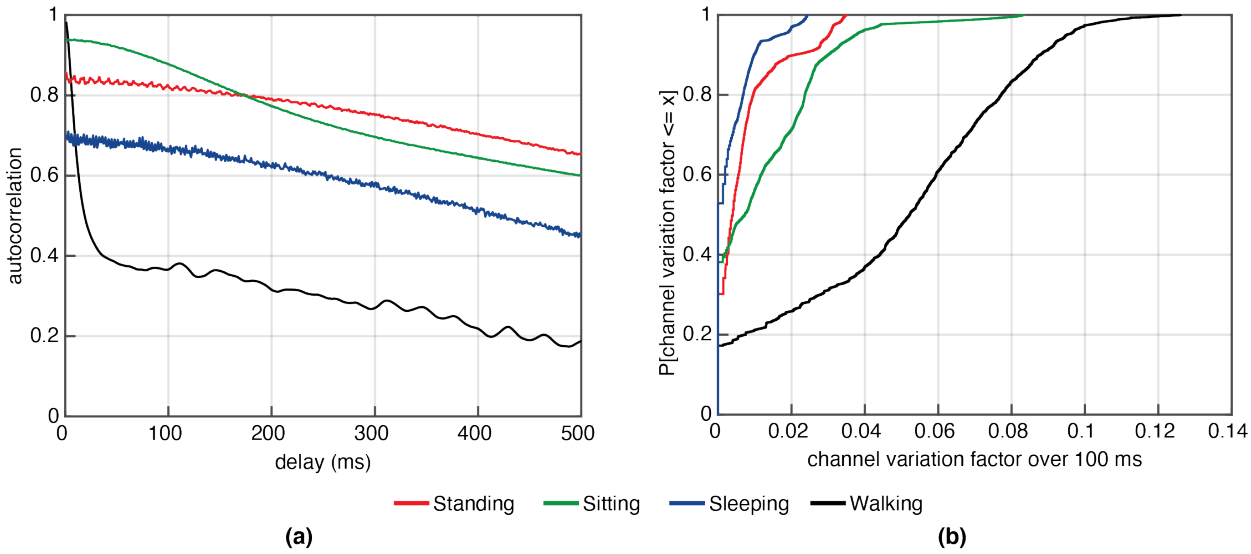


Figure 5.3: Analysis of wireless channel stability during the four activities shown in Fig. 5.2. (a) Temporal autocorrelation as a function of delay time. (b) Cumulative distribution function for the channel variation factor calculated over periods of 100 ms.

a combination of linear acceleration and rotational rate. Therefore, kinematics of the human body including the periodicity of walking would be reflected in the IMU reading. This signal has the same period as RSS and its peak times are correlated with RSS peaks.

Figure 5.4 illustrates a sample walking scenario generated by our channel emulator described in Section 5.3.1. The IMU readout (middle plot) is extracted from the wrist position (top plot) in BVH file by taking its second derivative. The path loss from left wrist to opposite side pants pocket (bottom plot) is calculated by our emulator.

The process starts with a calibration phase (shaded area in Fig. 5.4) during which we quantify the time difference between consecutive peaks in the IMU and find a coefficient to link that to RSS peaks. This coefficient is found by solving (5.4) for α:

$$t_{IMU,2} + \alpha(t_{IMU,2} - t_{IMU,1}) = t_{pathloss,p} \tag{5.4}$$

where $t_{IMU,1}$ and $t_{IMU,2}$ are the first two peaks in IMU readout and $t_{pathloss,p}$ is the first peak in RSS values that occurs after $t_{IMU,2}$. Note that RSS measurements are needed during the calibration phase which induces some overhead temporarily, but RSS values are no longer needed after the calibration is finished at $t_{pathloss,p}$. The i^{th} transmission will be scheduled at:

$$t_{Tx,i} = t_{IMU,i+2} + \alpha(t_{IMU,i+2} - t_{IMU,i+1}) \tag{5.5}$$

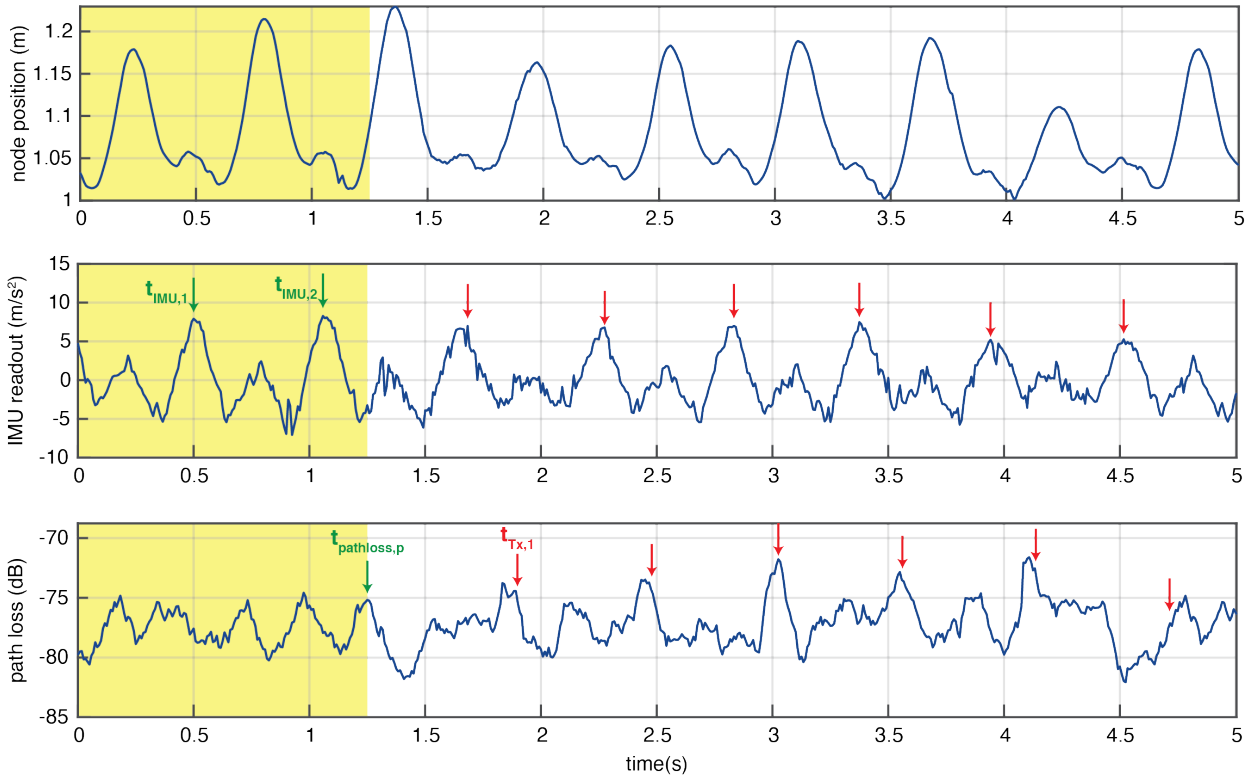


Figure 5.4: Left wrist position (top), IMU readouts (middle), and generated path loss (bottom) for the wireless link between left wrist and opposite side pants pocket in walking scenario. The shaded portion indicates the initial calibration phase. The red arrows in the middle plot show detected peaks of IMU and in the bottom plot indicate the calculated packet transmission times.

The mentioned algorithm is computationally light for implementation on wearable nodes. It also automatically adapts to varying walking pace without re-calibration since each transmission depends only on the previous two IMU peaks. Moreover, no additional control packet between receiver and transmitter is required after the calibration phase since the packet transmission schedule depends only on IMU readouts. This significantly reduces the control packets overhead that exists in most adaptive algorithms.

5.4.2 Electrical Biosignals

One of the main applications of WBAN is health monitoring by hosting either generic biosensors such as body temperature and heart rate, or more specialized nodes such as smart prostheses. The acquired biosignals provide indirect information on the network state. We propose methods to reconfigure network parameters based on these biosignals.

5.4.2.1 EMG

Smart prostheses consist of a sensing node placed on the residual limb, a controller node responsible to make decisions based on the sensed signals, and an actuator node to perform the mechanical movements. Having a robust wireless link between these components is crucial as a failure in connectivity may lead to catastrophic outcomes in the actuation, e.g. during a grasp task of handling a cup of coffee. The same challenges of wireless channel around human body exist here. Specifically, the interaction of the prosthesis with surrounding objects would most likely change the channel due to fading. We propose a learning-based adaptive algorithm that reconfigures the network when the prosthesis is engaged in an action.

EMG recorded using surface or implanted electrodes is one of the main input modalities to current prosthetic controllers [129, 130]. It enables the detection of patient's muscle contractions and interpretation of voluntary movement intentions. An increase in EMG activity implies subsequent movements of the prosthetic limb and as a result, changes in the wireless channel status are expected.

Figure 5.5(a) shows a scenario in which a prosthetic arm attached to the right arm is used to hold a metal bucket. EMG from the forearm is recorded using the test bed introduced in Section 5.3.2. The wireless link between the prosthetic node (containing both sensor and actuator) worn on the right forearm and the controller node placed in the opposite pants pocket is disrupted due to actuation movements and object shadowing effect. Our adaptive algorithm uses a binary classifier to detect muscle activity and compensates for the degradation in wireless channel quality by readjusting the transmission power level.

The radio transmission power is set to a low value of $P_{Tx,l}$ by default to keep the interference and power consumption low. The maximum (V_{max}) and minimum (V_{min}) values of EMG activity are calculated in windows of 100 ms. The node increases its transmission power to $P_{Tx,h}$ to guarantee wireless link quality under the following condition which happens during the engagement of prosthetic arm:

$$V_{max} - V_{min} > V_{EMG,thr} \tag{5.6}$$

where $V_{EMG,thr}$ is preset by training data as a threshold on EMG voltage. As experimental results show later in Section 5.5, such a low overhead algorithm can considerably improve the link robustness. It should be noted that this is a very basic classification of recorded EMG signals which can be significantly improved by using more complex algorithms [54].

5.4.2.2 ECG

Heart rate monitoring is one of the most common applications of WBAN achieved by optical or electrical measurements using nodes worn typically on wrist or chest. This signal varies based on human activity level. An increase in physical activity often leads to degradation in wireless channel quality. This degradation can be compensated by adaptively reconfiguring the network.

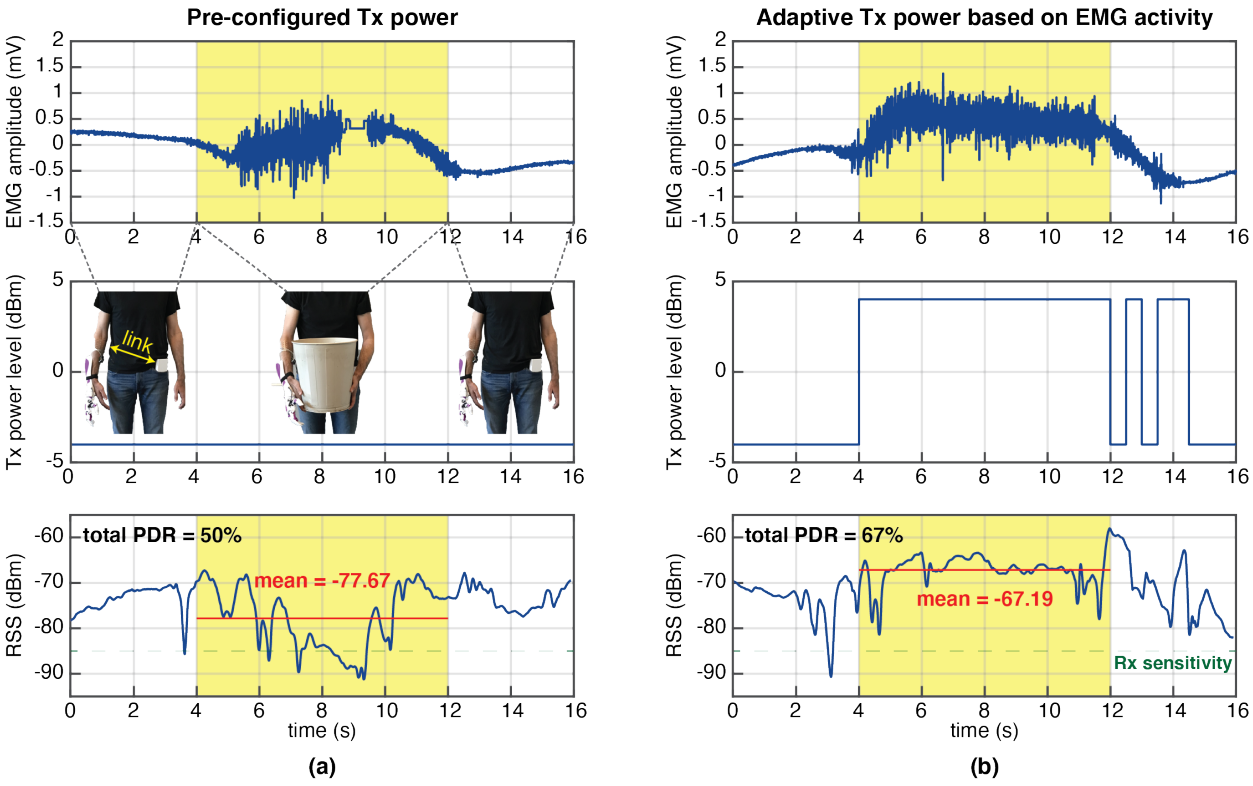


Figure 5.5: EMG-controlled TPC experimental results. The raw EMG signal (top), transmission power level (middle), and received signal strength (bottom) are plotted for: (a) a fixed transmission power scenario, and (b) an adaptive TPC based on EMG activity. Shaded areas correspond to holding the metal bucket.

Figure 5.6(a) shows ECG signal recorded from chest using our test bed explained in Section 5.3.2. We implemented a light peak detection algorithm on the radio processor to extract heart rate from ECG recordings (Fig. 5.6, second row). The occasional spikes in heart rate are due to the errors in our simple peak detection algorithm. The radio transmission power is kept at a low level of $P_{Tx,l}$ during body static state by default. When the heart rate goes above a threshold value (HR_{thr}) due to physical activity, the radio automatically increases the transmission power to $P_{Tx,h}$ to compensate for channel fading and variations. This improves the link quality as experimental results show in Section 5.5.

5.4.3 Integrated State Machine

The proposed adaptive schemes introduced in this section can all be integrated into WBAN stack as shown in the state diagram of Fig. 5.7. The network stays in *static* state by default which corresponds to activities such as sitting and sleeping. When the heart rate

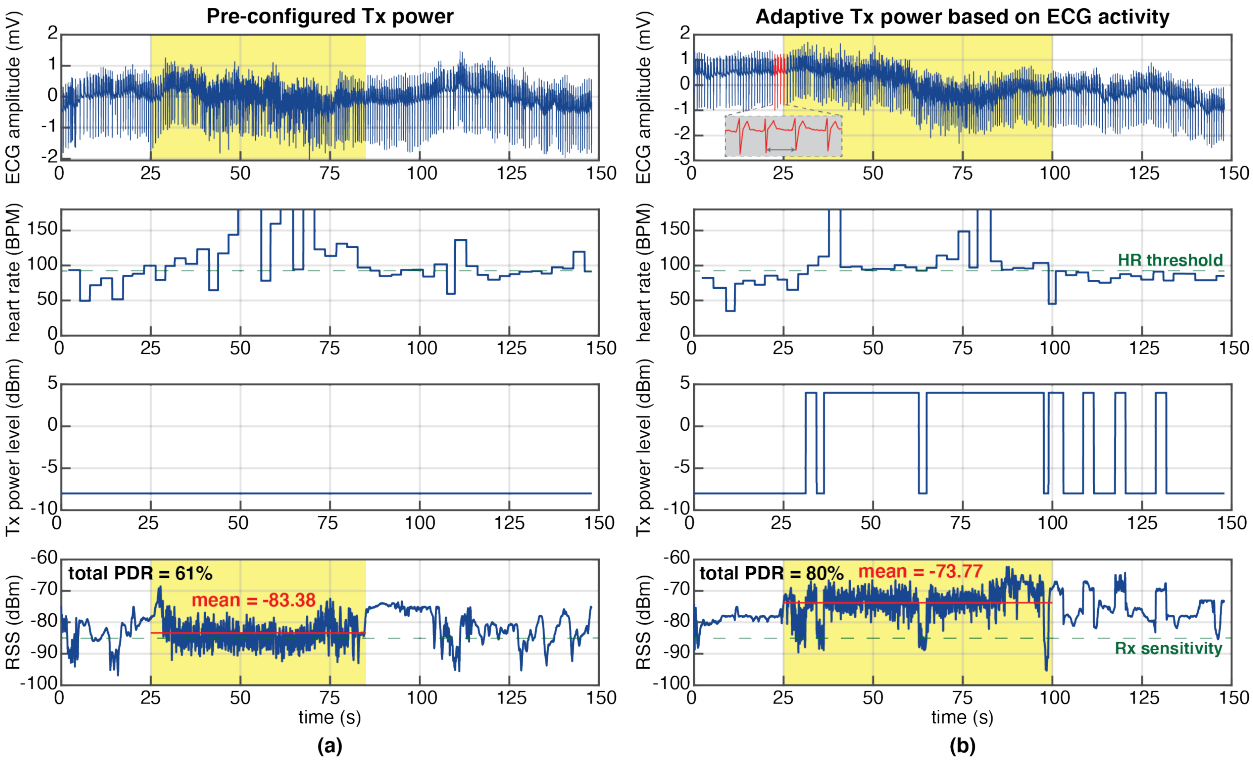


Figure 5.6: ECG-controlled TPC experimental results. The raw ECG signal (first row), extracted heart rate (second row), transmission power level (third row), and received signal strength (last row) are plotted for: (a) a fixed transmission power scenario, and (b) an adaptive TPC based on ECG activity. Shaded and unshaded areas correspond to increased physical activity level during walking and low activity during sitting down, respectively.

goes above its threshold, *IMU-based calibration* is triggered and the periodic pattern of movement is learned as described in Section 5.4.1.2. The network transmits packets based on IMU readouts as long as the heart rate stays high, otherwise it goes back to the *static* state. If EMG activity in the forearm is detected at any point of time, the network transitions to *EMG-based TPC* (Section 5.4.2.1). Note that IMU-based activity recognition for classifying static states (e.g. sitting vs. sleeping) can be done [131] to further optimize the network configurations such as routing tables and MAC layer parameters.

5.5 Experiment Results and Performance Evaluation

In this section, we present the experimental results of network efficiency and robustness after implementing our proposed adaptive scheme.

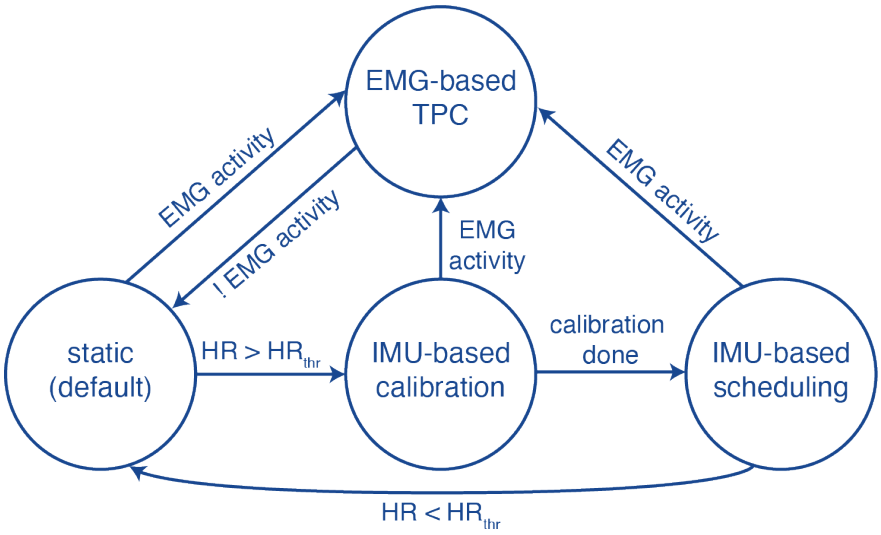


Figure 5.7: The state transition diagram of our adaptive WBAN.

5.5.1 Periodic Movements

We imported the emulated path loss from Section 5.4.1.2 into our network simulator [114]. The application layer was programmed to transmit packets in 100 ms intervals from left wrist to right pants pocket, mimicking a smart watch to smart phone communication scenario. The physical layer parameters are based on a Bluetooth chipset (nRF51822, Nordic Semiconductor) with −85 dBm receiver sensitivity in 2 Mbps data rate mode. The blue bars in Fig. 5.8 show packet delivery ratio (PDR) for different radio transmission power levels during two minutes of walking. While PDR can reach 100% by operating in a high-power mode of 0 dBm transmission power, it will go below 50% after decreasing the transmission power level to −8 dBm.

We then repeated the same simulations using our proposed adaptive algorithm based on kinematics (Section 5.4.1.2). The application layer generated packets at the same rate, but packets were buffered to be transmitted at RSS predicted peaks (red arrows in Fig. 5.4, bottom plot). The red bars in Fig. 5.8 show that using our adaptive scheduling, PDR remains at 100% after reducing transmission power level to −4 dBm, and remains acceptable (∼87%) after going down to −8 dBm. The reduced transmission power has the advantage of lower power consumption in wearable resource-limited nodes as well as less interference with other WBAN nodes.

5.5.2 EMG- and ECG-Controlled TPC

Figure 5.5 illustrates the experimental results of a subject in prosthetic arm scenario explained in Section 5.4.2.1. An increase in EMG amplitude can be observed in the top plot

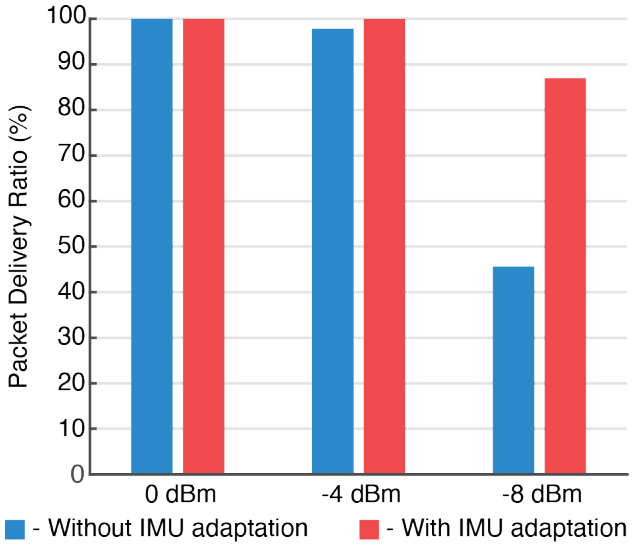


Figure 5.8: Packet delivery ratio for different transmission power levels during two minutes of walking with and without IMU-based adaptation.

of Fig. 5.5(a) between 4 s and 12 s when the subject transfers from rest state to holding the metal bucket. The bottom plot shows RSS when transmitting packets with fixed transmission power of $P_{Tx,l}$ =−4 dBm. The RSS is degraded during the hold task (shaded area of plots) which led to a total PDR of 50%.

Figure 5.5(b) shows the results for the same experiment after implementing our adaptive algorithm on the radio processor with $V_{EMG,thr}$ =610 µV. The transmission power level is automatically boosted to $P_{Tx,h}$ =+4 dBm upon detection of EMG activity to assure higher link reliability. Consequently, mean RSS is increased more than 10 dBm during the prosthetic arm engagement and total PDR is increased to 67%.

Figure 5.6 presents the experimental results for the ECG-controlled TPC scenario explained in Section 5.4.2.2. The subject is wearing the test bed node on the left wrist and the receiver node is placed in right pants pocket. The experiment starts with rest in sitting state followed by an increase in activity (walking with high pace) and back to rest. The bottom plot in Fig. 5.6(a) shows RSS when transmitting packets with fixed transmission power of $P_{Tx,l}$ =−8 dBm. The RSS is degraded during walking (shaded area of plots) which led to a total PDR of 61%.

The neural front-end of our test bed is connected to an Ag/Ag-Cl electrode placed on the subject's chest to measure their ECG. The implemented algorithm on the radio processor extracts the heart rate by measuring the distance between ECG peaks and adaptively toggles the transmission power level to $P_{Tx,h}$ =+4 dBm when the heart rate goes higher than HR_{thr} =92 bps threshold. Figure 5.6(b) shows that our proposed adaptive scheme increases the mean RSS during body movements by 9.6 dBm, causing the total PDR to reach 80%.

Table 5.1: Parameters used in sample scenarios.

P_{Tx} for Walking	-8 dBm
$P_{Tx,l}$ for Prosthesis	-4 dBm
$P_{Tx,h}$ for Prosthesis	$+4$ dBm
Power Consumption at -8 dBm	21 mW
Power Consumption at -4 dBm	24 mW
Power Consumption at $+4$ dBm	31.5 mW

5.5.3 Power Savings in Sample Scenarios

In order to show the energy efficiency of our adaptive method, we performed a comparison between scenarios with and without engaging our proposed algorithm. The parameters used for this comparison are listed in Table 5.1. We assumed failed packets would be successfully delivered after one re-transmission attempt.

In the case of walking, 27% reduction in power consumption was achieved by using IMU-based scheduling. For the prosthetic arm scenario, constantly transmitting at $P_{Tx,h}$ to have a reliable link caused 31% increase in power consumption compared to fix $P_{Tx,l}$ case. Incorporating the adaptive EMG-controlled TPC, however, led to only 16% increase in power consumption while the link reliability was preserved. Note that the overhead power of running our adaptive scheme is negligible since the IMU data and biosignals are already available in the network and the algorithms are computationally light.

5.6 Conclusion

This chapter demonstrated the efficacy of an adaptive WBAN approach which reconfigures the network based on body kinematics and biosignals. The body dynamic channel emulator presented here can be used in other WBAN studies that require a realistic channel model, especially considering the extensive available BVH datasets of various, more complex physical activities such as jumping and dancing. More biosignals and sensors may be added to this adaptive scheme in the future to increase the network energy-efficiency and robustness under various scenarios. These methods can ultimately be added as a vertical control plane to the existing protocols such as IEEE 802.15.6 standard to learn, identify and switch between the network states similar to Fig. 5.7.

Chapter 6

Final Thoughts

This thesis was a thorough demonstrator of a Human Intranet, featuring OMNI/WAND implantable node and a high-density EMG wearable node. It also tackled some of the challenges in designing an efficient and robust data communication network responsible for connecting the HI nodes, by proposing solutions for both before and after the network deployment.

There is no doubt in the explosive growth of the *smart* society, and the Trillion Sensor Universe [132] is becoming a reality. However, the input/output bandwidth of the human has not been increased with the same pace. We, as technology researchers, are responsible to fill this gap and allow the human to keep up with this smart world. Human Intranet tries to achieve this by extending our input/output modalities. Recent advent of wearable and implantable devices proves that the *augmented human* is no longer pure imagination. I believe that the concepts discussed in this thesis are among the key enablers for HI and can be used in its future developments:

- Closed-loop sense-interpret-actuate systems such as OMNI/WAND introduced in Chapter 2.
- Energy- and hardware-efficient in-sensor learning methods as demonstrated in Chapter 3.
- DSE algorithms that converge fast and find the optimum solutions for system and network design such as the one implemented in Chapter 4.
- Agile network protocols that learn from the network behavior and adapt to dynamic conditions, e.g. the one presented in Chapter 5.

Bibliography

[1] Jan M Rabaey. "The Human Intranet–Where Swarms and Humans Meet". In: *IEEE Pervasive Computing* 14.1 (2015), pp. 78–83.

[2] World Health Organization. *Neurological disorders: public health challenges*. World Health Organization, 2006. Chap. 2, p. 32.

[3] Helen S Mayberg et al. "Deep brain stimulation for treatment-resistant depression". In: *Neuron* 45.5 (2005), pp. 651–660.

[4] Scott Stanslaski et al. "Design and Validation of a Fully Implantable, Chronic, Closed-Loop Neuromodulation Device With Concurrent Sensing and Stimulation". In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 20.4 (July 2012), pp. 410–421.

[5] Felice T Sun and Martha J Morrell. "The RNS System: responsive cortical stimulation for the treatment of refractory partial epilepsy". In: *Expert review of medical devices* 11.6 (2014), pp. 563–572.

[6] J.J. Wheeler et al. "An implantable 64-channel neural interface with reconfigurable recording and stimulation". In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. Aug. 2015, pp. 7837–7840.

[7] Reid R Harrison et al. "A Low-Power Integrated Circuit for a Wireless 100-Electrode Neural Recording System". In: *IEEE Journal of Solid-State Circuits* 42.1 (2006), pp. 123–133.

[8] Fan Zhang, Jeremy Holleman, and Brian P Otis. "Design of ultra-low power biopotential amplifiers for biosignal acquisition applications". In: *IEEE transactions on biomedical circuits and systems* 6.4 (2012), pp. 344–355.

[9] Benjamin C Johnson et al. "An implantable 700μW 64-channel neuromodulation IC for simultaneous recording and stimulation with rapid artifact recovery". In: *2017 Symposium on VLSI Circuits*. IEEE. 2017, pp. C48–C49.

[10] C.K. Bjune et al. "Package architecture and component design for an implanted neural stimulator with closed loop control". In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. Aug. 2015, pp. 7825–7830.

[11] Ryan T Canolty, Karunesh Ganguly, and Jose M Carmena. "Task-dependent changes in cross-level coupling between single neurons and oscillatory activity in multiscale networks". In: *PLoS computational biology* 8.12 (2012), e1002809.

[12] Maryam Saleh et al. "Fast and slow oscillations in human primary motor cortex predict oncoming behaviorally relevant cues". In: *Neuron* 65.4 (2010), pp. 461–471.

[13] Jerome N Sanes and John P Donoghue. "Oscillations in local field potentials of the primate motor cortex during voluntary movement." In: *Proceedings of the National Academy of Sciences* 90.10 (1993), pp. 4470–4474.

[14] Annette Bastian, Gregor Schöner, and Alexa Riehle. "Preshaping and continuous evolution of motor cortical representations during movement preparation". In: *European Journal of Neuroscience* 18.7 (2003), pp. 2047–2058.

[15] Mark M Churchland et al. "Neural variability in premotor cortex provides a signature of motor preparation". In: *Journal of Neuroscience* 26.14 (2006), pp. 3697–3712.

[16] Mark M Churchland and Krishna V Shenoy. "Delay of movement caused by disruption of cortical preparatory activity". In: *Journal of neurophysiology* 97.1 (2007), pp. 348–359.

[17] John P Donoghue et al. "Neural discharge and local field potential oscillations in primate motor cortex during voluntary movements". In: *Journal of neurophysiology* 79.1 (1998), pp. 159–173.

[18] Jose L Cantero, Mercedes Atienza, and Rosa M Salas. "Human alpha oscillations in wakefulness, drowsiness period, and REM sleep: different electroencephalographic phenomena within the alpha band". In: *Neurophysiologie Clinique/Clinical Neurophysiology* 32.1 (2002), pp. 54–71.

[19] Hartmut Schulz. "Rethinking sleep analysis Comment on the AASM Manual for the Scoring of Sleep and Associated Events". In: *Journal of Clinical Sleep Medicine* 4.02 (2008), pp. 99–103.

[20] Leon F Heffer and James B Fallon. "A novel stimulus artifact removal technique for high-rate electrical stimulation". In: *Journal of neuroscience methods* 170.2 (2008), pp. 277–284.

[21] Felice T Sun, Martha J Morrell, and Robert E Wharen. "Responsive cortical stimulation for the treatment of epilepsy". In: *Neurotherapeutics* 5.1 (2008), pp. 68–74.

[22] Nicole C Swann et al. "Gamma oscillations in the hyperkinetic state detected with chronic human brain recordings in Parkinson's disease". In: *Journal of Neuroscience* 36.24 (2016), pp. 6445–6458.

[23] Jayoung Kim et al. "Wearable biosensors for healthcare monitoring". In: *Nature biotechnology* 37.4 (2019), pp. 389–406.

[24] Amay J Bandodkar, Itthipon Jeerapan, and Joseph Wang. "Wearable chemical sensors: Present challenges and future prospects". In: *Acs Sensors* 1.5 (2016), pp. 464–482.

[25] Yasser Khan et al. "Monitoring of vital signs with flexible and wearable medical devices". In: *Advanced Materials* 28.22 (2016), pp. 4373–4395.

[26] Shyamal Patel et al. "A review of wearable sensors and systems with application in rehabilitation". In: *Journal of neuroengineering and rehabilitation* 9.1 (2012), p. 21.

[27] Shogo Nakata et al. "A wearable pH sensor with high sensitivity based on a flexible charge-coupled device". In: *Nature Electronics* 1.11 (2018), p. 596.

[28] Zhenlong Huang et al. "Three-dimensional integrated stretchable electronics". In: *Nature Electronics* 1.8 (2018), p. 473.

[29] Wei Gao et al. "Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis". In: *Nature* 529.7587 (2016), p. 509.

[30] Somayeh Imani et al. "A wearable chemical–electrophysiological hybrid biosensing system for real-time health and fitness monitoring". In: *Nature communications* 7 (2016), p. 11650.

[31] Claire M Lochner et al. "All-organic optoelectronic sensor for pulse oximetry". In: *Nature communications* 5 (2014), p. 5745.

[32] Yasser Khan et al. "A flexible organic reflectance oximeter array". In: *Proceedings of the National Academy of Sciences* 115.47 (2018), E11015–E11024.

[33] Yasser Khan et al. "Flexible hybrid electronics: direct interfacing of soft and hard electronics for wearable health monitoring". In: *Advanced Functional Materials* 26.47 (2016), pp. 8764–8775.

[34] Yikun Gu et al. "Robust EMG pattern recognition in the presence of confounding factors: features, classifiers and adaptive learning". In: *Expert Systems with Applications* 96 (2018), pp. 208–217.

[35] Aaron J Young, Levi J Hargrove, and Todd A Kuiken. "The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift". In: *IEEE Transactions on Biomedical Engineering* 58.9 (2011), pp. 2537–2544.

[36] Xiaorong Zhang and He Huang. "A real-time, practical sensor fault-tolerant module for robust EMG pattern recognition". In: *Journal of neuroengineering and rehabilitation* 12.1 (2015), p. 18.

[37] Levi Hargrove, Kevin Englehart, and Bernard Hudgins. "A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control". In: *Biomedical signal processing and control* 3.2 (2008), pp. 175–180.

[38] Dennis Tkach, He Huang, and Todd A Kuiken. "Study of stability of time-domain features for electromyographic pattern recognition". In: *Journal of neuroengineering and rehabilitation* 7.1 (2010), p. 21.

[39] Bojan Milosevic, Elisabetta Farella, and Simone Benatti. "Exploring arm posture and temporal variability in myoelectric hand gesture recognition". In: *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE. 2018, pp. 1032–1037.

[40] Gea Drost et al. "Clinical applications of high-density surface EMG: a systematic review". In: *Journal of Electromyography and Kinesiology* 16.6 (2006), pp. 586–602.

[41] Dick F Stegeman et al. "High-density surface EMG: techniques and applications at a motor unit level". In: *Biocybernetics and biomedical engineering* 32.3 (2012), pp. 3–27.

[42] Christoph Amma et al. "Advancing muscle-computer interfaces with high-density electromyography". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 929–938.

[43] Weidong Geng et al. "Gesture recognition by instantaneous surface EMG images". In: *Scientific reports* 6 (2016), p. 36571.

[44] Xun Chen and Z Jane Wang. "Pattern recognition of number gestures based on a wireless surface EMG system". In: *Biomedical Signal Processing and Control* 8.2 (2013), pp. 184–192.

[45] Simone Benatti et al. "A versatile embedded platform for EMG acquisition and gesture recognition". In: *IEEE transactions on biomedical circuits and systems* 9.5 (2015), pp. 620–630.

[46] Xilin Liu et al. "The virtual trackpad: An electromyography-based, wireless, real-time, low-power, embedded hand-gesture-recognition system using an event-driven artificial neural network". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.11 (2016), pp. 1257–1261.

[47] Sidharth Pancholi and Amit M Joshi. "Electromyography-based hand gesture recognition system for upper limb amputees". In: *IEEE Sensors Letters* 3.3 (2019), pp. 1–4.

[48] Dario Farina et al. "Principles of motor unit physiology evolve with advances in technology". In: *Physiology* 31.2 (2016), pp. 83–94.

[49] Bert U Kleine et al. "Using two-dimensional spatial information in decomposition of surface EMG signals". In: *Journal of electromyography and kinesiology* 17.5 (2007), pp. 535–548.

[50] Vojko Glaser, Aleš Holobar, and Damjan Zazula. "Real-time motor unit identification from high-density surface EMG". In: *IEEE transactions on neural systems and rehabilitation engineering* 21.6 (2013), pp. 949–958.

[51] M Pozzo et al. "Sixty-four channel wearable acquisition system for long-term surface electromyogram recording with electrode arrays". In: *Medical and Biological Engineering and Computing* 42.4 (2004), pp. 455–466.

[52] Umberto Barone and Roberto Merletti. "Design of a portable, intrinsically safe multichannel acquisition system for high-resolution, real-time processing HD-sEMG". In: *IEEE Transactions on Biomedical Engineering* 60.8 (2013), pp. 2242–2252.

[53] Giacinto Luigi Cerone, Alberto Botter, and Marco Gazzoni. "A Modular, Smart, and Wearable System for High Density sEMG Detection". In: *IEEE Transactions on Biomedical Engineering* (2019).

[54] Ali Moin et al. "An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2018, pp. 1–5.

[55] Bernd G Lapatki et al. "A thin, flexible multielectrode grid for high-density surface EMG". In: *Journal of Applied Physiology* 96.1 (2004), pp. 327–336.

[56] Manfredo Atzori et al. "Electromyography data for non-invasive naturally-controlled robotic hand prostheses". In: *Scientific data* 1 (2014), p. 140053.

[57] Stefano Pizzolato et al. "Comparison of six electromyography acquisition setups on hand movement classification tasks". In: *PloS one* 12.10 (2017), e0186132.

[58] Manfredo Atzori et al. "Building the Ninapro database: A resource for the biorobotics community". In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2012, pp. 1258–1265.

[59] Yu Du et al. "Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation". In: *Sensors* 17.3 (2017), p. 458.

[60] Karthik Sivarama Krishnan et al. "Recognition of human arm gestures using Myo armband for the game of hand cricket". In: *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. IEEE. 2017, pp. 389–394.

[61] Yunyong Ploengpit and Tanasanee Phienthrakul. "Rock-paper-scissors with Myo Armband pose detection". In: *2016 International Computer Science and Engineering Conference (ICSEC)*. IEEE. 2016, pp. 1–5.

[62] Kristian Nymoen, Mari Romarheim Haugen, and Alexander Refsum Jensenius. "Mumyo–evaluating and exploring the myo armband for musical interaction". In: *Proc. Int. Conf. New Interfaces Music. Expr.)* 2015.

[63] João Gabriel Abreu et al. "Evaluating sign language recognition using the myo armband". In: *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*. IEEE. 2016, pp. 64–70.

[64] Gabriel Doretto Morais et al. "Application of Myo Armband System to Control a Robot Interface." In: *BIOSIGNALS*. 2016, pp. 227–231.

[65] Yanbin Xu et al. "Development of a hybrid motion capture method using MYO armband with application to teleoperation". In: *2016 IEEE International Conference on Mechatronics and Automation*. IEEE. 2016, pp. 1179–1184.

[66] Pentti Kanerva. "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors". In: *Cognitive computation* 1.2 (2009), pp. 139–159.

[67] Abbas Rahimi et al. "Hyperdimensional computing for blind and one-shot classification of EEG error-related potentials". In: *Mobile Networks and Applications* (2017), pp. 1–12.

[68] Alessio Burrello et al. "One-shot learning for iEEG seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing". In: *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE. 2018, pp. 1–4.

[69] Abbas Rahimi et al. "High-dimensional computing as a nanoscalable paradigm". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.9 (2017), pp. 2508–2521.

[70] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[71] Lauren H Smith et al. "Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.2 (2010), pp. 186–192.

[72] Kevin Englehart, Bernard Hudgins, et al. "A robust, real-time control scheme for multifunction myoelectric control". In: *IEEE transactions on biomedical engineering* 50.7 (2003), pp. 848–854.

[73] Pentti Kanerva. "Binary spatter-coding of ordered K-tuples". In: *International Conference on Artificial Neural Networks*. Springer. 1996, pp. 869–873.

[74] Manuel Schmuck, Luca Benini, and Abbas Rahimi. "Hardware Optimizations of Dense Binary Hyperdimensional Computing: Rematerialization of Hypervectors, Binarized Bundling, and Combinational Associative Memory". In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 15.4 (2019), p. 32.

[75] Yanjuan Geng, Ping Zhou, and Guanglin Li. "Toward attenuating the impact of arm positions on electromyography pattern-recognition based motion classification in transradial amputees". In: *Journal of neuroengineering and rehabilitation* 9.1 (2012), p. 74.

[76] Aaron J Young, Levi J Hargrove, and Todd A Kuiken. "Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration". In: *IEEE Transactions on Biomedical Engineering* 59.3 (2011), pp. 645–652.

[77] Ali H Al-Timemy et al. "Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees". In: *IEEE Trans. on Neural Systems and Rehab. Eng.* 24.6 (2015).

[78] Erik Scheme and Kevin Englehart. "Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use." In: *Journal of Rehabilitation Research & Development* 48.6 (2011).

[79] Rami N Khushaba et al. "Combined influence of forearm orientation and muscular contraction on EMG pattern recognition". In: *Expert Systems with Applications* 61 (2016), pp. 154–161.

[80] Jiayuan He et al. "Invariant surface EMG feature against varying contraction level for myoelectric control based on muscle coordination". In: *IEEE journal of biomedical and health informatics* 19.3 (2014), pp. 874–882.

[81] Jason WP Ng et al. "Ubiquitous monitoring environment for wearable and implantable sensors (UbiMon)". In: *Int. Conf. Ubiquitous Computing*. 2004.

[82] Bert Gyselinckx et al. "Human++: Emerging Technology for Body Area Networks." In: *VLSI-SoC*. 2006.

[83] Lan S Bai et al. "Automated construction of fast and accurate system-level models for wireless sensor networks". In: *Proc. Design, Automation and Test in Europe Conf.* 2011.

[84] Ivan Beretta et al. "Design exploration of energy-performance trade-offs for wireless sensor networks". In: *Proc. Design Automation Conf.* 2012.

[85] "IEEE Std 802.15.4-2006". In: IEEE Computer Society, 2006.

[86] Paolo Roberto Grassi et al. "Knowledge-based design space exploration of wireless sensor networks". In: *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis*. 2012.

[87] P. Nuzzo et al. "A contract-based methodology for aircraft electric power system design". In: *IEEE Access* 2 (2013), pp. 1–25.

[88] Nikunj Bajaj et al. "Optimized selection of reliable and cost-effective cyber-physical system architectures". In: *Proc. Design, Automation and Test in Europe Conf.* 2015.

[89] John Finn, Pierluigi Nuzzo, and Alberto Sangiovanni-Vincentelli. "A Mixed Discrete-Continuous Optimization Scheme for Cyber-Physical System Architecture Exploration". In: *Proc. Int. Conf. Computer-Aided Design*. 2015.

[90] Alberto Sangiovanni-Vincentelli. "Quo vadis, SLD? Reasoning about the trends and challenges of system level design". In: *Proceedings of the IEEE* 95.3 (2007).

[91] David B Smith, Athanassios Boulis, and Yuriy Tselishchev. "Efficient conditional-probability link modeling capturing temporal variations in body area networks". In: *Proc. Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2012.

[92] Dimosthenis Pediaditakis, Yuri Tselishchev, and Athanassios Boulis. "Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator". In: *Proc. Int. Conf. Simulation Tools and Techniques*. 2010.

[93] "IEEE Standard for Local and Metropolitan Area Networks - Part 15.6: Wireless Body Area Networks". In: IEEE Computer Society, 2012.

[94] A Rahman, W Olesinski, and P Gburzynski. "Controlled flooding in wireless ad-hoc networks". In: *Int. Workshop on Wireless Ad-Hoc Networks*. 2004.

[95] Yasser Khan et al. "Monitoring of Vital Signs with Flexible and Wearable Medical Devices". In: *Advanced Materials* (2015).

[96] `www.ibm.com/software/integration/optimization/cplex-optimizer`.

[97] `https://pypi.python.org/pypi/PuLP`.

[98] `https://castalia.forge.nicta.com.au/`.

[99] `http://www.opennicta.com.au/datasets`.

[100] *TI CC2650 Datasheet*. (Date last accessed 15-July-2014). URL: `http://www.ti.com/lit/gpn/cc2650`.

[101] Anirudh Natarajan et al. "To hop or not to hop: Network architecture for body sensor networks". In: *Proc. Conf. Sensor, Mesh and Ad Hoc Communications and Networks*. 2009.

[102] `https://github.com/perrygeo/simanneal`.

[103] Riccardo Cavallari et al. "A survey on wireless body area networks: Technologies and design challenges". In: *IEEE Communications Surveys & Tutorials* 16.3 (2014), pp. 1635–1657.

[104] Jan M Rabaey. "The Human Intranet–Where Swarms and Humans Meet". In: *IEEE Pervasive Computing* 14.1 (2015), pp. 78–83.

[105] Maulin Patel and Jianfeng Wang. "Applications, challenges, and prospective in emerging body area networking technologies". In: *IEEE Wireless communications* 17.1 (2010), pp. 80–88.

[106] Simone Benatti et al. "A prosthetic hand body area controller based on efficient pattern recognition control strategies". In: *Sensors* 17.4 (2017), p. 869.

[107] Kamya Yazdandoost. "Channel model for body area network (BAN)". In: *IEEE 802.15-08-0780-05-0006* (2009).

[108] Julien Ryckaert et al. "Channel model for wireless communication around human body". In: *Electronics letters* 40.9 (2004), pp. 543–544.

[109] Shuo Xiao et al. "Transmission power control in body area sensor networks for healthcare monitoring". In: *IEEE Journal on Selected Areas in Communications* 27.1 (2009), pp. 37–48.

[110] Seungku Kim and Doo-Seop Eom. "Link-state-estimation-based transmission power control in wireless body area networks". In: *IEEE journal of Biomedical and Health Informatics* 18.4 (2013), pp. 1294–1302.

[111] Weilin Zang and Ye Li. "Gait-Cycle-Driven Transmission Power Control Scheme for a Wireless Body Area Network". In: *IEEE journal of biomedical and health informatics* 22.3 (2017), pp. 697–706.

[112] Nathan E Roberts, Seunghyun Oh, and David D Wentzloff. "Exploiting channel periodicity in body sensor networks". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.1 (2012), pp. 4–13.

[113] K Shashi Prabh et al. "BANMAC: an opportunistic MAC protocol for reliable communications in body area networks". In: *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*. IEEE. 2012, pp. 166–175.

[114] Ali Moin et al. "Optimized design of a Human Intranet network". In: *Proceedings of the 54th Annual Design Automation Conference 2017*. ACM. 2017, p. 30.

[115] Athanassios Boulis. "Castalia: revealing pitfalls in designing distributed algorithms in WSN". In: *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM. 2007, pp. 407–408.

[116] Muhannad Quwaider, Jayanthi Rao, and Subir Biswas. "Body-posture-based dynamic link power control in wearable sensor networks". In: *IEEE Communications Magazine* 48.7 (2010), pp. 134–142.

[117] Seungku Kim, Seokhwan Kim, and Doo-Seop Eom. "RSSI/LQI-based transmission power control for body area networks in healthcare environment". In: *IEEE journal of biomedical and health informatics* 17.3 (2013), pp. 561–571.

[118] Qi Zhang. "Energy saving efficiency comparison of transmit power control and link adaptation in BANs". In: *2013 IEEE International Conference on Communications (ICC)*. IEEE. 2013, pp. 1672–1677.

[119] Bruce Moulton et al. "Body-area-network transmission power control using variable adaptive feedback periodicity". In: *2010 Australian Communications Theory Workshop (AusCTW)*. IEEE. 2010, pp. 139–144.

[120] Muhammad Alam et al. "Realistic simulation for body area and body-to-body networks". In: *Sensors* 16.4 (2016), p. 561.

[121] Thomas Zasowski et al. "UWB for noninvasive wireless body area networks: Channel measurements and results". In: *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*. IEEE. 2003, pp. 285–289.

[122] Zhen H Hu et al. "Measurements and statistical analysis of on-body channel fading at 2.45 GHz". In: *IEEE Antennas and Wireless Propagation Letters* 6 (2007), pp. 612–615.

[123] DB Smith et al. "Temporal correlation of dynamic on-body area radio channel". In: *Electronics Letters* 45.24 (2009), pp. 1212–1213.

[124] Maddock Meredith, Steve Maddock, et al. "Motion capture file formats explained". In: *Department of Computer Science, University of Sheffield* 211 (2001), pp. 241–244.

[125] *CMU Graphics Lab Motion Capture Database*. Accessed: Jan 20, 2019. URL: `http://mocap.cs.cmu.edu`.

[126] Cheryl D Fryar et al. "Anthropometric reference data for children and adults; United States, 2011-2014". In: *National Center for Health Statistics. Vital Health Stat* 3.39 (2016).

[127] Jian Zhang et al. "Stability of narrowband dynamic body area channel". In: *IEEE Antennas and Wireless Propagation Letters* 8 (2008), pp. 53–56.

[128] Jeroen Lecoutere et al. "Wireless fidelity electromagnetic field exposure monitoring with wearable body sensor networks". In: *IEEE transactions on biomedical circuits and systems* 10.3 (2015), pp. 779–786.

[129] Claudio Castellini and Patrick van der Smagt. "Surface EMG in advanced hand prosthetics". In: *Biological cybernetics* 100.1 (2009), pp. 35–47.

[130] Christian Cipriani et al. "On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction". In: *IEEE Transactions on Robotics* 24.1 (2008), pp. 170–184.

[131] Adil Mehmood Khan et al. "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer". In: *IEEE transactions on information technology in biomedicine* 14.5 (2010), pp. 1166–1172.

[132] Janusz Bryzek. "Roadmap for the trillion sensor universe". In: *Berkeley, CA, April* 2 (2013).