UNIVERSITY OF CALIFORNIA,
IRVINE

Power Optimization for Medical Sensing Systems

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Jun Luan

Dissertation Committee:
Professor Pai H. Chou, Chair
Professor Fadi Kurdahi
Professor Mohammad Al Faruque

2016

DEDICATION

To my wife, Jinglei Lou, and my son, Daniel.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

ACKNOWLEDGMENTS

First of all, I want to take this opportunity to say thank you to my PhD advisor, Dr. Pai H. Chou for being such a great mentor. I am grateful for his guidance and encouragement throughout my PhD study.

Secondly, I want express my gratitude to my committee members, professor Kurdahi and professor Al Faruque. I am honored to have them in my committee. I also wish to send my best regards to all my colleagues in Professor Chou's group. It was my great pleasure to work with such young and talented people. I wish them all the best in future. I would like to give special thanks to Seugnjae Lee for providing part of the experimental data used in this work.

I am truly blessed to have my family around me. I thank my parents, my brother, my wife and my son. Without their help and sacrifice I will accomplish nothing.

Finally, I would like to thank National Institute of Health (NIH) and QT Medical, Inc. for their support to this dissertation. This work was sponsored by the NIH STTR Phase I Grant R41HL112435, 1R41HL127868-01, 1R41GM113463-01A1, 1R41HD079051-01A1, Phase II Grant 2R42HL112435-03 through subcontracts by the QT Medical. The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

CURRICULUM VITAE

Jun Luan

Education

2016 Ph.D., Electrical Engineering and Computer Science, University of California, Irvine, CA, USA
2011 M.S., Computer Science, University of New Mexico, Albuquerque, USA
2001 B.E., Electrical Engineering, Harbin Institute of Technology, Harbin, China

Research Interests

Digital signal processing, embedded system design and low-power system design.

Publications

(1) Jun Luan, Ting-Chou Chien, Seungjae Lee, Pai H. Chou, "HANDIO: A Wireless Hand Gesture Recognizer Based on Muscle-Tension and Inertial Sensing," in *Proceedings of the Global Communications Conference: Selected Areas in Communications: E-Health ('GC' 15 - SAC - E-Health'),* San Diego, CA, USA, December 6-10, 2015.

(2) Jun Luan, Seungjae Lee, Pai H. Chou, "Fast Compressive Sensing Based on Dominant Frequency Estimation," in *Proceedings of the 12th IEEE International Conference on Mobile Ad hoc and Sensor Systems (IEEE MASS) ,* Dallas, TX USA, October 19-21, 2015.

(3) Jun Luan, Seungjae Lee, Pai H. Chou, "Low-Power Detection of Sternocleidomastoid Muscle Contraction for Asthma Assessment and Control" in *Proceedings of the International Symposium on Low Power Electronics and Design, (ISLPED) ,* Rome, Italy, July 22-24, 2015.

(4) Seungjae Lee, Jun Luan, Pai H. Chou, "ECG Signal Reconstruction from Undersampled Measurement Using A Trained Overcomplete Dictionary," in *Proceedings of the 4th International Conference on Convergence and its Application,* Jeju Island, Korea, November 12-14, 2014.

(5) Seungjae Lee, Jun Luan, Pai H. Chou, "New Approach to Compressing ECG Signals with Trained Overcomplete Dictionary," in *Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare (MobiHealth 2014),* Athens, Greece, November 3-5, 2014.

Patents

(1) Jun Luan, "Device for receiving and sending Email", CA 200910118585.

(2) Jun Luan, " Input device", CA 200810184816.

(3) Jun Luan, Zhuo Liu, " Vehicle Electronic Equipment", CA 200610172751.

ABSTRACT OF THE DISSERTATION

Power Optimization for Medical Sensing Systems

By

Jun Luan

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2016

Professor Pai H. Chou, Chair

Medical sensing system collect and analyze the patients' physiological data for monitoring, aid or diagnostic purposes. System designers are faced with stringent requirements on not only correctness and safety but also power. Reference designs and multi-purpose platforms help to significantly shorten the development cycle.

This work takes a cross-layer, system-level, platform-based approach to addressing the problem of saving power in a class of portable medical system. We propose a low-power medical sensing system that can be used to monitor Electrocardiography (ECG), Photoplethysmogram (PPG), and muscle tension. It also includes a hand gesture recognition system to aid mobility-impaired patients.

We explore the theory and application of a compressive sensing framework to medical signal processing. A novel compressive sensing-based ECG compression algorithm and a dominant frequency extraction-based PPG heart-rate calculation algorithm are proposed to reduce the system power. The unique combination of hardware structure and software signal-processing algorithms makes low-power design possible. The system test results

show that the proposed system is superior to existing works in terms of power consumption and system size.

Chapter 1

Introduction

The worldwide shortage of health care facilities and workers has posed great challenges to health care providers and government officials. The remote patient monitoring (RPM) [1] can be a potential solution to this problem and thus has drawn increasing interest as a research subject. Generally speaking an RPM implements a medical sensing system that collects and analyzes physiological data from the patient for monitoring, aid or diagnostic purposes in general. A typical system is shown in Fig. 1.1. The sensing front-end directly samples the signal from the patient using medical sensors and sends the raw or processed data to a mobile device such as a cellphone or a tablet. Depending on the bandwidth requirement, the sensing front-end can use WiFi, Bluetooth or even wired link to transfer the data. The mobile device subsequently sends the data to the back-end cloud server via the Internet. The data is analyzed on the server and the results are directly presented to the doctors. Front-end devices are required to be made small and light so as not to limit the patient's mobility. Usually simple data collection and sensor control are performed in the front end while complex data processing, analysis and profiling are carried out on the more powerful server side.

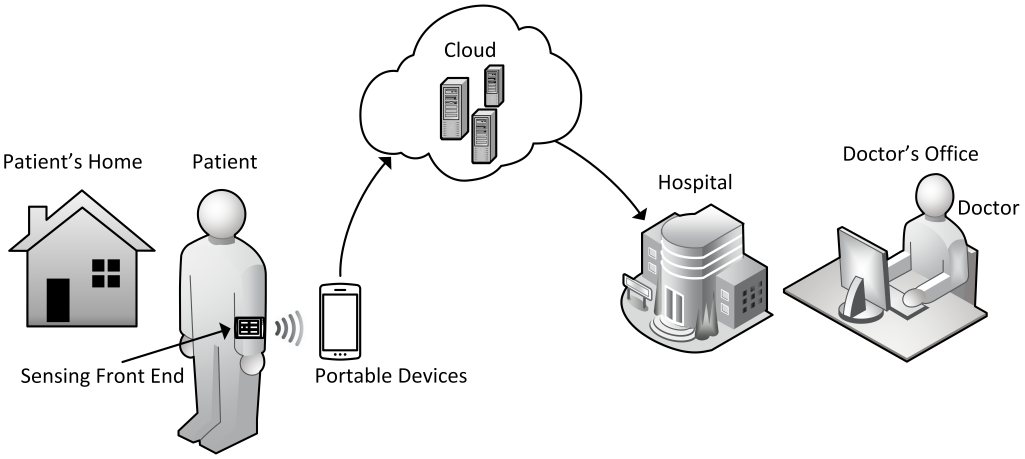


Figure 1.1: Medical Sensing System

Some general consumer electronic systems, such as sports and fitness monitors, can also serve the same purposes and the same technologies may appear in both. But medical grade systems are considered to be more accurate, reliable and responsive as many of these systems are more concerned with the patient's' health and are often used in life critical conditions. In the Untied States, the Food and Drug Administration (FDA) oversees the manufacturing of medical devices, including both software and hardware, to ensure the product quality and patient safety. These strict requirements place a huge pressure on system designers and manufacturers. A research [2] in 2009 shows that the average research and development time for these systems is around two years and some high risk systems take over 3 years. Thus design references and multi-purpose platforms are more important in medical system design as they help to significantly shorten the development cycle.

Designing a medical platform is very challenging as system designers have to make tradeoffs among several important factors: wearablity, responsiveness, development cost and time, etc. Power consumption apparently plays an important role. Large power consumption shortens the battery life, thus limits the wearablity as the battery tends to be the largest component in such systems. Many early generation medical devices are bulky, heavy due to the lack of high-speed low power micro-controllers (MCU) and miniature sensors. They can only be used in a controlled environment, such as research lab and hospitals. On the

other hand, low-power design tends to delay the system as the MCU usually has a simple structure thus slow in computation. To increase the speed, many signal processing and data analysis tasks can be moved to the front end but this will inevitably increase the power consumption and lead to an increased in system size. It is difficult to achieve significant energy reduction by isolated techniques.

This work addresses the ultimate challenge of power saving in medical sensing system design. We propose a low-power medical sensing system that can be used to monitor Electrocardiography (ECG), Photoplethysmogram (PPG) and muscle tension. It can be coupled wirelessly with mobile devices for Telecare or Telehealth services [3]. It also includes a hand gesture recognition system to facilitate mobility-impaired patients. The system block diagram is shown in Fig. 1.2. The detailed hardware and software configuration can be found in Chapter 4 Most of this work focuses on the sensing front end with the exception of a ECG compression algorithm running on the back-end server. The unique combination of hardware structure and software signal processing algorithms make the low-power design possible.

The dissertation is organized as follows: in the rest of this chapter we introduce two important techniques in power optimization followed by an overview of the proposed system featuring both of these two techniques. In Chapter 2, we present a motivation example of designing a heart rate monitor (HRM) to show how to actually optimize the power consumption. In Chapter 3 we provide a background and the related work on this subject. The detailed system design and power saving algorithms are described in Chapter 4 followed by the evaluation Chapter 5. We summarize the research and future topics in Chapter 6.

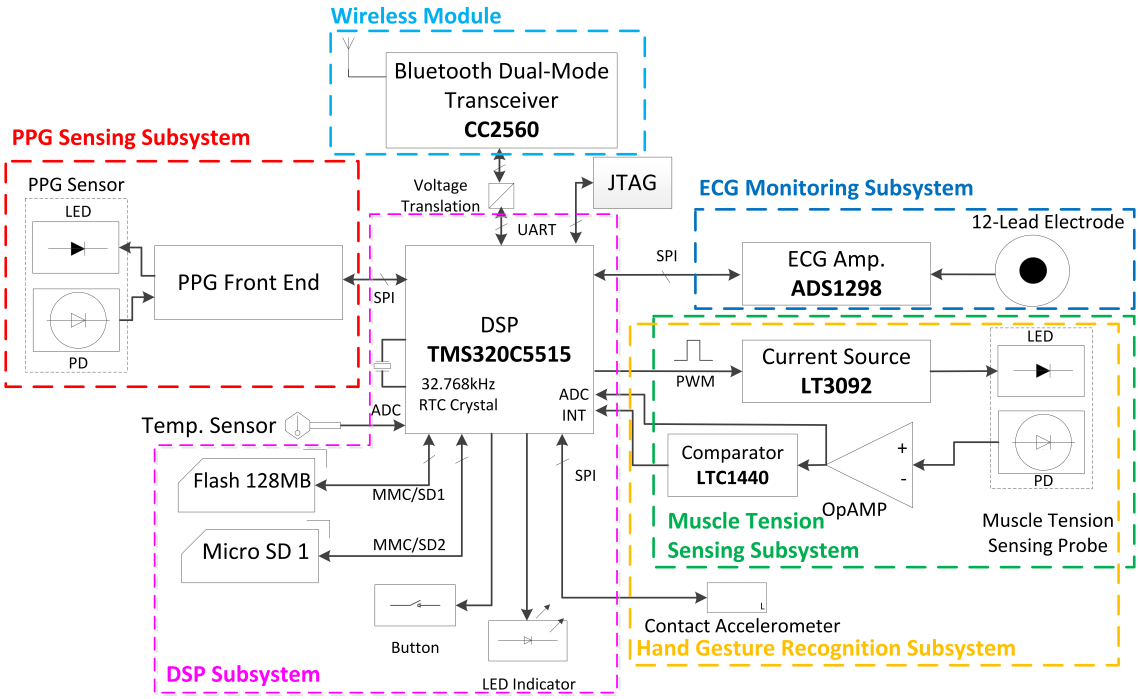


Figure 1.2: System Block Diagram

1.1 Power Optimization

In this section, we introduce two important power optimization techniques for medical sensing systems: low-power component selection and efficient digital signal processing. These two techniques actually help the designers to define the system architecture within the confinement of a certain power consumption budget. Hardware components together with the controlling software can be viewed as a software hardware codesign approach [4], which applied to general electronic systems. For medical sensing systems, however, efficient signal processing stands out as a key factor as it can further optimize the power consumption on top of both the hardware and software. We further explain these two techniques as follows.

Component Selection Component selection defines the hardware structure as well as the corresponding controlling software. Selecting low-power components obviously plays an

important role. In fact, it is often the first task facing the designers. This is a process involving investigating the power characteristic of each component and finding the best among many different options. The system requirement first leads to the medical sensor selection, which has a huge impact on the system power consumption. For example, a heart rate monitor can use either ECG based sensors or optical sensors. The power consumption of a ECG sensor mainly comes from the peripheral circuit such as the ECG amplifier, which can be controlled down to hundreds of microwatts. While the optical sensor utilizes LED which usually takes several miliwatts. This easily makes the optical sensor the most dominant component in the system in term of power consumption.

Most modern medical devices are also equipped with microcontroller units (MCUs). These integrated chips range from low-power MCU to high-end multi-core digital signal processors (DSPs). Low-power MCUs usually have a simple structure running at a low clock rate (MHz to 10s of MHz), thus are low power. They are not suitable to run complex, time-consuming tasks. To select the most suitable component, one also has to consider the software development cost, lead time, etc.

Signal Processing Using the right parts alone does not ensure good power performance. Modern MCUs usually have several power modes. The efficient digital signal processing helps to reduce the power consumption by lowering the sensor sampling rate. The fewer samples collected by the sensor, the less processing required by the software, which in turn enables the MCU to stay in the low-power mode more often. Even though a DSP may consume as much as hundreds of milliwatts when active, the standby power may be just tens of microwatts. Meanwhile, a low-power MCU may take more power if it is kept running actively. In a wireless system, low sampling rate also saves the power used to transfer the data wirelessly, which can be dominant in a low-power system. However, less signal samples usually leads to a more complicated processing algorithm, which also

impacts the power consumption. Clearly a balance has to be found between the complexity of the algorithm and the sampling rate.

1.2 System Structure

In this chapter, we introduce the system structure of the proposed low-power medical sensing platform that features both low-power component selection and efficient digital signal processing. As shown in Fig. 1.2, the system has an ECG sensor, a Photoplethysmogram (PPG) sensor [5], an optical muscle tension sensor, and an triaxial accelerometer (ACC). Depending on the requirements, some sensors can stand alone or be removed from the system. We divide the whole system into different subsystems to better present the system architecture.

1.2.1 Digital Signal Processor Subsystem

The main processor in this system is an embedded low-power DSP, TMS320C5515, or C5515 for short. The C5515 is a fixed point DSP running at 100MHz consisting of a general MCU core and a Fast Fourier Transform (FFT) co-processor. The reason we favor C5515 than other low-power MCUs or DSPs is that it strikes the most balance between processing speed and power consumption. In our medical sensing system, many signal processing tasks need to be done on the front end, such as real-time signal filtering, index computation, template matching, etc. These tasks cannot be done in real time by most of the low-power (i.e. low speed) MCUs. The average active power consumption of C5515 is only around tens of miliwatts. While an ARM cortex A processor would take hundreds of miliwatts to several watts of power [6]. For some of tasks requiring FFT, the hardware FFT acceleration makes C5515 run much faster than just a high speed MCU. Except for

the signal processing tasks, the DSP also runs the Bluetooth Protocol stack to communicate with the CC2560, which is a dual-mode Bluetooth (i.e., BR/EDR, also known as "classic", and low-energy) transceiver. The DSP has the option to either transmit the ECG and PPG data via BR/EDR or store the data locally to an external flash memory.

1.2.2 Muscle Tension Sensing Subsystem

Muscle tension sensing provides means to monitor the patient's movement. It cab also be used as a control interface to aid mobility-impaired patients. We select the optical muscle tension sensor (MTS) for its low power consumption and wearability. It consists of an LED-PD pair. The LED emits light into human tissue while the PD measures the backscattered light from human tissue. It is a fairly complicated process since the muscle movement has a triple effect: the muscle fiber contraction changes both the light absorbance and the reflected light path, while the blood volume in the muscle also affects the reflected light. As a result, muscle movements manifest as changes to the signal frequencies and amplitude in the PD output. The key to MTS is the low-power control algorithm carried out on the DSP. The DSP also samples the signal using the on-chip ADC and adds filtering before running the contraction detection algorithm as described in Section 4.2.1.2.

1.2.3 Hand Gesture Recognition Subsystem

The HGR subsystem fuses MTS and ACC signals. Both sensors are tiny in size. The add-on of MTS enables the system to detect wrist movements that makes it possible to detect a variety of common gestures. We propose a template matching based recognition algorithm for its low complexity and high accuracy. The sampled MTS and ACC signals are compared with the presets in the DSP memory and each is given a score based on the dynamic time warping (DTW) [7]. The gesture with the highest score is selected. The core

sensor fusion algorithm is presented in Section 4.2.2

1.2.4 ECG Monitoring Subsystem

ECG is one of the most important and basic types of physiological signal for cardiac diagnosis and analysis. The ECG sensor is a 12-lead electrode as shown in Fig. 5.15. The sampled signal is sent to ADS1298R [8], an 8-channel front-end chip integrated with a 24-bit ADC. The sampled ECG raw signal is processed on the back-end server instead of the sensing front end. This demands a high bandwidth to transfer the ECG data and huge server storage to save the data. For the doctors to make quick diagnoses, we propose a compressive sensing based ECG compression algorithm running on the back-end server to save both the bandwidth and storage. The algorithm is presented in Section 4.2.4.

1.2.5 Photoplethysmogram Monitoring Subsystem

The system can monitor PPG signal. PPG is the collection of vital signs from pulse oximetry. The most common ones are heart rate and blood oxygen level (SpO2). PPG monitors are commonly seen in intensive care, elderly care, sports and fitness, newborn screening, and other medical applications. A more detailed introduction of PPG can be found in Appendix B. A commercial pulse oximeter consumes 55-120 mW, most of which is consumed by the LEDs [9]. Compressive sensing (CS) [10] can directly save the LED power by reducing the LED on-time, thus making it a natural candidate. Here we propose a two-step algorithm consisting of prior estimation and Compressive Sensing Matching Pursuit (CoSaMP) [11]. The algorithm can be found in Section 4.2.3. Our algorithm can only report heart rate from PPG signal at this point but shows promising results in estimating SpO2 as well.

1.3 Contributions

There are several contributions of this work. First, the proposed framework can be used as a reference design for medical system designers for fast prototyping and product development. Chip manufacturers usually provide specific libraries and applications for development engineers to start with but the power optimization techniques and combined solutions are rarely found. The hardware and software provided in this work can serve as a power optimization guideline.

Secondly, this work contributes to the muscle tension sensing research field. Traditionally, the muscle tension is measured by surface Electramygraphy (sEMG). It is bulky, power consuming and barely wearable. The novel optical sensor design and low-power control greatly reduce the system size and overcome several shortcomings of sEMG, making long-time monitoring possible.

Thirdly, in the compressive sensing research area, the proposed dominant-frequency exaction based fast compressive sensing greatly increased the system response speed. We also utilize CS theories in real applications such as ECG and pulse oximetry.

Finally, we present a novel miniature hand-gesture recognition system utilizing both inertial and muscle tension sensing. By the fusion of the optical MTS and inertial sensors, the system size can be greatly reduced while still maintaining a high accuracy.

Chapter 2

Motivating Example

System power can be greatly reduced by low-power component selection and advanced signal processing. In this chapter, we present a motivating example of designing a wireless heart-rate monitor (WHRM) to explain how to utilize these power saving techniques. We show the hardware and software design, signal processing algorithms, and an evaluation.

2.1 Background

WHRM is a growing class of wearable devices for both personal health and fitness applications. The most popular type is worn under the shirt in the form of an elastic chest strap with two wet electrodes that contact the skin directly. Similar to ECG, a heart-rate monitor (HRM) also measures the electrical impulses generated by the polarization and depolarization of cardiac tissue; but unlike ECG, an HRM does not record raw data. Instead, it analyzes the waveform, extracts the heart beats over time, and transmits the running average heart rate within the given time window. By transmitting the heart rate data to the user's smartphone wirelessly, the smartphone can use the data in several possible ways, including

not only uploading the data to a cloud database but also providing musical feedback. For instance, the app can include a music player that adjusts the musical tempo or other audio signals to help the user maintain an optimal level of physical activity.

The most important metrics for evaluating a WHRM include the battery life and accuracy of heart-rate detection. The battery life is clearly important, because recharging or replacing the battery causes additional trouble to the users. Although one may attempt to target the most power efficient MCU and radio trans-receiver, in practice the opportunities are limited, because the radios such as ANT+ or Bluetooth 4.0 Low Energy Technology (BLE) are already very low power, and they are required to implement industry-standard profiles that prescribe the command format and rate. Therefore, system designers must seek opportunities in other subsystems.

The most fruitful areas are in the analog-to-digital converter (ADC) sampling subsystem and the heart-rate detection algorithm. An ADC consumes a significant amount of power when in operation. Moreover, the more data collected by the ADC, the more processing is required by the algorithm.

A heart-rate detection algorithm detects the presence of heat beats, records the timestamps, and calculates the overall heart rate. A heart beat is defined by an *ECG complex*, which is composed of 5 waves: P, Q, R, S, and T, as shown in Fig. 2.1. Q-wave and S-wave usually go downward. Together with R-wave, they form a *QRS complex*. The duration of QRS complex is usually 60 ms-120 ms. A longer duration may indicate a heart condition [12]. A good algorithm also needs to consider the other segments, such as the QT interval, to avoid false detection. The QT interval begins with the onset of Q wave and ends in the offset of T-wave. It is usually around 400 ms. A prolonged QT interval may increase the risk of sudden cardiac death [12].

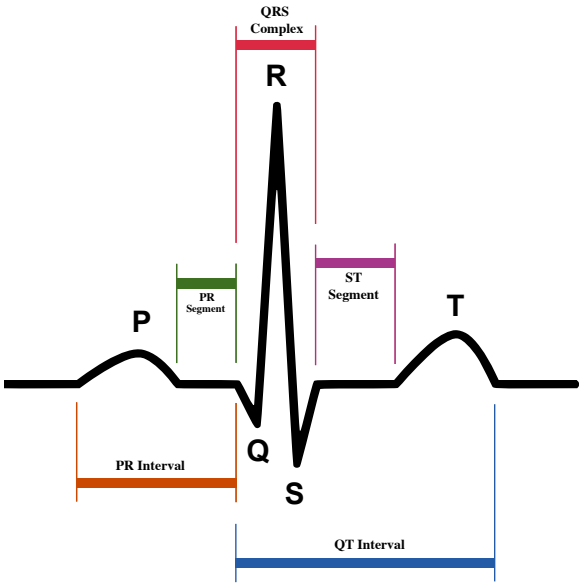


Figure 2.1: ECG Complex. Image from Wikipedia.

2.2 System Design

In this section, we introduce the system design of WHRM . As stated in Chapter 1, low-power component selection and efficient signal processing greatly reduces the system power consumption. Using WHRM as a real example, we further explain how to utilize these techniques.

2.2.1 Component Selection

The block diagram of a typical WHRM is shown in Fig. 2.2. It includes an MCU, an ECG front end, a low pass filter (LPF) and a wireless communication module. We use MSP430 [13] in this design for its low power consumption. It can have as many as five modes of operation. Low power mode 3 (LPM3) [13] can be used as the standby mode because the processor can be woken up periodically to detect the strip connection. An alternative would be to use a hard key to wake up the MCU, as this would allow the MCU to stay in power-off mode, which consumes much less power by turning off several modules

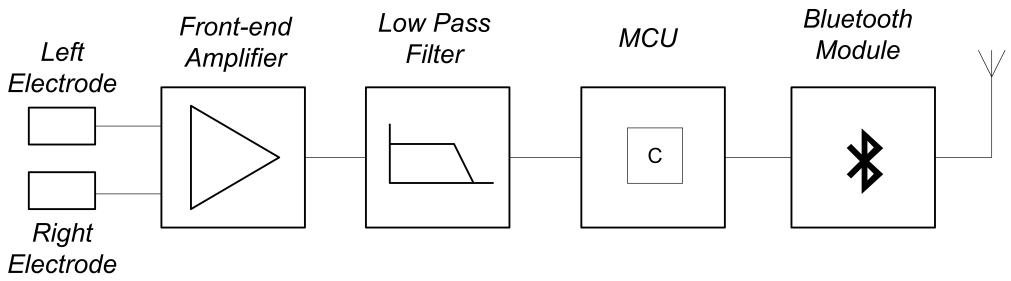


Figure 2.2: WHRM Block Diagram

that are active in LPM3, including the low power oscillator and the real time clock (RTC). However, it also increases the hardware cost and complicates the mechanical design as WHRM usually requires water resistance but has only a limited surface area. MSP430 also has a built-in ADC that can be used to sample the ECG signal.

The front-end ECG amplifier and the LPF block should also consume as little power as possible. The amplifier is a differential amplifier that amplifies the ECG signal picked from the electrode while rejecting common-mode noise. The signal can also be further amplified in the subsequent LPF stage. The cutoff frequency of the LPF is usually set to around 16 Hz to filter out the environmental noise and muscle noise, which are usually in the 50-60 Hz range. Another option is to implement an FIR or IIR digital filter in the MCU. However, the analog filter has near-real-time response speed while reducing the burden on the MCU.

The most common power source is the rechargeable Lithium battery that can be charged through a standard USB port. The battery output voltage is 4.2 V. A buck-type DC-DC converter is used to convert the battery voltage down to the required voltages to power the MCU, the analog frond-end and the wireless module. A main concern is that the DC-DC noise from the power block could become a continuous source of interference to the ECG signal and thus needs special attention. In the actual PCB routing, the ECG signal path is well separated from the DC-DC converter, and the two ECG input channels are very carefully balanced to avoid noise amplification.

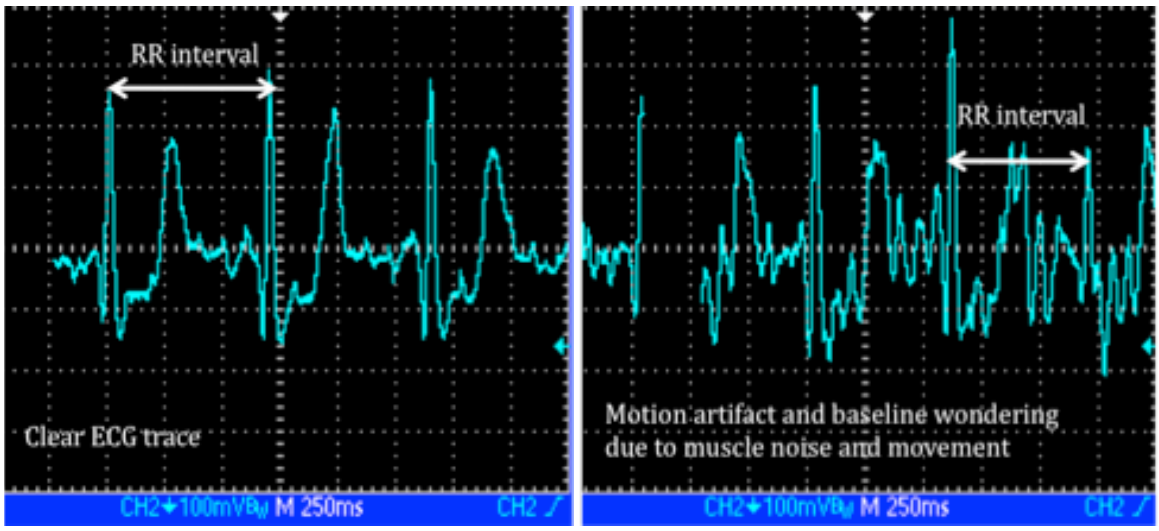


Figure 2.3: Observed ECG signal. Left: standstill; Right: in motion.

2.2.2 Signal Processing

The most common way to detect a heart beat from the digital samples is to detect the R wave peak, which is the most dominant signal in an ECG complex. To achieve real time performance, the peak detection can be only carried out to a small amount of buffered samples. This causes a serious problem when the sampling rate is not high enough. Fig. 2.3 shows the ECG signal in both standstill state and in motion. The waveforms are taken after the LPF in a real WHRM prototype. Significant motion artifact and muscle noise can be observed even after the smoothing and filtering while in motion. If the ECG signal is sampled at a relatively low rate, it becomes easy to be trapped into local maximum instead of the real R-wave peak.

Increasing the sampling rate also increases the power consumption, which further strains our already limited power budget. Adaptive sampling can help to satisfy the power constraint and meet the power specification. Saving ADC operations not only saves the power to operate ADC but also gives the processor more sleep opportunities. For example, if the sampling rate is reduced from 128 Hz to 40 Hz, 88 ADC operations are save per second. Each ADC operation takes about 4μs to complete (1μs for sample and 3μs for conversion).

The typical ADC power consumption is around 450μw. Total energy saved for 1 hour is:

$$450\,\mu\text{W} \times 4\mu\text{s} \times 88 = 0.16\text{mJ} \tag{2.1}$$

Meanwhile, the processor needs to stay in active mode to access the data. The current consumption is 1.7 mA at 16 MHz [13]. The interrupt subroutine is measured as around 15-20 μs as entering and exiting the interrupt routine take over 10 cycles and data processing requires at least another 10 cycles. Energy consumed per sample is

$$1.7\,\text{mA} \times 3.3\,\text{V} \times 0.6\,\mu\text{s} \times 14\,\text{cycles} = 47.12\,\text{nJ} \tag{2.2}$$

Energy saved per hour is

$$47.12\,\text{nJ} \times 88/\text{s} \times 3600\,\text{s/hr} \times 1\,\text{hour} = 14.92\,\text{mJ} \tag{2.3}$$

The total energy saved in 1 hour is around 15 mJ. This amount of energy may be small compared to the whole capacity of the battery, but the system will stay in standby mode most of the time. The average power consumption is measured to be around 0.3 mW in heart rate detection mode and 0.3 μW in standby. The amount of energy saved in 1 hour by adaptive sampling can support the whole system in standby mode for more than 40 minutes. If the original system works 4 hours and sleeps 12 hours, using our algoirhtm the system can last at least 4 hours more in standby mode. This is just a rough estimate with a minimum weight but it shows a great potential in saving power by adaptive sampling.

2.3 Heart Rate Detection Algorithm

We use a digital signal processing algorithm that assesses the noise level in the (2-lead) ECG signal to steer the sampling rate as a way to save a significant amount of power while maintaining high accuracy of heart-rate detection. This algorithm involves a double-threshold QRS complex detector and a mechanism for sampling-rate adjustment. In the QRS detector, all the local extremes are detected and compared with two sets of threshold: a signal amplitude threshold and a slope threshold. An R-wave peak point is identified only when the local extreme goes beyond both thresholds. The underlying rationale is that R-wave usually has the dominant peaks in an ECG complex, but amplitude alone is insufficient for the detection. Slope information serves as a good complement and discriminator from other high peaks such as T-wave peaks.

On top of this algorithm, we introduce an adaptive technique that dynamically adjusts the sampling rate in response to severe noise such as baseline wondering and muscle noise. Three sampling rates of 128 Hz, 64 Hz, and 40.8 Hz are implemented on the WHRM prototype. Our algorithm greatly reduces the power consumption by lowering the sampling rate while still maintaining a reliable detection by sampling the signal at a higher rate in the presence of noise. The performance of the algorithm is evaluated using standard MIT-BIH arrhythmia database [14, 15]. Our algorithm can also be used in other applications that require peak detection such as pulse oximeters.

2.3.1 Bluetooth Stack

Besides heart rate detection the system also needs to handle the wireless communication. System level coordination among all the tasks is necessary to achieve the lowest power consumption. To simplify the design, we select the Bluetopia protocol stack [16] for the

MCU to control the wireless module. It supports three communication protocols: ANT+, BLE, and Bluetooth Classic (BR/EDR). Industry-standard profiles such as SPP (serial port profile) and GAP (generic access profile) are already implemented. The WHRM works as a slave server while the mobile device (smartphone) works as a master client. The *slave latency*, which is defined to be the number of connection events that the slave is allowed skip, is set to the maximum so that it can remain in low-power mode if no heart rate update is needed. The stack also runs a non-emptive scheduler. Every task is literally a call-back function. No real context switching is needed upon scheduling. This greatly reduces the overhead.

The heart rate detection task (HRDT) performs ECG sampling, peak detection, and heart rate calculation. When scheduled, HRDT will set the watchdog timer (WDT) to the desired sampling rate and goes to LPM3. After the timer expires, HRDT will set the ADC to sample ECG signal and go back to LPM3 again. Upon ADC conversion, the ECG sample is then stored into a local buffer. Each HRDT takes only one sample at a time, and it is repeatedly scheduled until a certain number of samples have been accumulated.

Peak detection will be performed inside the local buffer by HRDT. The time of each R-wave peak will be calculated using the sampling timer and the local buffer index number. A heart rate then is calculated on three consecutive peaks. The system is designed to report the running-average heart rate every second. However, if the heart rate has not changed or has not been updated, the protocol stack can choose to ignore the request from the client and stay in sleep mode. The peak detection algorithm is described next.

2.3.2 Peak Detection Algorithm

We use the threshold subtraction algorithm proposed by [17] for its low complexity and simplicity. It can be easily incorporated into our adaptive mechanism because both the

slope and amplitude threshold comparisons are identical even under different sampling rates. As discussed in the latter part of this section, all the parameters are set in terms of time rather than number of samples. This greatly simplifies the sampling rate adjustment. The core of this algorithm only takes one comparison, one addition, and one multiplication. The downside of this algorithm is that it relies too heavily on the signal filtering and smoothing. It is simple and works well for stable ECG signals but cannot catch the complex in the presence of noise, or the complex also happens to have a high T-wave peak. The underlying reason is that it only extracts amplitude information, but amplitude alone is incapable to discriminate R-wave peaks from dominant T-wave peaks. We added slope detection to this algorithm to improve detection rate.

The proposed peak detection algorithm works as follows.

Step 1): Local extreme detection

A total of 180 ms of ECG data is buffered and 60 ms of the total data is copied to the beginning of the buffer from the previous frame. The buffer holds about 22 samples in 128 Hz sampling rate and only 6 samples in 40.8 Hz sampling rate. The sampling *interval* rather than the sampling *rate* will be recorded for each frame. When a change happens, the boundary of overlapped samples will be marked so that the timer interval between any two adjacent samples can be easily calculated. All the samples in the local buffer are then subject to a local extreme search. A sample is marked as a local extreme if it is a local maximum or a local minimum within a 60 ms interval of consecutive samples. Every sample taken between 31 ms-150 ms will be compared with 30 ms of the samples on its right hand side (later samples), and 30 ms on its left-hand side (earlier samples). Once an extreme is found, the samples within its immediate 30 ms of neighbors can be ignored. This step can also be implemented using a divide-and-conquer algorithm, but for such a small number of samples, the exhaustive way works fast enough as it takes only 6 comparisons at most for each sample and the lower the sampling rate, the faster it runs.

Step 2): Slope Detection

Every local extreme is subjected to a slope measurement. The slope is measured using the following equation.

$$slope = \frac{|X[30\,\text{ms}] - X[0]| + |X[-30\,\text{ms}] - X[0]|}{2} \tag{2.4}$$

where $X[0]$ is the local extreme, $X[30\,\text{ms}]$ the sample taken 30 ms after the extreme, and $X[-30\,\text{ms}]$ the sample taken 30 ms before the extreme.

A slope threshold is maintained as a weighted sum of the previous threshold and the current slope. Any local extreme whose slope is less than the threshold will be discarded.

Step 3): Amplitude Subtraction

The amplitude is calculated as the absolute value of the difference between the local extreme and bias voltage and further compared with a peak amplitude threshold. If greater than the threshold, the local extreme will be identified as a R-wave peak. The threshold is maintained using the approach proposed in [18]. Instead of just one set of threshold of the signal peak value (SPT), a noise peak value (NPT) is also maintained as a weighted average of the current extreme and the previous NPT. The final threshold is a weighted sum of SPT and NPT.

2.3.3 Sampling Rate Selection

We use WDT to trigger ADC operations as mentioned in Section 2.3.1. It is the only timer resource available to us in LPM3. With only one external 32-kHz crystal oscillator, WDT duration [13] can be set to 1.95 ms, 15.6 ms, or longer. The normal duration of QRS complex is 60 ms-120 ms [12]. This requires the sampling rate to go beyond 33 Hz. The

sampling rate is decided based on our test result. The possible combinations of timers are tested in simulation using the MIT-BIH arrhythmia database as shown in Fig. 2.3. No noticeable drop of detection rate is observed. As a result, three level of sampling rate are selected as 128 Hz (7.8 ms timer), 64 Hz (15.6 ms timer) and 40.8 Hz (24.5 ms timer).

2.3.4 SNR Measurement

Though the noise peak threshold is maintained during peak detection, this threshold cannot be used in signal-to-noise ratio (SNR) calculation since it contains the information of T-wave peaks. In some human ECG complex, the T-wave component can also be relatively high even after the smoothing filter. These T-wave peaks are detectable and can be removed by comparing with R-wave peaks. However, if it is counted as a noise peak, the sampling rate might stay at the highest level. For this reason, only the peaks that happen after 360 ms and identified as noise peaks can be taken into SNR calculation. An adjusted noise peak threshold (ANPT) is thus calculated as a weighted average of all the noise peaks detected after an R-wave peak. Unfortunately, this mechanism may not work for some ECG complex with extremely long QT interval. Long QT interval syndrome is a rare heart condition that may lead to sudden death. Detection of this type of QRS complex is out of the scope of this work. The final SNR is calculated as the ratio between SPT and ANPT.

2.3.5 Sampling Rate Adjustment

Different sampling rates are switched based on the measured SNR and RR interval. Sampling rates are controlled using the following lookup table.

Sampling Rate	40.8 Hz	64 Hz	128 Hz
SNR	$\leq 5\%$	$\leq 20\%$	$> 20\%$

If RR interval is less than 400 ms (corresponding to a heart rate of 150 bpm), the sampling rate will be adjusted to 125 Hz to avoid missing detection because the QRS complex becomes shorter than the usual fast heart rate.

2.4 Evaluation

The detection rate of our algorithm is evaluated using standard MIT-BIH arrhythmia database [19]. The database contains over 100,000 QRS complex taken from 48 patients. However, one problem is that our algorithm, like all other algorithms, cannot detect some abnormal QRS complexes. Our application only supports the heart rate up to 180 bpm, which corresponds to 330 ms RR interval. For example, in data tape 203, severe noise is observed in the middle of the tape and hundreds of RR intervals go much shorter than 330 ms. Irregular QRS complex is also observed in tape 228 as some of R-wave peaks have far greater amplitude than the others. This makes it difficult to find a proper threshold, and as a result, several beats are missing. To avoid these problems, we carefully picked up a small chunk of data that has 30 beats for each patient and evaluate our algorithm. This covers the variety of QRS complexes while avoiding cases caused by irregular beats.

2.4.1 Sampling Rate Study and Peak Detection Test

As mentioned in the previous section, different sampling rates are evaluated. Table 2.1 shows the results of 128 Hz, 64 Hz, and 40.8 Hz detection rates. The analog filter effect is simulated using the MATLAB digital filter. The original data is taken at 360 Hz sampling rate but is down-sampled by 3, 6, and 9 times to simulate our 128 Hz, 64 Hz and 40.8 Hz sampling rates, respectively. The detection result is compared with the database annotation and false positive peaks and false negative peaks are counted to the overall false detection

Table 2.1: Detection rate at 128 Hz sampling rate

Rec. #	QRS cpx	Avg HR	Avg SNR%	120 Hz FP/FN		60 Hz FP/FN		40 Hz FP/FN		Adapt. FP/FN	
100	35	73	1.54	0	0	0	0	0	0	0	0
101	35	59	3.84	0	0	0	0	0	0	0	0
112	35	73	8.18	0	0	0	0	0	0	0	0
103	35	67	5.55	0	0	0	0	0	0	0	0
104	35	74	13.11	0	0	0	0	0	0	0	0
105	35	95	18.90	0	0	0	0	0	0	0	0
106	35	60	15.03	0	0	0	0	0	0	0	0
107	35	70	10.40	0	0	0	0	0	0	0	0
118	35	55	2.74	0	0	0	0	0	0	0	0
109	35	85	3.85	0	0	0	0	0	0	0	0
111	35	69	13.49	0	0	0	0	0	0	0	0
112	35	86	3.81	0	0	0	0	0	0	0	0
113	35	62	2.89	0	0	0	0	0	0	0	0
114	35	60	8.76	0	0	0	0	0	0	0	0
115	35	63	2.66	0	0	0	0	0	0	0	0
116	35	82	1.79	0	0	0	0	0	0	0	0
117	35	51	15.51	0	0	0	0	0	0	0	0
118	35	72	2.08	0	0	0	0	1	0	0	0
119	35	70	2.31	0	0	0	0	0	6	0	0
121	35	58	6.25	0	0	0	0	1	0	0	0
122	35	79	0.37	0	0	0	0	0	6	0	0
123	35	50	5.77	0	0	0	0	0	0	0	0
124	35	53	1.05	0	0	0	0	0	0	0	0
200	35	101	0.26	0	0	0	0	0	0	0	0
201	35	54	0.56	0	0	0	0	0	4	0	0
202	35	54	5.11	0	0	0	0	0	2	0	0
203	35	100	18.49	0	0	0	6	0	10	0	0
205	35	89	1.28	0	0	0	0	0	0	0	0
207	35	61	5.49	0	0	0	0	0	0	0	0
208	35	107	6.69	0	0	0	1	0	7	0	0
209	35	111	11.12	0	0	0	0	0	0	0	0
210	35	91	10.92	0	0	0	0	0	0	0	0
212	35	91	7.68	0	0	0	0	0	0	0	0
213	35	106	8.72	0	0	0	0	0	0	0	0
214	35	74	1.00	0	0	0	0	0	0	0	0
215	35	111	11.72	0	0	0	0	0	0	0	0
217	35	71	0.71	0	0	0	0	0	0	0	0
219	35	79	12.55	0	0	0	0	0	0	0	0
220	35	64	4.40	0	0	0	0	0	0	0	0
221	35	84	12.26	0	0	0	0	0	0	0	0
222	35	72	3.12	0	0	0	0	0	0	0	0
223	35	82	2.02	0	0	0	0	0	0	0	0
228	35	64	8.85	0	0	0	0	0	0	0	0
230	35	74	3.62	0	0	0	0	0	0	0	0
231	35	63	13.10	0	0	0	0	0	0	0	0
232	35	58	4.84	0	0	0	0	0	0	0	0
233	35	101	23.51	0	0	0	0	0	0	0	0
234	35	90	0.09	0	0	0	0	0	0	0	0
Total False Detection				0	0	0	7	1	39	0	0
False Detection Rate				0%		0.42%		2.38%		0.00%	

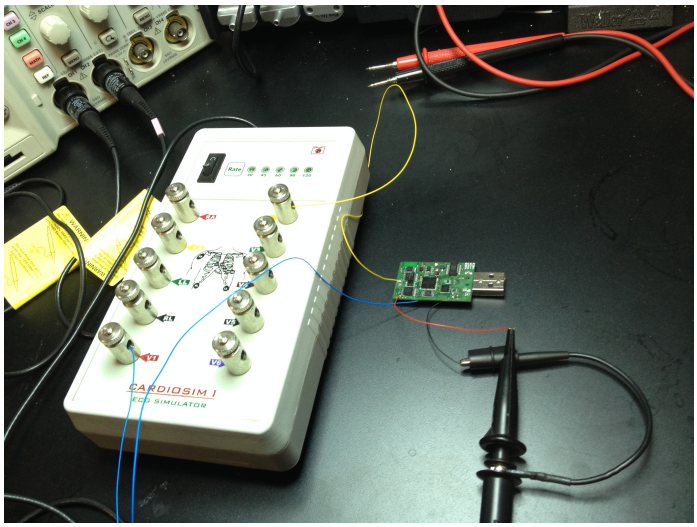


Figure 2.4: Prototype Test Environment

rate. This is a similar test approach in [20].

The trend is quite obvious that a high SNR does not necessarily leads to false detection but increases the false detection in 40.8 Hz sampling rate as shown in tape 203 and tape 105. The proposed algorithm sets the SNR for 40.8 Hz to be under 2% to decrease the misdetection rate. Even at 40.8 Hz sampling rate, our algorithm can still detect around 98% of the QRS complex. This is good enough for our application since we are not pursuing the absolute detection rate required by medical grade device.

2.4.2 Real Prototype Test

A prototype is implemented using MSP430 MCU. The accuracy of the prototype is tested using the heart rate simulation device CARDIOSIM I as shown in Fig. 2.4, which has five levels of heart rate ranging from 30-120 bpm. Our WHRM prototype tested with an iPhone4S and a Samsung Glaxy 3. In both cases, it can read all the levels accurately and is able to pickup the heart rate within a maximum of 3 seconds.

2.5 Discussion

In this section, we show that a noise adaptive algorithm for heart rate detection can help to greatly reduce the power consumption while maintaining the accuracy. We also present a WHRM prototype that implements this core algorithm. Both our simulation and system test shows that the WHRM meets the expectation and can effectively reduce the power consumption. The power reduction may seem small in numbers. This is because the baseline power consumption is large. However, our algorithm can be used in other devices that also require periodic peak detection.

The heart rate detection algorithm utilizes adaptive sampling to reduce the sampling rate. It is tempting to further cut down the sampling rate by precisely locating the signal segments of interest. However, this will inevitably introduce extra complexity into the system, thus narrowing down the error margin. This leads us to consider the more advanced compressing sensing algorithm utilizing random sampling. We will introduce it in Chapter 3.

Our design experience show that power optimization for medical sensing systems is a complex process that involves both component selection and digital signal processing. It is difficult to achieve significant energy reduction by isolated techniques.

Chapter 3

Background and Related Work

This section reviews the background and the related work on three technologies: muscle tension sensing, hand-gesture recognition and compressing sensing in physiological signal processing. We briefly discuss the limitations of the related work and how our proposed system may overcome them. Each related work only focuses on one aspect of the medical sensing. A contribution of this work is a fully integrated heterogeneous system, which has not been published in the literature to the best knowledge of the author.

3.1 Muscle Tension Sensing Technology

This section reviews two technologies of muscle tension: surface Electromyography (sEMG) and optical muscle tension sensing. The state-of-the-art in muscle tension sensing uses sEMG sensors. We first give an overview of sEMG and analyze its drawbacks. In comparison, we also introduce optical MTS as a promising replacement of sEMG.

3.1.1 Surface EMG

EMG measures the action potential generated by the contraction of muscle fiber, while sEMG replaces the needle-type electrodes with the non-invasive surface type, which is more comfortable to wear. sEMG is commonly used in medical sensing field for its high sensitivity. A typical sEMG system is shown in Fig. 5.1.

One significant drawback of sEMG is the power consumption. The state-of-the-art sEMG uses a precision instrumentation amplifier (in-amp) [21]. The in-amp is a special type of differential amplifier that adopts an input buffer to ease the impedance matching with the preceding stage. In-amps with low noise, low offset, and low drift are classified as precision in-amps. Commercially available precision in-amps consume milliwatts of power as shown in Table 5.1.

Another drawback is the bulkiness of the system due to the use of electrodes. Many applications require a minimum of three electrodes, two of which on the target muscle to measure the differential signal and a third ground electrode serving as the reference point for the precision in-amp. This ground electrode should be put on a bony or non-muscular part of the body as far as possible from the target muscle in order to generate a stable reference. Electrodes are big in size. Commercial electrodes are usually over 20 mm in diameter.

A third drawback is that an extra step of skin preparation is needed to get good signal quality. The purpose of skin preparation is to add moisture between the skin surface and the electrode to increase the input impedance of the in-amp. The general procedure includes cleaning and rubbing the skin, hair removal, and applying conductive gel for better contact. In fact, pre-gelled electrodes are the most often used in sEMG [22], but the gel can cause skin reaction and drying over time, making it less suitable for long-time monitoring.

3.1.2 Optical Sensor

The optical sensor requires no skin preparation and can be comfortably worn as long as it is put close enough to the skin surface. It utilizes an LED to emit light into human tissues and one or multiple PDs to measure the transmittance or reflectance of the light. The optical sensor has been widely used in pulse oximetry to obtain photoplethysmogram (PPG) [23]. Many health-critical physiological indices can be extracted from PPG, such as heart rate, blood oxygen concentration (SpO2), respiration rate, and more. An optical probe can be potentially made in tiny size using a miniaturized LED-PD combo chip [24].

Using optical sensors to monitor muscle activity is a relatively new research area. It is shown in [25] that optical sensors can differentiate between isometric and isotonic contraction and in work [26, 27], and similar optical sensors are used to monitor the upper limb movement for controlling prosthetic limbs. In all these works, the optical signal is only used for on-off control. Power consumption and power saving techniques have not been addressed.

Traditionally, active optical devices are considered high power consumption because of the LED. Pulse oximeters usually consumes 55-120 mW, most of which is by the LED [9]. To reduce power for long-term monitoring, some researchers resort to compressive sensing to decrease the sampling rate [28, 29]. Compressive sensing, however, is not suitable for this purpose, because it usually requires a powerful back-end computing unit to perform signal reconstruction [30] and the system often suffers from the subsequent delay. In this work, we design a low-power standalone system with real-time performance. We show that by careful selection and control of the sensor, the power consumption can be optimized without complicated high-level techniques.

3.2 Hand Gesture Recognition System

Most wearable HGRs to date perform inertial sensing (i.e., acceleration and rotation) as the primary modality for acquiring gesture data. Among them single ACC-based hand gesture recognition has been well studied. One advantage of ACC is its availability on many off-the-shelf portable devices. Examples include TI Chronos Watch [31], Nintendo Wii remote controller, and virtually all smartphones. A digital ACC consumes less than 1 mW in active mode. Acceleration data can be easily processed locally or transferred through the wireless interface.

On the software side, simple filters such as low-pass filter (LPF) or median filter [32] are usually used to smooth the signal before using either feature-based or template-based detection algorithms to detect a gesture. Feature-based methods use machine-learning techniques such as Hidden Markov Model (HMM) or Support Vector Machine (SVM) [33] to classify the signal according to the extracted features. High accuracy is reported [34] but a large training set is required to ensure high detection rate. In template recognition, a widely used technique is dynamic time warping (DTW). It can get started with only one template for each gesture [35]. The signal is usually divided into fixed-length segments, and dynamic programming is used to find the best matching subsequence with the preset templates. The complexity of DTW is $O(pST)$ with S being the segment size, T the template size, and p the number of preset templates. It can be easily implemented on a mobile platform such as a smartphone and many modern MCUs.

To overcome the limitations of single ACC-based systems' inability to detect wrist movements, multiple previous works suggest to couple sEMG with ACC. In [36, 37], HMM, K-means clustering, and decision fusion are used to fuse the multi-channel sEMG signal with ACC. A set of more than 18 gestures can be detected with a detection rate of over 90%. However, as mentioned in Section 3.1, a multi-channel sEMG can be uncomfortable

to wear. The closest one to our system is [38], which uses a single-channel sEMG with three electrodes 30 mm in diameter. The muscle contraction signal from sEMG is used as on-off control and only five gestures can be detected. It is made wearable with a much smaller size than the other two but still an order of magnitude larger than our proposed system and cannot solve the electrode problem. Therefore, ACC-sEMG-based systems are only suitable for a controlled environment.

New sensing technologies, especially low-power wearable sensors, provide potential solutions to this problem. Optical sensor-based MTS is very promising in replacing sEMG in this application due its size and wearing comfort. Our study also shows that much detail on the muscle movement are provided even in a single-channel MTS. We will show how to utilize the MTS signal in HGR in Chapter 4.

3.3 Compressive Sensing in Physiological Signal Processing

This section gives a brief of review of compressive sensing. Compressive sensing naturally combines sampling and compression in one step saving both the front-end power and transmission bandwidth. We start with the basic theory followed by an introduction to the existing CS applications in physiological signal processing.

3.3.1 Basic Theory

A discrete time-frequency signal of N samples can be represented as

$$T = D^{-1}(N)X$$

where T is the time-domain samples, X is the Discrete Cosine Transform (DCT) coefficient and $D^{-1}(N)$ is the inverse DCT matrix. A randomly sampled signal with s samples can be represented as

$$PT = PD^{-1}(N)X$$

The matrix P, shown as follows, takes the s random rows out of T.

$$P = \begin{vmatrix} 0 & \dots & 0 & 1 & 0 & \dots & & \\ 0 & \dots & & & 0 & 1 & 0 & \dots \\ & & \dots & & \dots & & & \\ \dots & 0 & 1 & 0 & & \dots & & \end{vmatrix}$$

Let $b = PT$ and $\Phi = PD^{-1}(N)$. Then

$$b = \Phi X$$

where b is the time-domain observation, and X is the frequency-domain signal that we want to recover from b. To differentiate, we call P the *time-domain selection matrix* and Φ the *sampling matrix*. Φ consists of k random rows from the inverse DCT matrix $D^{-1}(N)$.

When $k \ll N$, this linear system is *underdetermined*, i.e., having infinite solutions. However, if we know X is sparse, then we can search for the most sparse solution out of all the possible combinations of coefficients. The classic CS theory states that an exact reconstruction for a sparse signal is possible from partial knowledge of its Fourier coefficients [30]. An *k-sparse* discrete signal can be reconstructed from $O(k \cdot \log N)$ random samples in time domain with probability over $1 - O(N^{-C})$, where C is a given accuracy parameter and N the size of DFT basis. The actually sparse reconstruction can be solved by basis pursuit (BP) [30] (l_1-norm) minimization or matching pursuit (MP) [39]. In practice, this idea

can be extended to different transform bases such as wavelet by selecting a set of random support from the transform matrix.

It is worth noting that there are also other forms of sampling matrices such as Gaussian random matrix, random Bernoulli matrix [40], etc. The random sampling matrix has to obey the restrict isometry property (RIP) [41] defined in Definition 1.

Definition 1 *A matrix A satisfies the RIP of order k if there exists* $\delta_k \in (0,1)$ *such that for every k-sparse vector y,*

$$(1-\delta_k)\|y\|_{l_2}^2 \leq \|Ay\|_{l_2}^2 \leq (1+\delta_k)\|y\|_{l_2}^2 \tag{3.1}$$

where δ_k *is called the isometry constant.*

The RIP condition measures the Euclidean length of k-sparse signals. The random DCT matrix and random Gaussian matrix are used in this work. It is shown in [42] that both of them satisfy the RIP condition of order $2k$, which ensures the universal recovery of a k-sparse signal [41].

It is generally believed that BP-based algorithms yield higher accuracy if the signal is sufficiently sparse [43], while MP-based greedy algorithms only guarantee to recover an *k-sparse* signal when the number of measurements is proportion to k [39]. On the other hand, the complexity of MP-based algorithms can be only $O(sNk)$ while that of BP is generally much higher [39]. For our PPG application, we choose MP, because we are targeting low-power embedded platforms that usually do not possess the resource to perform convex optimization. While for the ECG compression algorithm running on the back-end server, we use BP to take the advantage of its higher accuracy.

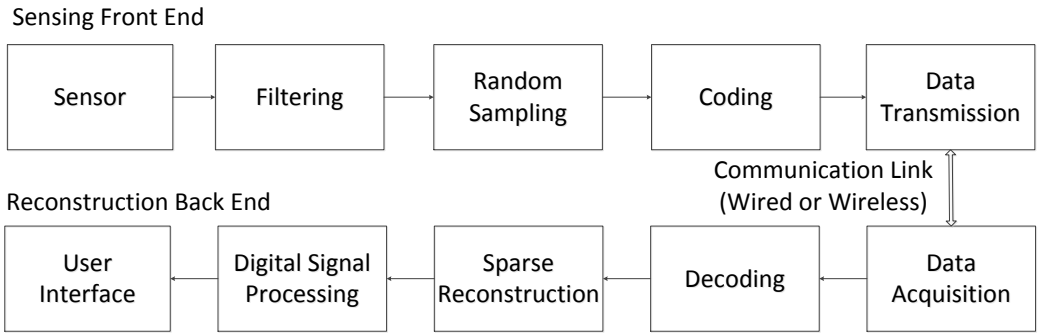


Figure 3.1: Compressive Sensing System Structure

3.3.2 Compressive Sensing Applications

Many CS applications have been developed over the past decade [9, 28, 44, 45]. The typical diagram of a CS-based sensing system is shown in Fig. 3.1. The transmission link can be either wired or wireless. Both the coding unit and decoding unit are optional. The economy of CS is acquired either from transmission link or the sensing unit. For transmission link, CS is used more as a compression than sampling method. A real-time ECG monitor based on CS [45] does uniform sampling but uses a sensing matrix during compression to reduce the bandwidth requirement. ECG sampling may consume only several hundred miliwatts by low-power ADC [13]. This is negligible even to the power consumption of a Bluetooth Low Energy SoC [46], which usually consumes more then tens of mW. Compared with ECG, the sampling process of PPG signal is more expensive due to LED usage. Besides PPG, magnetic resonance imaging (MRI) [44] also belongs to this category since CS could reduce the radiation exposure for patients. As opposed to the transmission link case, CS serves as a sampling paradigm to reduce the sampling power.

Several previous works attempted to recover PPG signal from compressed samples. The Gradient Projection based Sparse Reconstruction (GPSR) [47] is used in [28, 48]. GPSR is based on convex optimization, which is generally of higher complexity than MP-based greedy algorithms. OMP is used in [29]. A significant drawback of OMP is that it picks only one coefficient every iteration. This inevitably increases the number of iterations and

requires a higher order of computational resources. It is estimated that OMP would take tens to even hundreds of iterations for PPG reconstruction [29].

We believe in thoroughly exploring the properties of the source signal before applying the algorithm. Any reconstruction method within the CS framework should not be used as a black box. In our PPG application, the reconstruction is based on CoSaMP [11]. Different from OMP, CoSaMP can rapidly reconstruct the signal by adding multiple coefficients into the support. We customize several parameters based on the characteristics of PPG signals to further reduce the number of iterations.

Chapter 4

Technical Approach

In this chapter, we describe the proposed platform and show how to effectively manage the system power. The system is divided into ECG, PPG, MTS and HGR sub-systems. We first explain the hardware structure of each sub-system. We focus on the working mechanism of each medical sensor with the peripheral control circuit. Next, we present the featured power-saving algorithms and techniques for each subsystem. Each sub-system has different requirements of several factors that affect the overall power consumption, such as sampling rate, filter characteristic, etc. We explain in detail how these factors are determined to optimize the power consumption.

4.1 Hardware System

4.1.1 Optical Muscle Tension Sensing Subsystem

Fig. 4.1 shows the block diagram of the proposed low-power reflective optical muscle tension sensing subsystem. It consists of a light emitter with a current source, a PD with an

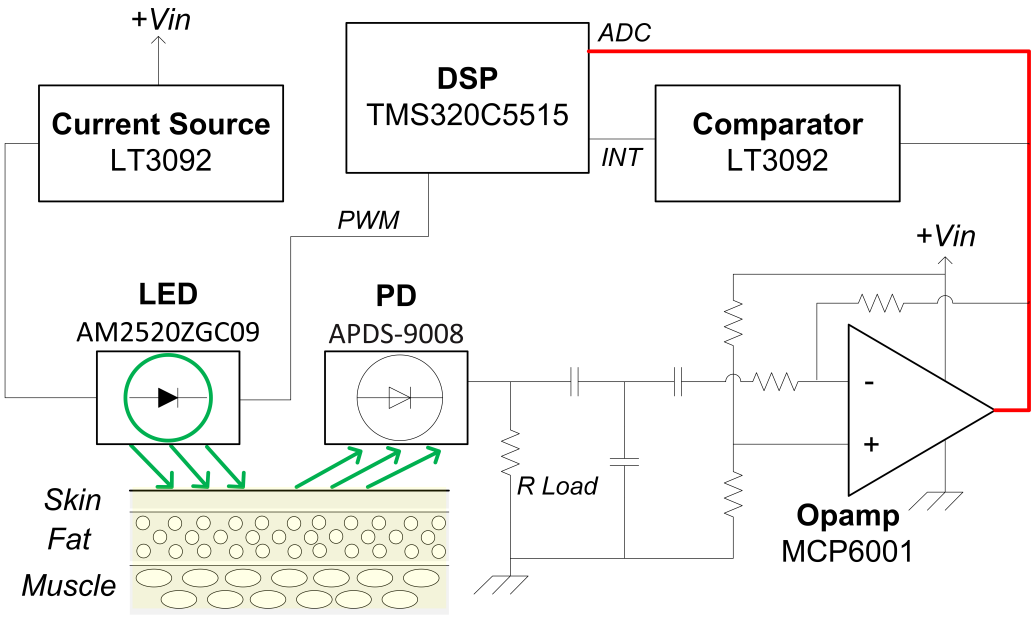


Figure 4.1: Optical Muscle Tension Sensing System

OpAMP, a comparator, a DSP, and a Bluetooth transceiver. The system can also be used in the hand gesture recognition (HGR) subsystem.

To explain how the muscle tension sensing works we give the example of using this system to monitor the sternocleidomastoid (SCM) muscle contraction. SCM is a paired muscle that stretches along both sides of the neck area. When acting alone, it is in charge of head tilting and rotation. It is also involved in various neck movements and aids in forced inhalation when acting together with other muscles. Monitoring SCM muscle contraction is of great significance to asthma assessment and control. Abnormal SCM contraction during asthma is usually a sign of further respiratory impairment. SCM retraction happens in 40% of the asthma episodes [55], and SCM contraction associated with asthma usually indicates a more severe disease in children [56].

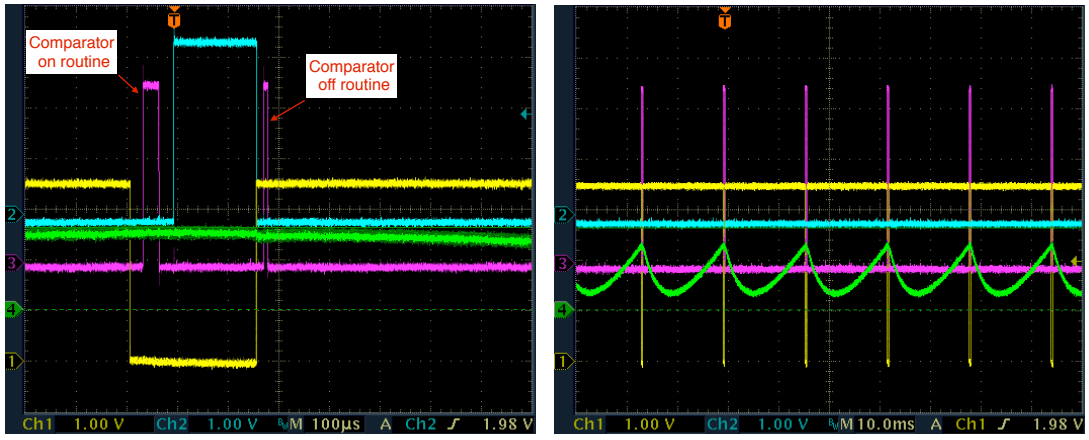
Light Emitter The light emitter subsystem consists of a pulse width modulation (PWM)-controlled LED (AM2520ZGC09) driven by a current source (LT3092) [49]. We choose the green LED with the peak wavelength of 515 nm, which has been shown to be more

robust to motion artifact than red or blue in the applications of heart rate and pulse volume detection [50]. The current from the LED is sinked into the PWM controller for brightness control.

Photo Detector and OpAMP The PD (APDS-9008) is chosen based on its peak response wavelength matching the LED wavelength. This will achieve the highest efficiency. The PD outputs a current signal that flows through the load resistor to generate the voltage signal. A low pass filter (LPF) further filtered the voltage signal to remove noise and motion artifact and the OpAMP (MCP6001) amplifies it for sampling by the DSP on-chip ADC.

Voltage Comparator The external comparator LTC1440 [51] is added to the system. The voltage comparator enables the DSP to sleep without sampling the signal and wakes it up by interrupt upon detecting muscle contraction As shown in Fig. 4.2a, when the amplified PD signal exceeds the reference voltage, the comparator wakes up the DSP. Two interrupt service routines (ISR) are set on the falling and rising edges of PWM to measure the timing of the on-time of DSP. The on-time is very short, and the timing of these routines are shown by a test pin in the same figure. The reference voltage of LTC1440 is fixed to $\frac{1}{2}V_{cc}$ by a voltage divider. This voltage is set higher than the highest respiration peak in order to prevent false triggering. By adding this comparator, the DSP can remain in sleep mode most of time, including walking, sitting, and sleeping. LTC1440 consumes only under 4 μA in active mode.

Digital Signal Processor The DSP will wake up periodically to set the PWM signal using timer interrupt. Once triggered by the comparator, the DSP will enter continuous-sampling mode by setting a timer interrupt to the falling edge (i.e. LED-on edge) of the PWM signal. The amplified PD signal is sampled by a 10-bit ADC every 16 ms, and the samples are further processed to detect SCM contraction as discussed in Section 4.2.1.2.



(a) Interrupt triggered (b) No trigger signal

Figure 4.2: Optical Sensor with comparator control. Yellow, PWM LED control, blue: comparator trigger, pink: comparator on and off ISRs, green: amplified PD output

Once the SCM contraction is no longer detected, the DSP will go back to standby mode and keep generating the PWM.

Fast switching of the optical sensor is made easier with the DSP timer. The LED on-time can be easily set and changed by setting the duty. The clock of PWM is sourced to its low-speed clock, which is still functioning in standby mode. Delay effect of the LPF can be seen on Fig. 4.2b. To overcome this delay, the signal is sampled 3 times in around 100 μs interval, and only the maximum value is taken. We also explored the option to switch off OpAMP periodically, but the system suffers from the severe delay caused by the LPF and OpAMP during fast switching. We ended up leaving the OpAMP on all the time instead of duty cycling it.

4.1.2 Hand Gesture Recognition Subsystem

Fig. 4.3 shows the block diagram of the proposed HGR system. It consists of an optical MTS with an OpAMP, an inertial sensor, a DSP and a Bluetooth transceiver.

The HGR subsystem uses both the MTS sensor and the accerlerometer. The MTS part is

similar to the subsystem describe in Section 4.1.1. We also use MPU-9250 [52] as the ACC controlled by the SPI interface of the DSP. The comparator is not need since user-programmable threshold can be set in MPU-9250 for motion detection.

MPU-9250 is actually a 9 degree-of-freedom (9-DoF) inertial sensor consisting of a triaxial ACC, gyroscope, and compass. We use only the ACC for this test while the gyro and compass are reserved for future use. The typical operating current of the gyroscope is 3.2 mA and only around 450 μA for the ACC. The compass takes around 280 μA and can be helpful to gesture recognition but it is susceptible to the disturbance of other magnetic materials in the vicinity [53]. We end up turning off the gyroscope and magnetometer to save power.

The DSP performs MTS sampling, ACC reading, and data transmission over Bluetooth. It remains in low-power mode normally. On each timer rollover, the DSP wakes up, reads its ADC for the MTS signal, and reads the ACC data from SPI. After all the data is acquired and stored into a local buffer, the DSP queues a sending event in the scheduler and go to standby mode again, from which the DSP can be waken by interrupts only. This process repeats at a rate of 62.5 Hz. The sending event will cause the queued data to be sent the mobile client through the transceiver.

4.1.3 Photoplethysmogram Monitoring Subsystem

Fig. 4.4 shows the structure of the PPG monitoring system. The system is of a typical transmittance-type pulse oximeter. The DSP is the central computing and control unit. More information on the hardware can be find in the reference design [54].

The system calculates the SpO2 and heart rate in real-time. A timer interrupt routine is set to wake up the DSP from the low-power mode. Upon awakening, the DSP will sample

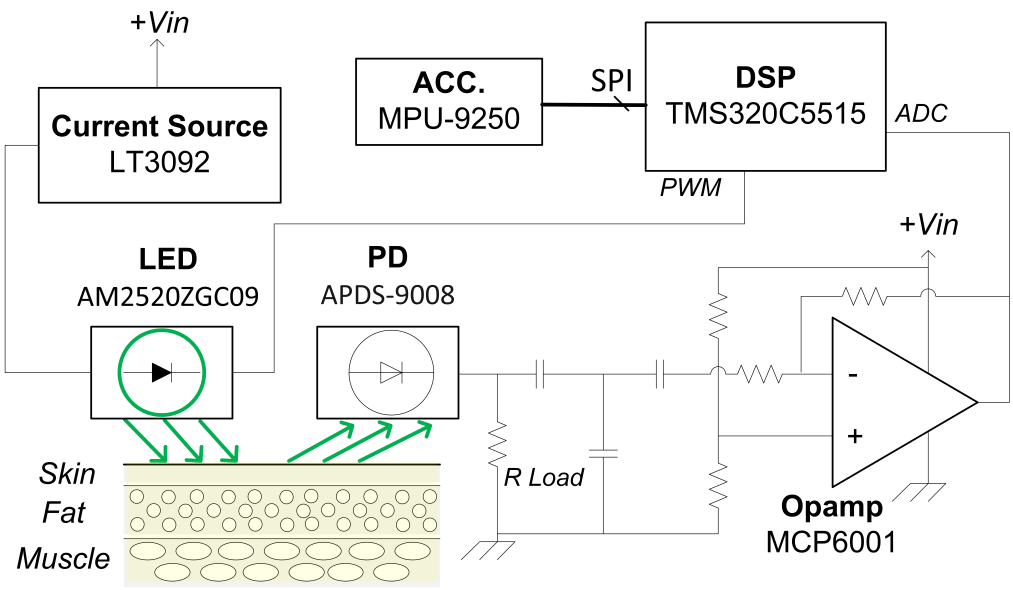


Figure 4.3: Hand Gesture Recognition System

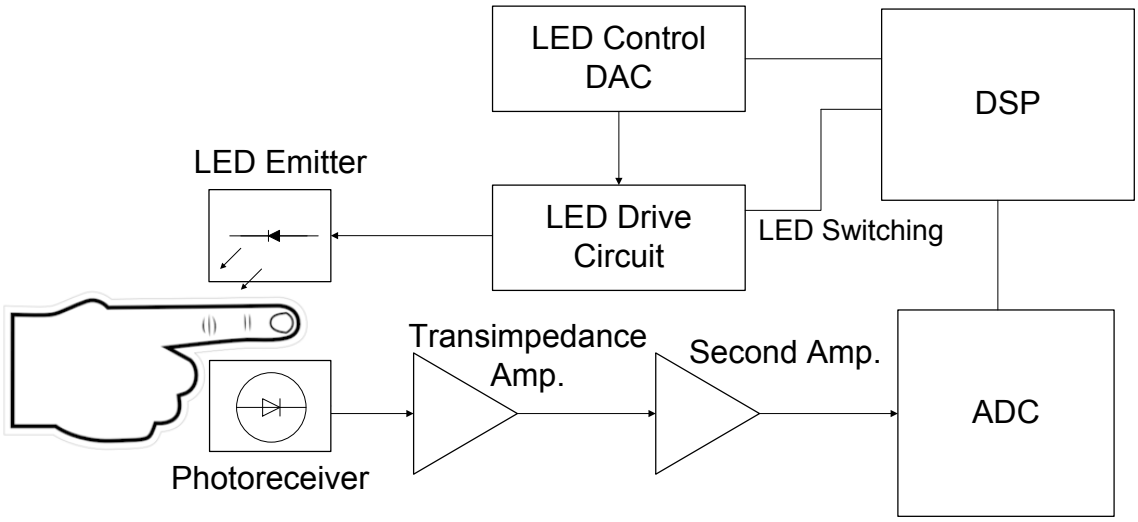


Figure 4.4: PPG System Block Diagram

the PD output. The DSP finds the best signal to noise ratio (SNR) by controlling the LED driver and both amplifiers based on the voltage level of both amplified and raw inputs from the photodetector. The LED brightness and the 2nd amplifier bias voltage are constantly updated and stored. Heart rate and SpO2 are extracted from infrared (IR) and red (R) signal by a threshold-based algorithm runs sample by sample. More detail of SpO2 calculation can be found in Appendix Section B.

4.1.4 ECG Monitoring Subsystem

The ECG monitoring system is quite simple. It utilizes a highly integrated front-end chip, ADS1298 to simplify the system design and reduce the hardware size. Many parameters can be adjusted in ADS1298, such as sampling rate, amplifier gain, filter characteristics, lead-off detection, etc.

4.2 Power Optimization

In this section, we present the power optimization algorithms. The system implements four algorithms in different subsystems. The first is low-power control of the optical sensor, which can be used in both MTS and hand gesture recognition system. The second is a Dynamic Time Warping (DTW) based hand gesture recognition algorithm. The third is a fast compressive sensing method based on dominant frequency extraction used in PPG sub-system. Currently this algorithm can only be used to rapidly extract the heart rate from the PPG samples without reconstructing the signal. But it shows great promise in fast SpO2 level calculation as well. Finally, in the ECG sub-system we introduce a compressive sensing based ECG compression algorithm. It is used on the back-end server side to save the storage and bandwidth.

4.2.1 Optical Muscle Tension Sensor Control

The MTS system developed in this work can be used in a long-term wearable monitoring device to detect and record SCM contraction events during an asthma attack. The recorded data will be later assessed by physicians to make a treatment plan. Low power consumption is essential for battery life and therefore longer monitoring time, as well as wearing comfort due to the size and weight of the battery. However, the SCM contraction is sporadic and fast. The duration of the contraction can be as short as 24 ms [57]. The traditional uniform sampling scheme cannot solve this problem since it takes full power for the DSP to sample and process the signal at such a high rate.

As mentioned in the previous section, the optical system consists of an LED and a PD, and a voltage comparator that enables the DSP to stay in low-power mode and wakes it up when contraction signal detected. The brightness of the LED can be controlled by DSP's pulse width modulator (PWM). A two-phase algorithm, learning and detecting, is also designed to detect the muscle contraction. In the learning phase, the LED brightness level is swept to determine the best PWM duty that controls the input signal under the desired threshold. In the detecting phase, the processed signal from the PD is compared with the comparator reference. Only when it exceeds the reference will the DSP be awakened from low-power mode to detect the signal peak and count the times of the contraction using a threshold-based algorithm. The contribution of this work lies in the novelty of the system design and the lower-power control of the optical sensor.

Because SCM is in the superficial layer of the neck, options capable of detecting muscle movements inlcude sEMG, the accelerometer and the optical sensor. Among them, sEMG is most mature technology for this purpose and can capture high quality signal [58], but its drawbacks include bulkiness, discomfort and power consumption. Commercial sEMG amplifiers consume miliwatts of power. The surface electrode is usually over 20 mm in

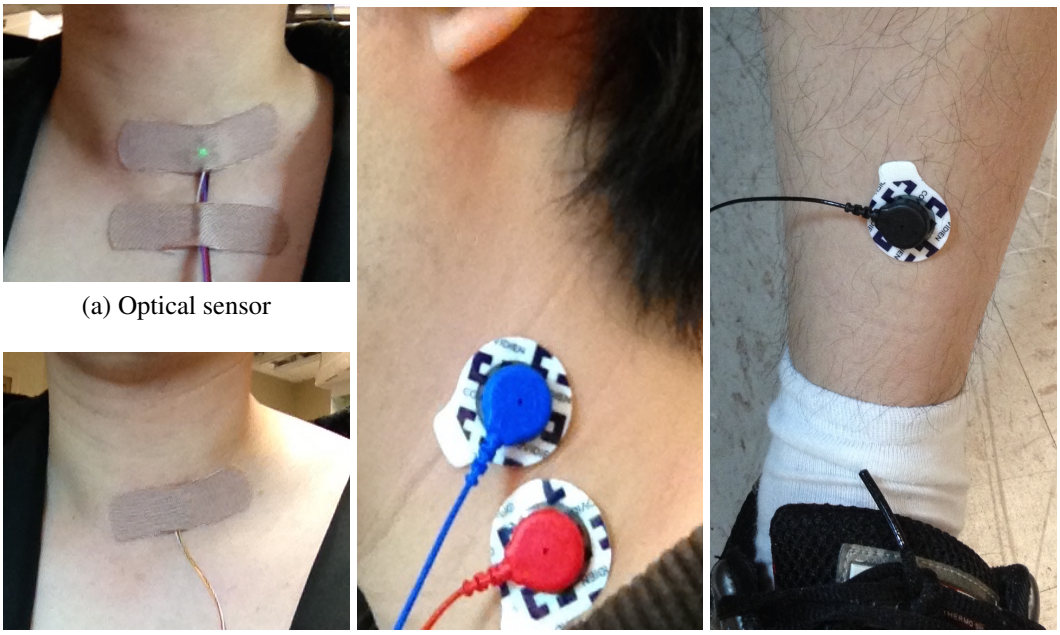
diameter. To measure the differential signal generated by muscle movement, and many applications require three electrode: one on the center target muscle, one on the end, and the third on bony or non-muscular part of the body as a reference. Skin preparation is also inevitable to increase the electrode-skin contact impedance to acquire a better signal. In contrast, the optical sensor and the accelerometer are small enough and can be easily integrated into a more comfortable system than sEMG. No special skin preparation is needed as long as they are placed close enough to the skin surface.

To select the most suitable sensor, we also conducted a comparative study among all three types of sensors. This study shows the optical sensor to be the best among the three for low-power monitoring of muscle activities. With proper control of the LED brightness, the optical sensor can consume much less power than sEMG, while the hardware design is more compact and more comfortable to wear. Even though the signal-to-noise ratio (SNR) of the optical signal is lower than that of sEMG, the optical signal is more stable with little baseline wandering. This feature simplifies the system structure and facilitates the use of a comparator, which further reduces the power consumption.

This section first provides the usage requirements, followed by a description of the detection algorithm. The last part shows the comparative study of the sensor selection.

4.2.1.1 Usage Requirements

The optical sensor is placed on top of the sternal head of left SCM muscle inside the anterior cervical triangle [59], close to the clavicle as shown in Fig. 4.5a. On this location, the greatest signal strength of voluntary contraction can be sensed while the respiratory signal noise is suppressed at a tolerable level. If the sensor is moved more to the left side, the contraction signal becomes weaker. If moved to the right side, the respiratory noise increases.



(a) Optical sensor

(b) Accelerometer

(c) sEMG: neck electrodes

(d) sEMG: leg electrode

Figure 4.5: Placement of sensors for testing muscle contraction.

4.2.1.2 Detection Algorithm

The detection algorithm can be summarized as a two-phase learning one that adapts the LED brightness based on the signal peak. The main reason for learning is that the peak level of SCM contraction varies among people. For example, adults are supposed to generate stronger contraction than youngsters. Neck or head movement can also trigger the interrupt as shown in Fig. 5.6 and Fig. 5.7 Such repetitive movements can be falsely detected as the SCM contraction. This is still tolerable since the recorded signal is supposed to be analyzed in parallel with other data such as stethoscope recordings [60] by physicians.

The designed algorithm, as shown in Algorithm 1, increases the coverage of our system. In the learning stage, the patient is asked to breathe normally. The device will scan the LED brightness by setting the PWM duty. In each duty level, the DSP simply detects the maximum peak of the signal for 5 s. The duty with the signal peak closest to but below the threshold is used in the next stage. A threshold-based algorithm is implemented to count

the peaks. As shown in Section 5.1.1, the signal generated by the optical sensor is robust to motion artifact and free from baseline wandering. This simple threshold based algorithm works well enough to detect the contraction.

Input: current signal sample: *current_sample*, sample data buffer: *data_buf*, detecting mode: *detect_mode*
Output: contraction flag: *contrc_flag*
learning
  while *max_sample* $<$ LO_THD **do**
    SAMPLE_SIGNAL(*data_buf*)
    max_sample $\leftarrow$ MAXIMUM(*data_buf*)
    SET_PWM_DUTY($++$*pwm_duty*); /* raise exception if out of the range and redo learning */
  end
  SET_PWM_DUTY($--$*pwm_duty*)
  SET_DEC_MODE(DETECTING)
end
detecting
  if *current_sample* $\geq$HI_THD **and** *first_peak* $==$ FALSE **then**
    first_peak $\leftarrow$ TRUE; /* first peak can also be set in the comparator ISR */
  else if *current_sample* $<$ HI_THD **and** *first_peak* $==$ TRUE **then**
    peak_count$++$
    first_peak $=$ FALSE
  else if *current_sample* $\geq$ HI_THD **and** *first_peak* $==$ TRUE **then**
    if $++$*peak_timer* $>$ PEAK_LIMIT **then**
      MODE_RESET()
    end
  else if *current_sample* $<$ HI_THD **and** *first_peak* $==$ FALSE **then**
    if $++$*valley_timer* $>$ VALLEY_LIMIT **then**
      MODE_RESET()
    end
  end
  if *peak_count* $>$ 2 **then**
    return TRUE
  end
  return FALSE
end

Algorithm 1: SCM Contraction Detection Algorithm

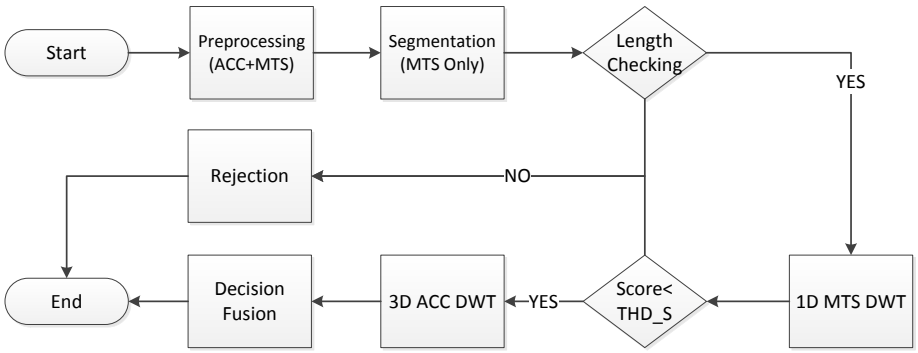


Figure 4.6: Detection Algorithm

4.2.2 Hand Gesture Recognition

We propose a recognition algorithm based on DTW and decision fusion. The flowchart of the algorithm is shown in Fig. 4.6.

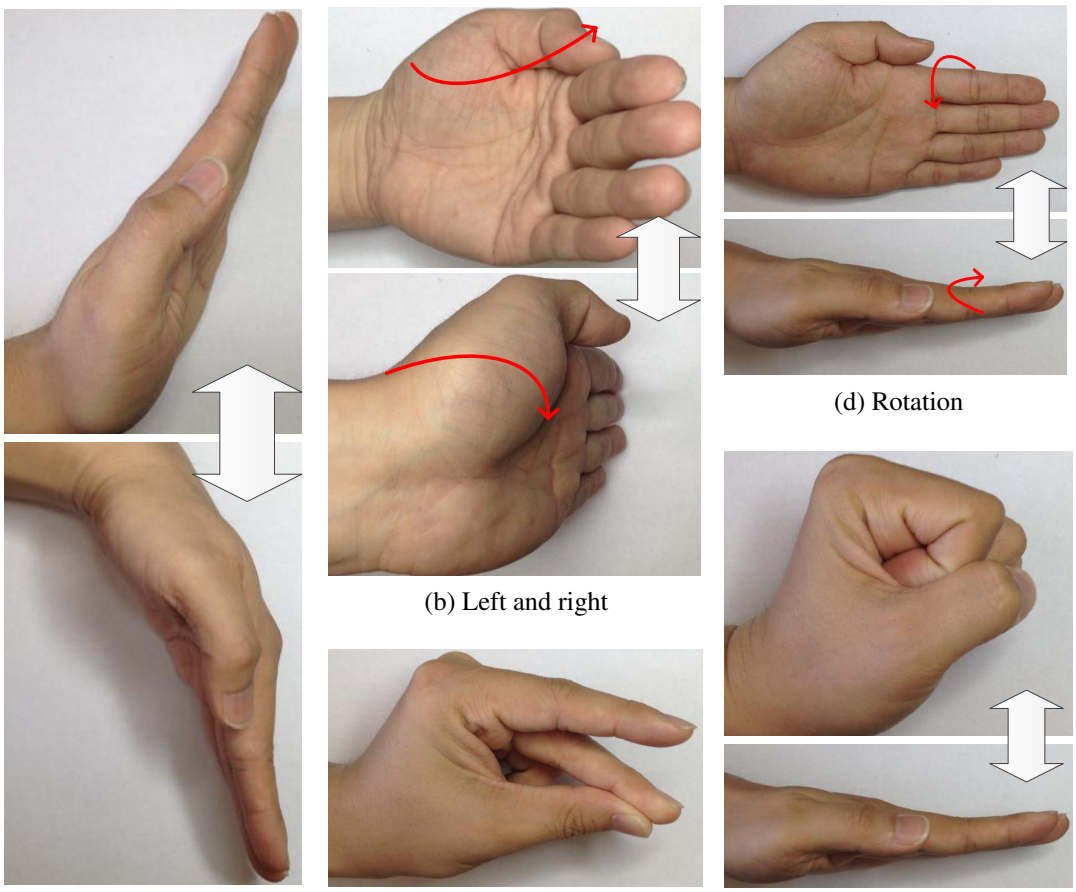
4.2.2.1 Gesture Set

Fig. 4.7 shows the tested gestures. What distinguishes them from those that can be handled by previous ACC-based systems [34, 35, 61] is their use of significant wrist movements, such as hand up, clicking, and making a fist.

4.2.2.2 Preprocessing and Segmentation

In the preprocessing stage, a median filter is used for both MTS and ACC to remove the high-frequency noise. A high-pass filter (HPF) is then used to remove the gravity factor in all three axes of ACC. After the filtering, only the MTS signal is normalized to reduce the influence of signal amplitude variation due to different speeds of performing the same gesture. The raw and the filtered signals are shown in Fig. 4.8.

Segmentation is based solely on the amplitude of the MTS signal. A double-thresholding



(a) Up and down (release)

(b) Left and right

(c) Clicking

(d) Rotation

(e) Fist and release (down)

Figure 4.7: Gestures set. (No distinction between the down gesture from the up position and the release gesture from the fist position.)

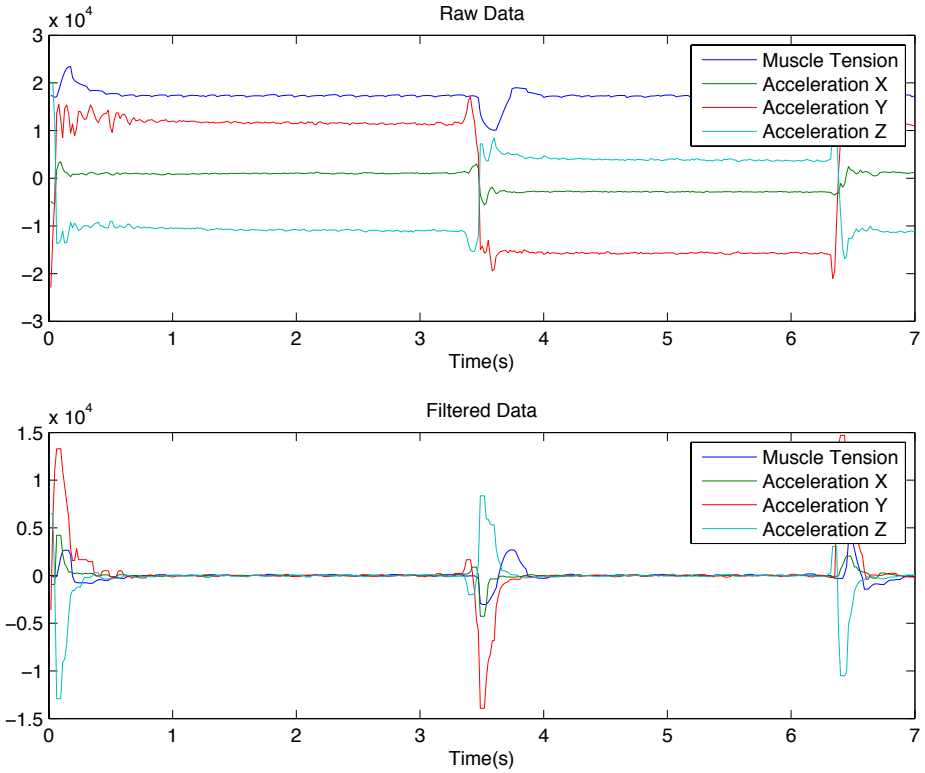


Figure 4.8: Filtered Signal

algorithm, as shown in Algorithm 2, runs point by point to divide the signal into segments. Previous single ACC-based systems [34, 35] usually use a fixed-size sliding window with overlap to segment the signal. The signal content within the window has to be checked constantly to see if any valid gestures exist. Longer delay will be caused by the fixed window size. By setting the starting point and ending point in the signal of interest, our MTS-based segmentation improves the system response time while naturally rejecting some segments based on their length. Any signal that is too short or too long will be automatically discarded and will not go to the computationally complex dynamic programming.

The algorithm keeps counting the samples across or below the corresponding amplitude threshold. Based on a series of counting conditions, the algorithm resets the counter or outputs the starting and ending points of the given segment. Fig. 4.9 shows a segmented MTS signal with its starting and ending points. This algorithm requires that the gestures be done separately with enough resting interval. Otherwise, several segments may be counted

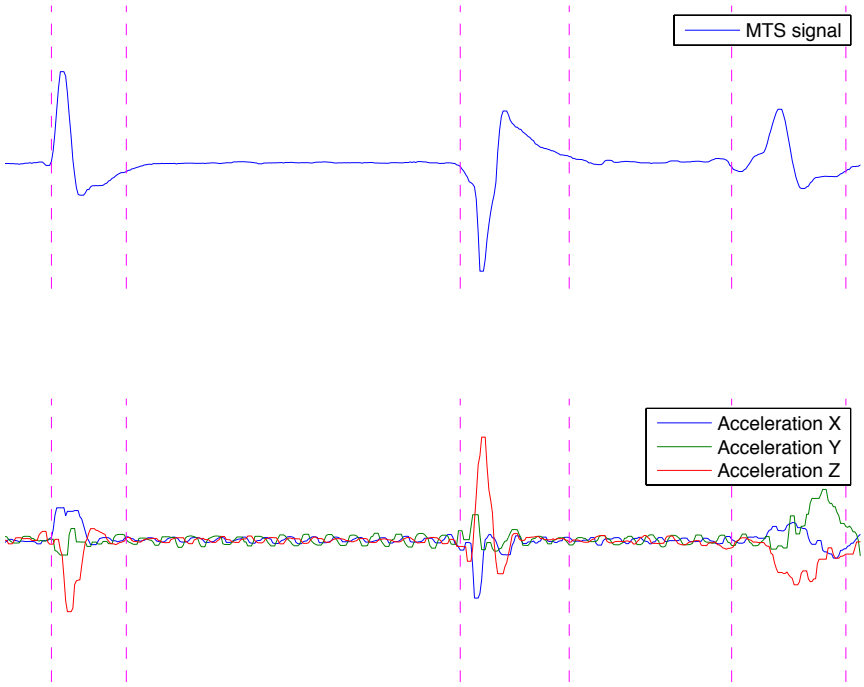


Figure 4.9: Signal Segmentation

as one and may be discarded due to oversize.

4.2.2.3 Dynamic Time Warping Algorithm

As shown in Fig. 4.6, DTW is the core algorithm of detection. It is performed to both MTS and ACC signals. After preprocessing and segmentation, MTS signal is first checked with all stored templates using one-dimensional DTW. Only those templates with sufficiently high scores can be sent to the next step. If the segmented signal does not match any pattern over the threshold, then it will be discarded. We find this rejection step to be able to effectively reduce the number of false-positive detections due to casual, unintended movement. The basic DTW algorithm is shown in Algorithm 3.

DTW finds a match between two time sequences by dynamic programming. The idea is to find the shortest path in the cost matrix as shown in Fig. 4.10, where each cell represents the similarity score between the two corresponding subsequences. The distance between two

Input: current signal sample: *current_sample*, current sample position: *current_position*
Output: segment starting and ending points: idstart_point, *end_point* or NONE
if *current_sample* $<$ LO_THD **and** *start_flag* $==$ FALSE **then**
  if *jump_flag* $==$ TRUE **then**
    start_point $\leftarrow$ *current_position*
  else
    reset_cont $\leftarrow$ *reset_cont* $+1$
    if *reset_cont* $>$ RESET_COUNT **then**
      RESET()
    end
  end
else if *current_sample* $\geq$HI_THD **and** *start_flag* $==$ FALSE **then**
  reset_cont $\leftarrow 0$
  jump_flag $\leftarrow$ TRUE
  signal_cont $\leftarrow$ *signal_cont* $+ 1$
  if *signal_cont* $>$ SIGNAL_COUNT **then**
    start_flag $\leftarrow$ TRUE
  end
else if *current_sample* $<$ LO_THD **and** *start_flag* $==$ TRUE **then**
  if *jump_flag* $==$ TRUE **then**
    end_point $\leftarrow$ *current_position*
    jump_flag $\leftarrow$ FALSE
  end
  ending_cont $\leftarrow$ *ending_cont* $+1$
  if *ending_cont* $>$ ENDING_COUNT **then**
    return *start_point, end_point*
  end
else if *current_sample* $\geq$HI_THD **and** *start_flag* $==$ TRUE **then**
  if *jump_flag* $==$ FALSE
  then
    ending_cont $\leftarrow 0$
    end_point $\leftarrow$ *current_position*
  end
end
return NONE

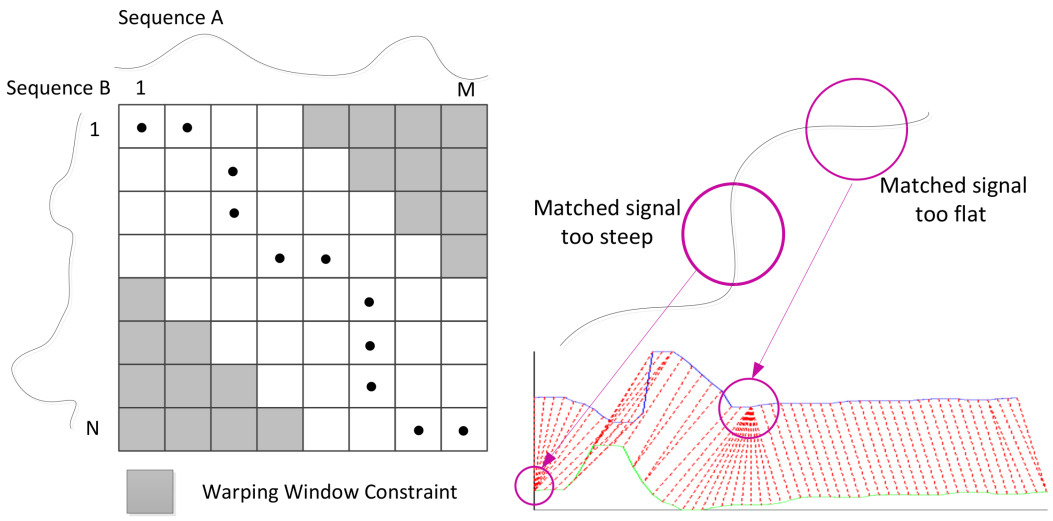
Algorithm 2: MTS segmentation algorithm

Input: signal sequence A: *seq_a* of length M, signal sequence B: *seq_b* of length N
Output: similarity score: *matching_score*
initialization
    cost matrix *cost_matrix*
    slope matrices *slope_x* and *slope_y*
end
for *i=2 to M+1* **do**
    for *j=2 to N+1* **do**
1   **if** ABS(*i-j*)≤WINDOW_SIZE **then**

$$neighbors \leftarrow \begin{pmatrix} cost\_matrix(i\text{-}1,j) \\ cost\_matrix(i,j\text{-}1) \\ cost\_matrix(i\text{-}1,j\text{-}1) \end{pmatrix}$$

2   **if** *slope_x*(*i-1*,*j*) == SLOPE_THRESHOLD **then**
                neighbors(*1*) ← INF
            end
            if *slope_y*(*i*,*j-1*) == SLOPE_THRESHOLD **then**
                neighbors(*2*) ← INF
            end
            cost_matrix(*i*,*j*) ← MIN(*neighbors*) + DISTANCE(*seq_a*(*i*-1),*seq_b*(*j*-1)))
3   SLOPE_UPDATE()
        end
    end
end
return *cost_matrix*$(M+1,N+1)$

Algorithm 3: Dynamic Time Warping



(a) Cost matrix and warping window constraint (b) Slope constraint

Figure 4.10: Dynamic Time Warping with Constraints

MTS samples is just the absolute value, while for 3D acceleration samples, the distance is calculated based on Euclidean distance as in shown in Eq. (4.1). The matched signals in all four dimensions are shown in Fig. 4.11.

$$\sqrt{(ACCX_1 - ACCX_2)^2 + (ACCY_1 - ACCY_2)^2 + (ACCZ_1 - ACCZ_2)^2} \tag{4.1}$$

Two constraints are implemented here: warping window constraint and slope constraint [62]. Warping window constraint (line 1) prevents one point from matching any point too far away by setting global forbidden area, shown as the gray cells in Fig. 4.10, to eliminate the path far off the diagonal. The local slope constraint (line 2) avoids the alignment paths that are too steep or too shallow as shown in Fig. 4.10. The slope value along the optimal path from the starting point to each cell is stored in separate slope matrices. When exceeding the threshold, the path will be discarded along the way. The slope matrix is updated on line 3. If the two sequences are too different from each other, then the template sequence will be automatically rejected before DTW.

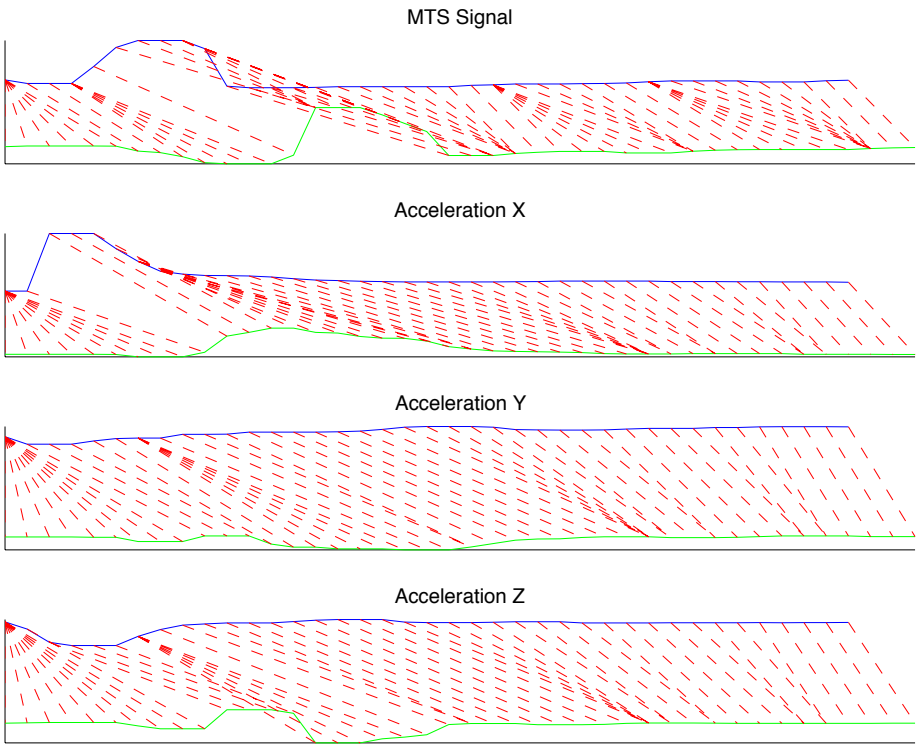


Figure 4.11: Signal matched with Dynamic Time Warping

4.2.2.4 Decision Fusion

The final matching score of the segmented signal to each template is given by Formula (4.2). It is a weighted average of both MTS and ACC scores with the weight $\gamma = 0.66$. The template with the best score is selected as the output gesture.

$$score\_final = \gamma \cdot score\_ACC + (1 - \gamma) \cdot score\_MTS \tag{4.2}$$

4.2.3 Dominant Frequency Extraction Based Fast PPG Compressive Sensing

4.2.3.1 Motivation

Compressive sensing (CS) can directly reduce the sampling rate by random sampling. Some CS systems can work even at a sub-Nyqvist rate. However, the potentially complex reconstruction process can increase the system's response time and power consumption by more than can be saved in sampling. In this section, we explore the inherent signal characteristic to address this problem. One of the most fruitful areas of CS is reconstruction with *known support*, also called *prior knowledge*, which can generally reduce the complexity of CS. In particular, many signals in the natural world possess strong periodic components. In frequency domain, these periodicities appear as one or multiple large DCT coefficients. We introduce a new *frequency-dominant model* to better represent these signals. On top of this model, we show that dominant frequencies can be discovered using partially computed DCT. We also propose a novel system structure with a prior estimation unit as shown in Fig. 4.12. Comparing with the traditional CS structure shown in Fig. 3.1, this unit enables signal reconstruction to be bypassed in some applications, thus increasing the system responsiveness and further reducing the power consumption. A background review for DCT can be found in Appendix A.

4.2.3.2 Dominant Frequency Signal Model

To describe our algorithm, we first start with a formal definition of dominant frequency:

Definition 2 *A frequency f_k is* dominant *to frequency $f_k^{'}$ with degree $M > 0$ if $|a_k| > |a_{k'}|$ and $|a_k - a_{k'}| = M$.*

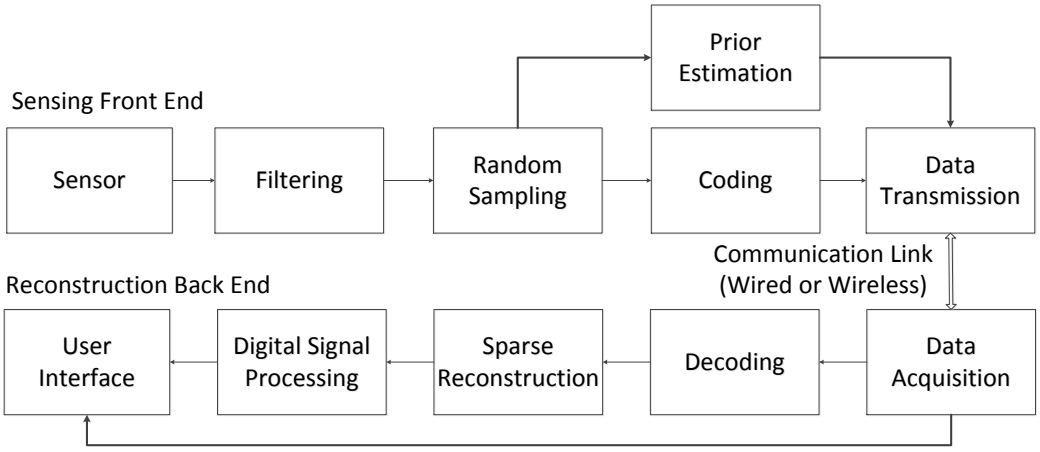


Figure 4.12: Proposed Fast Compressive Sensing System Structure

In real situations, spectral leakage [63] happens due to the windowing effect of DCT. Spectral leakage represents the energy from a non-integer frequency leaked into the adjacent DCT bins. Spectral leakage reduces the dominance degree, as can be seen in Fig. B.1. To overcome this influence, we introduce an extra factor of tolerance β into the definition of a dominant frequency over a range. In practice, β can be set to the frequency range of 1-2 DCT bins based on tolerance.

Definition 3 *Let* $f_k \in \{f_{n_1}, f_{n_2}\}$, $\Gamma = \{f_k - \beta, f_k + \beta\}$. *Let* $\Gamma_{|a|}$ *be the set of all absolute values of DCT coefficients within* Γ. *We say* f_k *is dominant in* $\{f_{n_1}, f_{n_2}\}$ *with degree* $M > 0$ *if the following holds for all* $f_{k'} \in \{f_{n_1}, f_{n_2}\} - \Gamma$.

$$|a_k| = \sup\{\Gamma_{|a|}\}, \qquad |a_k - a_{k'}| \geq M \tag{4.3}$$

As the above definition shows, a dominant component can be either plus or minus as long as it stands up in the DCT spectral. The advantage of this model is that it very well quantifies the dominance degree of a DCT component. It is also related to the sparse model. We can loosely interpret a sparse signal under our defined model. It is easily shown that if a signal is s-sparse in DCT domain, then it has s dominant frequencies with a certain degree. We can approximate a signal with s dominant frequencies using the s-sparse model by discarding

all the coefficients that are not dominant. In fact, it has been shown in [64] that a signal is compressible if the sorted transform coefficients decay exponentially. This is equivalent to exponentially decayed dominance degrees.

4.2.3.3 Dominant Frequency Estimator

We use partially computed DCT transform from random samples to estimate the dominant frequencies defined in Eq. (4.4).

$$C_{\text{partial}} = D P^T b \tag{4.4}$$

where D is the DCT matrix, P is the selection matrix, and b is the time-domain sample. Letting Φ^+ denote the pseudo-inverse of the sensing matrix Φ, Eq. (4.4) is equivalent to

$$C_{\text{partial}} = \Phi^+ b \tag{4.5}$$

Fig. 4.13 shows the original DCT and the partially computed DCT with a different number of samples. It is easy to see the high resemblance of the dominant frequencies. If one frequency is dominant enough in the original DCT, it may be still dominant in the partially computed DCT. This method is mentioned in different contexts from the previous works [11, 65], but neither shows that it can be used to directly extract a significant parameter. In this section, we analyze the correctness of this approach. Particularly, we show that it will pick up the dominant frequency with high probability when the dominance degree M is sufficiently large. We first present the following theorem and also provide a sketch of the proof. We assume that the source $\{x_n\}_{0 \le n \le N-1}$ is normalized to $[-1, 1]$. We define the following notations:

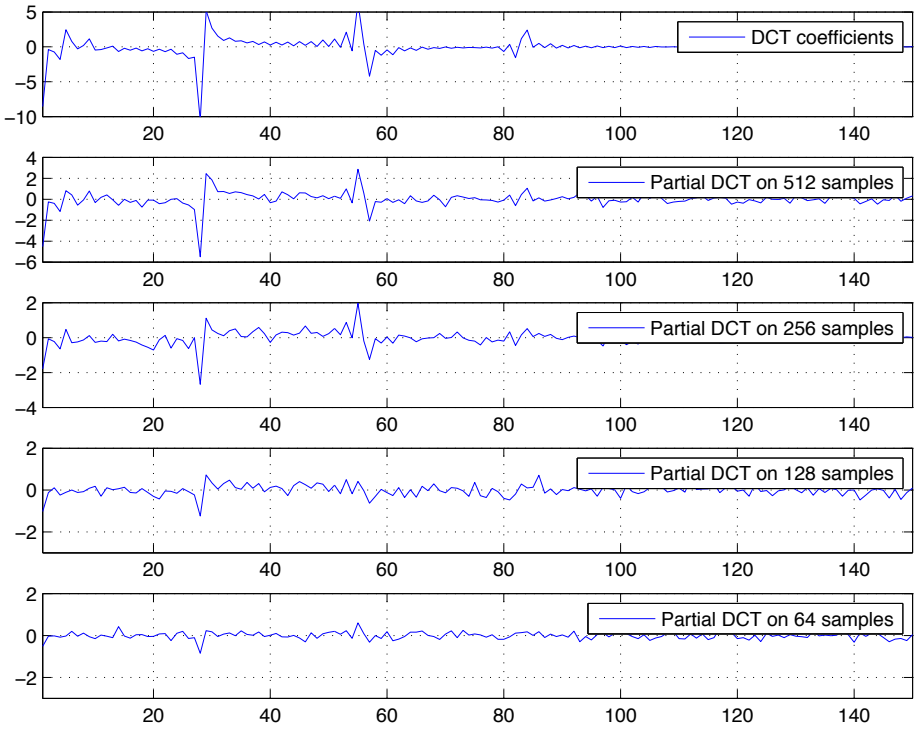


Figure 4.13: Partially computed DCT using different number of random samples

X: $\sum_{i=0}^{N-1} a_i x_i$

Y: $\sum_{i=0}^{N-1} b_i x_i$

$\{x_n\}$: the time-domain samples,

$\{a_n\}, \{b_n\}$: the two rows from the DCT matrix, $0 \leq n \leq N-1$

$\{K_1, \ldots, K_k\}$: a subset of size k randomly taken from $\{0, 1, \ldots, N-1\}$ with $k \ll N$

X_K: $\sum_{i=K_1}^{K_k} a_i x_i$

Y_K: $\sum_{i=K_1}^{K_k} b_i x_i$.

Theorem 1 *If $X - Y = M$ for some $M > 0$, then*

$$\Pr(X_K > Y_K) \geq \frac{M^2}{\frac{N-K}{N-1}\frac{2N-M^2}{K} + M^2} \tag{4.6}$$

.

Proof We define random variable (r.v.) Z to take on $\{(a_n - b_n)x_n\}_{n=0,1,\ldots,N-1}$. We also use the following notations.

$$
\begin{aligned}
Var[Z] &= \frac{\sigma_Z^2}{N}, \\
\bar{Z} &= \sum_{i=0}^{N-1} (a_i - b_i)x_i, \\
Z_K &= X_K - Y_K
\end{aligned}
$$

Obviously, $E[X_K] = \frac{K}{N}X$ and $E[Y_K] = \frac{K}{N}Y$. From Chebyshev-Cantelli inequality [66] we have:

$$
\begin{aligned}
\Pr(X_K - Y_K > 0) &= \Pr(X_K - Y_K - \frac{k}{N}M > -\frac{k}{N}M) \\
&\geq 1 - \frac{Var[X_K - Y_K]}{Var[X_K - Y_K] + \frac{k^2}{N^2}M^2} \\
&\geq \frac{\frac{k^2M^2}{N^2}}{Var[Z_K] + \frac{k^2}{N^2}M^2} \qquad (4.7)
\end{aligned}
$$

We define another r.v. Z' to take on $\{Z_k\}_{k=0,1,\ldots,N-1}$. Z_K is the sum of k samples drawn out of N without replacement. We denote them as $\{Z'_i\}_{i=1,2,\ldots,k}$. We have $Cov(Z'_i, Z'_j)_{i\neq j} = -\frac{\sigma_Z^2}{N-1}$. It follows that

$$
\begin{aligned}
Var[Z_K] &= Var\left[Z'_1 + \ldots + Z'_k\right] \qquad (4.8) \\
&= k^2 \frac{N-k}{N-1} \frac{\sigma_Z^2}{k} \qquad (4.9)
\end{aligned}
$$

We need to develop an upper bound for σ_Z^2.

$$\begin{aligned}
\sigma_Z^2 &= \sum_{i=0}^{N-1} \frac{[(a_i - b_i)x_i - \bar{Z}]^2}{N} \\
&= \sum_{i=0}^{N-1} \left[\frac{(a_i - b_i)^2 x_i^2}{N} \right] - 2\bar{Z} \sum_{i=0}^{N-1} \frac{[(a_i - b_i)x_i]}{N} + \bar{Z}^2 \\
&= \sum_{i=0}^{N-1} \frac{[(a_i - b_i)^2 x_i^2]}{N} - \frac{M^2}{N^2} \\
&\leq \sum_{i=0}^{N-1} \frac{[(a_i - b_i)^2]}{N} - \frac{M^2}{N^2} \\
&\leq \frac{2}{N} - \frac{M^2}{N^2} \qquad (4.10)
\end{aligned}$$

Here, we use the fact that $x_i \leq 1$ and $\bar{Z} = \frac{M}{N}$. Since DCT matrix is orthogonal, $\sum_{i=0}^{N-1} a_i^2 = \sum_{i=0}^{N-1} b_i^2 = 1$ and $\sum_{i=0}^{N-1} a_i b_i = 0$. Substituting σ_Z^2 into Eq. (4.9) and then into Eq. (4.7), we have:

$$\Pr(X_K > Y_K) \geq \frac{\frac{k^2 M^2}{N^2}}{k^2 \frac{N-k}{N-1} \frac{\sigma_Z^2}{k} + \frac{k^2 M^2}{N^2}} \qquad (4.11)$$

$$\geq \frac{M^2}{\frac{N-k}{N-1} \frac{2N - M^2}{k} + M^2} \qquad (4.12)$$

■

This problem is actually about two random variables that have a gap between their expectations. We want to find the relationship between them in terms of the gap. The theorem implies that we can extract the dominant frequencies with the maximum of partial DCT coefficients as $X_K = \sum_{i=K_1}^{K_k} a_i x_i$. $x_{K_1}, \ldots, x_{K_k}$ is our time domain observation and $a_{K_1}, \ldots, a_{K_k}$ is the corresponding partial row in DCT matrix. In the example of PPG signal, we use 1024-point DCT and continuously monitored the PPG signal in MIMIC database [19]. When the heart rate is stable, the dominance degree M is over 9. The probability of success is 93.5% for 256 random samples and 85.8% for 128 random samples using Eq. (4.12) .

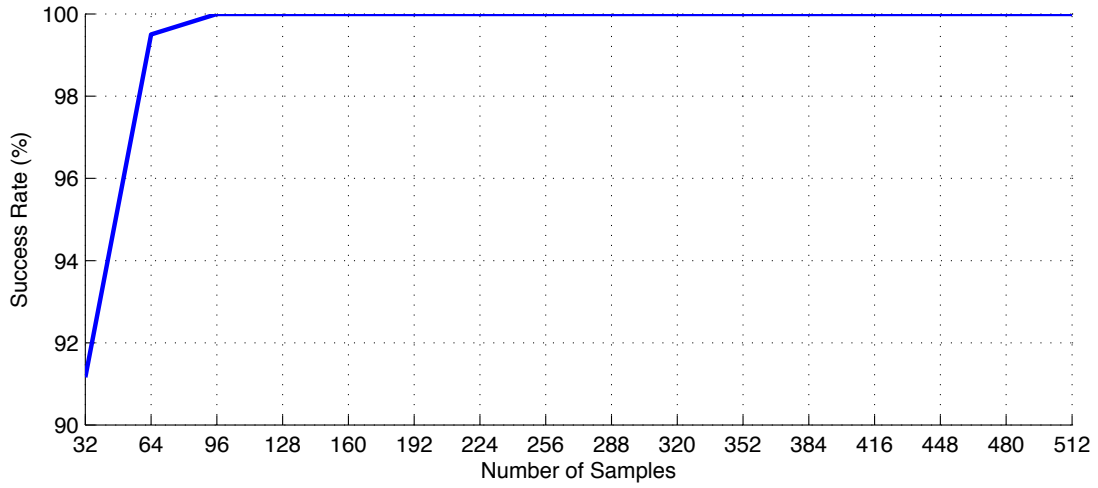


Figure 4.14: Success rate of the frequency estimator

This is just a loose bound derived from all the samples at the maximum value in Eq. (4.10). It is hard to put a tight bound without knowing the distribution of the source signal. However, if we can assume that $Ex[\{x_n\}_{n=0,1,\ldots,N-1}]$ is around 0, then we can approximate $\sum_{i=0}^{N-1}[(a_i-b_i)^2x_i^2]/N$ with $(\sum_{i=0}^{N-1}[x_i^2]/N)(\sum_{i=0}^{N-1}[(a_i-b_i)^2]/N)$. According to Popoviciu's inequality [66] on variances, this would be less than $1/N$. That would give us 96.7% for 256 random samples and 92.6% for 128 samples. The real situation should be much higher. Fig. 4.14 shows the success rate of 10,000 tests using a 1024-point segment with a dominance degree of 9.8 from patient No. 55 in MIMIC database. The success rate is overwhelmingly high when the number of samples is over 96.

In reality, however, M is unknown. We will next develop an estimator for M and its confidence interval as defined in [66]. Given $\{Z_i'\}_{i=1,2,\ldots,k}$ and $Z_K = X_K - Y_K = M'$, we now consider how closely we can estimate M. It is proved in [66] that we can estimate M using the expectation and variance of Z' for the signal that has a normal distribution.

$$\Pr\left(\frac{\bar{Z}' - \frac{M}{N}}{S/\sqrt{k}} \leq z_{\alpha/2}\right) = 1-\alpha \tag{4.13}$$

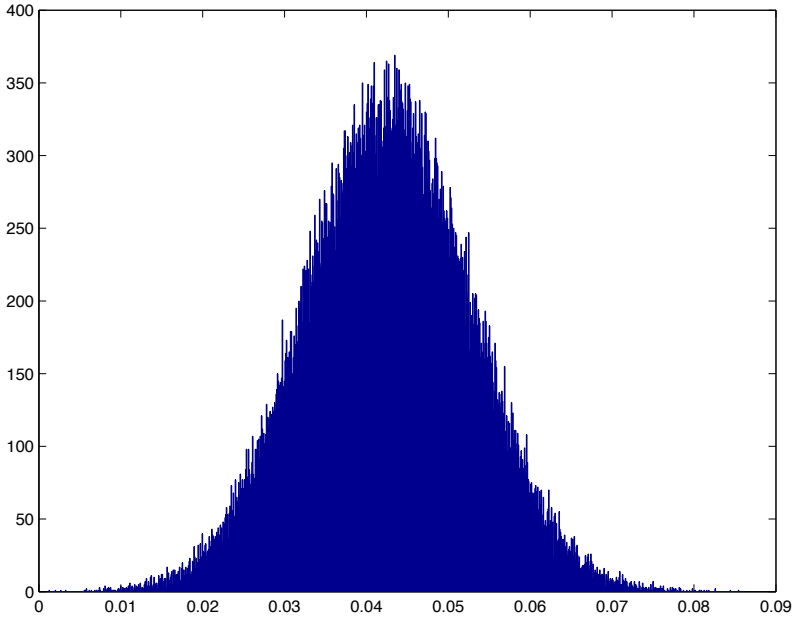


Figure 4.15: Distribution of $(\bar{Z}' - M/N)/(S/\sqrt{k})$

Here we use the notation

$$\bar{Z}' = \frac{\sum_{i=1}^{k} Z_i'}{k} = \frac{M'}{k}$$

$$S = \sqrt{\sum_{i=1}^{k} \frac{(Z_i' - \bar{Z}')^2}{k}}$$

$z_{\alpha/2}$ can be derived from the normal distribution $N(0,1)$. For example, $z_{.025} = 1.96$ and $z_{.050} = 1.645$. This can usually be done by a lookup table. Even if the original signal is not normal, $(\bar{Z}' - M/N)/(S/\sqrt{k})$ has an approximate $N(0,1)$ when sample size $k \geq 50$ [66]. Fig. 4.15 clearly shows this distribution, which is created from 100,000 tests using the data of patient No. 55.

In practice, we can fix α and lower bound of M as M_L. We also put a threshold on M' to decide whether M is large enough. The following has to be satisfied for $M \geq M_L$.

$$M' \geq \frac{M_L k}{N} + \frac{z_{\alpha/2} S k}{\sqrt{k}} \tag{4.14}$$

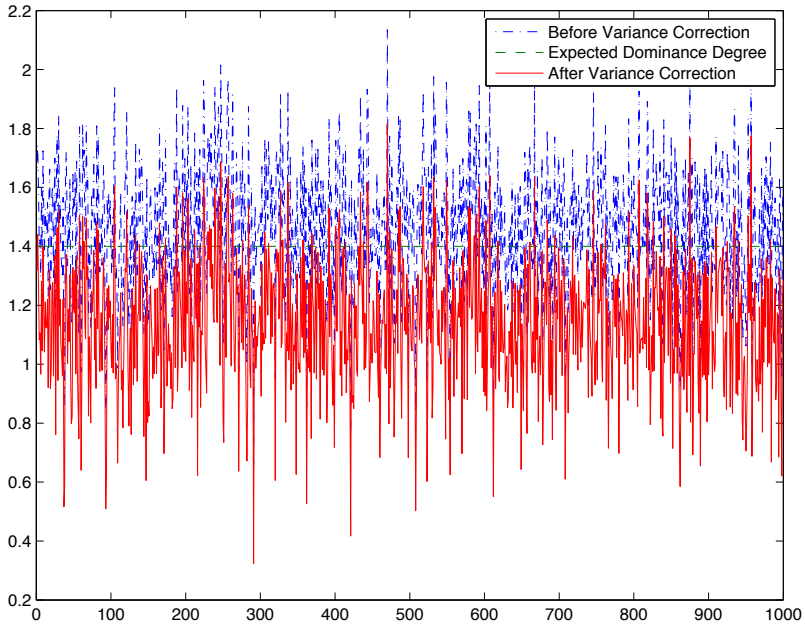


Figure 4.16: Effect on the variance correction

For example, if we want $M_L = 5$ with 90% of confidence then we would need $M' > 1.0226$ with $S = 0.0213$ for the data shown in Fig. B.1.

4.2.3.4 The Role of Randomness

It is obvious that if all the time-domain samples are zeros or close to zero, then the algorithm will fail, since the samples do not provide enough information to extract the dominant frequency or to reconstruct the signal. The randomness actually ensures that this happens with a very small probability. The variance term in Eq. (4.14) also adds a correction factor to reduce the chance of false detection. Fig. 4.16 shows this effect from the data of 1000 partial DCTs from random samples. Before the variance correction, a large percentage of the computed dominance degree is beyond the expected line. After the variance correction, the threshold is raised to prevent wrong detection.

4.2.3.5 DCT Basis

The performance of sparse reconstruction relies heavily on the transform basis for two reasons. One is that the basis directly decides the sparsity of the signal. The other is that complex computation can sometimes be optimized on certain bases. In this work, we use DCT basis. In previous works [28, 48], however, wavelet basis is used to represent PPG signals. We favor DCT for PPG compressive sensing for the following reasons.

1. DCT coefficients can be computed fast from FFT [67]. FFT algorithm runs in $O(N \lg N)$ time. Many DSPs are equipped with hardware butterfly units [68] to accelerate FFT. A great advantage over wavelet is that only a linear processing step is needed to compute DCT from FFT.

2. The PPG signal in DCT domain is sparse when only few dominant frequencies exist as shown in Fig. B.1. Our study also shows that most PPG signals can be represented using around 60 coefficients in the case of 1024-point DCT.

3. The DCT coefficients can be used directly to compute heart rate and SpO2 level. Thus, no extra time-domain processing is required. In previous works [28, 29, 48], the heart rate and SpO2 are detected by the extra beat-by-beat analysis after the reconstruction, therefore consuming extra power.

4.2.3.6 Heart Rate Estimation Algorithm

Our heart-rate detection algorithm has two parts: prior estimation and sparse reconstruction. We want to find the greatest peak in the cardiac frequency range of 0.5–2.5 Hz and the dominance degree of the peak frequency. If it is larger than the threshold, then we are confident that it is the heart rate. If not, we need to reconstruct the original signal for further signal processing. The sparse reconstruction will be discussed in the next section.

A partial DCT transform actually performs dot products of size k for N times. Instead of doing this to the whole DCT basis $D_{N\times N}$, we can pick up the rows in the range {R: 0.5 Hz $\leq f \leq$ 2.5 Hz} and store it locally. The algorithm is shown in Algorithm 4. We assume that $Max(C)$ operation finds the maximum absolute value in set C and outputs a 2-tuple (c_{max}, f_{max}) for the DCT coefficient and the corresponding frequency. We use the same notation for error β and Γ as in the Section 4.2.3.2. An acceptable error range for heart rate is usually ± 1 beat per minute (bpm), $\beta = 1/60$ Hz.

Input: selection matrix P, time domain sample Y, partial DCT matrix D', confidence interval α, error β
Output: heart rate H_{bpm} beats per minute over the sampling period of N
begin
  $C_{partial} \leftarrow D'Y$; /* partial DCT */
  $(c_{max}, f_{max}) \leftarrow Max(C_{partial})$;
  $\Gamma = [f_{max} - \beta, f_{max} + \beta]$;
  for (f, c_f) *in* $\{R - \Gamma\}$ **do**
    $M' \leftarrow |c_{max} - c_f|$;
    $TH_L \leftarrow \frac{Mk}{N} + \frac{z_{\alpha/2} Sk}{\sqrt{k}}$; /* Eq. (4.14) */
    if $M' < TH_L$ **then**
      $CoSaMP()$;
      $Peak\_Detection()$;
      return;
    end
  end
  $H_{bpm} \leftarrow 60 \times f_{max}$;
  return;
end

Algorithm 4: Heart Rate Detection Algorithm

The actual range of the PPG signal is usually determined by the range of ADC. There is no need for normalization but to adjust the threshold of the dominance level. The running time of the prior estimation part is dominated by variance calculation in Eq. (4.14), which takes $O(k \cdot c_R)$ time with c_R being the size of R. Since $c_R, k \ll N$, it can be easily implemented on a mobile device or a low-power DSP [68].

4.2.3.7 Sparse Reconstruction

Heart rate variation or significant noise will result in an insufficient dominance degree. The original needs to be reconstructed in order to perform a beat-by-beat analysis. Our intention is not to develop a universal algorithm but one that works effectively for this special type of signal.

We use CoSaMP for signal reconstruction. The standard CoSaMP algorithm is shown as follows. $Max_n(S)$ finds the best n support from set S. Φ_T denotes the matrix restricted to the columns in set T.

Input: sampling matrix Φ, time domain sample y, sparsity level s, stopping criteria P
Output: s-sparse representation vector x
begin
  $x^0 \leftarrow 0$; /* x_i is the i^{th} approximation */
  $e \leftarrow y$; /* current error */
  $i \leftarrow 1$; /* iteration count */
  while P *not true* **do**
    $C \leftarrow \Phi^+ y$; /* formula (4.5) */
    $\Omega \leftarrow Max_{2s}(C)$;
    $T \leftarrow \Omega \cup x^{i-1}$;
    $b_T \leftarrow \Phi_T^+ y$; /* least square */
    $x^i \leftarrow Max_s(b)$;
    $r \leftarrow y - \Phi x^i$; /* time domain residual */
    $i \leftarrow i + 1$;
  end
end

Algorithm 5: Basic CoSaMP

Fig. 4.17 shows the reconstructed signal using 30, 60, and 90 coefficients. The original algorithm does not work well for our application as expected, for two reasons.

1. As mentioned above, CoSaMP also uses partial DCT to approximate the original. It can easily be seen in Fig. 4.13 that the gap between each pair of frequencies becomes much smaller or even gets reversed. Even if a component is not dominant enough, it can still

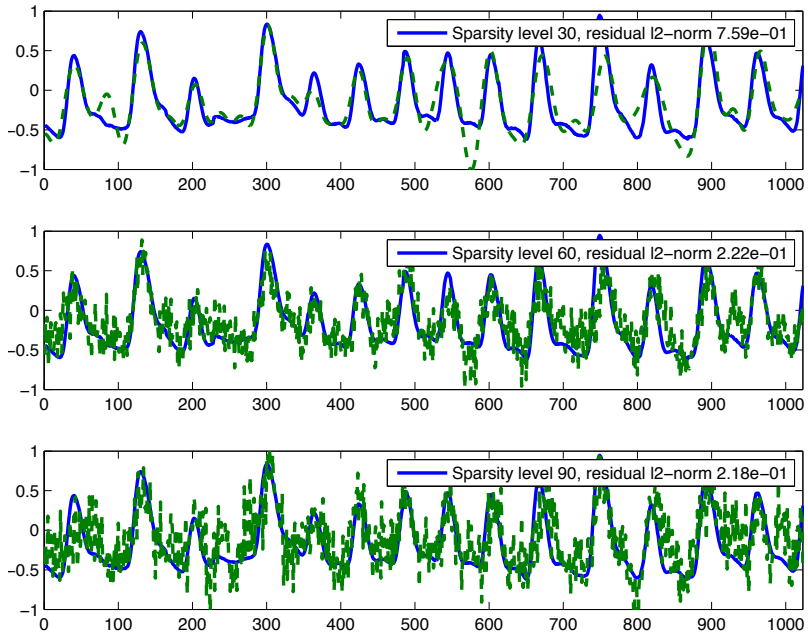


Figure 4.17: PPG reconstruction with 192 samples

be wrongly selected into the best $2s$ support. This is especially significant when we do not have an accurate estimate of the sparsity level.

2. The l_2-norm of the time domain residue r is commonly used as a stopping criterion, but it does not correctly reflect the accuracy of the reconstruction as shown in Fig. 4.17. We also observed that increasing the sparsity level does not always help when the total number of samples is relatively small (below 200) compared to the cardinality of the basis. In Fig. 4.17, the signal reconstructed with sparsity level of 60 actually generates more error than 30 does, even though the residual l_2-norm is smaller.

We reduce the size of the basis to solve this problem. Fig. 4.18 shows the histogram of 1024-point DCT coefficients over the PPG data of 55 patients in MIMIC database. Every count in the histogram is weighted by the absolute value of the corresponding coefficient. The majority of the coefficients is concentrated in bins 1–120. To ensure that our algorithm can also work for signals with severe noise, we set up a separate threshold for the frequencies outside the above range. We still pick them up if they are dominant enough.

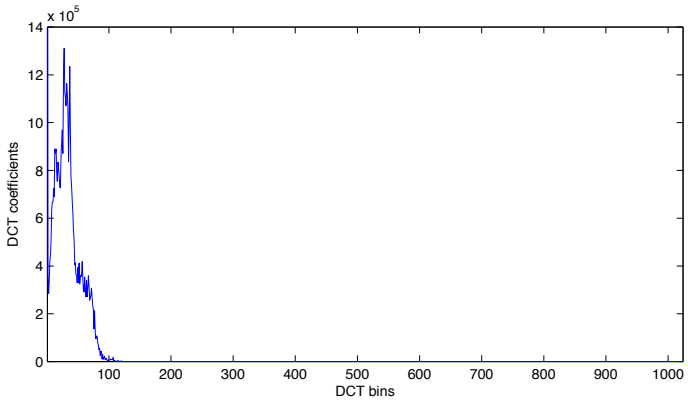


Figure 4.18: Weighted histogram of DCT coefficients

This works much better than just restricting the basis to bins 1–120.

Fig. 4.19 shows the reconstruction over different sparsity level after reducing the basis. We use the record of patient No. 208 because of its low dominance degree throughout the whole data. As can be seen from Table 5.6, 399 out of 433 segments require reconstruction. We compare the reconstructed signal with the original to compute l_2-norm instead of the residue. This gives us the accuracy but it is not available in the real application. It is obvious that adding more support will improve the accuracy, but it will also increase the complexity of the reconstruction. In practice, we find that a support of size around 60 is enough for heart rate detection. The main test result is shown in the next section. Fig. 4.20 shows the reconstruction of one DCT segment. We could still see the difference that generates large l_2-norm of the time domain signal, but there is no need to add more coefficients to the support for this application.

4.2.4 Compressive Sensing Based ECG Compression

This section presents a compressive sensing based ECG compression algorithm. ECG signal is periodic in time domain and sparse in frequency domain. In fact most of the significant frequency contents in a typical QRS complex are contained in frequencies below

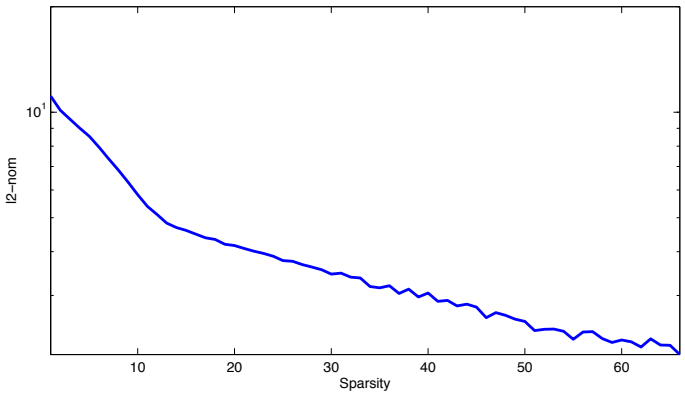


Figure 4.19: Reconstruction with different sparsity level. Data from patient No. 208 in MIMIC database. l_2-norm is computed using the original signal.

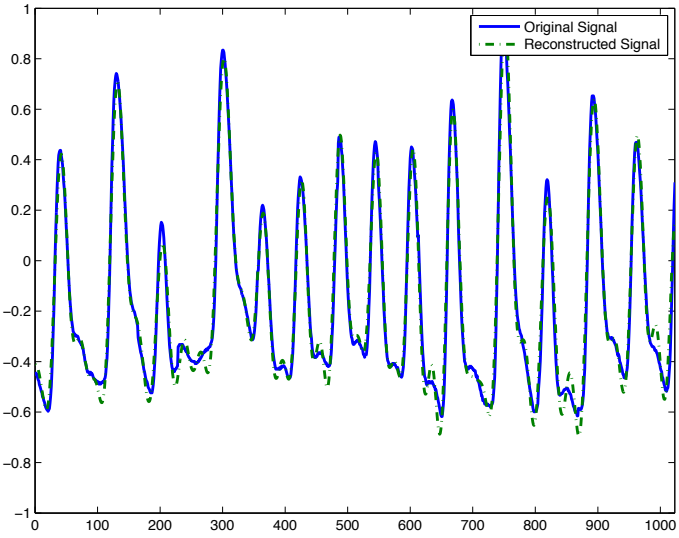


Figure 4.20: Reconstruction with the proposed CoSaMP

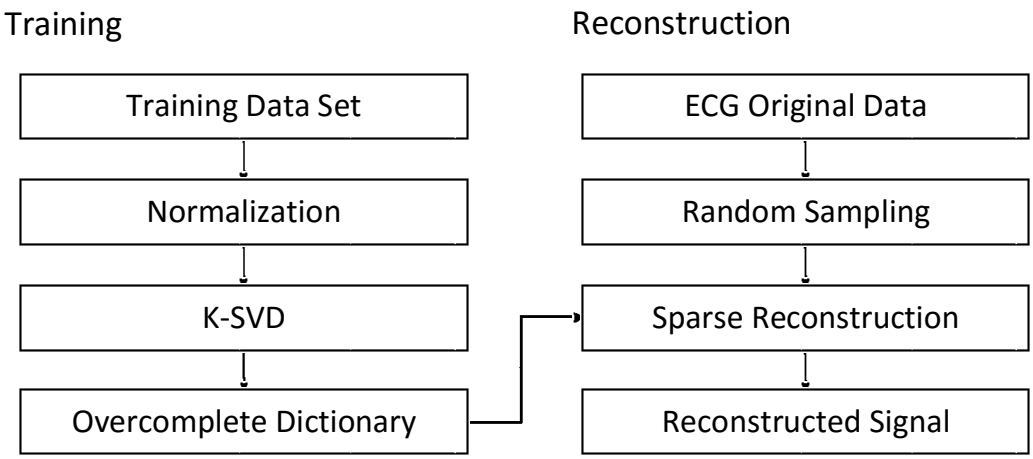


Figure 4.21: Compressive Sensing Based ECG Compression Algorithm

30 Hz with the peak power in the range of 4 to 12 Hz [69]. Our novel algorithm belongs to the lossy compression category. As shown in Fig. 4.21, this two-step algorithm consists of a training and a reconstruction step. During the training, an overcomplete dictionary is built from the training data. It will be subsequently used for reconstruction. In the sparse reconstruction stage, a small number of atoms in the dictionary is chosen to approximate the ECG signal based on random samples. This is done by using Least Absolute Shrinkage and Selection Operator (LASSO) [70], a basis pursuit algorithm.

4.2.4.1 Overcomplete Dictionary

The traditional lossy compression uses orthogonal bases such as DCT or wavelet basis. After the transform only the significant coefficients in the transformed domain will be used to reconstruct the signal. The components with less significant coefficients will be discarded. This approach works extremely well when the signal spectrum power purely concentrates on a few components. For ECG signal, however, this is not the case. Even though the signal is sparse in the frequency domain but there are still too many components that are clinical useful.

To overcome the above limitation of the orthogonal basis, we adopt an overcomplete dictio-

nary [71] approach which combines orthogonal bases with a trained dictionary built offline from the clinical data. The trained dictionary fills in part of the information discarded by the traditional method, keeping the balance between compression ratio and signal distortion.

4.2.4.2 Dictionary Training

As shown in Fig. 4.21, in the training stage the input ECG records are first divided into a number of segments with each segment containing only one complete ECG complex. The segmentation is done by R-wave peak detection [18]. Each segment is than normalized to a fixed length. The K-SVD [71] algorithm is then used to select a small set of segments to build the dictionary. A random set of atoms is chosen as the initial dictionary. Then K-SVD will recursively optimize the dictionary by updating only one atom in each iteration until the stopping criteria is met. The basic K-SVD algorithm is shown in Algorithm 6. Singular Value Decomposition (SVD) in line 1 is used to find the rank-one approximation to the residual matrix E_k. The trained dictionary is further combined with the standard orthogonal bases to build an overcomplete dictionary.

4.2.4.3 Signal Reconstruction

With the overcomplete dictionary we are now ready to reconstruct the signal from a very small number of random samples. This process is shown in Fig. 4.21. We remove the bias from the signal to compute the zero mean signal. This is because we are only interested in the shape of the ECG signal other than the bias.

Let b be the zero bias observation and T_0 be the sparsity level. The sparse reconstruction process is to calculate coefficient x that minimize the l_2-norm as shown in 4.15

Input: Data set Y, sparsity level K, stopping criteria P
Output: Dictionary D of size K
Initialization;
begin
  while *P is not met ;* `/* iteration t */`
  do
    Use OMP to compute coefficient matrix X for Y using $D^{(t-1)}$;
    for *each column d_k in $D^{(t-1)}$ with $k = 1,2,\ldots,K$* **do**
      Define ω_k as the group of examples that use d_k;
      Compute $E_k = Y - \sum_{j \neq k} d_j {x_T}^j$;
      Restrict E_k to the columns with indices in ω_k to obtain $E_k^{R_k}$;
1   Do SVD to $E_k^{R_k} = U \Delta V^T$;
      Substitute d_k with the first column of U;
      Update $D^{(t-1)}$ to $D^{(t)}$;
    end
  end
end

Algorithm 6: Basic K-SVD Algorithm

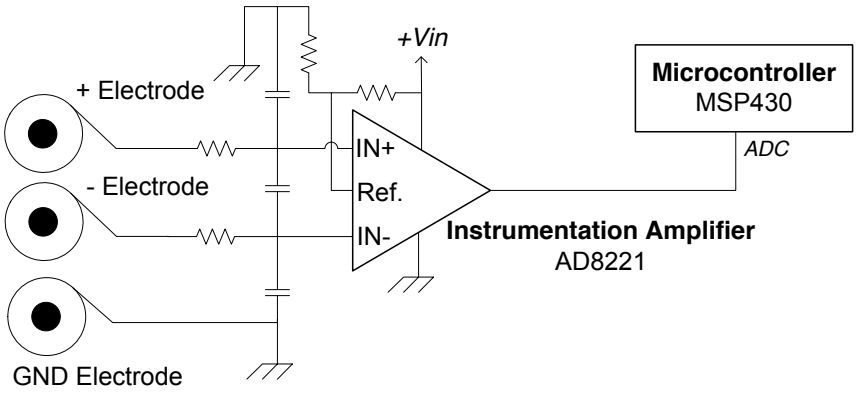
$$\min_x \{\|b - Dx\|_{l_2}\}, \text{ subject to } \|x\|_0 \leq T_0 \tag{4.15}$$

We use LASSO to solve this convex optimization problem. The off-the-shelve MATLAB toolbox [72] is directly used as a solver. The original length of the signal needs to be restored after the sparse reconstruction.

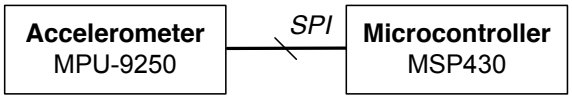
Chapter 5

Implementation and Experimental Results

Without the financial commitment from customers it is difficult to make prototypes to test the whole system. What is more challenging is to collect the physiological signal from the real patients. This can only be done by qualified organizations with the consent from the patients. To avoid these challenges, we test the each sub-system with either recorded data or patient simulators. We either fabricate prototypes or use existing systems. We perform both accuracy test and power measurement. The result is also compared with the existing technologies or systems to show the advantage of the proposed system. The records from MITBIH Database are used to test ECG subsystem and MIMIC database [19] to test PPG subsystem. We have shown the CARDIOSIM ECG simulator in Fig. 2.4. We also use the Fluke Pulse Oximter Functional Tester [73] for PPG testing. The hardware platform is kept as identical as possible to give an accurate estimation.



(a) Surface EMG system



(b) Accelerometer system

Figure 5.1: Other Sensing Modalities for Muscle Contraction Monitoring

5.1 Muscle Tension Sensing Subsystem Test

To evaluate our muscle tension sensor, we compare the results with different sensing modalities for their power consumption and signal quality. Power consumption varies with the details of the hardware and software design. We consider not only power at the component level, but also the complexity of the software algorithm, which directly affects the energy consumption. The signal quality of the sensor also has an impact since more processing may be required to recover the key component of interest.

To make our comparison fair and practical, we stick to the specific purpose of monitoring SCM contraction rather than general muscle movements. We also use commercially available components rather than custom designs for this purpose. We evaluate the power consumption as the primary objective and accurate detection of contraction in real time as the secondary, rather than pursuing the best noiseless signal.

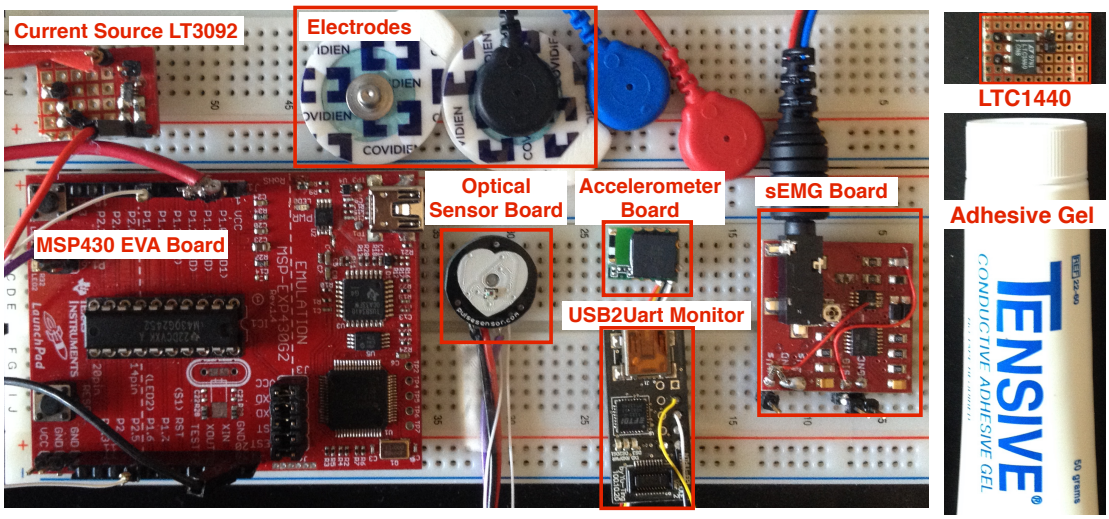
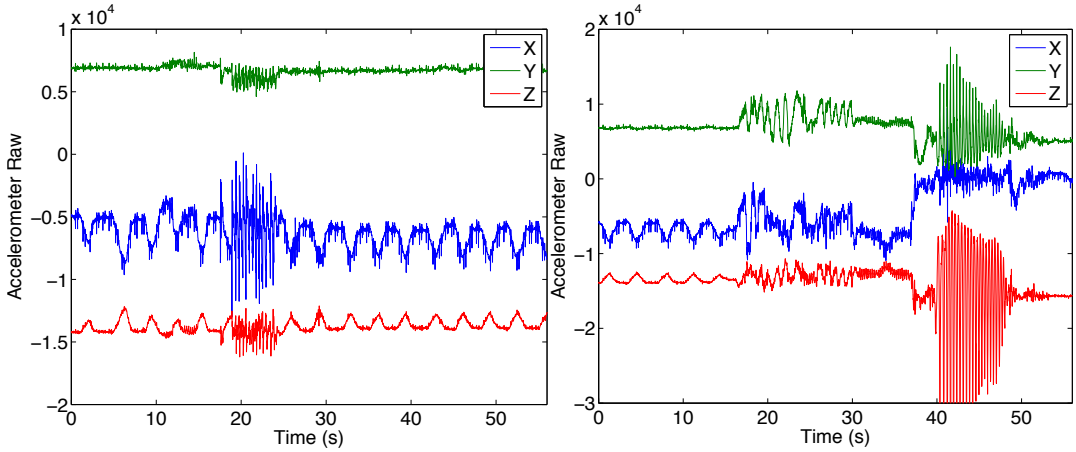


Figure 5.2: Hardware Prototypes for Comparative Study

5.1.1 Comparative Systems

For fast prototyping of the system, we use the pulse sensor board in [74] for the optical sensor and muscle sensor board in [75] for sEMG. We adapted both boards for our comparison as shown in Fig. 5.1. We use an MSP430G2452 [13] evaluation board as the MCU board for sEMG and the optical sensor with the MCU running at 1 MHz on a 3.3 V power supply. For the accelerometer, we designed a prototype using TI's CC2541 Bluetooth Low Energy (BLE) single-chip MCU [76] to control the MPU-9250 accelerometer [52]. A UART monitor is used to transfer the signal to a computer. All the hardware used in this study is shown in Fig. 5.2. The sampling rate is set to 62.5 Hz (16 ms period) as a representative number. The same contraction detecting algorithm is running on the MCUs for all sensors. It is a simple threshold-based algorithm doing sample-by-sample analysis described in Section 4.2.1.2.

Accelerometer The system is shown in Fig. 5.1b. The sensor is placed in the same spot as the optical sensor as shown in Fig. 4.1. MPU-9250 is a low-power 9 degree-of-freedom (9-DoF) inertial sensor with a triaxial accelerometer, gyroscope, and compass. It is controlled



(a) Still state (b) Light to moderate movement

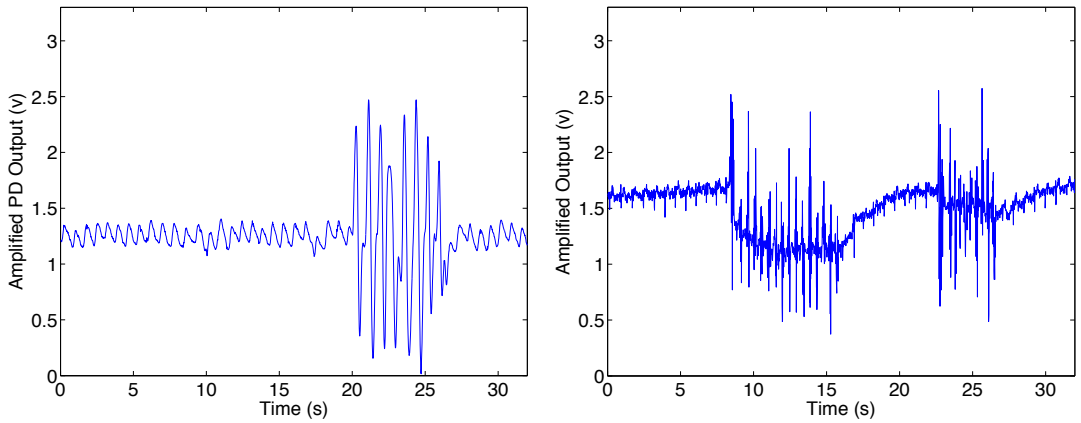
Figure 5.3: Voluntary contraction signal from accelerometer

by the SPI interface of the CC2541 MCU. Acceleration data is sent out from the MCU through UART interface. The size of whole system including CC2541 is only 12 mm$\times$ 9 mm, making it highly wearable.

Surface EMG A two-stage design consisting of an instrumentation amplifier plus OpAMP is common for sEMG. The in-amp serves as a pre-amplifier, rejecting the common mode noise and amplifying the signal only with a small gain. The second stage further amplifies the signal and is usually built in with strong filters to deal with motion artifact and baseline wandering. To simplify the circuitry, we remove the second stage amplifier while increase the gain in the first stage. This inevitably reduces the signal amplitude but saves power.

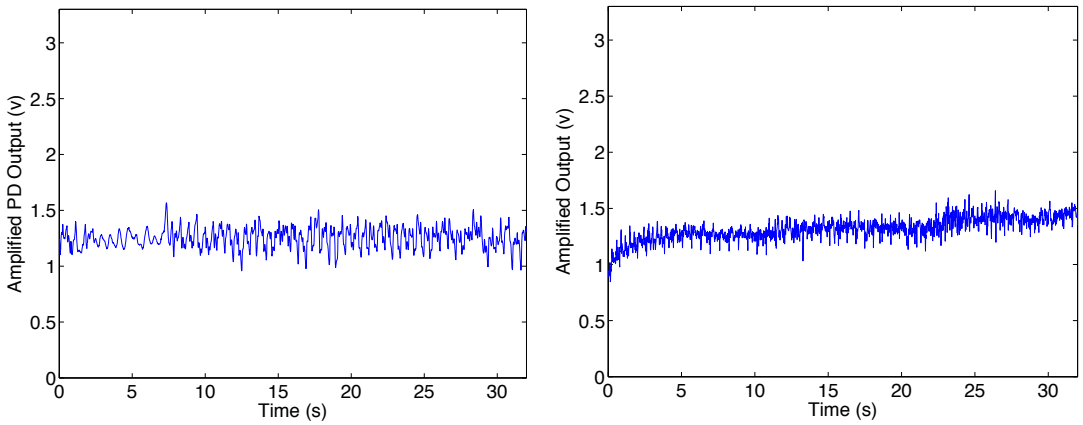
5.1.2 Signal Quality

Accelerometer When the body is in a still state, voluntary contraction can be easily seen. However, SCM muscle contraction cannot be detected during body movement. The signals of voluntary contraction during body movements are shown in Fig. 5.3. It is extremely difficult to detect the contraction even under light upper body movement such as walking



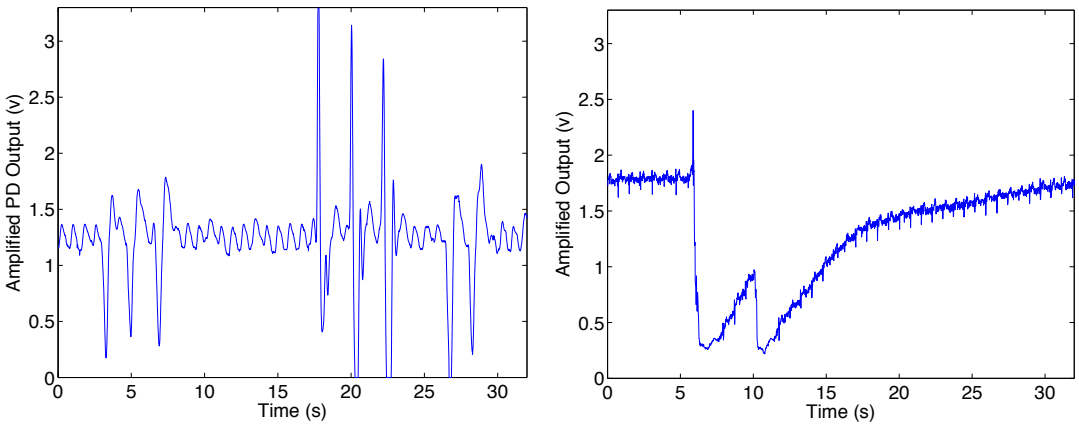
(a) Optical sensor (b) sEMG

Figure 5.4: Voluntary contraction



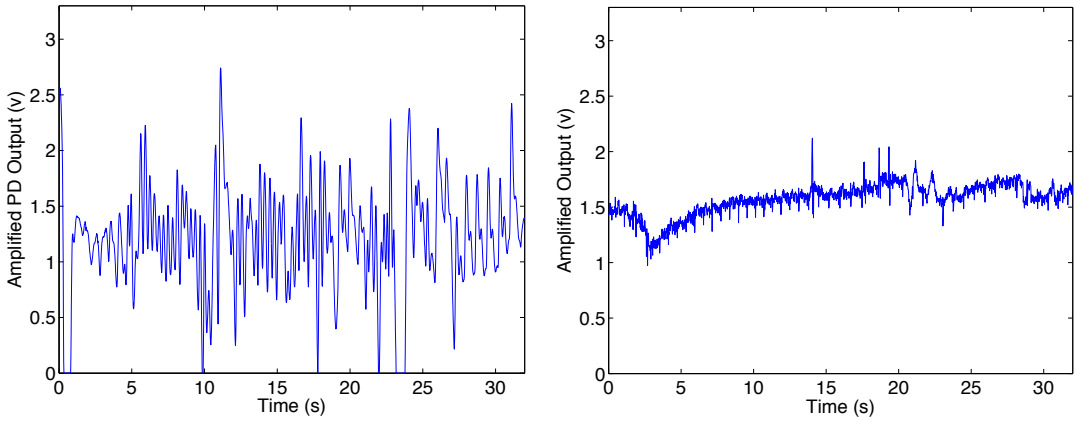
(a) Optical sensor (b) sEMG

Figure 5.5: Jogging with no neck movement



(a) Optical sensor (b) sEMG

Figure 5.6: Turning right and then left



(a) Optical sensor (b) sEMG

Figure 5.7: Continuous head up and down

and jogging, while these movements are common in daily life.

5.1.3 Surface EMG vs. Optical

Figs.5.4-5.7 compare the sEMG signals with those from the optical sensor over different scenarios. sEMG generates better signal in term of amplitude and SNR. The in-amp of the sEMG can also remove body noise such as electrocadiac and respiratory signals very well compared to the optical sensor. However, great baseline wandering can be observed from the graph. Under this influence, the simple thresholding algorithm does not work well on sEMG. Further hardware or software filtering is definitely needed to remove it. For the optical sensor, clear respiratory cycles can be seen in the signal in Fig. 5.4a, but the optical signal is more stable with little changing to the baseline. Similar to sEMG, most of the moderate movements such as walking and jogging do not necessarily cause motion artifact unless the neck area moves. Among all the movements, right turn triggers the greatest peak. Lifting the head up and down generates similar signal peaks to voluntary contraction.

We realized from this study that sEMG is not suitable for this application for two main reasons. First is the use of adhesive gel. The signal changes over time due to the drying

of the gel that changes the skin-electrode impedance. The other is that a third ground electrode is critical to the signal and has to be put far away to generate a stable reference. These constraints inevitably increase the discomfort level and inconvenience to the patient's daily life.

5.1.4 Power Consumption

Accelerometer The current consumption of MPU-9250 is just 19.8 μA in active accelerometer mode and 8 μA in standby mode. Unfortunately, as shown in Section 5.1.2, the MPU-9250 is too susceptible to body movement for this application, even though it would otherwise be perfect in terms of power consumption and wearing comfort.

Surface EMG vs. Optical The dominant power consumer in sEMG is the precision in-amp. AD8221 [77] is used in our sEMG design. Table 5.1 compares it with some other commercially available products that basically have the same function set. It shows that AD8221 is one of the best among the list with a relatively low supply current but good performance in term of noise and bandwidth. There are also so-called micro-power precision in-amps that consume only hundreds of micro-watts, but they are usually optimized for very low input stage current [21]. Thus, the noise level is usually higher than the higher current amplifiers shown in Table 5.1.

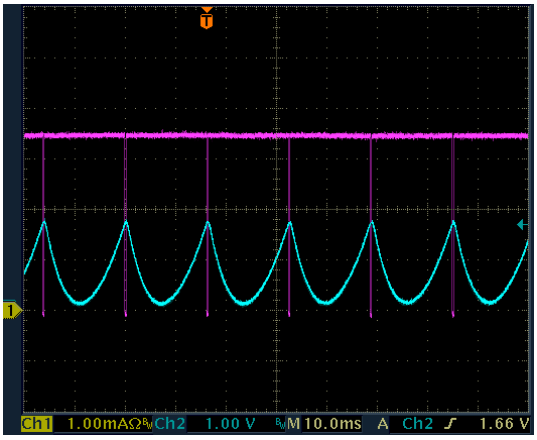
Frequent switching of in-amp is not an option for power saving. Power supply should be turned on before the input signal to avoid overloading the amplifier. In the case of AD8221, once the output voltage goes beyond the supply rails, the protection diodes will be turned on. Even if the in-amp is not damaged, significant noise and delay might occur due the fast switching. As a result, we keep the AD8221 on all the time. A 16-ms timer is set to wake up the MSP430 from LPM3 and sample the signal. The ISR duration and the in-amp output

Table 5.1: Commercial Precision Instrumentation Amplifiers

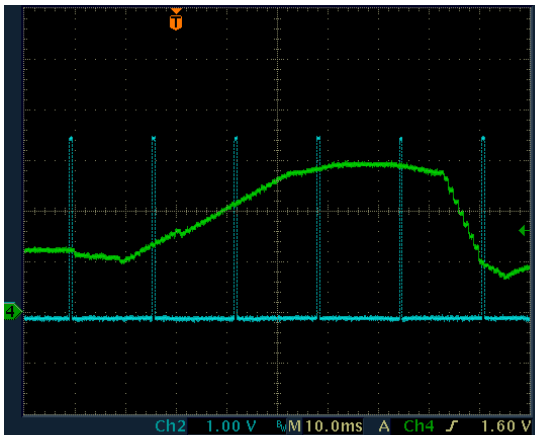
P. N.	AD8221	AD624	INA326	LTC2053	LT1167	MAX4208
Supply Current	0.9mA	2.5mA	2.4mA	0.75mA	1.3mA	0.75mA
Gain	$1-10^3$	$1-10^3$	$0.1-10^4$	$1-10^3$	$1-10^4$	100
Bandwidth (0.1-10Hz)	825kHz	25MHz	1kHz	200kHz	200kHz	750kHz
Input Noise ($\mu VP-P$)	0.25	0.2	0.8	2.5	0.28	2.5
CMRR DC G=1	90dB	80dB	70dB	116dB	86dB	135dB
CMRR 10kHz G=1	80dB	40dB	70dB	¡100dB	62dB	120dB

signal are shown on Fig. 5.8b. An MCU port is pulled up at the beginning and pulled down at the end of the interrupt. The duration of the ISR varies depending on the input signal and the stage of signal processing. For the optical sensor, the interrupt is set on the falling edge of PWM to sample the signal without activating the comparator. This is shown in Fig. 5.8c with the current signal from the power supply.

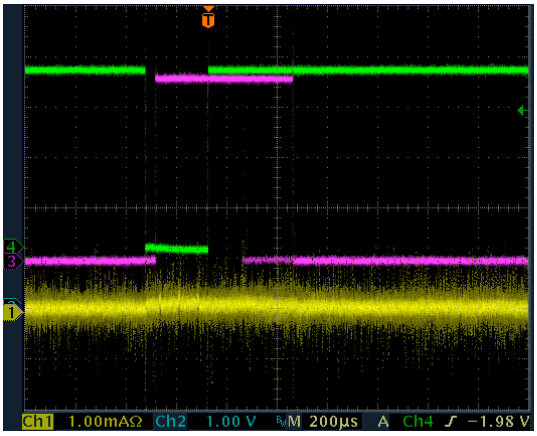
The power consumption is compared by attaching sEMG to one side of the SCM muscle and the optical sensor to the other. In this way, approximately the same input signal can be generated by the symmetric left and right movements. The current signal is monitored and recorded using an oscilloscope. The data is very noisy as shown in Fig. 5.8, and current consumption is mostly in the form of surge current due to the charging and discharging of the capacitive components in the circuit. We are able to calculate the average power by doing integral. For sEMG, the average current during the ISR is 2578 μA and a current of over 2 mA is constantly drawn by the in-amp with an average power of over 6.6 mW. The ISR duration is measured as from 170 μs to 280 μs. For the optical sensor, the average current consumption from LED on to the end of the ISR is calculated as 589 μA with the duration varies from 320 μs to 540 μs. Even though the duration is much longer than



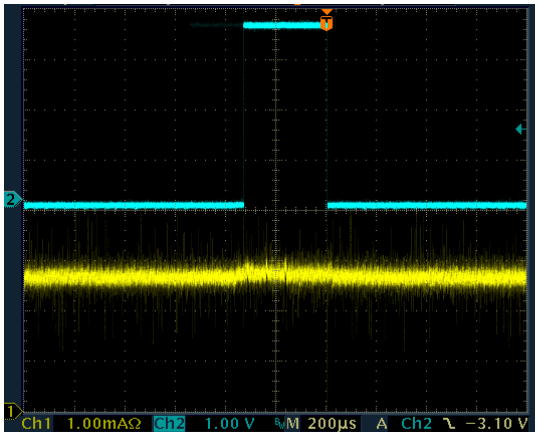
(a) Optical sensor interrupt. Pink: PWM, blue: amplified PD output



(b) sEMG interrupt. Blue: interrupt flag, green: in-amp output



(c) Optical sensor current consumption. Yellow: system current (mA), pink: interrupt flag, green: PWM



(d) sEMG current consumption. Yellow: system current (mA), blue: interrupt flag

Figure 5.8: Signal comparison between optical sensor and sEMG

sEMG because of the sampling interval, the average power consumption in a 16 ms cycle is only around 450 μW. The current probe definitely introduces large error in measuring the noisy sub-mA current, but the comparison clearly shows that sEMG consumes much more current than the optical sensor does.

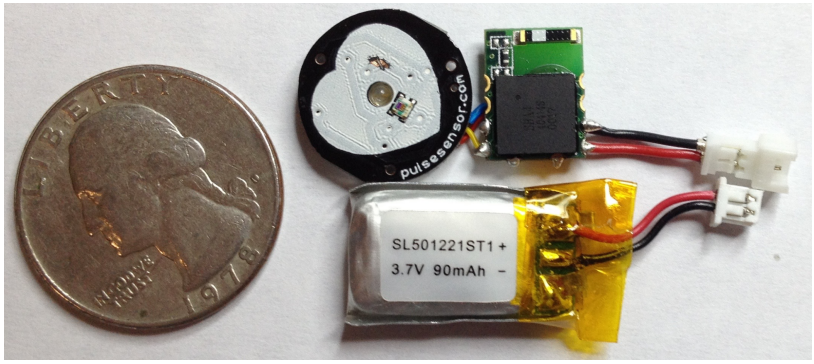


Figure 5.9: Hand Gesture Recognition Hardware Prototype

5.1.5 Discussion

In this section we have presented the evaluation for the proposed low-power, compact, optical MTS system. The system is tested on SCM muscle contraction monitoring for asthma symptoms and other respiratory diseases. This is currently an important area with sorely unmet need for wearable monitors. We compared common sensing modalities, including accelerometry and surface Electromyography, and we showed the optical one to be the most suitable by striking the best balance among the size, power, and sensitivity. In addition to the hardware, our detection algorithm can also handle variations ranging from adults to young children. Together, our hardware and software techniques will enable a new generation of highly wearable monitors for precise monitoring of muscle tension.

5.2 Hand Gesture Recognition Subsystem Test

5.2.1 Prototype Implementation

The prototype of the HGR system is shown in Fig. 5.9. To quickly validate our design, we implement our sensor board using a commercially available pulse-sensor design [74] but customize the board to our design. For the accelerometer and BLE board, we designed

Figure 5.10: Sensing location on the wrist area.

a board as shown in Fig. 5.9. Both boards are connected using jumpers. The overall dimension of the system including the battery is 30 mm (L) $\times$ 15 mm (W) $\times$ 8 mm (H). The optical sensor board for prototyping purpose may appear bulky but can be shrunk easily by fabricating a flexible PCB, as the optical sensors themselves are small. Even without optimization, the whole system is lightweight and can be easily adhered onto the wrist area or further integrated into a wearable sensing platform such as a wrist band or a watch. The sensing location of the optical sensor is shown in Fig. 5.10.

The LED on-time is set to approximately 200 μs in every 16 ms by PWM control. Fast switching of LED causes the current surge due to the change of LED forward voltage. To suppress the inrush current, we use the current source to drive the LED. The driving current is fixed to 500 μA. Changing both the LED and the PD would make it hard to take the advantage of commercially available prototype.

5.2.2 Detection Accuracy

We tested our HGR system on three subjects, two male and one female. One important aspect of HGR is the noise rejection. In real situations, gestures coexist with random activities. These random activity should be rejected instead of being recognized as a valid

gesture. Otherwise, the system will become hard to use due to many false positives. We collected 40 non-gesture noise patterns, including random hand or wrist shake with both short and long wrist activities from each subject. Another 40 samples of each gesture from each person are also recorded. The total collection time is over 1 hour each. The system is implemented on a laptop computer to repeatedly study the data. The detection rate is defined in Eq. (5.1).

$$Detection\_Rate = 1 - \frac{True\_Negtives + False\_Positives}{Total\_Gestures} \tag{5.1}$$

As shown in Table 5.2, the detection rate is over 93% with the highest over 95%. We also implemented a single ACC-based HGR with the sliding window algorithm. A threshold is set to reject the noise based on the DTW score. The result clearly indicates that our system is superior to single ACC-based system in terms of noise rejection and detection accuracy.

The detailed result for each gesture is shown in Table 5.3. The detection rate is just the true positive rate, since no noise gesture is involved. Even though for some gestures, ACC-only results in slightly higher detection rate than the fusion method does, the fusion method helps improve the accuracy of the majority at a very small cost. The overall result shows the advantage of decision fusion over the single ACC-based method. The same gesture is usually done multiple times with different stopping points or angles to record multiple templates for each gesture. This multi-template setting helps eliminate ambiguity. Fig. 5.11 shows the influence of the weight γ in Eq. (4.2). The detection rates for all three people go beyond 93% when $\gamma = 66\%$.

Table 5.2: Total Detection Rate

Det. Alg.	Sub.	TN	FP	Dec. Rate
Acc+MTS	Male 1	18	4	93.89%
	Male 2	11	5	95.56%
	Female	18	3	94.17%
Acc	Male	25	17	88.33%
	Male 2	14	9	93.61%
	Female	31	7	89.44%

Table 5.3: Detection Rate of Each Gesture

Sub.	Gesture	Fusion Rate	Acc. Rate	Temp.	Length
Male 1 $\gamma = 0.66$	Up	92.5%	87.5%	4	240*ms*-928*ms*
	Dn/Rel.	90.0%	60.0%	6	304*ms*-1312*ms*
	Rot.$\leftarrow$	92.5%	95.0%	2	304*ms*-1120*ms*
	Rot.$\rightarrow$	90.0%	90.0%	2	288*ms*-1200*ms*
	Fist	97.5%	67.5%	2	448*ms*-1120*ms*
	Clicking	97.5%	32.5%	4	336*ms*-1040*ms*
	Left	95.0%	92.5%	1	480*ms*-1200*ms*
	Right	97.5%	100.0%	1	288*ms*-768*ms*
	Tol.	94.4%	78.1%	22	240*ms*-1312*ms*
Male 2 $\gamma = 0.82$	Up	97.5%	100.0%	5	512*ms*-1408*ms*
	Dn/Rel.	92.5%	95.0%	5	480*ms*-1264*ms*
	Rot.$\leftarrow$	97.5%	97.5%	4	592*ms*-2880*ms*
	Rot.$\rightarrow$	100.0%	100.0%	4	560*ms*-2384*ms*
	Fist	97.5%	67.5%	1	656*ms*-1376*ms*
	Clicking	92.5%	100.0%	3	1072*ms*-1488*ms*
	Left	100.0%	50.0%	2	784*ms*-1040*ms*
	Right	95.0%	45.0%	2	672*ms*-1296*ms*
	Tol.	96.56%	81.9%	26	480*ms*-2880*ms*
Female $\gamma = 0.75$	Up	97.5%	100.0%	4	784*ms*-1392*ms*
	Dn/Rel.	97.5%	100.0%	5	624*ms*-1424*ms*
	Rot.$\leftarrow$	92.5%	82.5%	2	1024*ms*-1840*ms*
	Rot.$\rightarrow$	92.5%	90.0%	2	1264*ms*-2544*ms*
	Fist	92.5%	80.0%	2	448*ms*-1408*ms*
	Clicking	90.0%	80.0%	3	688*ms*-2480*ms*
	Left	95.0%	77.5%	2	912*ms*-1488*ms*
	Right	90.0%	80.0%	1	912*ms*-1264*ms*
	Tol.	94.4%	86.3%	21	448*ms*-2544*ms*

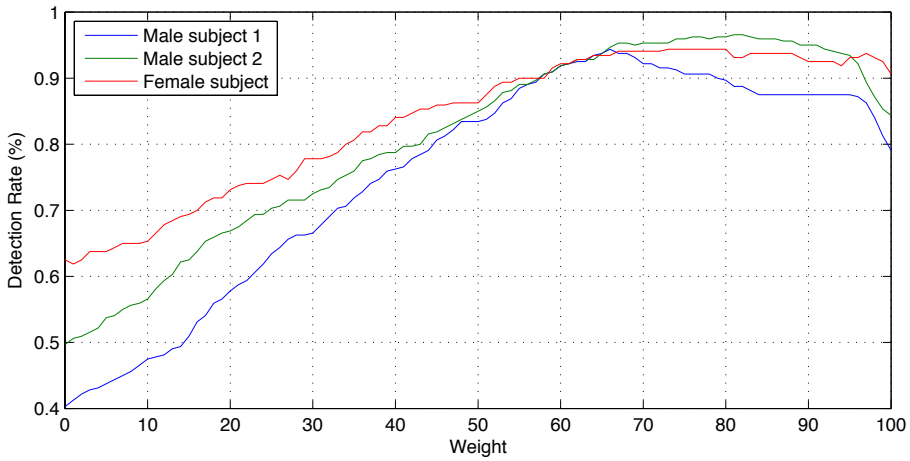


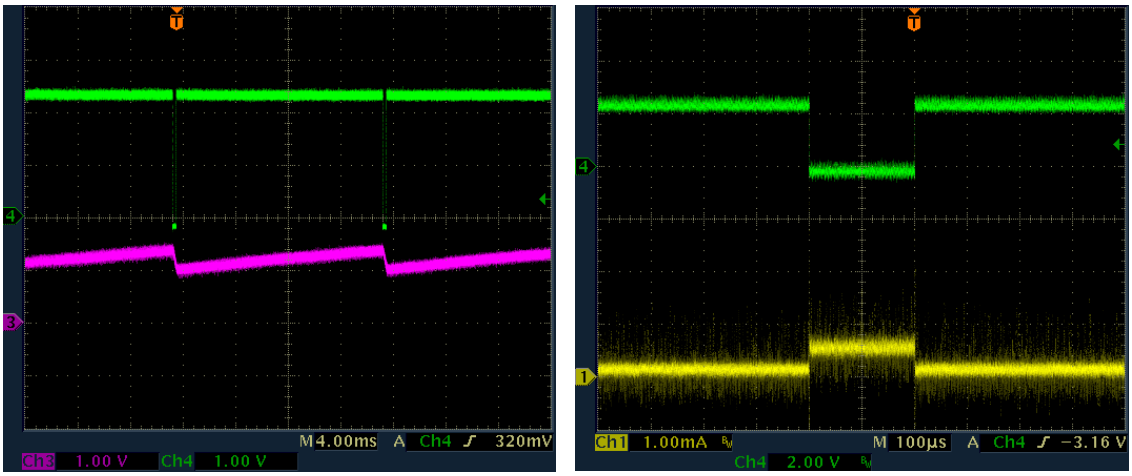
Figure 5.11: Decision Weight

5.2.3 Power Consumption

The power consumption of the system is measured using a current probe. The current values are integrated over a period of 5 minutes to calculate the average power consumption. Fig. 5.12 shows the periodic current signal of the optical sensor board. Even though the LED takes around 500 μA when turned on, the duty of PWM is around 1.25%. The optical sensor consumes around 258μW on average. The majority of the power is consumed by BLE communication sending and receiving data. The average power consumption of the CC2541 and MPU-9250 is around 57.5 mW. Thus, the power consumption increased by the MTS is only around 0.4% of the system power. The current system can last for more than 6 hours without recharging. Further optimization for future work includes lowering the SoC's Tx power, adjusting the sampling rate, and enabling threshold detection in the accelerometer.

5.2.4 Time Delay

The recognition system has also been ported to an iPhone4s. The measured computation delay is around 50 ms in the worst case with an average of 14 ms. Even with an expansion



(a) PWM and PD output. Green: PWM, pink: PD output

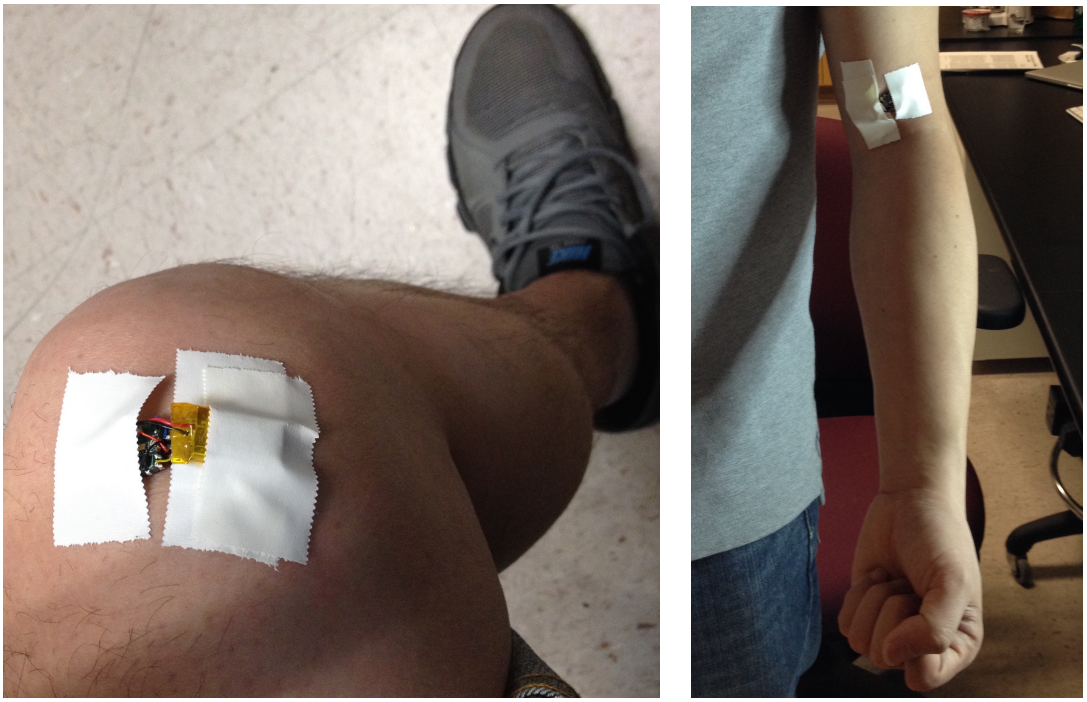
(b) Current consumption. Green: PWM, yellow: optical sensor board current signal

Figure 5.12: LED control and current consumption.

of around 30 gestures, the delay is under 100 ms.

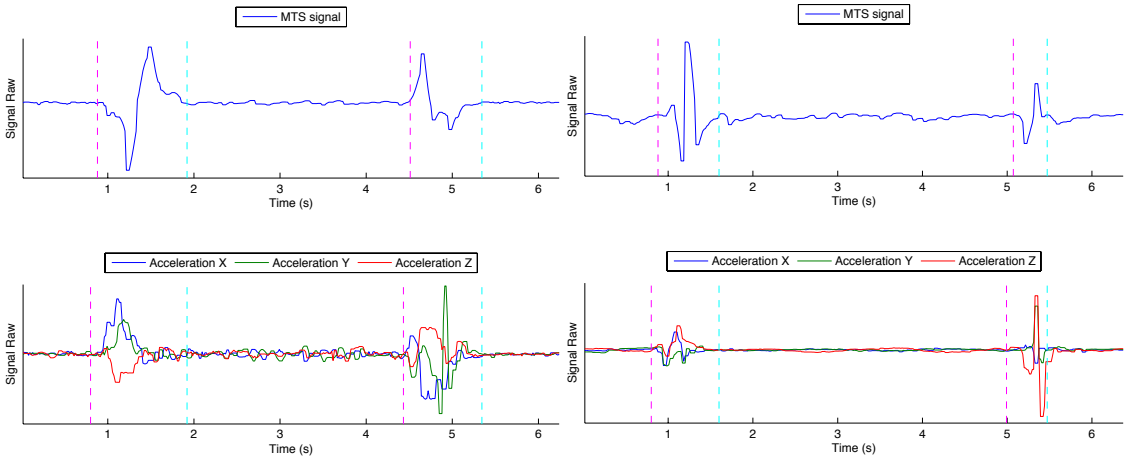
5.2.5 Other Applications

We also tested the system on elbow and knee joints. The sensing locations are shown in Fig. 5.13. Signal samples are shown in Fig. 5.14. The degree of freedom of movement is generally less than the wrist joint. The MTS signal can be used to detect the muscle movement around the joint while the ACC data provides the detailed movement, such as moving direction, timing, etc. The fusion of both MTS and ACC can be synergistic in classifying the movements.



(a) Knee (b) Elbow

Figure 5.13: Sensing locations on the knee and elbow.



(a) Knee bending and recovering while sitting (b) Elbow bending and recovering

Figure 5.14: Signals from the knee and elbow joint.

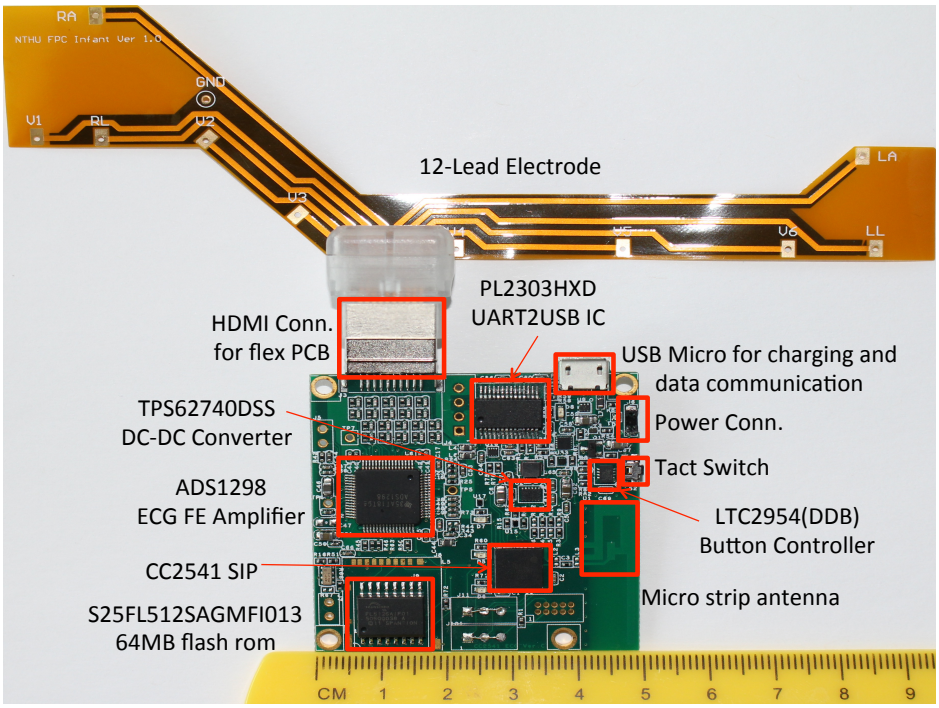
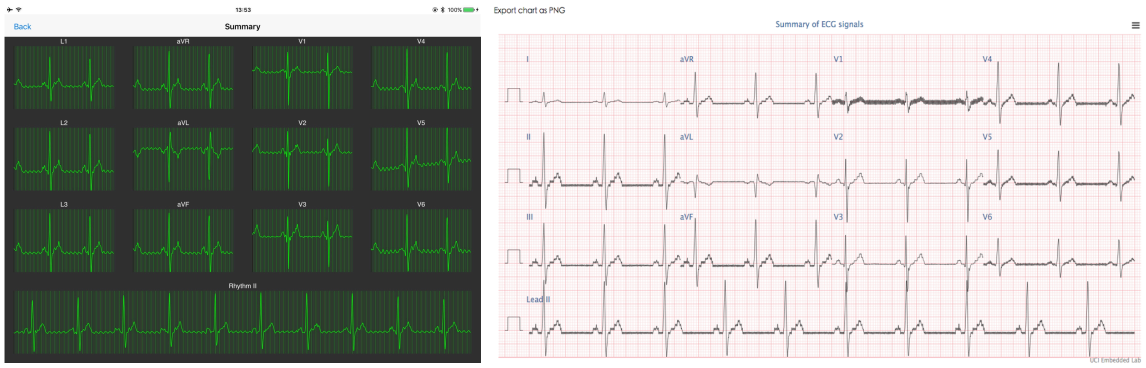


Figure 5.15: ECG hardware prototype

5.3 ECG Subsystem Test

5.3.1 ECG Prototype

The fabricated ECG board and a 12-lead electrode are shown on Fig. 5.15. The actual board uses TI's BLE SoC CC2541 [76] instead of CC2560. CC2541 is an integrated MCU and radio frequency transceiver SoC that runs TIs BLE protocol stack and the application code on a single chip. The device can record the ECG data into the flash memory and transfer the data to a server via BLE after recording. Fig. 5.16 shows a recorded signal and the real-time display on an iPad.



(a) iOS App. Display

(b) Web Server Display

Figure 5.16: ECG Signal

5.3.2 Compression Algorithm Evaluation

To evaluate the proposed compression algorithm, we use the records from MIT-BIH arrhythmia database [19]. A total of 48 records of 2-lead ECG signal from the database is partitioned into two sets for cross validation. The reconstruction results are compared with traditional CS approaches. The percentage root-mean-squared distortion (PRD) and compression ratio (CR) are calculated based on the original and reconstructed signal. PRD is given by

$$PRD = \sqrt{\frac{\sum_{i=1}^{n}(x[i]-\hat{x}[i])^2}{\sum_{i=1}^{n}x[i]^2}} \times 100 \tag{5.2}$$

where n is the total number of samples, and $x[i]$ and $\hat{x}[i]$ denote the original samples and the reconstructed samples respectively. CR is defined as

$$CR = \frac{B_{original} - B_{compressed}}{B_{original}} \times 100 \tag{5.3}$$

Table 5.4: Training and Testing Data Set

Training Set	Testing Set
100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 118, 119, 121, 122, 123, 124, 200, 201	117, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 217, 219, 220, 221, 222, 223, 228, 230, 231, 232, 233, 234

where $B_{original}$ and $B_{compressed}$ stand for the number of bits for the original and compressed signals respectively.

As shown in Table 5.4, 24 out of 48 records are used to train our signal model to build the overcomplete dictionary, while the rest are used to evaluate the performance. Some of the noisy data is avoided manually during the training process. This is because they do not represent the typical ECG complex. The final dictionary has a total of 5140 supports combining the trained 5012 atoms, 64 DCT basis and 64 wavelet basis (Daubechies 4) [78]. Each atom has 64 elements. Only 10 supports were used to reconstruct each record.

The original and reconstructed signals are shown in Fig. 5.17. Comparing with the traditional DCT, wavelet and the trained dictionary, the combined dictionary method achieved $PRD = 1.34\%$ with $CR = 50\%$, which means only half of the samples are randomly selected for the reconstruction.

The relation between CR and PRD for each reconstruction algorithm is shown in Fig. 5.18. The trend is obvious that a high CR leads to a high PRD. With combined dictionary, the PRD is greatly improved comparing with other methods.

5.4 Photoplethysmogram Subsystem Test

The PPG monitoring system is tested using an existing prototype as shown in Fig. 5.19. The system structure is the same as shown in Fig. 4.4 The DSP, TMS320C5515, will sample the PD output at a high rate of $1kHz$. A 351-tap finite impulse response (FIR) and infinite

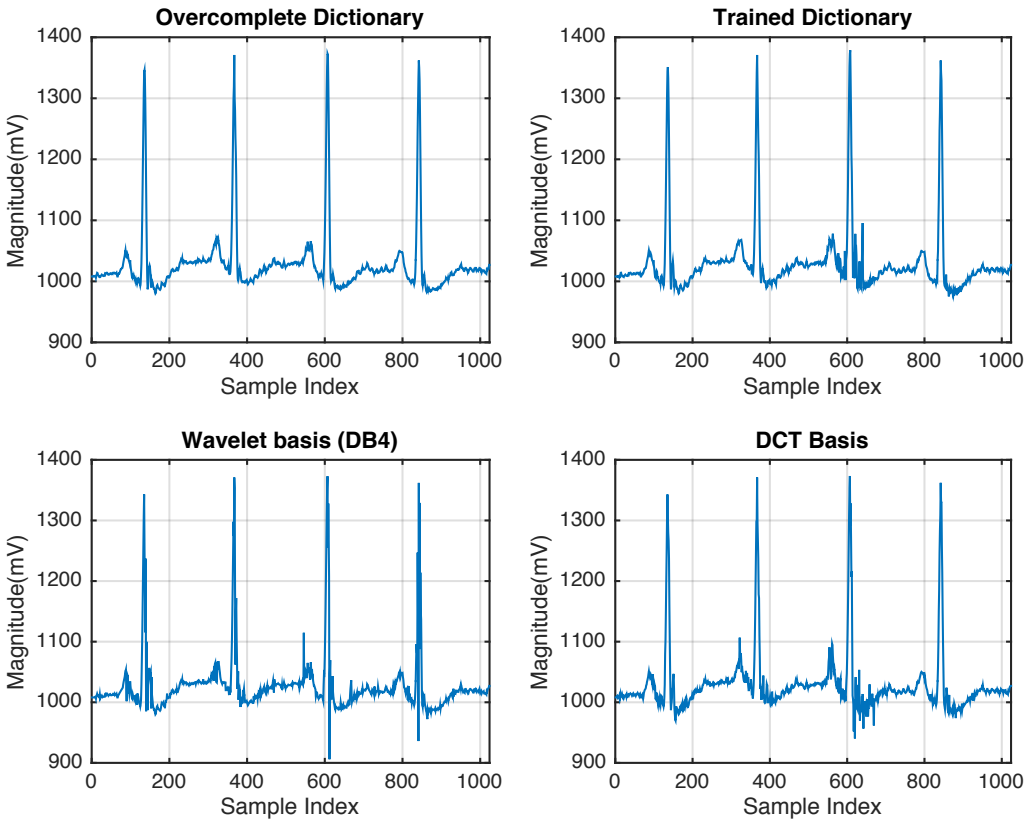


Figure 5.17: Reconstructed Signal

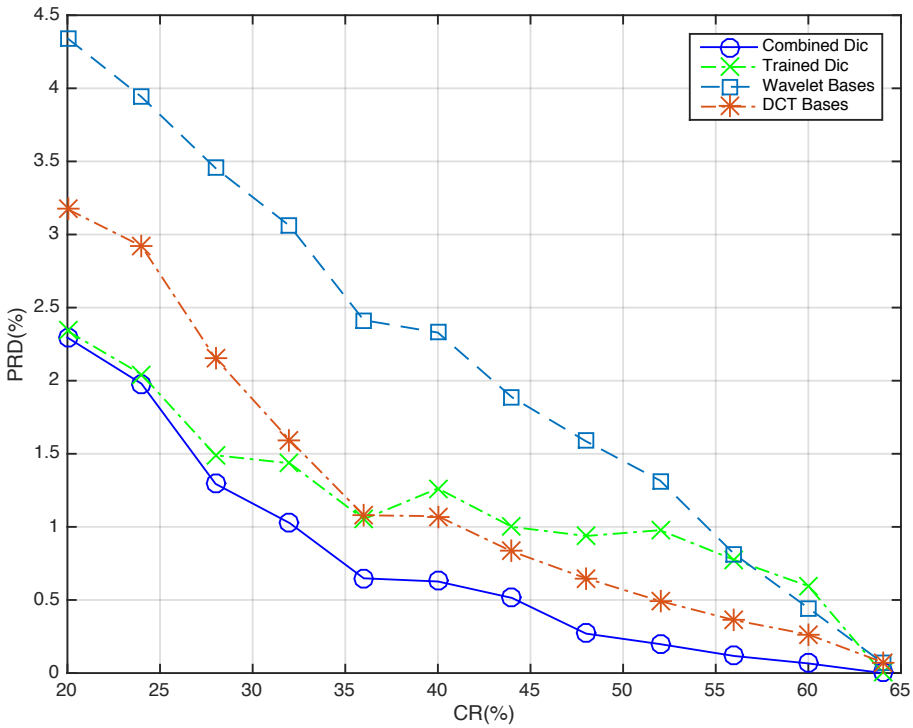


Figure 5.18: Compression Performance Comparison

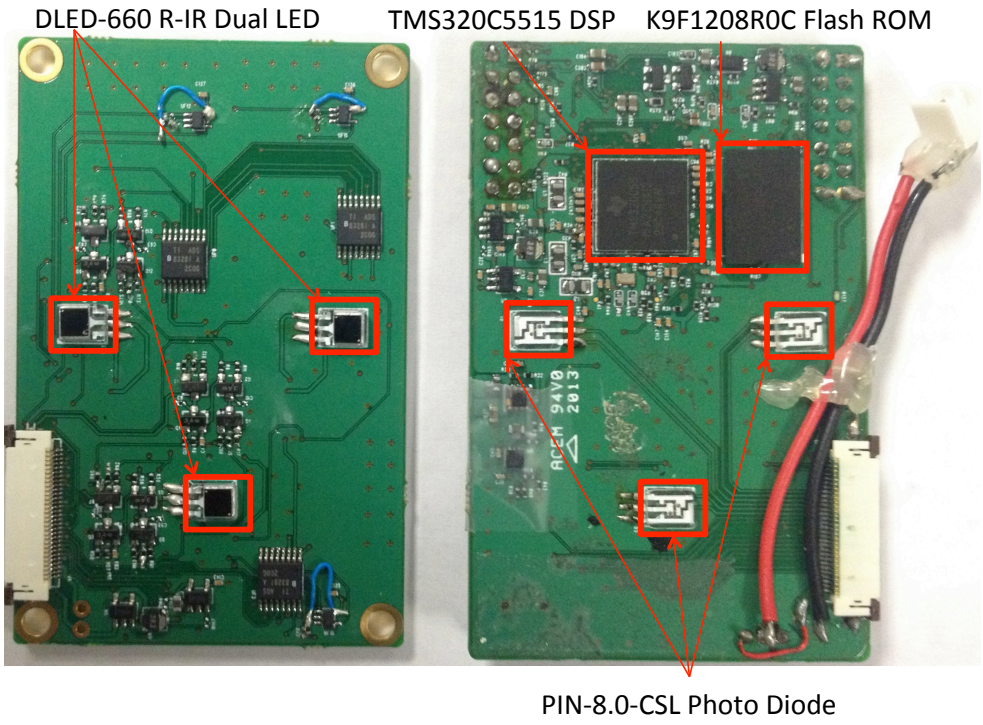


Figure 5.19: PPG Monitor Prototype

impulse response (IIR) based band-pass filter of $0.1Hz - 10Hz$ is implemented in DSP to remove the noise and motion artifact. More information about the filter can be found in [54].

5.4.1 Real Time SpO2 and HR Calculation

A real-time sample-by-sample heart rate and SpO2 detection algorithm is implemented to test the accuracy of the SpO2 sensor. The DSP keeps a dynamic threshold equals to 70 % of the maximum AC peak value. Once the sampled value passes this threshold for a second time, a valid cardiac cycle is detected and an instantaneous heart rate is calculated using (5.4). The red to infrared RMS ratio of the detected cardiac cycle is then calculated using (B.4) and (B.7). An average of 10 instantaneous heart rates is reported to the user as in (5.5).

Table 5.5: SpO2 Calibration Table

ratio	SpO2
>97%	$ratio + \frac{ratio}{50}$
96%-97%	$ratio + \frac{ratio}{75}$
92%-95%	$ratio + \frac{ratio}{100}$
92%-95%	$ratio + \frac{ratio}{100}$
89%-92%	$ratio$
87%-89%	$ratio - \frac{ratio}{100}$
85%-87%	$ratio - \frac{ratio}{50}$
<85%	$ratio - \frac{ratio}{33.3}$

$$INST\_HR = \frac{60 * sampling\_rate}{(cont_{2^{nd}\_peak} - cont_{1^{st}\_peak})} \tag{5.4}$$

$$HR = \frac{\sum_{i=1}^{10} INST\_HR_i}{10} \tag{5.5}$$

The curve of the ratio versus SpO2 is further calibrated using Table 5.5.

The evaluation is carried out using Fluke Pulse Oximter Functional Tester [73]. The tester works like a reversed pulse oximeter. It mimics the real fingers by using a group of PDs to detect the light intensity and using LEDs to emit light according to the PD output and the preset SpO2 level. The SpO2 level can be easily set from 70 %-100 %. The reading from our prototype is within $\pm 2\%$ of the tester which shows the correctness of the sensor and the calibration.

5.4.2 Compressive Sensing Based Heart Rate Detection

To test the compressive sensing based heart rate detection algorithm, we continuously divide the digital samples from MIMIC database into 1024-point segments and run our algorithm on them. The sampling rate is 125 Hz. The number of random samples is 192 over each DCT segment. We find it is very difficult to get a stable performance of the sparse reconstruction when given fewer samples. Some other parameters for the experiment are shown as follows.

a) interval $\alpha = 0.1$ with $z_{\alpha/2} = 1.645$

b) Threshold on dominance degree M_L: 5

c) Threshold out of the range M_R: 4

d) Tolerance β: 1 DCT bin

e) CoSaMP stopping criteria: residual l_2-norm $< 10^{-10}$, or residue l_2-norm difference between current and last iteration $< 10^{-10}$, or iteration count > 100

The main result is shown in Table 5.6. The heart rate on some segments is detected just by prior estimation while others by beat-by-beat analysis after reconstruction. The error columns in the table are average error in the unit of bpm. We manually removed some segments severely polluted by noise. Examples of such segments are shown in Fig. 5.20. The results show that our algorithm is effective to over 99% of the data.

The reference heart rate is calculated by two algorithms, a beat-by-beat peak searching algorithm [79] for the reconstructed signal and a DCT algorithm for the signal with the prior estimation. For the peak searching algorithm, the signal first goes through a band pass filter (BPF). Then each sample is compared with the thresholds for peak detection.

Table 5.6: Detection rate

Patient #	Seg. Tol.	Det. Rate.	Recons. Tol.	Recons. Err.	Prior Esti. Tol.	Prior Esti. Err.
55	14206	99.58%	5363	0.08	8843	0.25
208	433	95.61%	399	0.30	34	0.76
209	8590	94.99%	8037	0.25	553	0.41
210	5835	95.29%	5326	0.23	509	0.89
211	8866	99.37%	4750	0.05	4116	0.29
212	17570	99.88%	1951	0.09	15619	0.32
216	10107	98.43%	2773	0.28	7334	0.45
218	8792	97.71%	4357	0.19	4435	0.45
219	9638	99.22%	2935	0.10	6703	0.35
220	482	99.99%	190	0.00	292	0.41
221	10087	99.96%	917	0.07	9170	0.28
224	20199	99.68%	4879	0.16	15320	0.28
225	17935	99.64%	6357	0.07	11578	0.34
226	12204	99.56%	2346	0.13	9858	0.29
230	3400	99.94%	1006	0.00	2394	0.28
231	18083	99.89%	15441	0.01	2642	0.27
237	16318	99.53%	4590	0.10	11728	0.34
252	10901	99.75%	2371	0.10	8530	0.30
430	2753	98.80%	1658	0.13	1095	0.30
437	19978	99.26%	5601	0.19	14377	0.27
438	19746	98.27%	18074	0.08	1672	0.56
439	19872	99.82%	2855	0.11	17017	0.24
446	10511	97.47%	9043	0.13	1468	0.19
449	6353	99.98%	1659	0.01	4694	0.53
451	11661	99.81%	4599	0.02	7062	0.29
452	13861	97.35%	10154	0.21	3707	0.28
453	17461	96.08%	14554	0.21	2907	0.46
454	14374	95.11%	11776	0.25	2598	0.53
456	15776	99.32%	6428	0.14	9348	0.27
466	5927	99.95%	1132	0.02	4795	0.33
471	25291	98.66%	10036	0.16	15255	0.37
472	2997	94.56%	2145	0.31	852	0.60
474	3850	99.92%	97	0.14	3753	0.31

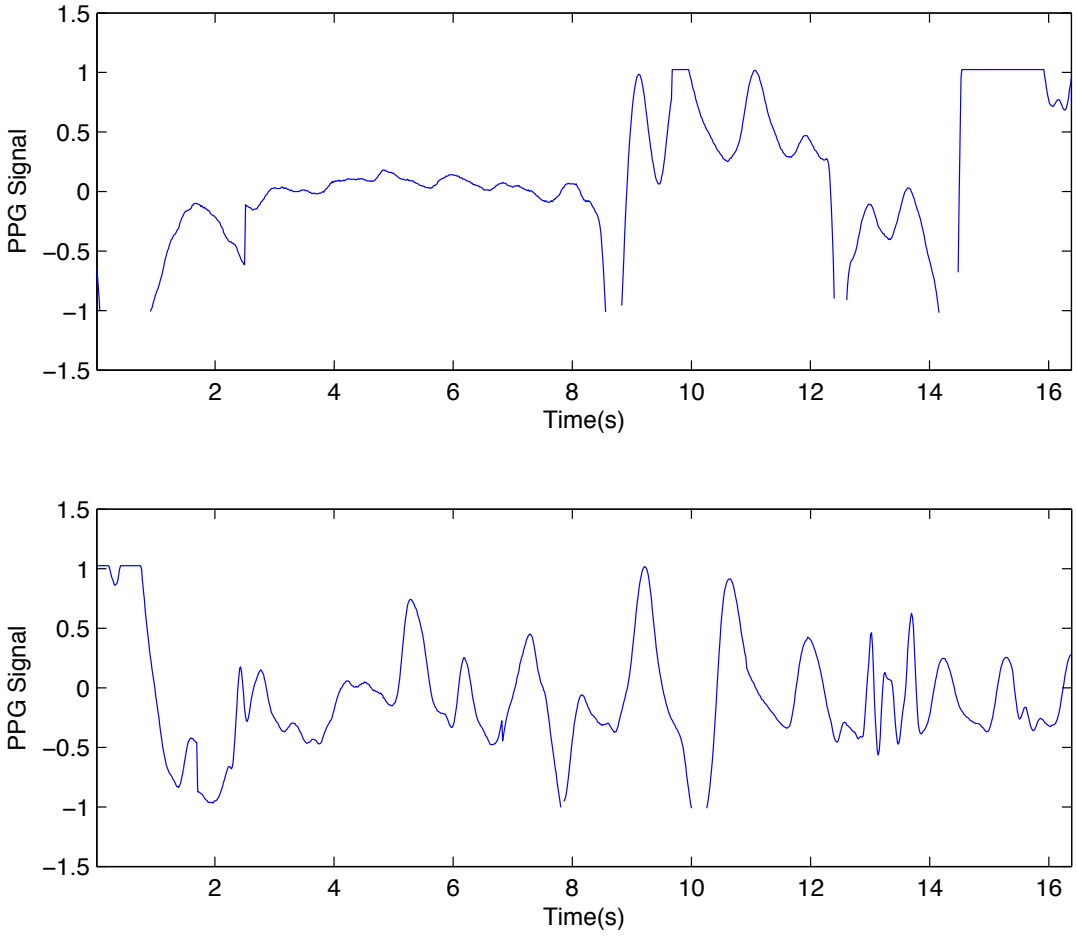


Figure 5.20: Corrupted and noisy PPG signal segments. Top: data containing NaN (not a number), bottom: data with high noise

Two sets of adaptive threshold are maintained for peaking searching, one for heart beat and the other for noise peaks. For the DCT algorithm, a full DCT is performed to the original signal to compute the heart rate.

The average iteration count of the reconstruction over different sparsity levels is shown in Fig. 5.21. Again, the data of patient No. 208 is used. The traditional OMP algorithm would take at least 50% more iterations since the OMP iteration count is roughly the same with the size of support.

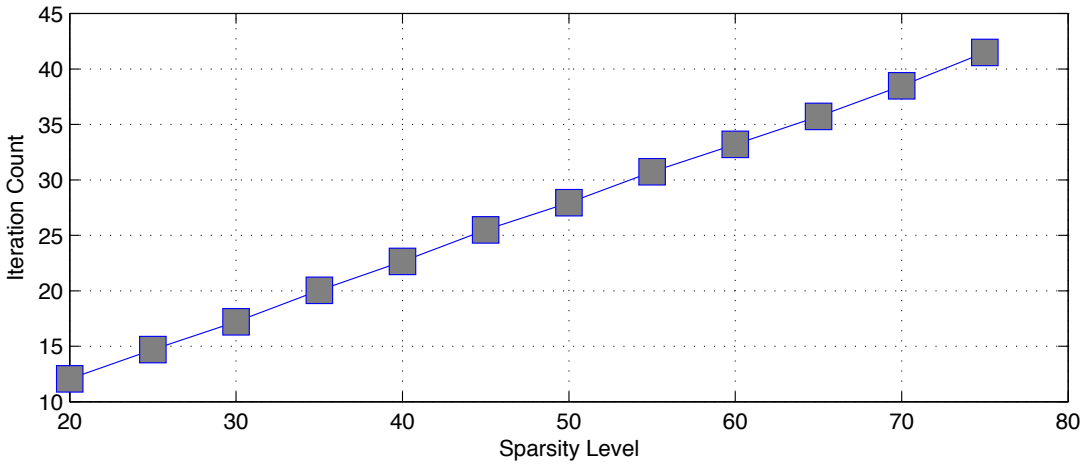


Figure 5.21: Iteration count vs. sparsity level

Power Consumption The under sampling rate (*USR*) is generally used to quantify the rate of random sampling. It is equal to N/k with the same N and k defined in Appendix Section 3.3. For our work, the *USR* is around 5. The previous works claim they can reconstruct the PPG signal with a *USR* of under 20 without losing much information [29]. However, the target of our reconstruction algorithm is the data with high level noise and high heart rate variation. The prior estimation stage will naturally screen the signal to avoid reconstruction for the more sparse signal generally shown in other works. We believe the setting of our experiment is closer to the real applications.

The power saving mainly comes from the optional reconstruction in the back end. It is hard to estimate the actual effect without the detailed information. However, as shown in Table 5.6, for PPG signal with a stable heart rate, over 99% of the reconstruction can be saved with very little power increase in the front end. The system response is also greatly improved.

As a justification for the power saving effect of CS, the LED power saved at the *USR* of 5 is around 0.99 mW. This alone is already around 16% of the system power. The PPG samples are taken and processed during the interrupt service routine (ISR) by the DSP. The parameters used for this calculation are shown as follows. These data is measured from the

real prototype board.

Average ISR time:	600 μs [28]
DSP active power:	59.1 mW@100MHz [80]
DSP standby power:	0.7 mW [80]
IR LED on power:	19.8 mW (60mA at 3.3V)
IR LED on time:	50 μs
Sampling rate:	125 Hz

The system takes 6.32 mW overall before applying CS. Over 50% of the power is actually saved from setting the DSP to standby mode than turning off the LED.

One would think to uniformly sample the signal at 25 Hz and detect the heart rate. We run our peak searching algorithm to the down-sampled data from patient No. 208 and find a maximum heart rate error of over 1.8 bpm for the same data set used in Table 5.6. This is much larger than the error after the sparse reconstruction, which is around 0.6 bpm. Several heart beat peaks are detected as noise peaks when using down-sampled signal. Even though the frequency range of interest is 0.4-5 Hz [23] for the PPG signal, detection algorithms require a higher sampling rate to tolerate noise and irregular heart beats. Most of the systems are actually running at high sampling rates of 100-1k Hz [54, 81–85].

Discussion Increasing the number of random samples inevitably increases the power consumption, but it actually discourages reconstruction since the probability of discovering the dominant cardiac frequency in prior estimation becomes higher. It would be very interesting to find an optimal point for power consumption on a real platform. This will be done for our future work.

For the segments needing reconstruction, we did observe some significant deviation from beat-by-beat analysis as large as 7–10 bpm, but we ascribe that to the environment noise and

corrupted samples such as the data shown in Fig. 5.20. Our implementation of the detection algorithm could not even yield a reliable result under these conditions. The overwhelmingly large number of the total segments still validates the correctness of our algorithm.

For the segments estimated with the prior, the error is mainly caused by the spectral leakage as discussed in Section 4.2.3.2. One obvious way to deal with the error is to decrease the minimum sampling interval, i.e. shorten the width of each DCT bin. Another is to change the signal to a zero-padded DCT as discussed in [64]. This topic is beyond the scope of this work.

Our sparse reconstruction obviously takes advantage of the low-pass filtering to the source signal, as it makes the source signal sparse. This could be implemented in hardware [86] without significantly increasing the power consumption. The intuition is that the proposed CoSaMP should also be effective on unfiltered data. The components outside the cardiac frequency range have little effect on the heart rate. As our future work, we will acquire unfiltered data using real hardware to confirm the performance of our algorithm.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation describes a low-power wearable medical sensing platform. A combination of physiological signals can be monitored by this system including ECG, PPG and muscle tension. It also contains a HGR system that can be used as a human-machine interface to aid mobility impaired patients. The system power is optimized by low-power component selection and advanced digital signal processing. Several hardware prototypes are developed to test the performance of this system. Simulation and real hardware measurements show that our platform is superior to the existing systems in terms of power consumption and system size. Several sensing sub-systems presented in this work can be used directly to build stand-alone medical devices. The software and hardware presented in this work can serve as a design guideline or reference for medical system designer to greatly shorten the development cycles.

This work also explores the theory and application of compressive sensing framework in physiological signal processing research. A novel compressive sensing based ECG com-

pression algorithm and a dominant frequency extraction based PPG heart rate calculation algorithm are proposed to reduce the system power. Simulation results show the compressive sensing effective in power saving without compromising the performance.

Our design experience shows some profound implications:

- The signal properties should be thoroughly studied in order to reduce the complexity of the processing algorithm.
- No need to make universal applicable system. The compatibility often poses overhead to the system power and response speed.
- No single isolated technique can work while dealing with the integrated system as a whole. System level coordination and trade-offs have to be made to realize the power optimization.

6.2 Future Work

Except for the tasks already mentioned in previous chapters, we have a few more to be done in future. The first is to actually implement the compressive sensing based PPG HRM in a real hardware platform to evaluate the power consumption and accuracy. We also want to perform a similar evaluation on SpO2 calculation. We can not present it in this dissertation due to the lack of clinical data even though Equation (B.4) in Section 5.4.1 shows this is highly possible. Secondly, we want to prototype the whole system including the back-end server to test the overall performance. Due to the cost and time considerations, we are unable to finish it at this point. Finally, the clinical trail is essential to medical system design. We will seek opportunities to perform the trail.

Appendices

A Discrete Cosine Transform

DCT is widely used in image and audio signal compression. It is similar to Discrete Fourier Transform (DFT) basis but using only real numbers.

$$x_n = \sum_{k=0}^{N-1} \alpha(k) a_k \cos\left[\frac{\pi(2n+1)}{2N}k\right], n \in [0, N-1]$$
$$a_n = \alpha(n) \sum_{k=0}^{N-1} x_k \cos\left[\frac{\pi(2k+1)}{2N}n\right], n \in [0, N-1] \quad \text{(A.1)}$$

In both equations, $\alpha(n)$ is defined as:

$$\alpha(n) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } n = 0, \\ \sqrt{\frac{2}{N}} & \text{if } n \neq 0 \end{cases} \quad \text{(A.2)}$$

$x = [x_1 x_2 \cdots x_N]^T$ is the time-domain observation, and $a = [a_1 a_2 \cdots a_N]^T$ is the DCT coefficient vector. The following equation transfers an index in N-point DCT into the corresponding frequency. id is the DCT index number, f_s is the sampling rate, and f_t is the frequency.

$$f_t = \frac{f_s \times id}{2N}; \quad \text{(A.3)}$$

For simplicity, we directly use frequency as the DCT index omitting the projection.

B SpO2 Calculation

Photoplethysmogram uses pulse oximetry to measure the absorbance of light due to the pulsatile arterial blood. A single infrared LED or a pair of infrared and red LEDs emit light into body tissues, and one or multiple photo receivers measure the transmittance or reflectance of the light. The system structure of a typical transmittance-type pulse oximeter based on a DSP is shown in Fig. 4.4. The DSP finds the best SNR by controlling the LED driver and both amplifiers based on the voltage level of both amplified and raw inputs from the photoreceiver. Heart rate and SpO2 are the two most basic indices that can be extracted from PPG. Among several signal-processing techniques used, the most common one is called *weighted moving average* (WMA) [23]. It is a beat-by-beat algorithm that detects signal peaks in a PPG signal and computes the average HR and SpO2. SpO2 can be estimated from the ratio of R to IR signal. A traditional approximation formula from work [23] is shown in (B.4). In practice, RMS value is usually used for approximation (B.7).

$$\text{Ratio} = \frac{AC(\text{Red})}{DC(\text{Red})} \Big/ \frac{AC(\text{IR})}{DC(\text{IR})} \qquad \text{SpO2} = -2.5\text{Ratio} + 110 \tag{B.4}$$

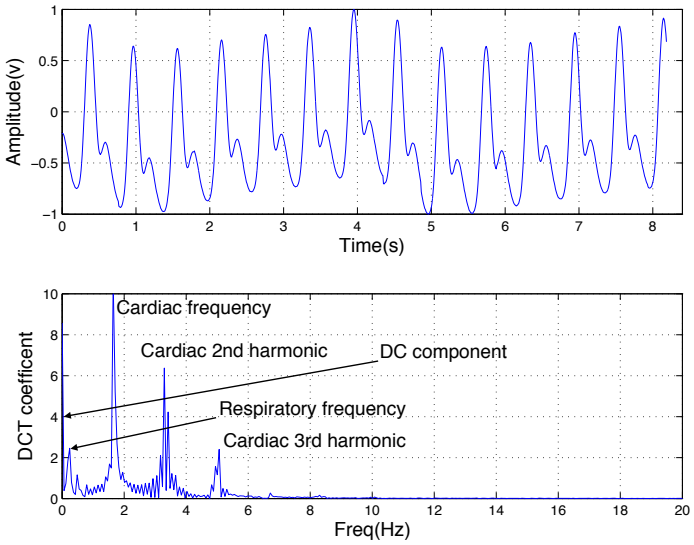


Figure B.1: Typical PPG signal and its DCT transform. Top: PPG signal from patient No. 55 in MIMIC database, bottom: absolute value of the DCT coefficients

$$R_{rms} = \sqrt{\frac{\sum_{i=1}^{n} R_i^2}{n}} \tag{B.5}$$

$$IR_{rms} = \sqrt{\frac{\sum_{i=1}^{n} IR_i^2}{n}} \tag{B.6}$$

$$ratio = 1.1 - 0.25\frac{R_{rms}}{IR_{rms}} \tag{B.7}$$

The SpO2 also needs to be calibrated with a reference curve and a co-oximeter [5].

DFT and DCT are also shown as alternative measures [5, 23, 87]. In contrast to WMA, these are known as transform-based algorithms. They transform a series of time domain samples to a frequency domain signal and perform spectral analysis. A typical PPG signal and the magnitude of DCT coefficients are shown in Fig. B.1. HR can be calculated from the highest peak between 0.5-2.5 Hz. The AC and DC components can be calculated from the cardiac line and DC line, respectively. Thus, SpO2 can be calculated from the ratio.

Bibliography

[1] Catherine Klersy, Annalisa De Silvestri, Gabriella Gabutti, François Regoli, and Angelo Auricchio. A meta-analysis of remote monitoring of heart failure patients. *Journal of the American College of Cardiology*, 54(18):1683–1694, 2009.

[2] Rich Nazarian. Developing medical devices in a consumer-driven market. *Medical Device & Diagnostic Industry (MDDI)*, 2009.

[3] Anthony F Jerant, Rahman Azari, and Thomas S Nesbitt. Reducing the cost of frequent hospital admissions for congestive heart failure: a randomized trial of a home telecare intervention. *Medical care*, 39(11):1234–1245, 2001.

[4] Jürgen Teich. Hardware/software codesign: The past, the present, and predicting the future. *Proceedings of the IEEE*, 100(Special Centennial Issue):1411–1430, 2012.

[5] J.E. Scharf, S. Athan, and D. Cain. Pulse oximetry through spectral analysis. In *Biomedical Engineering Conference, 1993., Proceedings of the Twelfth Southern*, pages 227–229, 1993.

[6] Emily Blem, Jaikrishnan Menon, and Karthikeyan Sankaralingam. Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures. In *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pages 1–12. IEEE, 2013.

[7] S.M.A. Hussain and A.B.M.H. Rashid. User independent hand gesture recognition by accelerated DTW. In *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, pages 1033–1037, May 2012.

[8] ADS1298 ECG Amplifier Info. Page. `http://www.ti.com/product/ADS1298`.

[9] M. Tavakoli, L. Turicchia, and R. Sarpeshkar. An ultra-low-power pulse oximeter implemented with an energy-efficient transimpedance amplifier. *Biomedical Circuits and Systems, IEEE Transactions on*, 4(1):27–38, Feb 2010.

[10] S. Qaisar, R.M. Bilal, W. Iqbal, M. Naureen, and Sungyoung Lee. Compressive sensing: From theory to applications, a survey. *Communications and Networks, Journal of*, 15(5):443–456, Oct 2013.

[11] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *ArXiv e-prints*, March 2008.

[12] Frank G Yanowitz. Characteristics of the normal ECG, 2004.

[13] Texas Instruments. MSP430. `http://www.ti.com/lsds/ti/microcontroller/16-bit_msp430/overview.page`.

[14] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, PhysiotoolKit, and PhysioNet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

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

[16] StoneStreetOne. BlueTopiaLE. `http://www.stonestreetone.com/bluetopiaLE.cfm`.

[17] Murugavel Raju. Heart rate and EKG monitor using the MSP430FG439. *Texas Instruments, Dallas, Tex, USA*, 2005.

[18] Jiapu Pan and Willis J. Tompkins. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, (3):230–236, 1985.

[19] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[20] Sourabh Ravindran, Steven Dunbar, and Bhargavi Nisarga. Real-time, low-complexity, low-memory solution to ECG-based heart rate detection. In *Conference proceedings:... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, volume 2009, pages 1371–1374, 2008.

[21] L Couts. A designer's guide to instrumentation amplifiers. chap. 1, 2006.

[22] Peter Konrad. The ABC of EMG. *A practical introduction to kinesiological electromyography*, 1, 2005.

[23] T.L. Rusch, R. Sankar, and J.E. Scharf. Signal processing methods for pulse oximetry. *Computers in Biology and Medicine*, 26(2):143 – 159, 1996.

[24] COBP photo reflector with RED and IR LED. `http://www.njr.com/semicon/products/NJL5501R.html`.

[25] Alessio Chianura and Mario E Giardini. An electrooptical muscle contraction sensor. *Medical & biological engineering & computing*, 48(7):731–734, 2010.

[26] Hyonyoung Han and Jung Kim. Novel muscle activation sensors for estimating of upper limb motion intention. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3767–3770, Sept 2009.

[27] Ashu K. Bansal, Shuoben Hou, Olena Kulyk, Eric M. Bowman, and Ifor D. W. Samuel. Wearable organic optoelectronic sensors for medicine. *Advanced Materials*, 2014.

[28] Pawan K. Baheti, Harinath Garudadri, and Somdeb Majumdar. Blood oxygen estimation from compressively sensed photoplethysmograph. In *Wireless Health 2010*, WH '10, pages 10–14, New York, NY, USA, 2010. ACM.

[29] Pawan K. Baheti and Harinath Garudadri. Heart rate and blood pressure estimation from compressively sensed photoplethysmograph. ICST, 5 2010.

[30] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, Feb 2006.

[31] TI Chronos Watch. `http://www.ti.com/tool/ez430-chronos`.

[32] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, Jan 2006.

[33] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, May 2007.

[34] David Mace, Wei Gao, and Ayse K Coskun. Improving accuracy and practicality of accelerometer-based hand gesture recognition. In *2nd Workshop on Interacting with Smart Objects, in conjunction with the ACM International Conference on Intelligent User Interfaces (IUI)*. Citeseer, 2013.

[35] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657 – 675, 2009. PerCom 2009.

[36] Xu Zhang, Xiang Chen, Yun Li, V. Lantz, Kongqiao Wang, and Jihai Yang. A framework for hand gesture recognition based on accelerometer and emg sensors. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(6):1064–1076, Nov 2011.

[37] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 401–406, New York, NY, USA, 2009. ACM.

[38] J. Cannan and Huosheng Hu. A multi-sensor armband based on muscle and motion measurements. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 1098–1103, Dec 2012.

[39] J.A. Tropp and A.C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, Dec 2007.

[40] Van H Vu and Terence Tao. The condition number of a randomly perturbed matrix. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 248–255. ACM, 2007.

[41] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[42] Pier Luigi Dragotti and Yue M Lu. On sparse representation in fourier and local bases. *Information Theory, IEEE Transactions on*, 60(12):7888–7899, 2014.

[43] Stefan Kunis and Holger Rauhut. Random sampling of sparse trigonometric polynomials, ii. orthogonal matching pursuit versus basis pursuit. *Found. Comput. Math.*, 8(6):737–763, November 2008.

[44] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing mri. *Signal Processing Magazine, IEEE*, 25(2):72–82, March 2008.

[45] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza. A real-time compressed sensing-based personal electrocardiogram monitoring system. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1–6, March 2011.

[46] CC2540 Bluetooth SoC. `http://www.ti.com/product/cc2540`.

[47] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597, Dec 2007.

[48] Pawan K. Baheti and H. Garudadri. An ultra low power pulse oximeter sensor based on compressed sensing. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 144–148, June 2009.

[49] Programmable current source LT3092. `http://www.linear.com/product/LT3092`.

[50] Kenta Matsumura, Peter Rolfe, Jihyoung Lee, and Takehiro Yamakoshi. iPhone 4s photoplethysmography: Which light color yields the most accurate heart rate and normalized pulse volume using the iPhysioMeter application in the presence of motion artifact? *PLoS ONE*, 9(3):e91205, 03 2014.

[51] Ultralow Power Comparator LTC1440. `http://www.linear.com/product/LTC1440`.

[52] MPU9250 motion sensor. `http://www.invensense.com/mems/gyro/mpu9250.html`.

[53] Daniel Roetenberg, Henk J Luinge, Chris TM Baten, and Peter H Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 13(3):395–405, 2005.

[54] Pulse Oximeter Implementation on the TMS320C5515 DSP. `http://www.ti.com/lit/an/sprab37a/sprab37a.pdf`.

[55] S G Kelsen, D P Kelsen, B F Fleeger, R C Jones, and T Rodman. Emergency room assessment and treatment of patients with acute asthma. adequacy of the conventional approach. *Am J Med*, 64(4):622–628, Apr 1978.

[56] J O Commey and H Levison. Physical signs in childhood asthma. *Pediatrics*, 58(4):537–541, Oct 1976.

[57] Isabelle Stockman. Neck muscle influence in rear impacts–a sled test study using the BioRID. 2010.

[58] M.A. Mananas, J.A. Fiz, J. Morera, and P. Caminal. Analyzing dynamic EMG and VMG signals of respiratory muscles. *Engineering in Medicine and Biology Magazine, IEEE*, 20(6):125–132, Nov 2001.

[59] Manuel Grodinsky and Edward A. Holyoke. The fasciae and fascial spaces of the head, neck and adjacent regions. *American Journal of Anatomy*, 63(3):367–408, 1938.

[60] A. Homs-Corbera, J.A. Fiz, J. Morera, and R. Jane. Time-frequency detection and analysis of wheezes during forced exhalation. *Biomedical Engineering, IEEE Transactions on*, 51(1):182–186, Jan 2004.

[61] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Di Marca. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, 10(5):285–299, July 2006.

[62] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, Feb 1978.

[63] Steven M. Kay. *Modern spectral estimation : theory and application*. Prentice Hall signal processing series. Prentice Hall, Englewood Cliffs (N.J.), 1988.

[64] Marco F. Duarte and Richard G. Baraniuk. Spectral compressive sensing. *Applied and Computational Harmonic Analysis*, 35(1):111 – 129, 2013.

[65] A.C. Gilbert, M.J. Strauss, and J.A. Tropp. A tutorial on fast fourier sampling. *Signal Processing Magazine, IEEE*, 25(2):57–66, March 2008.

[66] R.V. Hogg and E.A. Tanis. *Probability and Statistical Inference*. Number v. 978, nos. 0-58475 in Probability and Statistical Inference. Prentice Hall, 2010.

[67] Wen-Hsiung Chen, C. Smith, and S. Fralick. A fast computational algorithm for the discrete cosine transform. *Communications, IEEE Transactions on*, 25(9):1004–1009, Sep 1977.

[68] C5000 DSP Info. Page. `http://www.ti.com/lsds/ti/dsp/c5000_dsp/overview.page`.

[69] Vrudhula K Murthy, Thomas M Grove, George A Harvey, and L Julian Haywood. Clinical usefulness of ecg frequency spectrum analysis. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 610. American Medical Informatics Association, 1978.

[70] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[71] Michal Aharon, Michael Elad, and Alfred Bruckstein. Svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[72] Sparselab Toolbox for MATLAB. `http://sparselab.stanford.edu/`.

[73] Fluke Pulse Oximter Functional Tester. `http://www.flukebiomedical.com/Biomedical/usen/Patient-Simulators/ProSim-SPOT-Light-SpO2-Tester-Pulse-Oximeter-Analyzer.htm?PID=74249`.

[74] Pulse Sensor. `http://pulsesensor.com`.

[75] Muscle Sensor v3. `http://www.advancertechnologies.com/p/muscle-sensor-v3.html`.

[76] CC2541 SimpleLink Bluetooth Smart and Proprietary Wireless MCU. `http://www.ti.com/product/cc2541`.

[77] Low power instrumentation amplifer. `http://www.analog.com/en/products/amplifiers/instrumentation-amplifiers/ad8221.html`.

[78] Ali N Akansu, Richard A Haddad, and Hakan Caglar. The binomial qmf-wavelet transform for multiresolution signal decomposition. *IEEE transactions on signal processing*, 41(1):13–19, 1993.

[79] FC Chang, CK Chang, CC Chiu, SF Hsu, and YD Lin. Variations of hrv analysis in different approaches. In *Computers in Cardiology, 2007*, pages 17–20. IEEE, 2007.

[80] Power Consumption Summary for C5000 DSP. `http://processors.wiki.ti.com/index.php/TMS320C5504/05/14/15/32/33/34/35_Power_Consumption_Summary`.

[81] Zhilin Zhang, Zhouyue Pi, and Benyuan Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *Biomedical Engineering, IEEE Transactions on*, 62(2):522–531, Feb 2015.

[82] Jaywoo Kim, Mi-hee Lee, Hyoung-Ki Lee, Kiwan Choi, SeokWon Bang, and Sangryong Kim. Heart rate monitor for portable mp3 player. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5207–5210. IEEE, 2006.

[83] Sokwoo Rhee, Boo-Ho Yang, and Haruhiko Harry Asada. Artifact-resistant power-efficient design of finger-ring plethysmographic sensors. *Biomedical Engineering, IEEE Transactions on*, 48(7):795–805, 2001.

[84] R Couceiro, P Carvalho, RP Paiva, J Henriques, and J Muehlsteff. Detection of motion artifacts in photoplethysmographic signals based on time and period domain analysis. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 2603–2606. IEEE, 2012.

[85] Rajet Krishnan, Balasubramaniam Natarajan, and Steve Warren. Two-stage approach for detection and reduction of motion artifacts in photoplethysmographic data. *Biomedical Engineering, IEEE Transactions on*, 57(8):1867–1876, 2010.

[86] Wrist based HRM Reference Design. `http://www.ti.com/lit/ug/tidu125/tidu125.pdf`.

[87] J.E. Scharf and T.L. Rusch. Optimization of portable pulse oximetry through fourier analysis. In *Biomedical Engineering Conference, 1993., Proceedings of the Twelfth Southern*, pages 233–235, 1993.