Spring 2014

Adaptive Filtering for Heart Rate Signals

Samira Jaafari
San Jose State University

ADAPTIVE FILTERING FOR HEART RATE SIGNALS

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Samira Jaafari

May 2014

The Designated Thesis Committee Approves the Thesis Titled

ADAPTIVE FILTERING FOR HEART RATE SIGNALS

by

Samira Jaafari

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2014

Dr. Shahab Ardalan	Department of Electrical Engineering
Dr. Robert Morelos-Zaragoza	Department of Electrical Engineering
Dr. Amin Mobasher	Samsung Semiconductor, Inc.

ABSTRACT

ADAPTIVE FILTERING FOR HEART RATE SIGNALS

by Samira Jaafari

This work focused on the effects of suppressing the motion artifact of wrist photoplethysmographic heart rate signals. Monitoring of the heart can offer important insight with regard to health and wellness. The objective of the experiment conducted here was to recover the distorted signal resulting from body movement while measuring the heart rate signal non-invasively from the wrist.

The class of filters, known as adaptive filters, that can extract meaningful information from the distorted signal, used predetermined initial conditions to equalize the signal distortion due to motion. These filters do not require prior knowledge about the system.

Adaptive filters of LMS-type and RLS-type were used in this study to recover the distorted heart rate signal. This study also presented a comparison on short-range wireless technologies, such as Bluetooth and ANT+, that can be used for data transmission of the heart rate signal.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor Dr. Shahab Ardalan for his motivation and support. I have been challenged by Dr. Ardalan's wish to extract my enthusiasm about this work into more than just a thought, but as a small ripple in the wave of biotechnology and health awareness. I am grateful that the like-mindedness we share in modern technology has elevated this work from beginning to end. Most importantly, I am thankful for all that Dr. Ardalan has taught me and for his guidance to test risk.

This thesis came about from my passion for fitness and appreciation for well-being. The heart is one of the body's most vital organs. It is essential to our life and longevity, and it has given me great inspiration and determination to educate others.

Lastly, I would like to show appreciation for my family. Without their continued support and push to further myself, I would be nowhere today. I contribute all of my achievements and successes to my family, friends, and sports coaches who have influenced me to get me where I am today. This accomplishment is directly dedicated to my mother, my father, and my older brother who have not doubted me once. I extend my tremendous thanks to you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Heart Rate Signal

The heart rate signal is one of various biomedical signals transmitted throughout the human body. Monitoring this signal leads to important information about the state of this critical organ, with important implications for an individual's health [1].

Heart rate and exercise are directly related to losing weight. Non-invasive heart rate monitors have been integrated into wristwatches and fitness machines that display caloric expenditure during exercise and body movement. Fitness machines have metal sensors that are able to detect and display the heart rate only when in contact with both palms simultaneously. Wristwatches that measure heart rate require that a chest strap sensor be worn at all times. However, holding onto metal sensors without interruption during exercise can be problematic in the context of high intensity movement. Chest strap sensors represent an alternative for overcoming this problem, as they remain in contact with the body even during the most intensive exercise.

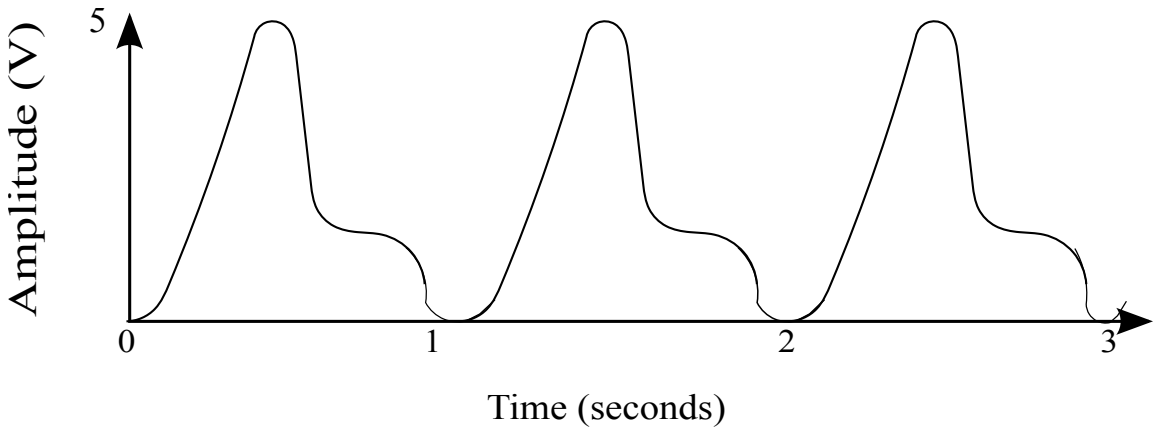

Figure 1.1: Normal Heart Pulse

Figure 1.1 illustrates a normal heart pulse, showing consistent peaks and valleys. Heart rate, together with other biomedical signals, gives important information about various aspects of an individual's health. Devices that can accurately measure the hemoglobin oxygen saturation content together with the heart rate are known as pulse oximeters [2]. Pulse oximetry refers to the non-invasive measurement of the level of oxygen saturation and of the pulse rate in veins. Pulse oximeters are generally used in the medical field for taking a variety of body measurements. Blood circulates throughout the body in rhythm with the heartbeat [3]. Thus, measuring the signal of the rate that blood moves throughout the body is one way of gauging the heart rate [4]. A non-invasive optical method of measuring the heart rate with the use of light-emitting diodes (LEDs) and photodiodes is known as photoplethysmography [5]. Using this technique, light directed toward the skin is partially absorbed by hemoglobin, with the remainder of the light reflected off of the skin or anatomical layers beneath the skin, as blood pulsates

throughout the veins [6]. With photoplethysmography, heart rate can be gauged by measuring the amount of reflected light.

Pulse oximeters use photoplethysmography to direct the light through the fingertip, measuring the heart rate signal based on the volume of the backscattered light received by the photodiode. The photodiode then converts the amount of backscattered light to an output voltage.

Similarly, photoplethysmography can be used to acquire the heart rate signal from the wrist, thereby eliminating the need to constantly hold onto the metal sensors or wear a chest strap sensor. In this way, the need for wearing a chest strap sensor in combination with a wristwatch is eliminated. Although the heart rate signal that is acquired from the wrist using photoplethysmography is distorted by body movement, the signal can be recovered [1], [5]. The heart rate signal is distorted in the process of body movement by intersymbol interference (ISI), dispersion of the pulse in the transmission channel, as a result of light absorption by the skin, tissues, and muscles. Although the backscattered light is not in sync with the heart beat when the body is in motion, the signal can be recovered using an adaptive filter.

1.2 Motivation

The motivation for this work originated from heart rate monitoring for health awareness. Devices that have integrated heart rate monitoring are typically used in the

area of sports monitoring and for various health applications. The two main benefits of adaptive filtering of the distorted heart rate signal are active noise cancellation and improved accuracy of measurement during body movement.

The wrist, where the heart rate signal is acquired, is a common location for wearable electronic devices such as wristwatches and wristbands. The integration of this precise method of heart rate acquisition in similar devices is consistent with the current trend toward greater health awareness and health consciousness.

The heart rate signal reconstructed by an adaptive filter can be used in sports monitoring or health related applications, not limited to any single purpose. Photoplethysmography allows for mobility in implementation, which is advantageous relative to other methods that lack mobility. The myriad potential usages are what inspired this study.

1.3 Literature Survey

Clean signals transmitted through a noisy channel, when recovered at the receiver contain noise, as a result of ISI. In the experiment conducted for this study, the measured heart rate signal was passed through a noisy channel. This noisy channel results from the backscattered light due to a combination of the skin, blood, tissues, and muscles.

The heart rate signal that was passed through the noisy channel was distorted at the receiving end. In order to determine a reliable measurement, it is necessary to recover

the signal or to reconstruct the original signal, unaltered by channel noise. Although other devices use non-invasive photoplethysmography, they retrieve the heart rate signal from other locations of the body, such as the palm, finger, forehead, or earlobe [7].

An adaptive filter, which is a digital filter, can reconstruct the distorted heart rate signal. This type of filter is self-designing and relies for its operation on a recursive algorithm. Starting with predetermined initial conditions, an adaptive filter converges to the optimum solution to some desired value [8]. While general digital filters have fixed coefficients, the coefficients of an adaptive filter adjust themselves, based on the magnitude of the error signal. Several algorithms, including the LMS-type and the RLS-type, estimate the coefficients of an adaptive filter.

1.4 Research Objective

The main objective of this study was to examine the effects of suppressing the motion artifact of wrist photoplethysmographic heart rate signals. The method for equalizing the ISI from the distorted heart rate signal was to adaptively filter the output response from the noisy channel. An adaptive filter is unique in the sense that its filter coefficients are driven to adjust by the error signal magnitude and converge to some desired value, representing an optimum solution [8]. The output result was the recovered heart rate signal.

The heart rate signal, recovered by the adaptive filter, can be transmitted wirelessly

to other devices, as it is acquired from the wrist. Popular short-range wireless communication technologies include Bluetooth and ANT+. Although Bluetooth and ANT+ technologies are short-range wireless standards, they are both high-demand technologies that are ubiquitous today. ANT+ technology is commonly used in health applications and sports monitoring, whereas Bluetooth technology is integrated across several platforms of electronic equipment and is a more common form of electronic communication in mobile devices.

1.5 Overview of Thesis

This study focused primarily on the adaptive filtering of the heart rate signal that is distorted by ISI as a result of body movement. A supplemental comparison presented short-range wireless standards for transmitting the reconstructed heart rate signal to mobile devices.

Chapter 2 discusses the fundamental adaptive filtering theory. A block diagram is used to describe an ideal adaptive filter, as well as the governing equations. The more commonly used algorithms that tune the filter coefficients, especially LMS-types and RLS-types, are discussed.

Chapter 3 compares short-range wireless standards, in particular Bluetooth and ANT+. It is hoped that combining data transmission with signal recovery will broaden the potential uses and applications of the study findings. The comparison was done

primarily to determine which technology would be most suitable for applications integrating the photoplethysmography methodology.

Chapter 4 reviews the performance of the filter algorithms. Practical results are examined relative to the fundamental adaptive filtering theory. The impact on the adaptive filtering of heart rate signals is investigated, and the conclusions with regard to each adaptive algorithm are summarized.

2 Adaptive Filters

2.1 Signal Properties

Various types of interference distort the heart rate signal during photoplethysmographic (PPG) monitoring as a result of body movement. ISI that distorts the signal is caused by light absorption in the skin, tissues, and muscles. Light absorption reduces the amount of backscattered light at the receiving end, as well as the peak-to-peak amplitude of the signal.

The heart rate signal is measured by the circulation of blood in veins during body movement. Specifically, the amount of light that is reflected back to the photodiode is directly proportional to the frequency at which the heart pumps blood throughout the body. The amplitude of the signal, based on the amount of backscattered light measured by the photodiode, increases or decreases at the same frequency that blood is pumped by the heart. The properties of the resulting signal can aid in digital processing and assist in gauging a reliable heart rate.

Two components of the signal properties are the DC component and the AC component. The DC component is relatively invariant in regard to pulsating blood and provides a constant signal amplitude. The AC component directly corresponds to the amount of backscattered light and is time-varying. Figure 2.1 illustrates that the AC component of the signal provides important information about the signal frequency [9].

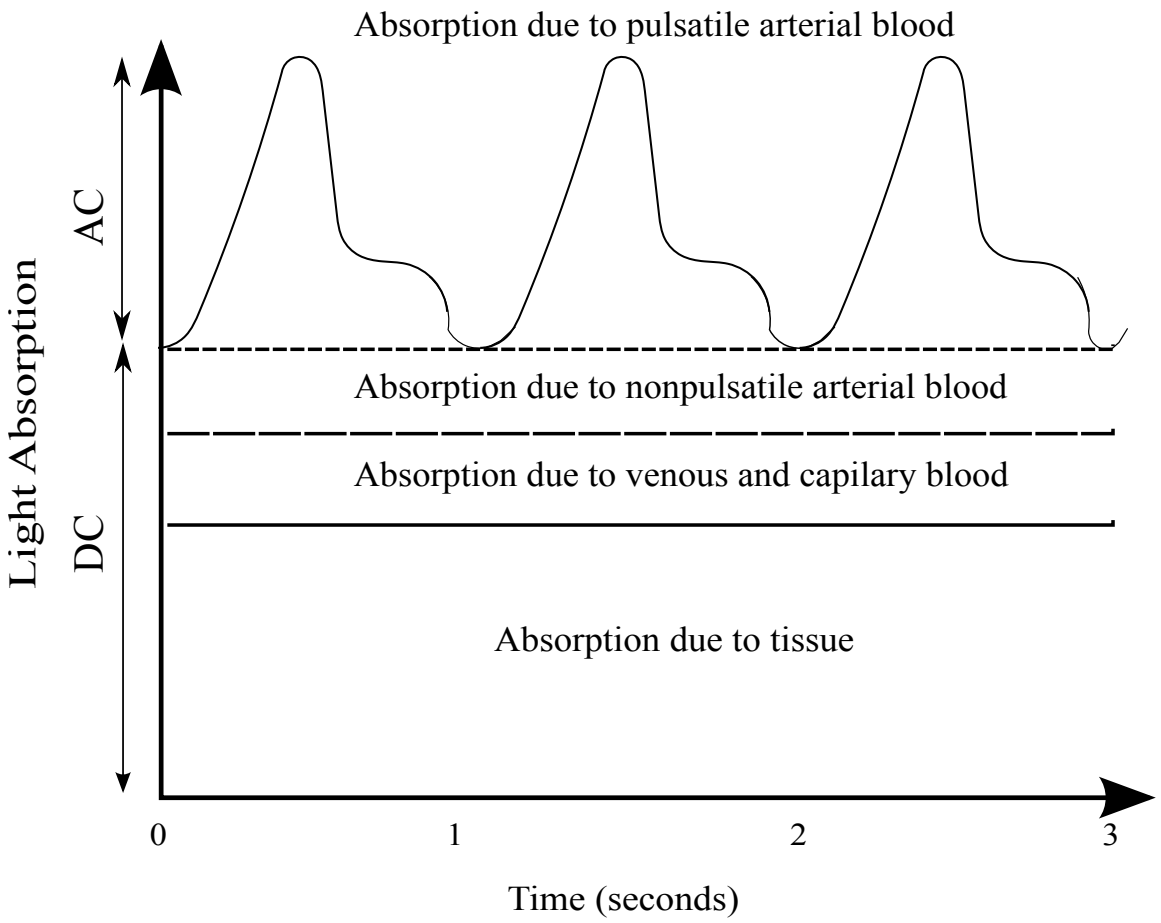

Figure 2.1: AC and DC Components in a PPG Signal

The AC component of the PPG signal is distorted as a result of body movement. Absorption of light in the skin, tissues, and muscles attenuates and distorts the AC component of the signal during body movement. One method to recover the distorted AC component is by adaptively filtering the heart rate signal.

An adaptive filter, further discussed in Section 2.2, can reconstruct the AC component such that the original signal properties are nearly recovered. Signal reconstruction is more easily done in the digital domain than the analog domain because

it is more cost efficient. By passing the AC component of the signal through an analog-to-digital converter it can be converted to corresponding digital values.

The microcontroller used in this experiment, known as an Arduino Micro, converted the AC component to digital values and passed them through adaptive filters. The microcontroller contains a 10-bit analog-to-digital converter and an Atmel ATmega32u4 microprocessor. The adaptive filters in this experiment operated on two input signals. The two input signals were the desired heart rate signal plus ISI and accelerometer signals. The adaptive filters used reference input signals from an accelerometer, known as the ADXL335 triple-axis accelerometer. Both the heart rate sensor and the accelerometer were placed on the wrist. The accelerometer signals, measuring accelerations of −3G up to +3G, isolated ISI and motion artifacts in the distorted heart rate signal during body movement. The accelerometer signals assisted in the recovery of the heart rate signal.

2.2 Adaptive Filter Theory

Many types of filters are prevalent in signal processing. Filters manipulate signals to reject unwanted characteristics. Noise, interferences, and erroneous frequencies are undesirable characteristics that affect data in signals. These unwanted attributes can be extensively minimized or nearly eliminated with the use of filters. An adaptive filter, also known as a transversal filter, is one that self-adjusts its parameters to converge to

specified target values. Adaptive filters are useful when the characteristics of the incoming signal are unknown or likely to vary.

An adaptive filter uses a recursive algorithm to adjust its parameters based on predetermined initial conditions. The filter output converges to target values after several parameter iterations. Choosing which recursive algorithm to implement is determined by the prioritization of the following adaptive filter characteristics: rate of convergence, misadjustment, tracking, robustness, computational requirements, structure, and numerical properties [8].

The idealities of an adaptive filter should be considered as practical frames of references for the application that it is integrated into. Above all, computational cost, performance, and robustness are the three main factors for the type of adaptive filter to implement in an application [8].

The least-mean squares (LMS) algorithm is simple to implement and aids in identifying areas of improvement for the characteristics of the adaptive filter. After areas of improvement have been identified, more sophisticated algorithms can be substituted in place of the LMS algorithm [8]. A simple block diagram of an adaptive filter is depicted in Figure 2.2.

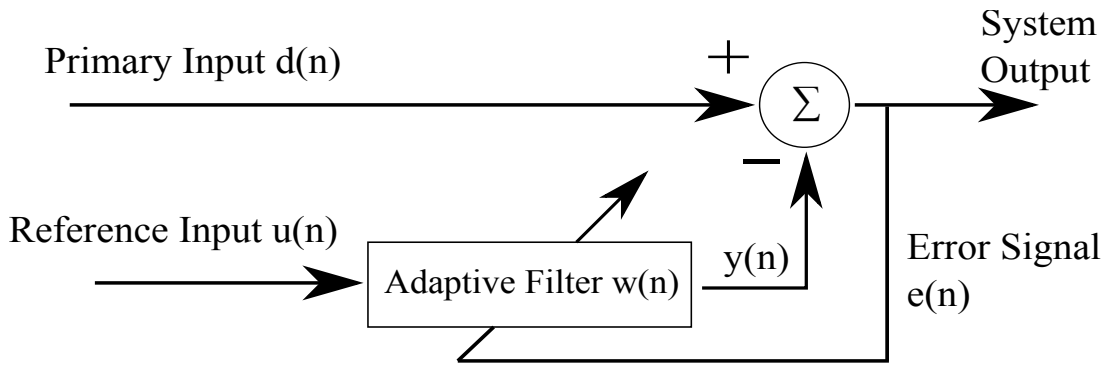

Figure 2.2: Adaptive Filter Block Diagram

The block diagram in Figure 2.2 represents one of four classes of adaptive filters. Referred to as the interference-cancelling type, the fourth class of adaptive filters cancels unknown interferences contained in a signal. A reference signal from a separate sensor provides correlated information that is used to cancel the interference [8].

The primary input *d(n)* is the incoming signal that contains the original signal *x(n)* plus noise *m(n)*. However, the noise is uncorrelated with the original signal. The primary input *d(n)* is defined in Equation 2.1.

$$d(n) = x(n) + m(n) \tag{2.1}$$

The reference input *u(n)* is measured separately and contains noise that is correlated with the noise *m(n)* from the primary input. The noise from the reference input passes through an adaptive filter that estimates the noise of the original signal, known as *y(n)* in Figure 2.2. The noise estimate is subtracted from the primary input to produce an estimate of the original signal.

On each iteration, *y(n)* is estimated by the adaptive filter coefficients and is driven by the error signal *e(n)*. The error signal, the difference between the desired signal and the incoming signal, is the estimated system output. Noise *y(n)* is subtracted from the primary input *d(n)* to provide an estimate of the system output. Over several iterations, the system output converges to the primary input signal without noise [2].

The error signal is given by Equation 2.2, and the estimated noise signal *y(n)*, which is the addition of the PPG signal *d(n)* and the motion artifact signal *w(n),* is given by Equation 2.3 [2].

$$e(n) = d(n) - y(n) \tag{2.2}$$

$$y(n) = d(n) + w(n) \tag{2.3}$$

In Equation 2.3, *d(n)* and *w(n)* are known on each iteration of the algorithm [10]. Haykin refers to *y(n)* as the model output, formulated in Equation 2.4.

$$y(n) = \sum_{k=0}^{M-1} \hat{w}_k(n) u(n-k) \tag{2.4}$$

In Equation 2.4, $\hat{w}_0(n)$ through $\hat{w}_{M-1}(n)$ are the estimated filter coefficients, also known as tap weights, *u(n)* through *u(n-M+1)* are the reference input values at time *n*,

and *M* is the filter order length in the recursive algorithm [8]. The filter output is computed on each iteration and adjusted by the error signal *e(n)*. The recursive algorithm iterates until the error signal magnitude is smaller than a desired value [8].

2.3 Adaptive Filter Type

2.3.1 Least Mean-Square (LMS) Type

The LMS algorithm has low computational complexity and functions as a widely used algorithm for FIR adaptive filters [10]. The algorithm is a member of the family of stochastic gradient algorithms and produces the least mean-square of the error signal [8].

The primary signal noise *m(n)* in Figure 2.2 is uncorrelated with the noise from the reference input. The system output is expressed by Equation 2.5.

$$\hat{s}(n) = d(n) + m(n) - y(n) \tag{2.5}$$

The term on the left of Equation 2.5 is the system output and is the same as the error *e(n)*. The terms in Equation 2.5 are squared and expressed in Equation 2.6. The expected value of the terms in Equation 2.6, also known as the mean value, is given by Equation 2.7.

$$s^2 = d(n)^2 + (m(n) - y(n))^2 + 2d(n)(m(n) - y(n)) \tag{2.6}$$

$$E[\hat{s}^2] = E[d(n)^2] + E[(m(n) - y(n))^2] + 2E[d(n)(m(n) - y(n))] \tag{2.7}$$

The third term on the right in Equation 2.7 reduces to zero, due to the fact that the input signal *d(n)* is not correlated with any of the noise components. Minimizing the output noise power, which is the rightmost term in Equation 2.7, subsequently minimizes the output power of the estimated signal. The signal-to-noise ratio (SNR) increases when the output noise power is minimized because the signal power is independent of the noise. Hence, the least mean-square value is simplified in Equation 2.8.

$$\min E[\hat{s}^2] = E[d(n)^2] + \min E[(m(n) - y(n))^2] \tag{2.8}$$

The simplicity of the LMS algorithm lies in the fact that it does not require correlation functions or matrix inversions that other algorithms use [8]. Two features of an adaptive or a transversal filter are that the filter derives the estimated output *y(n)* and filter coefficients or tap weights adjust in a recursive format. Table 2.1 summarizes the input and output parameters of the LMS algorithm. Figure 2.3 illustrates an ideal LMS filter structure with a delay line for each tap-weighted coefficient.

Table 2.1: Mth-Order FIR Adaptive Filter Parameters

Inputs	M = filter length μ = step-size factor *d(n)* = input data to adaptive filter *w(0)* = initialization vector = 0
Outputs	*y(n)* = adaptive filter output = $\hat{d}(n)$ $e(n) = d(n) - y(n) =$ error $\hat{w}(n+1) = \hat{w}(n) + \mu u(n) e(n)$

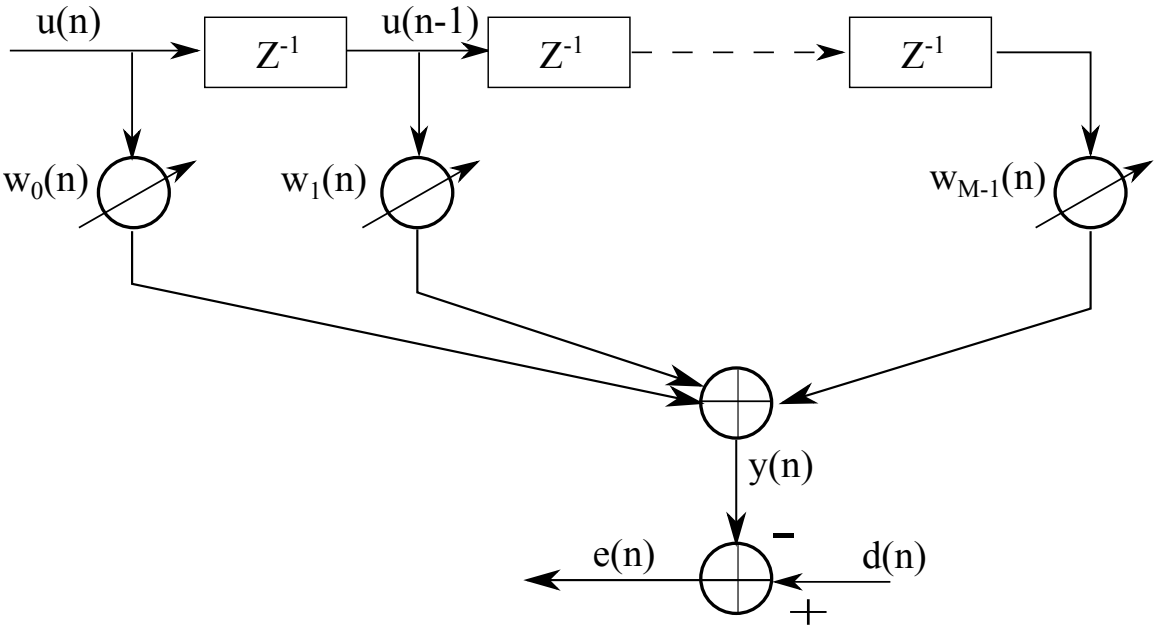

Figure 2.3: LMS Filter Structure

The filter length corresponds to the number of coefficients that are computed in the algorithm, and the initialization value for the first coefficients are zero. The step-size factor μ is a constant value that is applied to the error signal on each iteration to update the subsequent filter coefficient $w_k(n)$. The value chosen for μ must be small enough such that the misadjustment is minimized and the filter coefficients progress slowly instead of producing large jumps at the filter output [8]. The output $w(n+1)$ updates the initial value of the filter coefficient or tap weight with a correction factor based on the step-size μ.

The estimated PPG signal, also known as the system output, $e(n)$ in Figure 2.2 loops back to the LMS filter input. The input passes through the tapped-delay line in the

filter, and the filter coefficients are adjusted based on the values in the reference input *u(n)* as well as in the step-size factor μ. The estimated noise output *y(n)* is subtracted from the primary input *d(n)* to produce an estimate of the output *e(n)*. The iterative procedure continues until the error is smaller than a desired value. The LMS filter structure has low computational cost and is simple to implement for PPG heart rate reconstruction.

2.3.2 Recursive Least-Squares (RLS) Type

The recursive least-squares (RLS) algorithm recursively derives the filter coefficients that minimize a weighted least-squares cost function, also known as error signal, in an adaptive filter. Unlike the LMS-type, the RLS-type belongs to the class of deterministic input signals and has a rate of convergence that is typically an order of magnitude faster [8]. However, the high convergence rate consists of high computational complexity.

The RLS algorithm derives the filter coefficients recursively in time. As incoming data are processed, the influence of old data decreases based on the algorithm memory. The forgetting factor λ represents the memory of the algorithm [11]. The forgetting factor is commonly exponentially weighted, such that older data have less impact on the current filter coefficients [8]. The filter coefficients are optimized from the time the filter starts to the time that the current values of the error are minimized [10]. Figure 2.4

illustrates the RLS algorithm structure.

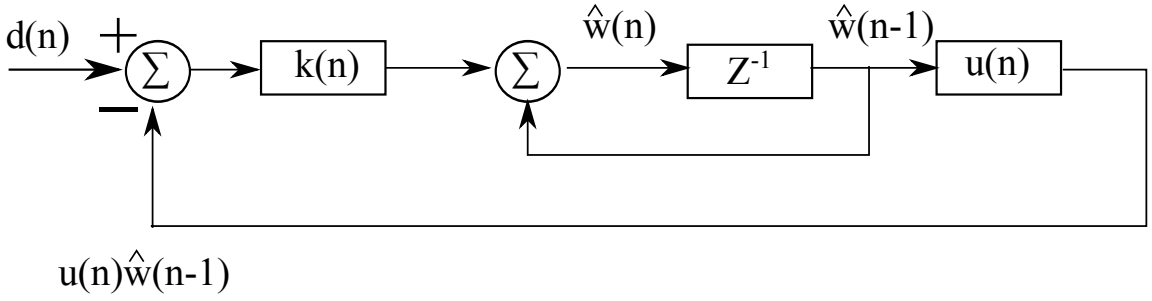

Figure 2.4: RLS Filter Structure

The incoming signal *d(n)* contains the original signal plus noise. The time-varying gain vector *k(n)* updates the filter coefficients on each time iteration *n*. In Equation 2.9, ***P*** is the correlation matrix and is initialized to ***P(0)*** in Equation 2.10.

$$\mathrm{k}(n) = \frac{\mathrm{P}(n-1)\lambda^{-1}\mathrm{u}(n)}{(1+\mathrm{u}(n)^{H}\mathrm{P}(n-1)\lambda^{-1}\mathrm{u}(n))} \tag{2.9}$$

$$\mathrm{P}(0) = \delta^{-1}\mathrm{I} \tag{2.10}$$

The $u(n)^{\mathrm{H}}$ in Equation 2.9 refers to the Hermitian transpose of the reference input *u(n)*, also known as the conjugate transpose, and λ is the exponentially weighted forgetting factor of the algorithm. The forgetting factor is valued between $0 < \lambda \leq 1$ typically, and the special case when λ=1 corresponds to infinite memory [8].

Furthermore, the minimum square error (MSE) is given by Equation 2.11.

$$E[e^2(n)] = E[x^2(n)] + E[\{m(n) - y(n)\}^2] + 2E[x(n)\{m(n) - y(n)\}] \tag{2.11}$$

The third term on the right side of Equation 2.11 reduces to zero, and Equation 2.12 expresses the minimum MSE (MMSE) [11].

$$\min E[e^2(n)] = E[x^2(n)] + E[\{m(n) - y(n)\}^2] \tag{2.12}$$

The algorithm updates the correlation matrix with the filter coefficients after the matrix is initialized. The RLS algorithm is summarized in Table 2.2 [8].

Table 2.2: RLS Algorithm Summary

Initialization	$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$ $\quad \delta =$ small positive constant $\hat{\mathbf{w}}(0) = 0$
Equations	At each time n = 1, 2, ..., $\mathbf{k}(n) = \dfrac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)}$ $e(n) = d(n) - \hat{\mathbf{w}}^H(n-1)u(n)$ $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)e(n)$ $\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)u^H(n)\mathbf{P}(n-1)$

The RLS algorithm retains memory from the very start of the process. Because of this, the RLS algorithm has the ability to converge to the optimum solution quicker than the LMS algorithm. However, the RLS algorithm has more computational cost and

requires large storage capacity [10]. The algorithm should be selected based on an application frame of reference in which it is integrated into and on the prioritization of the filter characteristics listed in Section 2.2.

2.3.3 Other Filters

Among other filters, the LMS and RLS filter types are in the family of adaptive filters. Other common filters that are used to reject noise and interference are the Wiener solution, the method of steepest descent, and Kalman filters.

The LMS algorithm is derived from the Wiener solution, first derived by Wiener [8]. Wiener filters are grouped in the class of linear optimum discrete-time filters. The requirement for Wiener filters is to minimize the MSE with the following two restrictions: The filter is linear, and the filter operates in discrete-time [8]. The main differences between Wiener filters and adaptive filters are that Wiener filters have no *a priori* knowledge on the statistics of the signal nor do they require a separate reference input signal [2]. If the statistical characteristics of the incoming data do not match the *a priori* information for which the filter was designed to expect, then the filter will not provide the optimum solution [8]. Although both the SNR and the correlation coefficient in [2] were found to be higher for Wiener filters than LMS adaptive filters, the complete data set on which the algorithm was performed on was recorded before it was actually processed. Thus, the real-time implementation for Wiener filters in the recovery filtering

of the heart rate signal is not feasible.

Another adaptive filter is the method of steepest descent, which refers to finding the bottom point of a bowl-shaped surface, also known as the error-performance surface [8]. The method of steepest descent is a recursive algorithm that minimizes the error on each iteration as it progressively steps toward the bottom point. The filter coefficients or tap-weighted vector takes on the minimum value w_O, which is defined by the Wiener-Hopf equation in Equation 2.13 [8].

$$Rw_o = p \tag{2.13}$$

In Equation 2.13, the correlation matrix $\boldsymbol{R}$ is nonsingular and p is the cross-correlation vector between the reference inputs of the filter [8]. The issue of stability is critical for the method of steepest descent and relies on the step-size parameter μ and the correlation matrix $\boldsymbol{R}$. The value of the step-size parameter μ dictates how the transient response of the loop behaves, yet there is no direct mathematical model to solve for the precise value [8]. Moreover, Kalman filters are another type of common filters. The Kalman filter structure is illustrated Figure 2.5.

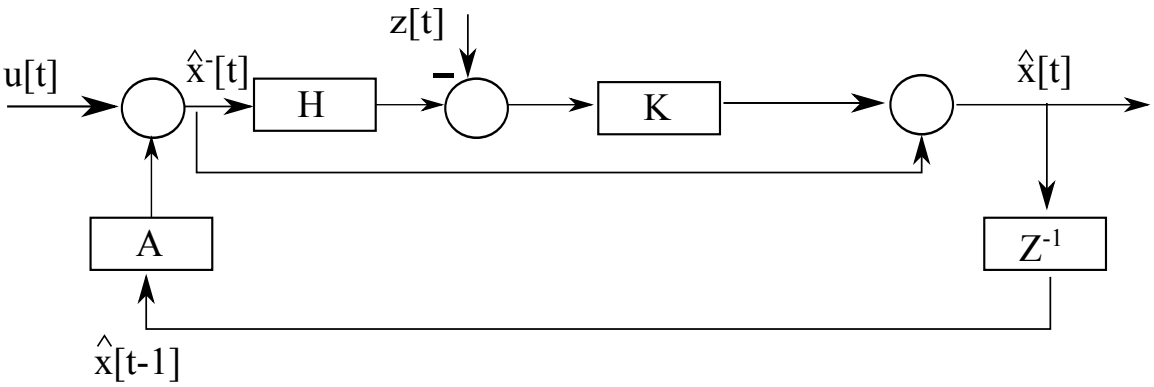

Figure 2.5: Kalman Filter Structure

A Kalman filter is a linear recursive filter characterized by state-space concepts. The current state is computed by the previous estimate and the new data. This implies that the previous state estimate value requires storage [8]. In Figure 2.5, the matrix ***A*** predicts the next state $\hat{x}^-[t]$ by using the current estimate $\hat{x}[t-1]$. The matrix ***H*** calculates the error signal by generating a prediction that is based on the value of the reference signal subtracted from *z[t]*. The error signal is multiplied by the Kalman gain matrix ***K*** to produce a correction toward the optimum solution and yield the final estimate $\hat{x}[t]$.

A Kalman filter was used in [12] to successfully reduce the motion artifact from PPG signals that were measured from the finger. However, the distorted heart rate signal in [12] had fewer motion artifacts than PPG signals that are measured from the wrist, resulting from anatomical differences in these two body locations.

Similarly, in [13] a fixed-interval Kalman smoother showed superior performance to normalized LMS (NLMS) and RLS-type filters for PPG signals that were measured

from the finger. In general, Kalman and RLS-type filters showed similar performances, which are both superior to NLMS [13]. In addition, RLS-type filters can be represented in the state-space concept. Kalman filters can also be used in real-time processing, while the fixed-interval Kalman smoother in [13] must be used in offline post-processing.

Furthermore, several families and classes of filters can be used to cancel noise in signals. Variations of each algorithm have also been implemented, such as the NLMS-type that was aforementioned. Each algorithm must be evaluated for the particular application it is integrated into. As demonstrated in [12], [13], different techniques have proven to be superior to adaptive filters for PPG signals and to suppressing the motion artifacts as a result of body movement. The difference between the experiment conducted in this study and the work in [12], [13] was the body location of where the PPG signal was measured from; This experiment acquired the signal from the wrist and [12], [13] acquired the signal from the finger.

2.4 Summary

The requirement to adaptively filter the PPG signal originates from the motion artifact that is embedded in the heart rate signal, as a result of body movement. The causes of the motion artifacts are light-absorption variations in the pulsatile blood during movement and ISI from the skin-to-air channel between the body and the PPG sensor. Figure 2.6 illustrates the skin-to-air channel.

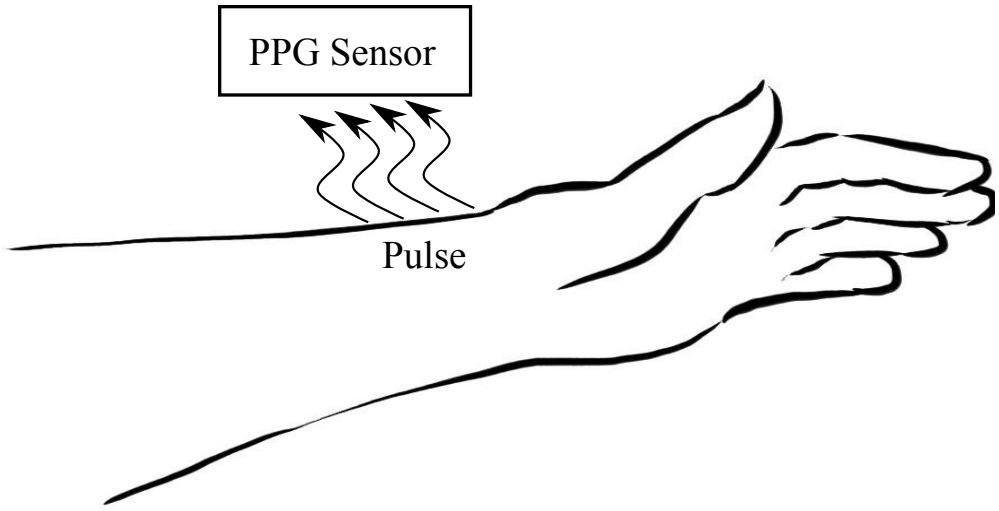

Figure 2.6: Wrist Source and Sensor Placement

The skin-to-air channel is represented by the arrows. Figure 2.6 illustrates the location of the wrist PPG signal in regard to the sensor placement. In this experiment, the PPG sensor was susceptible to noise and light absorption variations as a result of body movement. In order to recover the corrupted heart rate signal, an adaptive filter processed primary and reference inputs.

Two types of adaptive filters, LMS-types and RLS-types, were used on the same data sets to compare their performances. The incoming data to the PPG sensor were supplemented with a separate reference input from an accelerometer that was also located next to the PPG sensor. The accelerometer is an ADXL335 triple-axis accelerometer that works with Arduino microcontrollers. The PPG sensor, which is an open-source sensor,

and the accelerometer were linked to the Arduino Micro microcontroller that was connected to a computer for data transfer and storage. Furthermore, Chapter 4 presents the hardware and software test setups for gathering the data, as well as the experiment results. The next chapter compares short-range wireless technologies, such as ANT+ and Bluetooth, that are commonly used in health monitoring and mobile electronic devices.

3 Wireless Technology

3.1 Short-Range Wireless Technology

Wireless communication is a major demand among consumer electronic devices. Some forms of communication are more popular than others, and some are specific in regard to certain applications. However, common features that all wireless technologies provide are convenience, flexibility, and remote control. Almost all consumer electronic devices today are compatible with one or more short-range wireless technologies. Choosing a particular wireless technology for a device depends on several factors such as range, data rate, complexity, environment, size, space, security, user experience, and other notable considerations like licensing fees. A few of the popular short-range wireless technologies include radio-frequency identification (RFID), near-field communication (NFC), ANT+, and Bluetooth. Each of these technologies consists of distinct networks.

3.1.1 Networks

Networks are formed by wireless communication over the air between multiple devices. A wireless local area network (LAN) is a computer network that uses wireless connections for connecting network nodes [14]. LANs connect laptops and other portable devices in the local area, and they are designed for high data rates [15]. The

major standards for wireless LANs are defined in the IEEE 802.11 family, although smaller-scale standards such as Bluetooth exist [15]. Figure 3.1 illustrates the topology for a star wireless network.

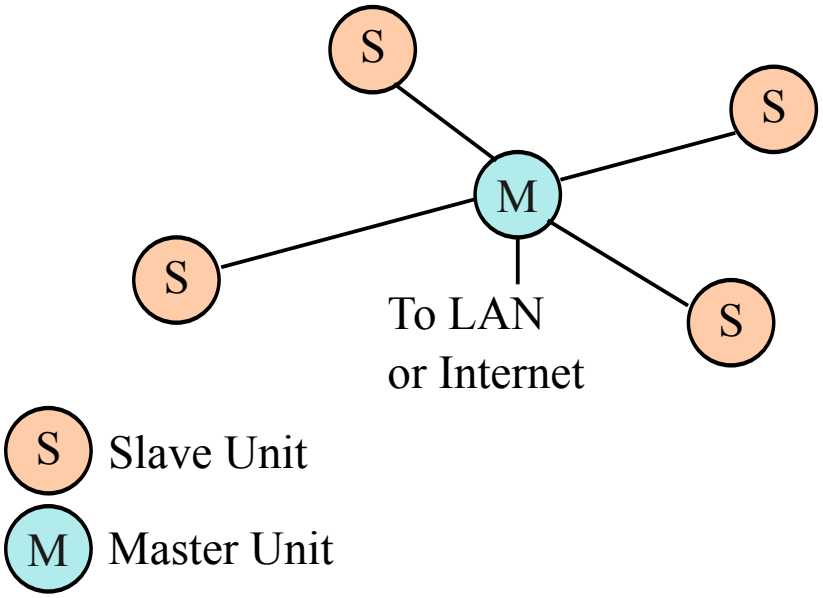

Figure 3.1: Star Wireless Network Topology

Although the slave units in Figure 3.1 cannot communicate to each other directly, they can communicate and transfer data through the master unit. The star network configuration is one out of many, and other topologies, illustrated in Figure 3.2, include the ring, bus, tree, and mesh structures.

Other short-range wireless networks include personal area networks (PANs) and body area networks (BANs), also known as body sensor networks (BSNs). The IEEE 802 standards committee mandates wireless PAN (WPAN), which includes a subset task

group that governs BANs [16], [17].

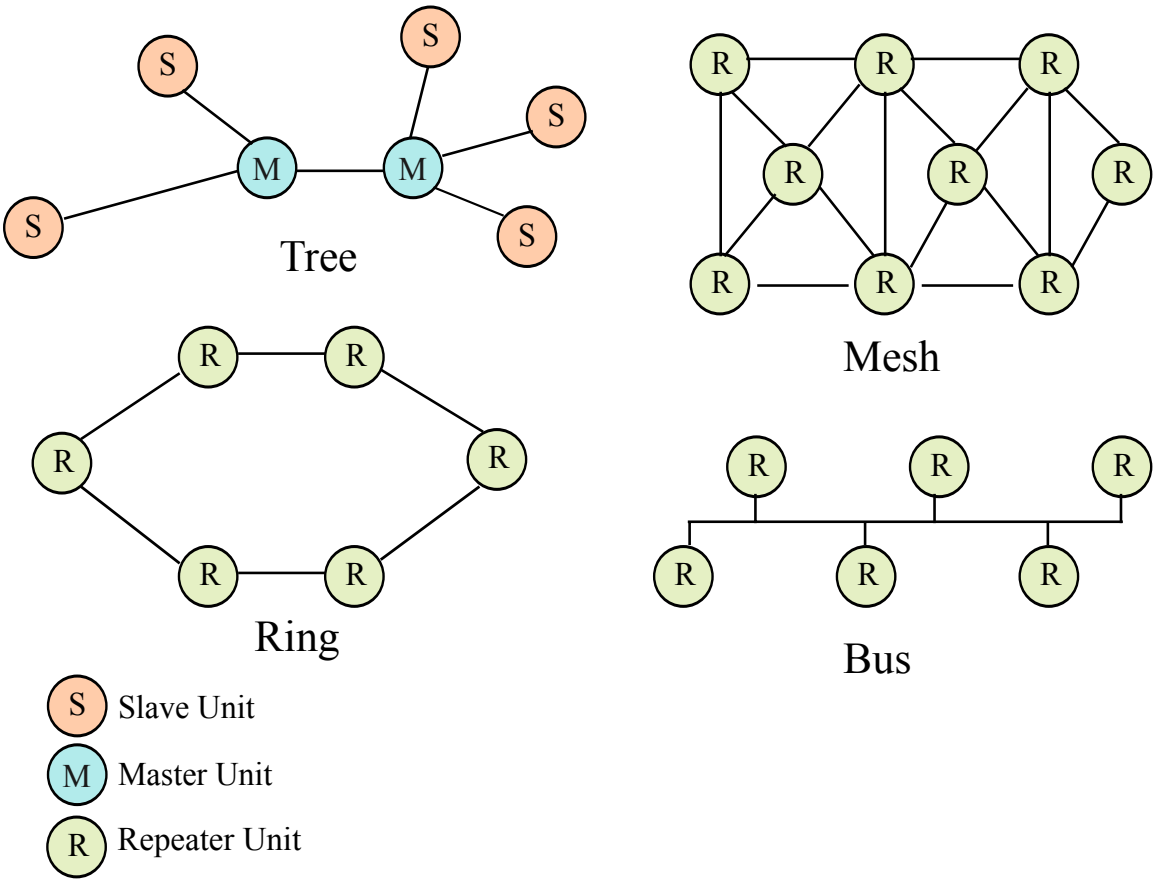

Figure 3.2: Network Topologies

A PAN, consisting of consumer devices, enables wireless communication with existing infrastructure networks [18]. The devices that typically create PANs, such as mobile phones, personal digital assistants, and laptops, have large batteries and powerful processors that can provide high data rates [18]. Figure 3.3 illustrates a wireless BAN with sensor nodes around the human body.

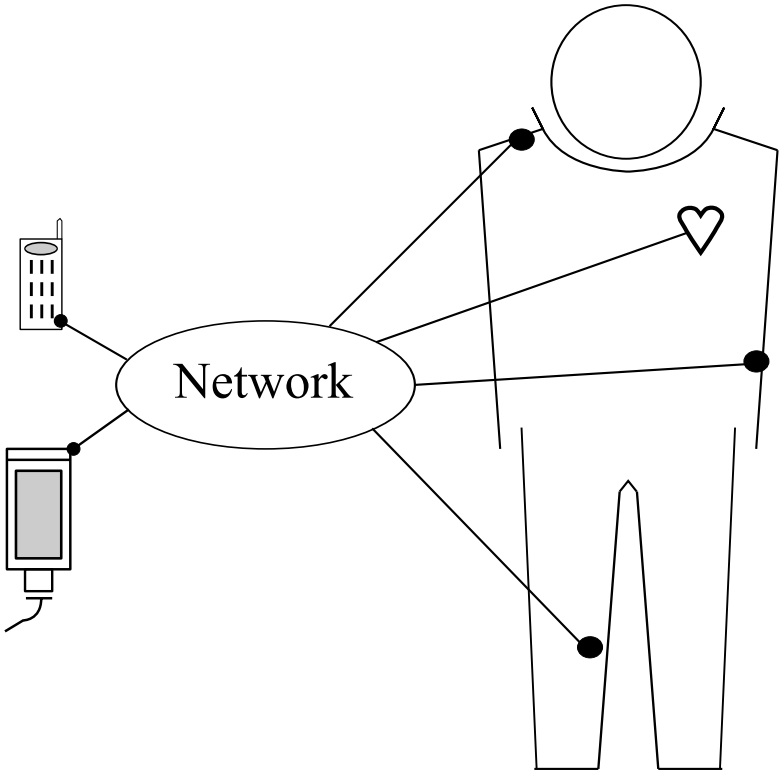

Figure 3.3: Wireless Body Area Network (BAN)

A BAN is a subset of PANs that relays body sensor information to a mobile phone or a remote server. These sensor nodes monitor body signals and body movement. BANs enable remote health monitoring at a level of convenience and personal comfort. Multiple wireless communication standards and data-exchange technologies for the health field are detailed in the following sections. Key features, such as power, range, and speed, are compared for each technology.

3.2 Wireless Technologies for Health Applications

The awareness of health monitoring has surged in recent years partly due to technological advancements in consumer devices. As more attention regarding health and wellness emerges, the demand for monitoring physiological activity and body signals has increased. Nowadays, the bond between wireless data exchange and health and wellness is inseparable. Many wireless technologies are suitable for applications in health and fitness sectors.

RFID is a widely used wireless technology that uses radio-frequency electromagnetic fields to transfer data. The ability to transfer data lies in the identification process of using tags that are attached to the objects to be identified. The signaling between a tag and a reader, which is the counterpart of a tag, is when data exchange occurs. RFID has been used in health and medical applications for patient tracking and positioning. This wireless technology is moving beyond the perception of being solely used as an asset tracker. Instead, RFID is being viewed as a technology that can improve individual care by tracking vulnerable patients or staff in medical and health facilities [19].

Another wireless technology that is growing in popularity among consumer devices is NFC. As an extension of RFID, NFC is a form of contactless communication that utilizes electromagnetic radio fields based on tags and readers. Standardized in ISO/IEC 18092 [20], NFC is interoperable with other wireless communication technologies.

As its name indicates, NFC can only be used for short ranges of up to 10 centimeters [21]. It is more suitable for access-verification applications in vehicles, offices, airports, stores, and theaters as a result of its range boundary. As integration for this technology increases, the robustness of security becomes a major concern [21].

Although RFID and NFC are both widely used, their applications are limited. Both technologies utilize active and passive tags. The tradeoffs between the two types of tags are power consumption and range. An active tag consumes more power and can receive and transmit data over greater distances, whereas the opposite is true for passive tags.

In addition, the RFID and NFC tags are susceptible to electrostatic discharge (ESD). Over a period of time, ESD can degrade the device that these technologies are integrated into. Despite the drawbacks for certain applications, RFID and NFC are commonly used in medical fields and by health professionals [19].

Moreover, many health care and technology companies support ZigBee Health Care. Detailed in [22], the IEEE 11073-10441 standard defines the set of protocols for personal health device communication by ZigBee networks [23]. ZigBee network is a mesh network based on the IEEE 802.15.4 standard for the low data rate WPANs (LR-WPAN) [24]. There are slightly different variations of this network, including ZigBee PRO and ZigBee Smart Energy.

Although ZigBee wireless technology is considered to have low power consumption, the complexity and power requirements do not make it acceptable for long term and unmaintained devices that run on limited power sources [25]. To combat the

power consumption issue, ZigBee and other technologies have started to support energy harvesting and energy-efficient mechanisms. ZigBee PRO is optimized for low power consumption and has the ability to support large networks with thousands of devices. The quality of performance has led ZigBee to become a standard by a large part of the health care industry [24], [25].

Unlike the aforementioned technologies, the ANT protocol is a proprietary wireless technology for sports, fitness, and health sectors [27]. ANT+ is the low-power wireless standard built on top of the ANT protocol, and it evolved from the need to communicate wireless sensor data that run on limited power sources [27]. ANT is used extensively in sports and health monitoring systems. ANT+ ensures that all devices are interoperable in a managed network, allowing reliable data exchange and transfer [27]. As future technology aims to implement BSNs, ANT+ has the potential to grow with new devices.

Bluetooth is a wireless technology standard that is most commonly found in consumer electronic devices today. Bluetooth technology exchanges data over short distances, forming PANs. One of the earliest applications of Bluetooth was the communication link between mobile phones and hands-free headsets. Since then, Bluetooth has grown into many sectors and has been used to transmit health sensor data from medical devices to mobile phones and tele-health devices [28]. Bluetooth technology is suitable for connecting two devices that are nearby each other, such as a hands-free headset and a mobile phone. Over the last two decades, Bluetooth technology has been improved to accommodate for the speed and power bottleneck.

3.3 Bluetooth Standard

Bluetooth is a WPAN technology that operates on the 2.4 GHz industrial, scientific, and medical (ISM) radio band. The maximum number of devices that Bluetooth can support in its point-to-point topology, also known as a piconet, is eight, one master and seven slaves [29]. The connection between devices is called a pairing, and each piconet in one network can be connected to seven slaves, allowing them to create their own network, and so on [28]. Thus, a large network of Bluetooth-enabled devices can communicate simultaneously, as illustrated in Figure 3.4.

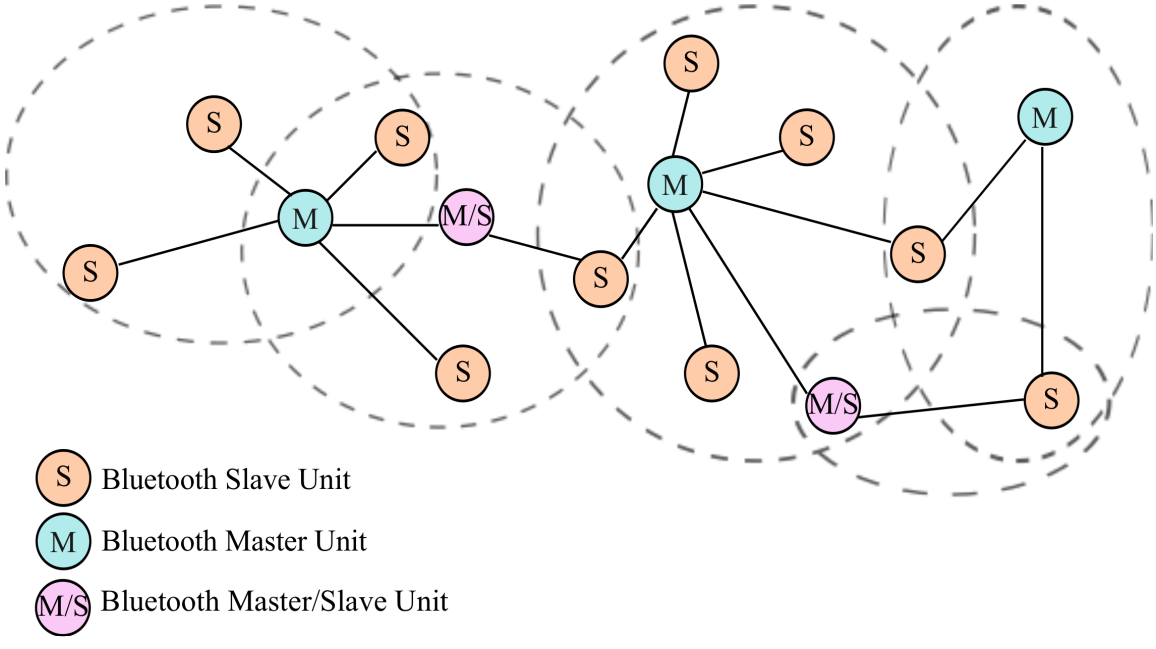

Figure 3.4: Bluetooth Piconet

Bluetooth has been integrated in devices at a fast rate over the last decade, and many consumer electronics are compatible with this technology. Due to the minimizations in device dimensions, power sources have been reduced in size, which imposes a need for consumer electronics to implement low-power and low-energy wireless technologies.

3.3.1 Frequency Spectrum

Bluetooth technology includes 79 frequency channels that are spaced at 1 MHz apart and have the ability of frequency-hopping. The frequency band spreads from 2400 MHz to 2483.5 MHz. The initial format for frequency modulation was chosen to be Gaussian frequency-shift keying (GFSK). To support higher data rates, Bluetooth version 2.0 implemented phase-shift keying modulation. However, Bluetooth version 3.0 did not achieve higher data rates by changing the modulation, but instead by working with the IEEE 802.11g physical layer [28].

For wireless technologies that share the ISM band, interferences and collisions are important issues on crowded frequencies. Multiple technologies using the same frequency band results in a slower data-exchange rate. However, the adaptive frequency-hopping (AFH) mechanism was designed to reduce the amount of interferences and collisions. AFH scans among the 79 frequencies at 1 MHz intervals to provide better performance interaction when other technologies are operating simultaneously [28].

3.3.2 Range

Different from other short-range wireless technologies mentioned in Section 3.2, Bluetooth technology has various ranges that depend on the radio signal strength. The Bluetooth Core Specification protocol defines a minimum range that must be adhered to [28]. The ranges for the different radio classes are defined in Table 3.1.

Class two radios are most commonly found in mobile devices, and class one radios are used primarily in industrial applications.

Table 3.1: Bluetooth Radio Class Versus Range [28]
(Adapted from Bluetooth Basics)

Class of Radio	Range
3	Up to 1 meter or 3 feet
2	Up to 10 meters or 33 feet
1	Up to 100 meters or 300 feet

3.3.3 Power

Low power consumption is one of the most challenging aspects among the design of wireless technologies. High data-exchange rates require an increase in power consumption. Restrictions in the design layout, such as the form factor, limit the capacity of a power source that can be used as the power supply. Bluetooth was designed for low-power consumption, and the class two radios use 2.5 mW of power, equivalent to 4 dBm

[28], as listed in Table 3.2.

Table 3.2: Bluetooth Class Versus Output Power [28]
(Adapted from Bluetooth Basics)

Class of Radio	Maximum Output Power
3	1 mW (0 dBm)
2	2.5 mW (4 dBm)
1	100 mW (10 dBm)

Bluetooth Low Energy (BLE), also known as Bluetooth SMART or Bluetooth Ultra Low Power (ULP), is a feature of Bluetooth version 4.0 intended to power applications running off of a coin battery cell. The current version of Bluetooth, version 4.1, extends version 4.0 devices to act as both a peripheral and as a hub [30]. Table 3.3 summarizes the data rates for different operations of version 4.1.

Table 3.3: Bluetooth 4.1 Data Rates [30]
(Adapted from Bluetooth Specification Version 4.1)

Operation	Speed
Basic Data Rate	721.2 Kbps
Enhanced Data Rate	2.1 Mbps
High Speed 802.11 AMP	54 Mbps

For BLE, the data rates of 1 Mbps are attained on 37 frequency channels of 2 MHz spacing [30], and low power consumption between one-half and one-hundredth of the

power of the basic Bluetooth technology is implemented [28].

3.4 ANT Technology

ANT is a low-power proprietary protocol in the 2.4 GHz ISM band. It was derived from the need to transfer running-distance data from a tiny device placed in a shoe to a local wearable device. ANT supports scalable network topologies, illustrated in Figure 3.5. The supported network structures span from simple two-node networks to full-complex point-to-multipoint communication networks [31].

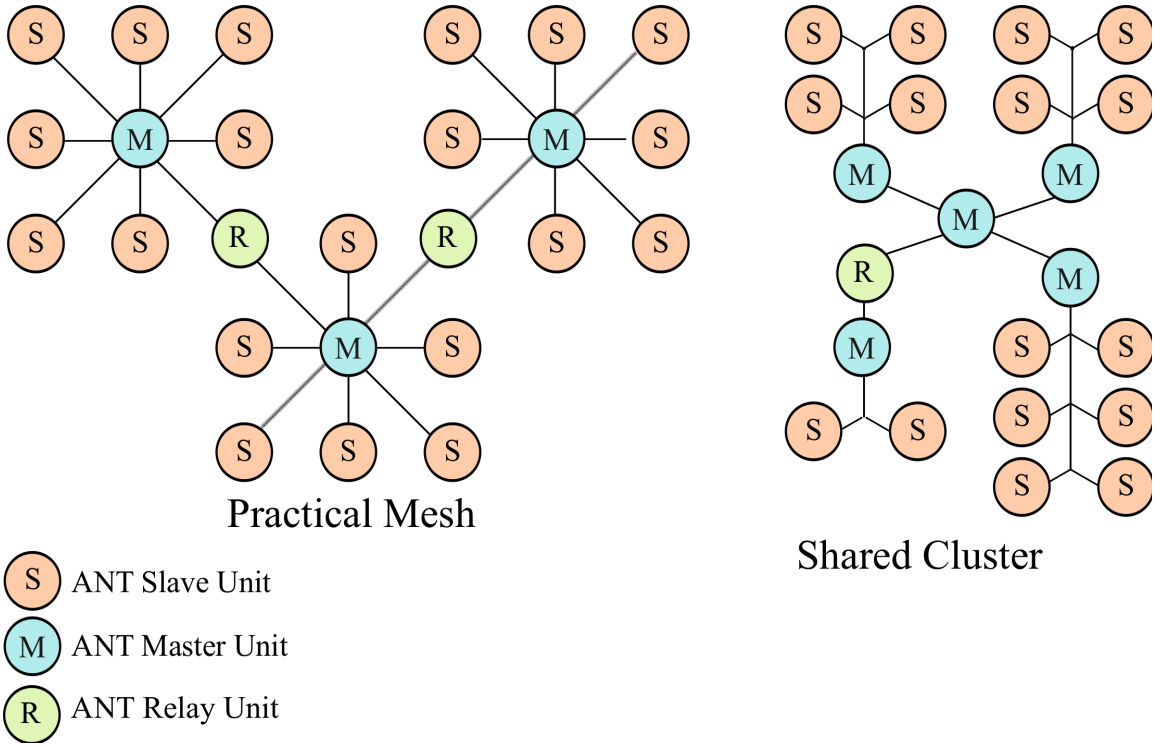

Figure 3.5: ANT Networks

The primary sector that implements ANT is the sports sector, specifically fitness and health monitoring sectors, where the update of information does not require data

transfer of more than a few times per second.

3.4.1 Frequency Spectrum

The frequency spectrum that ANT technology supports spreads from 2400 MHz to 2524 MHz. Each channel operates on a single frequency, every 1 MHz apart unless otherwise specified by the application host [31]. The specific frequency band of 2457 MHz is reserved for ANT+ devices [27]. Similar to the frequency-hopping mechanism that Bluetooth implements, ANT has the ability to switch the operating frequencies, improving the coexistence with other wireless devices, which is referred to as the frequency agility. However, this functionality monitors channel performance and only changes operating frequencies when signal integrity drops below predefined values [31].

3.4.2 Range

The ANT protocol emphasizes ultra low power (ULP) because of the low data rate. The general message rates, which are the time periods between transmissions, span from 0.5 Hz to 200 Hz for the basic rate. In a burst transfer mode, the maximum data throughput of 20 Kbps to 60 Kbps is attained on the raw data rate of 1 Mbps depending on the ANT chip that is used [27].

Detailed in [32], ANT can support a range between 1 and 30 meters or reach kilometer distances depending on the topology and the number of nodes in the network.

The modules are designed to work for line-of-sight, and the range can be extended with external power amplifiers and antennas [27].

3.4.3 Power

The ANT protocol developed the low-power form factor needed to run off of a coin-sized battery [29]. To save power, the ANT protocol uses very short duty cycles and utilizes asynchronous transmission when required by the application. Asynchronous transmission only occurs when there are new data to send, allowing the devices to be in a sleep mode for extended periods of time, drastically reducing the power consumption [31]. When the data rate changes to a burst transfer mode, the power consumption increases. Similarly, a high duty search, which is when a slave unit searches for a master unit, consumes more power [31].

Each device has the ability to control the transmit power levels. In addition, it is possible to decimate the message rate of the slave unit relative to the master unit, as long as the possibility of missed data packets is tolerable [31]. Table 3.4 summarizes the power level settings for various ANT devices.

The nRF24AP2 ANT chip, made by Nordic Semiconductor, possesses ultra low-power consumption, down to an average current draw of 9 μA when operating in a simple broadcast mode [33].

Table 3.4: ANT Output Power Level Settings per Device [31]
(Adapted from ANT Message Protocol)

Setting	nRF24Ap1	ANT11TRx1	AT3 chipsets, nRF24AP2 modules	nRF51xxx	CC257x modules
0	-20 dBm	-20 dBm	-18 dBm	-20 dBm	-20 dBm
1	-10 dBm	-10 dBm	-12 dBm	-12 dBm	-10 dBm
2	-5 dBm	-5 dBm	-6 dBm	-4 dBm	-4 dBm
3	0 dBm	0 dBm	0 dBm	0 dBm	0 dBm
4	N/A	N/A	N/A	4 dBm	4 dBm

3.5 Other Technologies

Among technologies such as Bluetooth and ANT, lies another low-power, short-range technology that was introduced in Section 3.2. ZigBee creates highly reliable mesh networks that allow all participating devices to communicate with each other by acting as repeaters, as illustrated in Figure 3.2 [34]. Other supported network topologies include the star and the cluster tree.

Globally, the ZigBee raw data rate is limited to 250 Kbps on the unlicensed 2.4 GHz ISM band. In North America, the data rate is limited to 40 Kbps in the 915 MHz band [24]. ZigBee has 16 frequency channels that require 5 MHz of bandwidth in the 2.4

GHz band and 10 channels in the 915 MHz band [25], [35], which is a significant difference from the Bluetooth and ANT frequency spectrums. The main objective was intended for short-range reliable operation, low cost, and low power consumption [26].

To minimize the interferences and collisions, ZigBee implements a "listen before you talk" strategy [35]. This technique in the 802.15.4 standard is known as carrier-sense, multiple-access (CSMA). The device first listens to see if the channel is busy. Only when the channel is not busy, the device begins to transmit messages. Furthermore, the ZigBee device that initiates formation, known as the ZigBee Coordinator (ZC), is required to scan the available channels to automatically select the channel with the least interference [35].

The frequency spreading technique of the IEEE 802.15.4 standard, called direct-sequence spread spectrum (DSSS), promotes coexistence in the ISM band. DSSS implements a pseudo-random code sequence that modulates the basic carrier signal and encodes the data being transmitted [35].

Transmission distances range from 10 to 1600 meters depending on the power output and environmental conditions [36]. The low power consumption stems from the ability of powered devices to go to a sleep mode for hours or even days. In addition, the duty cycle of ZigBee was designed to be very low. Operation modes are beacon mode and non-beacon mode. In beacon mode, the ZC wakes up other nodes to check for messages. If there are no messages, the nodes and the ZC go back to a sleep mode [25]. Thus, the device batteries can last for years.

The ZigBee Health Care standard already supports medical devices such as pulse oximeters, blood pressure monitors, pulse monitors, weight scales, and glucose meters [34]. The benefits of the large mesh networks and sleep modes enable ZigBee to be one of the main contenders among the short-range wireless technologies.

3.6 Summary

Several short-range wireless technologies use the 2.4 GHz ISM band. Each technology may have been created for specific applications initially, but they are now beginning to overlap and compete in similar sectors. A few of the areas of overlap include home automation and control, fitness, medical, and more. Nonetheless, each technology is dominant in the sector that it serves. For example, Bluetooth is popular in consumer electronic devices, ANT in sports and fitness, and ZigBee in health and automation. These technologies are varying their capabilities and features to reach out to new areas, and many of the variations include low-power modes, sleep modes, and burst modes. Table 3.5 summarizes the technologies discussed in the previous sections.

Table 3.5: Comparison of Short-Range Wireless Technologies
(Adapted from Bluetooth Basics, ANT Message Protocol, and ZigBee Specifications)

Market Name	Bluetooth	ANT	ZigBee
Standard	IEEE 802.15.1	Proprietary	IEEE 802.15.4
Network Topology	Ad hoc piconet	Mesh, star, tree	Mesh, star, tree
Devices per Network	8	2 to thousands	2 to thousands
Power Profile	Days	Many years	Years
Data Rate	1 Mbps	1 Mbps	20 to 250 Kbps
Approx. Range (m)	1 to 100	1 to 30	10 to 1600
Success Metrics	Low cost Convenience	Ultra low power Low cost	Low power Low cost

The comparison of wireless short-range technologies presented in this section considers important factors such as the frequency band, interference avoidance, range, and power. Although they are similar in these features, each technology is superior in their own regard. In comparison to ANT and ZigBee, Bluetooth may appear inferior due to its ability to manage only eight devices in one network. Contrary to that belief, Bluetooth is a very popular short-range technology because it is reliable for a handful of devices and can combine many piconets in one network. Moreover, ANT and ZigBee have many similar features with each other, but ANT consumes less power in

comparison. Thus, devices running off of coin cells can last for many years. Figure 3.6 illustrates the range relationship between ANT, Bluetooth, and ZigBee.

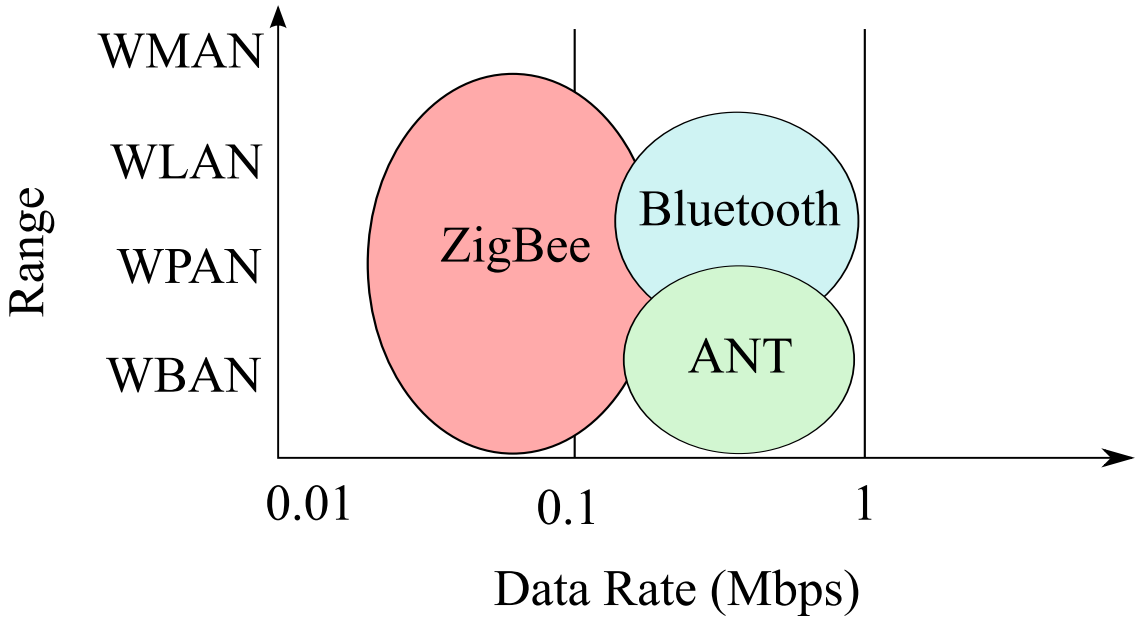

Figure 3.6: Range Versus Data Rate for ANT, Bluetooth, and ZigBee Technologies

These three technologies partially overlap with each other, while each technology maintains their own network range. WMAN is the abbreviation for wireless metropolitan area network.

It is notable to mention that the factors that were not presented, such as security level, complexity, size, space, user experience, and licensing fees, must also be considered for a thorough analysis of a wireless technology. However, in the context of consumer mobile devices, the power attributes discussed in the previous sections are very significant. Due to the limitations in device power-source size, power consumption is one of the most critical factors in the design and varies for each application.

4 Methods

The previous chapter compared the short-range wireless technologies that are typically integrated in most commonly used consumer electronic devices. The number of applications that the recovered heart rate signal can be used in is likely to increase if this signal can be wirelessly transmitted. This chapter presents the hardware and software test setups for data acquisition, the analysis, the test results, and the summary.

4.1 Setup

The heart rate signal was measured by a PPG sensor for Arduino microcontrollers, known as the Pulse Sensor®, and is pictured in Figure 4.1.

Figure 4.1: Pulse Sensor Front

The reference input to the adaptive filter was the accelerometer signal. Both of the signals were inputs to the microcontroller that communicated to a computer by USB. The data were stored on a computer and processed by MATLAB. Figure 4.2 illustrates the test setup configuration, in which the accelerometer was placed next to the PPG sensor on the wrist that is depicted in Figure 2.6. Both of the signals were inputs to the microcontroller that transmitted data to a computer.

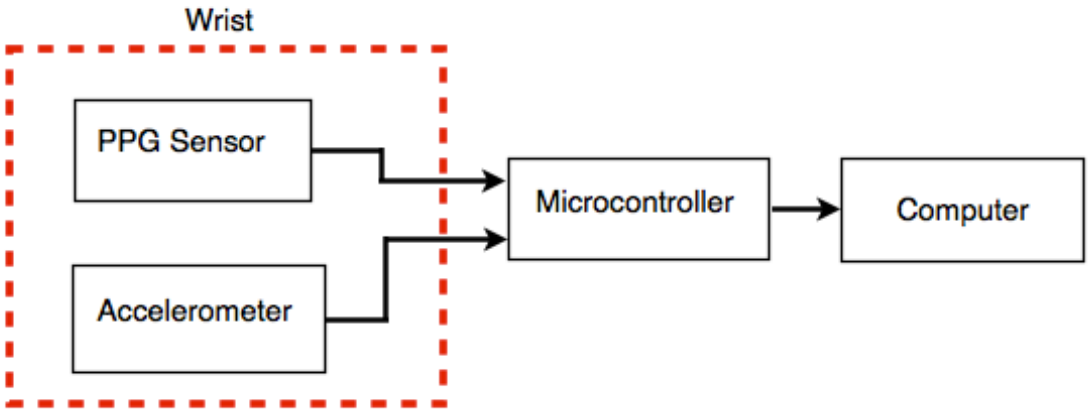

Figure 4.2: Setup Configuration

4.1.1 Hardware

The PPG sensor used an ambient light sensor made by Avago®, with a peak spectral response between 500 nm and 600 nm. For the optimum results, the PPG sensor has integrated into it a green Super Bright LED from Kingbright®, a passive resistor-capacitor filter, an operational amplifier, and a few other components. The PPG sensor

was optimized for both 3 V and 5 V. Figure 4.3 pictures the back of the Pulse Sensor, which contains its circuit components.

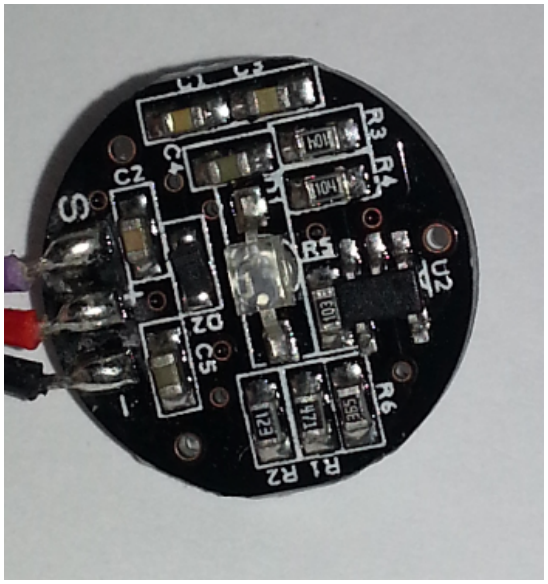

Figure 4.3: Pulse Sensor Back

The pulse sensor integrated a Microchip MCP6001 low-power operational amplifier of approximately 4.5 mA that has a gain-bandwidth product of 1 MHz. Thus, the gain of the signal depended on the frequency range of the heart rate. Heart rate is measured in beats per minute (BPM) and can be roughly between 40 and 200 BPM, depending on the individual and their level of activity and health condition. The BPM range corresponds to a signal frequency of 0.66 Hz and 3.3 Hz. Therefore, the gain of the operational amplifier, calculated directly from the gain-bandwidth product, was between 100 dB and 120 dB for the heart frequency range.

The accelerometer block, illustrated in Figure 4.2, provides analog output signals

proportional to the acceleration that it senses. The accelerometer has a small form factor of 4 mm by 4 mm by 1.45 mm and consumes low power, 350 µA. The magnitude of acceleration can be measured as a result of any motion, shock, or vibration. The bandwidth of the accelerometer signal can be adjusted by adding capacitors across the output pins and the ground reference. Equation 4.1 defines how to set the capacitor value of the bandwidth for each axis signal.

$$F_{-3dB} = 5\mu\text{F} / C_{(x,y,z)} \tag{4.1}$$

The capacitor implements low-pass filtering for antialiasing and noise reduction. To achieve a 3 dB bandwidth of 5 Hz, capacitors with values of 1 µF were selected. Figure 4.4 pictures the front of the accelerometer and the direction of each axis.

Figure 4.4: Accelerometer Front

After viewing the results and reading product reviews, it was discovered that the *X* and *Y* pins were switched than what was marked on the front of the accelerometer. The

X, *Y*, and *Z* accelerometer outputs and the heart rate signal were four inputs to the Arduino microcontroller.

The microcontroller board was an Arduino Micro with an Atmel ATmega32u4 processor, depicted in Figure 4.5. The primary considerations for choosing the microcontroller were form factor and software compatibility.

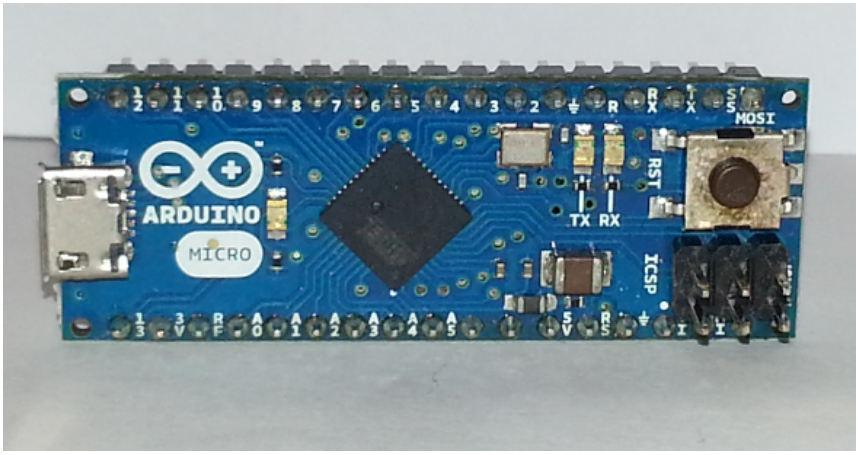

Figure 4.5: Arduino Micro Front

The microcontroller circuit board dimensions are 4.8 cm by 1.77 cm, which allows for easy placement on a breadboard or an electronic module holder. The Arduino-based microcontrollers allow for serial communication with a computer over USB, appear as virtual COM ports to software on computers, and are easily programmable with Arduino software.

The three outputs of the accelerometer and the single output of the Pulse Sensor were four analog inputs to the microcontroller. The analog inputs were passed through a 10-bit analog-to-digital converter (ADC) that was integrated into the microcontroller

analog channels. The ADC provides 2^{10}, or 1024, different values to represent the analog voltage as a decimal number. The ADC equally divides 1024 by the full swing of the input voltage, also known as the full-scale range.

The default voltage that powered the Pulse Sensor was 5 V. However, the upper end of the full-scale range was changed from 5 V by applying an external voltage to the AREF pin on the microcontroller and by utilizing the *analogReference()* function in Arduino software. An external voltage of 3.3 V supplied power to the Pulse Sensor.

4.1.2 Software

The Arduino software received the transfer of the analog signals to the computer. The Arduino integrated development environment (IDE) is an open-source program environment, written in Java, and made to work across all Arduino microcontroller platforms. The Arduino IDE contains built-in libraries, functions, and example code that is compatible with any Arduino microcontroller.

The Pulse Sensor, used as the PPG sensor in this experiment, was delivered with programming code that is compatible with the Arduino Uno microcontroller. The open-source code for the Pulse Sensor was modified to be compatible with the Atmega32u4 processor on the Arduino Micro microcontroller.

The software code was programmed to sample the heart rate and the accelerometer signals at 125 Hz, or every 16 milliseconds. The signals were in the range of 0.66 Hz to

3.3 Hz. Thus, the sampling rate obeyed the Nyquist criterion, such that oversampling occurred at over 20 times the Nyquist sampling rate of 6.6 Hz. The digital values of the analog outputs were transferred to the computer through USB ports and processed in MATLAB® R2010b.

MATLAB is a very powerful computing tool that has the ability to operate on large data sets. The basic element of MATLAB is the matrix, and mathematical operations, such as cross-products, dot-products, and determinants, are built-in functions of the language. All recorded data sets from this experiment were passed through adaptive filters in MATLAB. The MATLAB graphical user interface allowed for ease of complex computation and data plotting.

4.2 Data Acquisition

The PPG sensor and accelerometer signals were wired to the microcontroller analog inputs. The four signals were passed through ADCs that quantized the voltage levels to digital values in the range of 0 to 1024. The values for each signal were transmitted to the Arduino serial monitor and saved in a text file for processing.

Processing the signals required importing the recorded data from the text file into the MATLAB workspace. It is important to note the condition of the heart rate signal before processing the distorted signal. The heart rate signal, illustrated in Figure 4.6, clearly shows the peaks of each pulse in a six-second window.

Figure 4.6: Wrist PPG Heart Rate

Seven noticeable pulses were recorded when there was no body movement. The Arduino IDE controlled the PPG sensor to maintain a mean voltage of VDD/2, where VDD was 3.3 V. The ADC midpoint value was 512 and each peak above that value corresponded to one heart pulse.

The heart rate signal was distorted during body movement, as a result of ISI. The anatomy of the wrist also affected the signal integrity during body movement. The reconstruction of the heart rate signal was possible with the implementation of adaptive filters. The interference-cancelling adaptive filters used the reference input to minimize noise from the primary input and to guide the output to the correct values as if the primary input was not distorted to begin with. The heart rate signal during body

movement is illustrated in Figure 4.7.

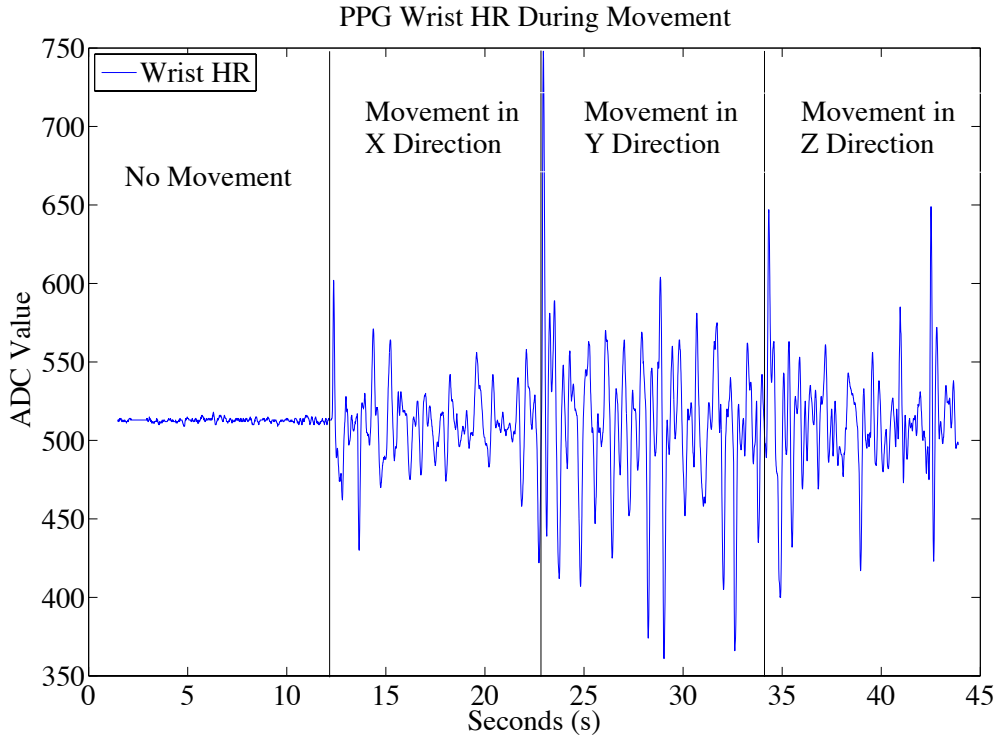

Figure 4.7: PPG Wrist HR During Movement

The heart rate signal was recorded for roughly 45 seconds during four distinct periods of time. The four distinct periods include no movement and movement in each axial direction alone that are roughly 10 seconds each. It is clearly visible in Figure 4.7 that the heart rate signal was distorted during arm movement in the X, Y, and Z directions, as this signal should ideally resemble the characteristics of those in Figure 4.6. The measurement of the heart rate signal during body movement was not possible without a supplemental input or additional signal processing. The adaptive filters were applied to the distorted signal in MATLAB for signal reconstruction. The accelerometer signals during body movement are illustrated in Figure 4.8.

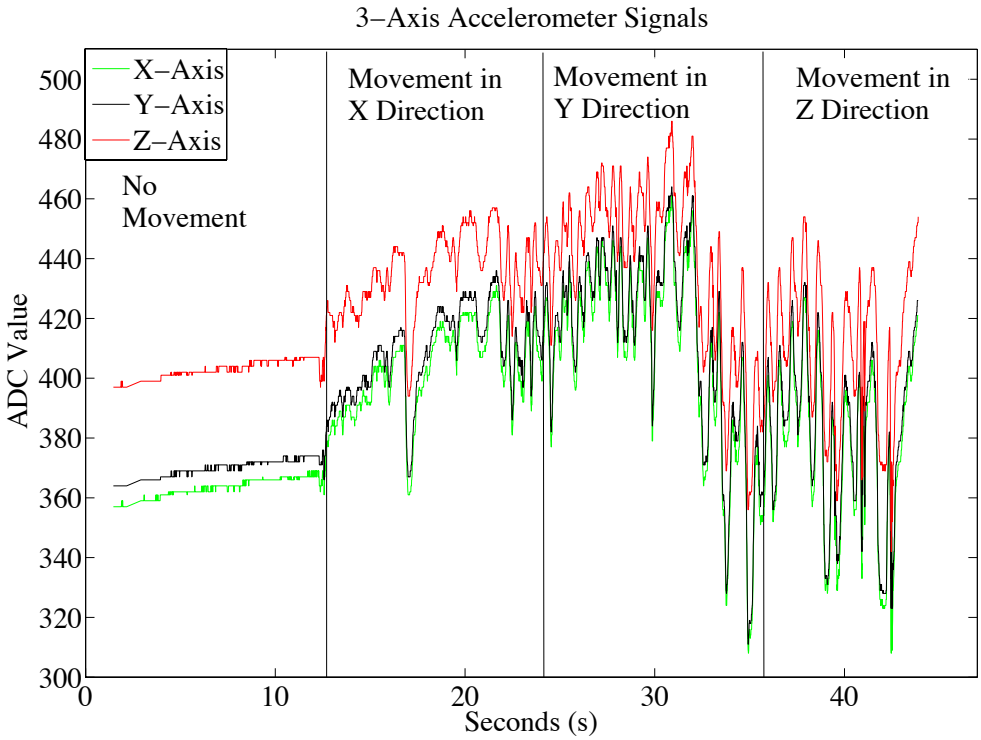

Figure 4.8: Accelerometer Signals During Movement

The three accelerometer signals during the last three distinct time periods were unique in comparison to each other. Each of the three signals in each of the last three distinct periods were used as the reference input to the adaptive filter, one at a time. The adaptive filter produced a system output for each axial direction.

The heart rate and accelerometer signals were superposed in one MATLAB plot, illustrated in Figure 4.9, to show their relationship in each window of time.

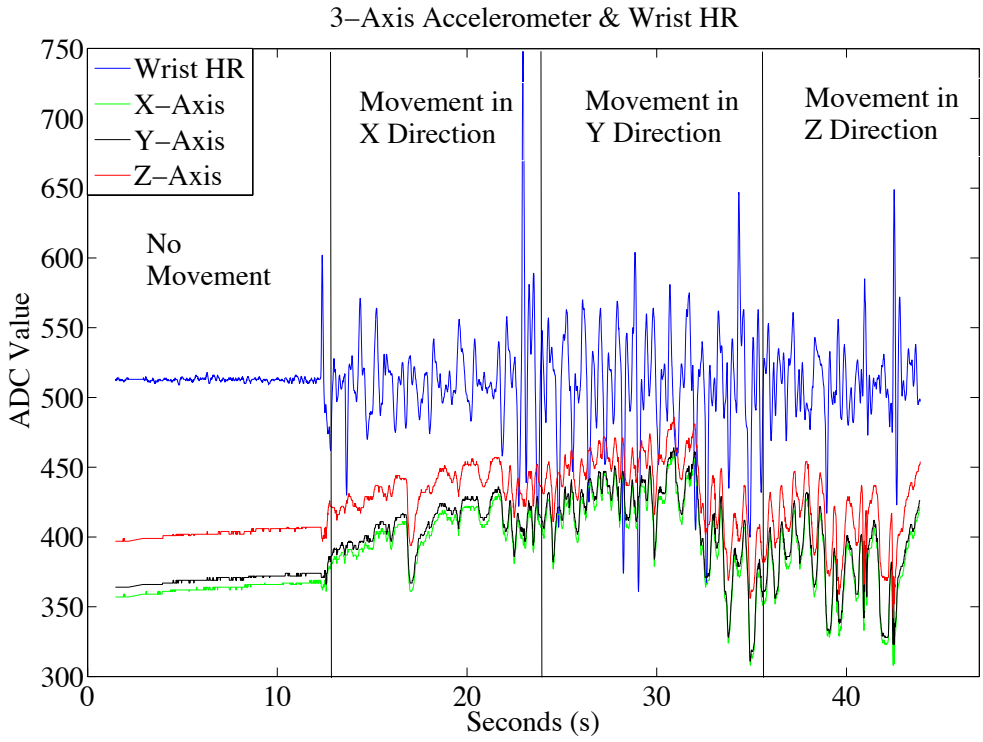

Figure 4.9: Accelerometer Signals and Wrist HR During Movement

The axis signal, corresponding to the direction of movement, had a high correlation to the heart rate signal in that respective window of time. For example, the *X-axis* signal was closely related to the heart rate signal when movement in the *X* direction occurred. Each individual accelerometer signal was used as the reference input to the adaptive filters, providing nine unique data sets to use as the reference input.

4.3 LMS Adaptive Filter

The LMS adaptive filter was applied to nine sets of data to produce nine unique system outputs. The nine sets correspond to the *X*, *Y*, and *Z* axes in the last three time periods, illustrated in Figure 4.9.

The system output is the error signal from the adaptive filter, depicted in Figure 2.2. The filter output for the accelerometer *X-axis* with movement in the *X* direction is shown in Figure 4.10. It is important to note that the signals were superposed on the same plot and reduced to unit amplitude for signal processing.

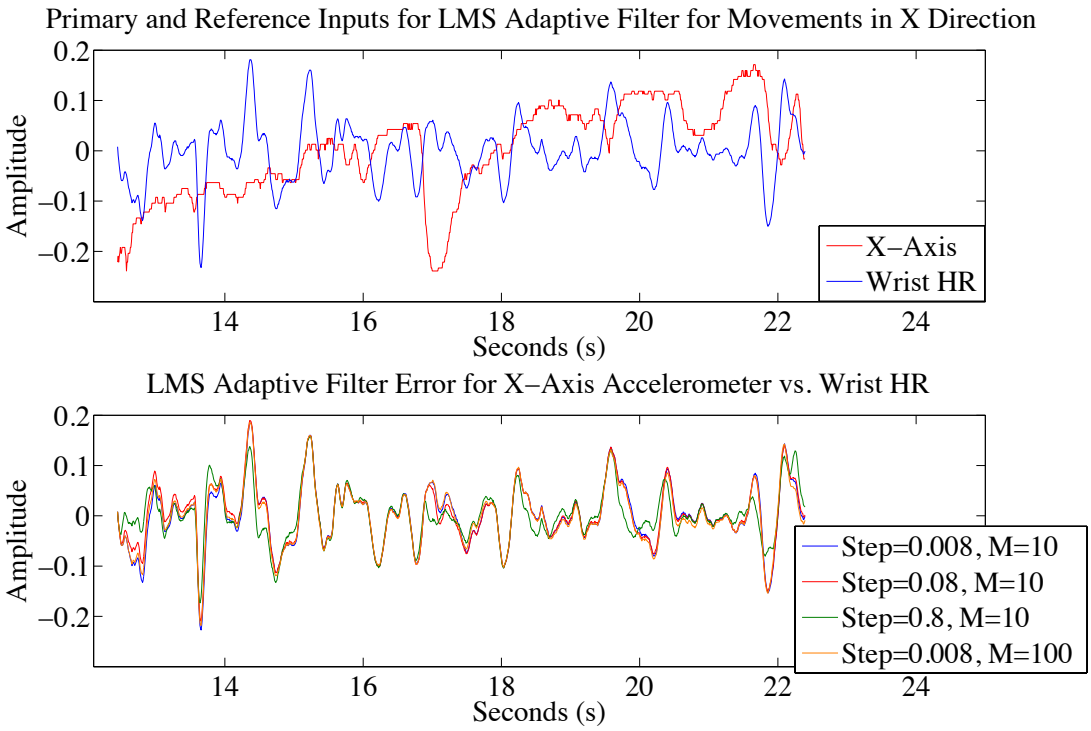

Figure 4.10: LMS Output for X-Axis in X Direction

The adaptive filter was tested on different step-sizes μ and filter lengths M, with only one parameter changed per test. The step-size μ and the filter length M influenced the filter coefficients or tap weights, as referred to in Chapter 2. The step-sizes of 0.008, 0.08, and 0.8 were used to compare the system output to the value of the step-size. The smaller the step-size was, the more time it took for the output to converge. Conversely, the larger the step-size was, the greater the jump was in the system output. A series of

tests can provide a reasonable value for the step-size. Nonetheless, it is favorable to have a slower convergence rate and a more accurate output than a fluctuating and a noisy output due to a large step-size value.

The filter order length is another critical factor in the performance of the adaptive filter. Filter order lengths of 10 and 100 with the same step-sizes were tested to compare differences in the system output. The greater the filter length was, the more complex the computations were. A series of tests may provide a more accurate filter order length for the optimum system output. The filter output for the accelerometer *Y-axis* with movement in the *X* direction is shown in Figure 4.11, and the filter output for the accelerometer *Z-axis* with movement in the *X* direction is shown in Figure 4.12.

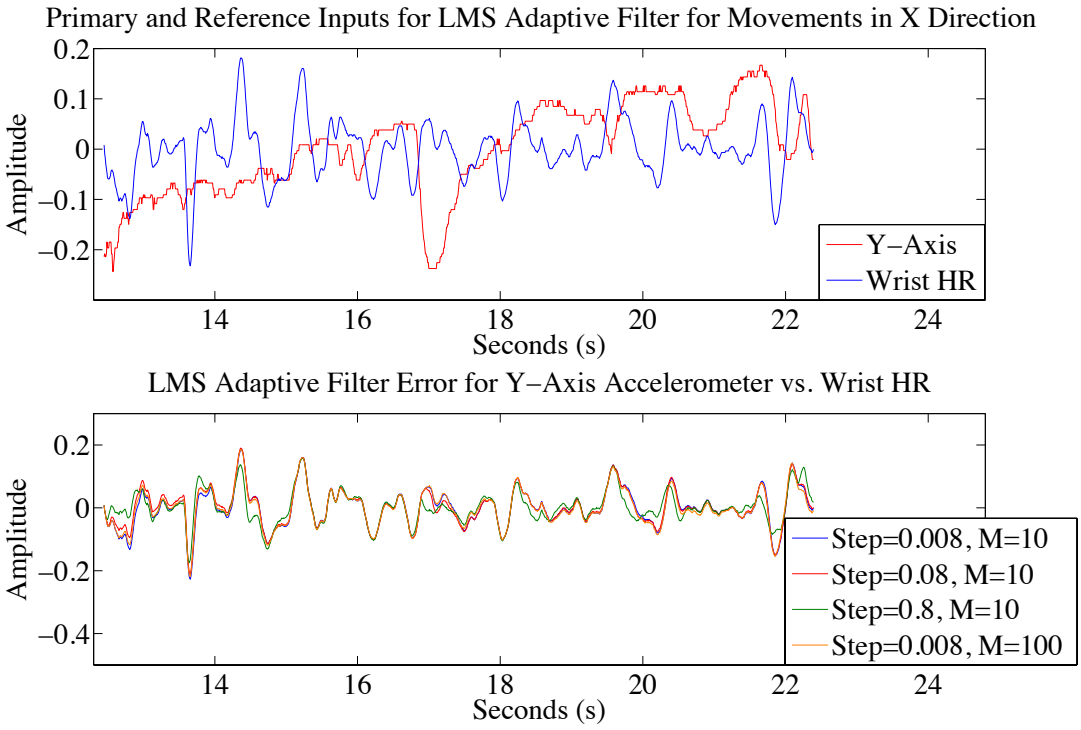

Figure 4.11: LMS Output for Y-Axis in X Direction

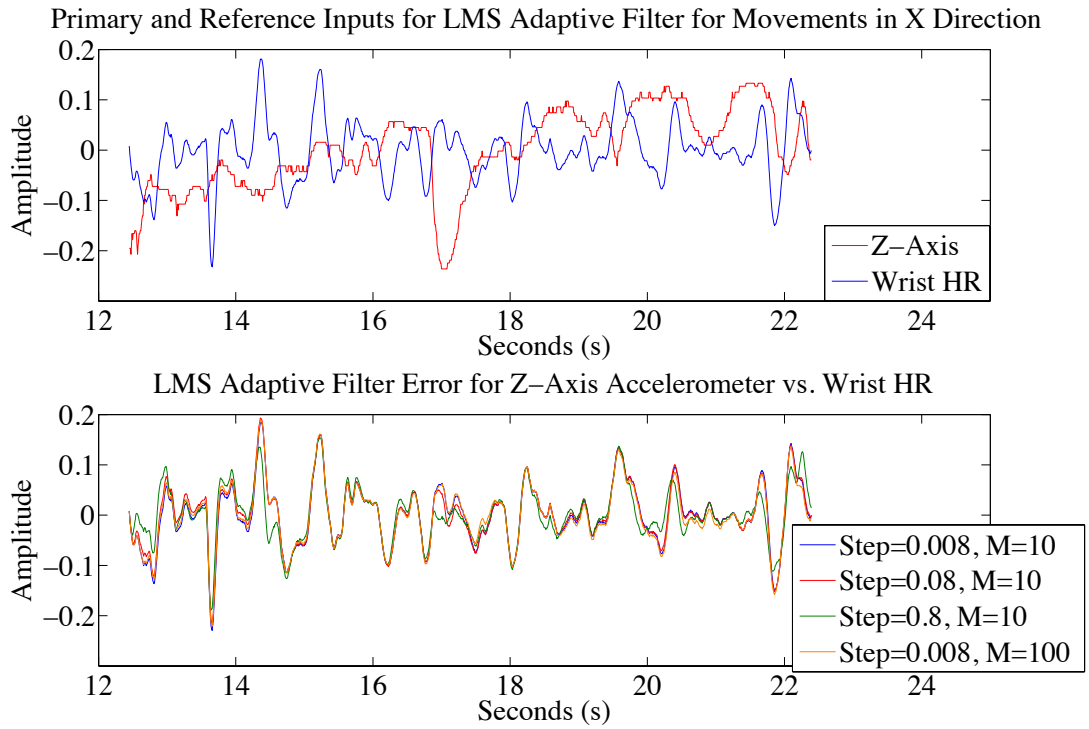

Figure 4.12: LMS Output for Z-Axis in X Direction

In the time period for movement in the *X* direction, it did not appear that the heart rate was fully reconstructed in any of the three system outputs. The different step-sizes and filter order lengths reduced the noise content of the primary input signal, but the peaks of the heart rate signals in these results were not as clearly visible as they were when there was no body movement, as in Figure 4.6.

The accelerometer signals for movement in the *Y* direction were tested with the same LMS adaptive filter as they were for movement in the *X* direction. Based on the results, it appears that the correlations between the accelerometer heart signal were higher than they were for movement in the *X* direction. The filter output for the accelerometer *X-axis* with movement in the *Y* direction is shown in Figure 4.13, and the filter output for the accelerometer *Y-axis* with movement in the *Y* direction is shown in Figure 4.14.

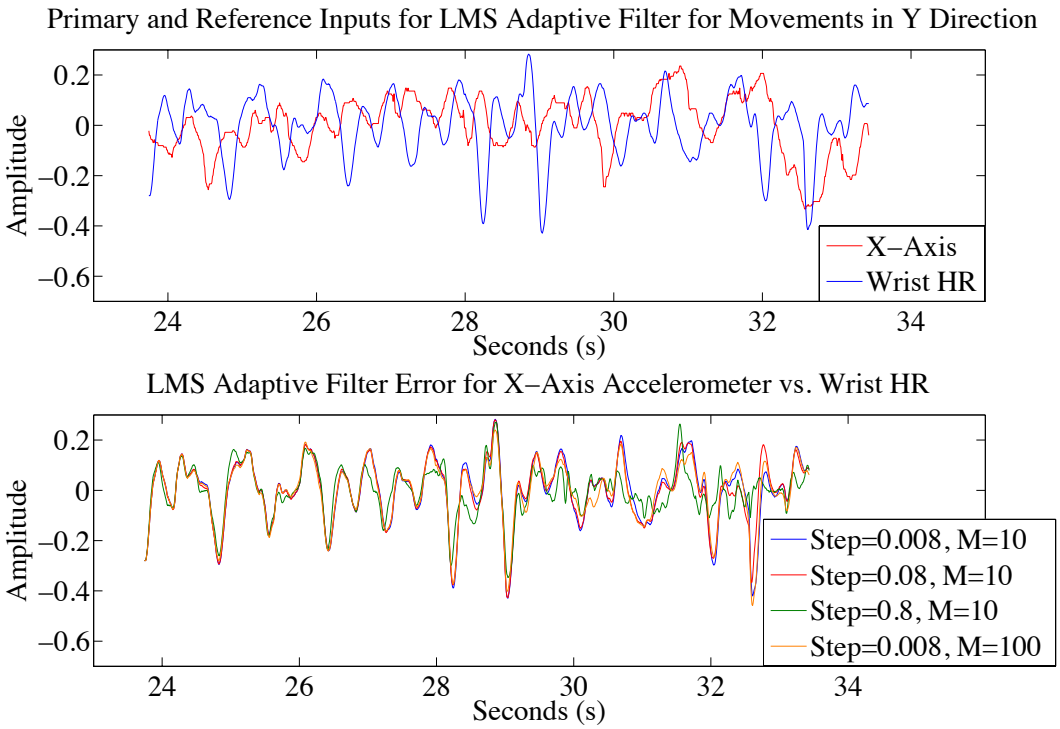

Figure 4.13: LMS Output for X-Axis in Y Direction

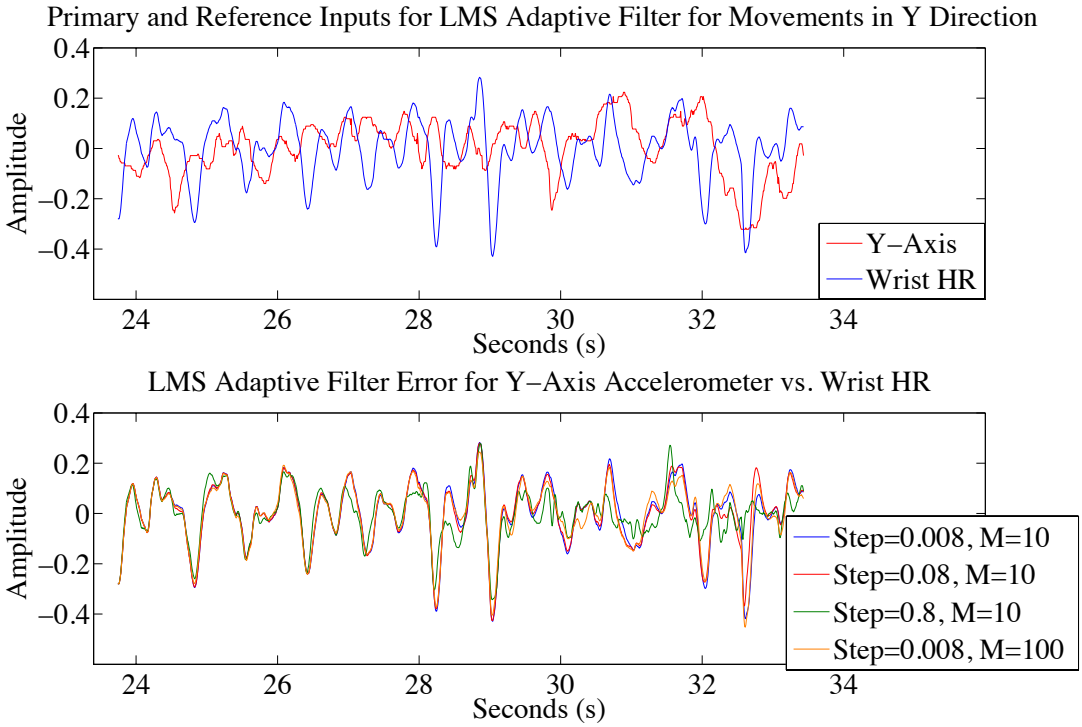

Figure 4.14: LMS Output for Y-Axis in Y Direction

The correlation for the *X* and *Y* accelerometer signals with movement in the *Y* direction appeared to be higher than any of the same signals with movement in the *X* direction. There was a clear time lag between the accelerometer signals and the heart rate. Intuitively, this lag implied that there was a change in the heart rate after some movement had occurred. The heart rate was partially reconstructed in Figure 4.13 and Figure 4.14, and the BPM calculations may be possible with additional signal processing. The filter output for the accelerometer *Z-axis* with movement in the *Y* direction is shown in Figure 4.15.

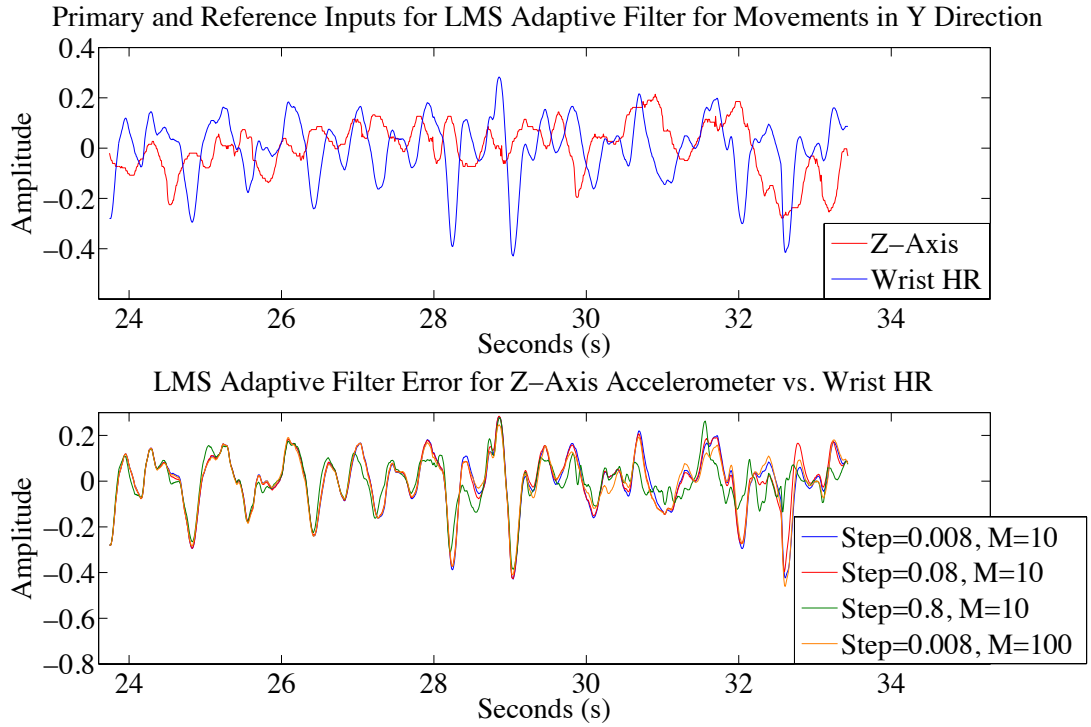

Figure 4.15: LMS Output for Z-Axis in Y Direction

The system output appeared to have some partial reconstruction, but it was also not fully reconstructed. The large step-size of 0.8 produced a system output that had many

fluctuations. This is an undesired effect of adaptive filters of any kind. The filter output for the accelerometer *X-axis* with movement in the *Z* direction is shown in Figure 4.16, and the filter output for the accelerometer *Y-axis* with movement in the *Z* direction is shown in Figure 4.17.

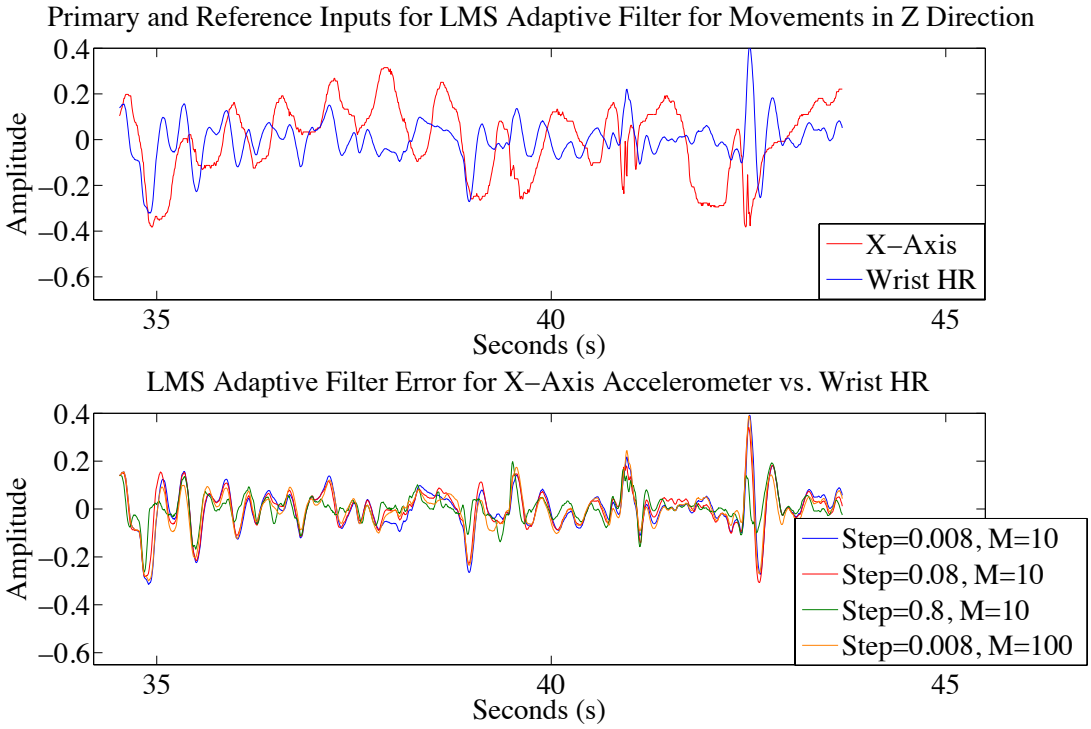

Figure 4.16: LMS Output for X-Axis in Z Direction

The system output for the *X* and *Y* axes did not provide reconstruction of the heart rate signal. The peaks and valleys of the system output did not line up to a reconstructed signal. The filter output for the accelerometer *Z-axis* with movement in the *Z* direction is shown in Figure 4.18.

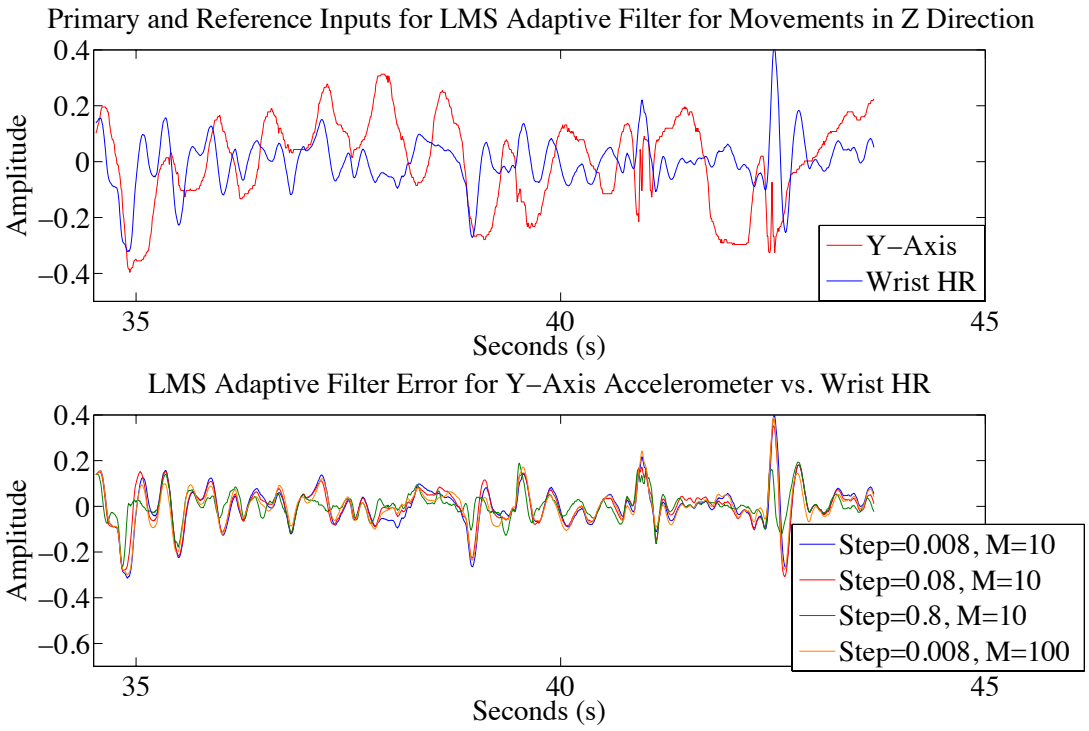

Figure 4.17: LMS Output for Y-Axis in Z Direction

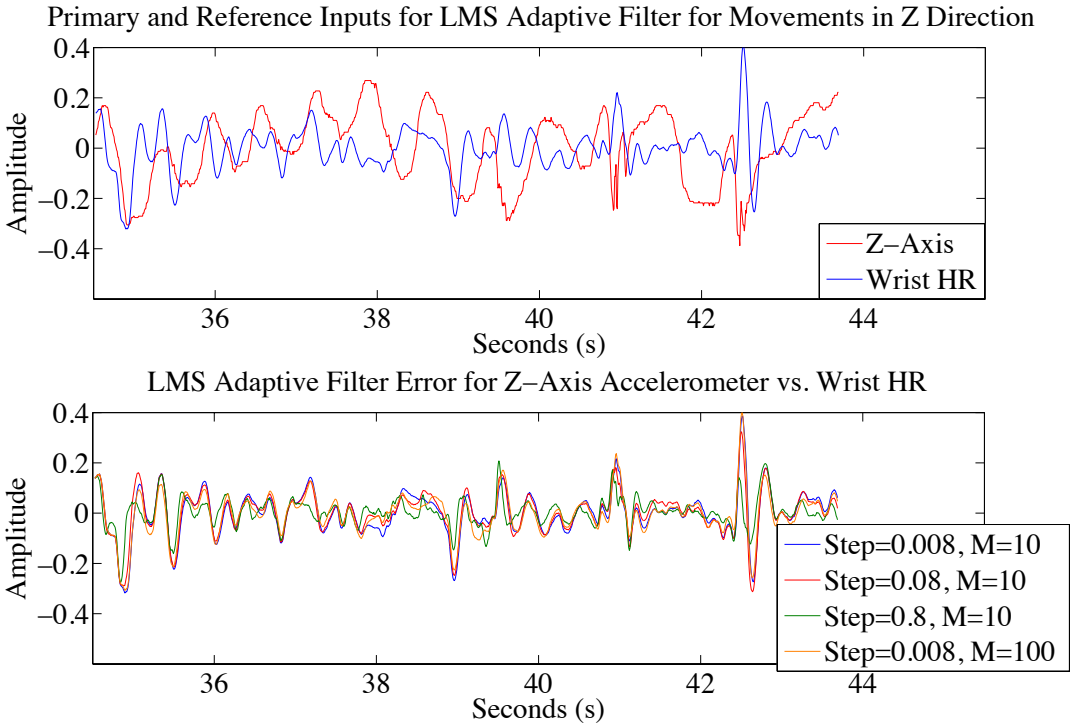

Figure 4.18: LMS Output for Z-Axis in Z Direction

The system output for the *Z-axis* was similar to the system outputs for the *X* and *Y* axes. There was no reconstruction of the heart rate signal, and it would be very difficult for any subsequent signal processing to recover the distorted signal. Of the nine data sets, sets with the system outputs with movement in the *Y* direction yielded the most acceptable results, although the heart rate signal was only partially reconstructed. The next section used the same nine data sets on the RLS adaptive filter.

4.4 RLS Adaptive Filter

The RLS adaptive filter is computationally more complex than the LMS adaptive filter. However, the tradeoff for the increased complexity is the improved accuracy. The nine data sets that were used on the LMS adaptive filter were also used on the RLS adaptive filter for comparison between the two filter types.

The parameters for the RLS adaptive filter included the filter order length, the forgetting factor, and the inverse covariance matrix. These three parameters are outlined in Chapter 2; M is the filter order length, λ is the forgetting factor, and $\boldsymbol{P(0)}$ is the inverse covariance matrix initialization as defined in Table 2.2. The filter output for the accelerometer *X-axis* with movement in the *X* direction is shown in Figure 4.19. The system outputs were varied by changing the parameters, one at a time. The forgetting factor of values 0.99, 0.49, and 0.09 were used. The RLS adaptive filter in this experiment used an upper value of 0.99 and a lower value of 0.09 for the forgetting factor.

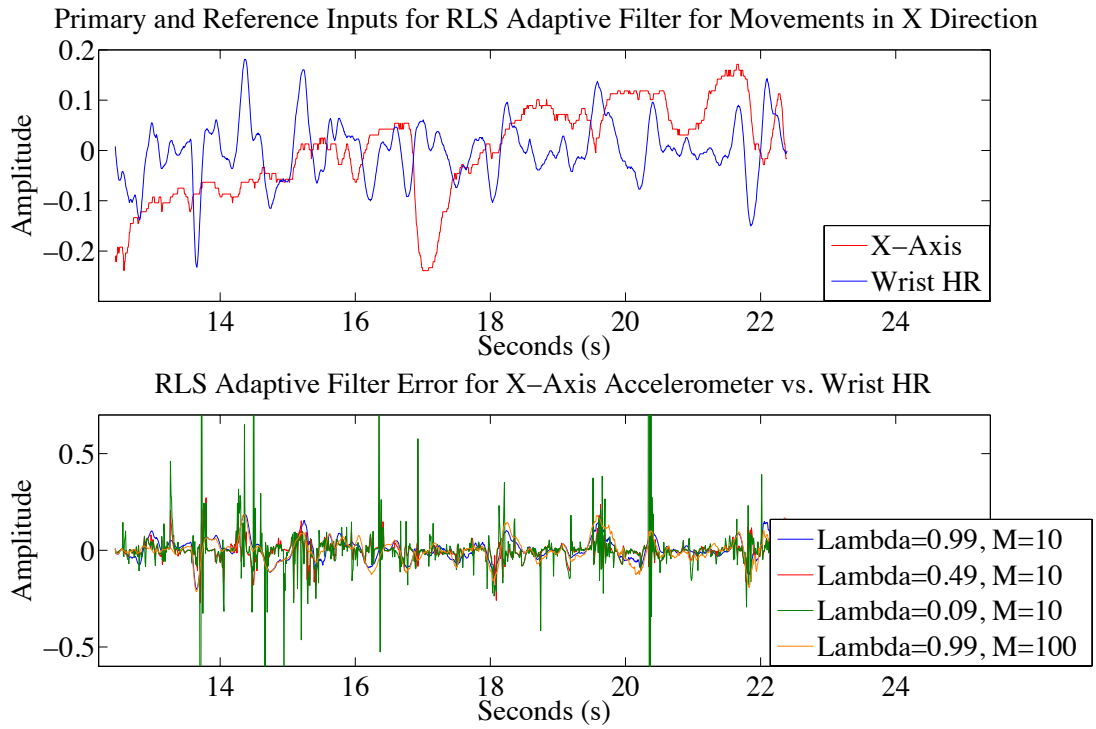

Figure 4.19: RLS Output for X-Axis in X Direction

When the forgetting factor λ is zero, the RLS adaptive filter has no memory. Conversely, when the forgetting factor λ is one, the RLS adaptive filter has an infinite memory. This means that the current value that is being processed by the adaptive filter contains information on the previous output values from the very instant the algorithm starts processing the data. It is important to note that infinite memory is very difficult to implement in reality and a λ value close to one, but not equal to one, is more realistic.

The system output was very noisy with a forgetting factor of 0.09 and a filter order length of 10. This implied that the lack of memory of the previous output values caused large overcompensations in the current value that was being processed by the adaptive filter. The filter outputs in Figure 4.19, without the two smaller forgetting factors of 0.09

and 0.49, are shown in Figure 4.20.

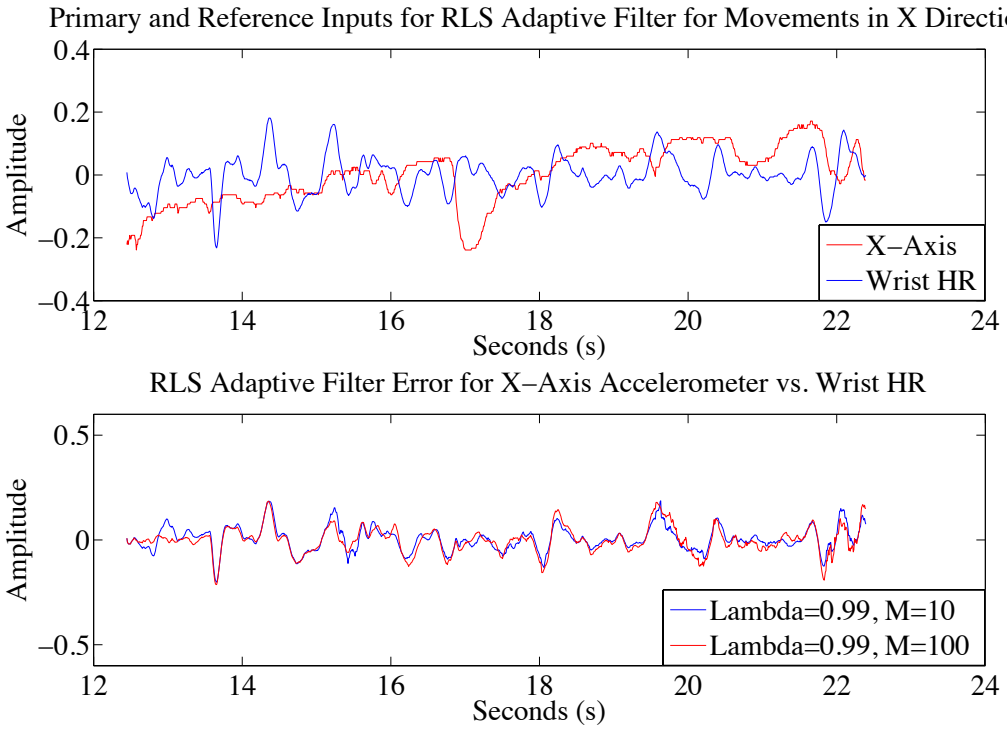

Figure 4.20: RLS Output for X-Axis in X Direction Without Smaller Forgetting Factors

The system outputs that had a large forgetting factor of 0.99 were more stable than the system outputs that had a smaller forgetting factor. This was due to the fact that the RLS adaptive filter retained more information with a larger forgetting factor. In addition, the differences between the two filter outputs were their filter order lengths. The output with a greater filter order length minimized the error more than the output with a smaller filter order length. The filter outputs for the accelerometer *Y-axis* with movement in the *X* direction are shown in Figure 4.21, and the filter outputs for the accelerometer *Z-axis* with movement in the *X* direction are shown in Figure 4.22.

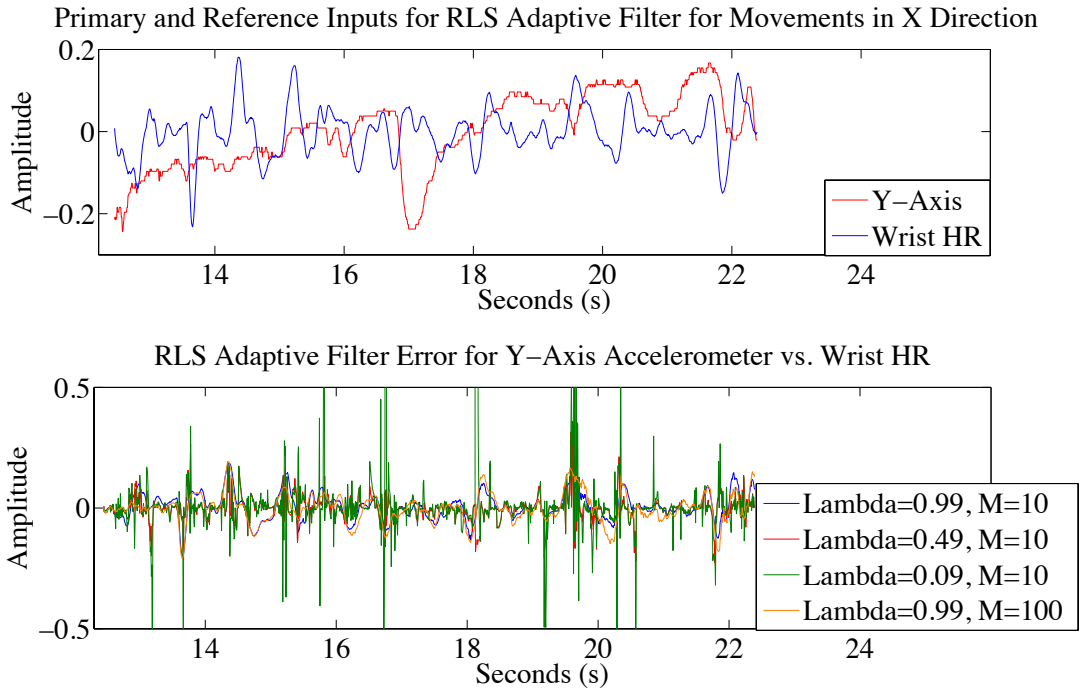

Figure 4.21: RLS Output for Y-Axis in X Direction

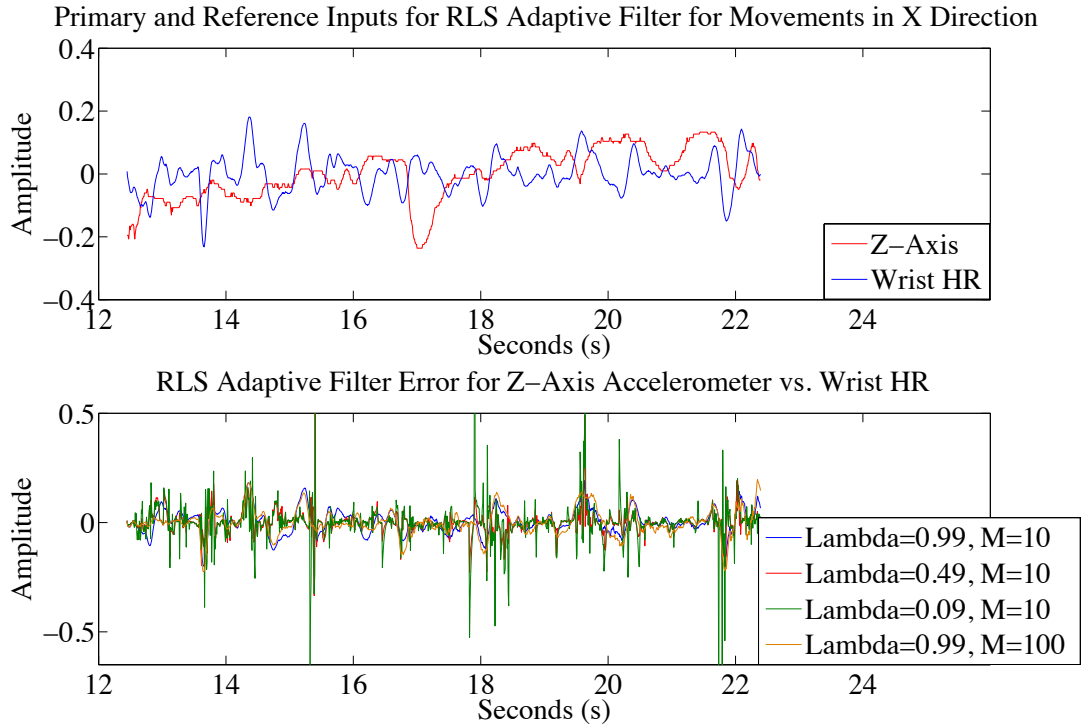

Figure 4.22: RLS Output for Z-Axis in X Direction

The filter outputs for three data sets were very similar to each other with movement in the *X* direction. The primary and reference inputs to the adaptive filter were not highly correlated with each other, which led to poor results. The RLS adaptive filter did not successfully reconstruct the heart rate signal. Although the filter output in Figure 4.20 minimized the error, the BPM calculation requires additional signal processing.

With movement in the *Y* direction, the filter outputs for the accelerometer *X-axis* are shown in Figure 4.23, the filter outputs for the accelerometer *Y-axis* are shown in Figure 4.24, and the filter outputs for the accelerometer *Z-axis* are shown in Figure 4.25. The outputs are the results of forgetting factors equal to 0.99, as the results of lower forgetting factors had poor performance and were not plotted in the following tests.

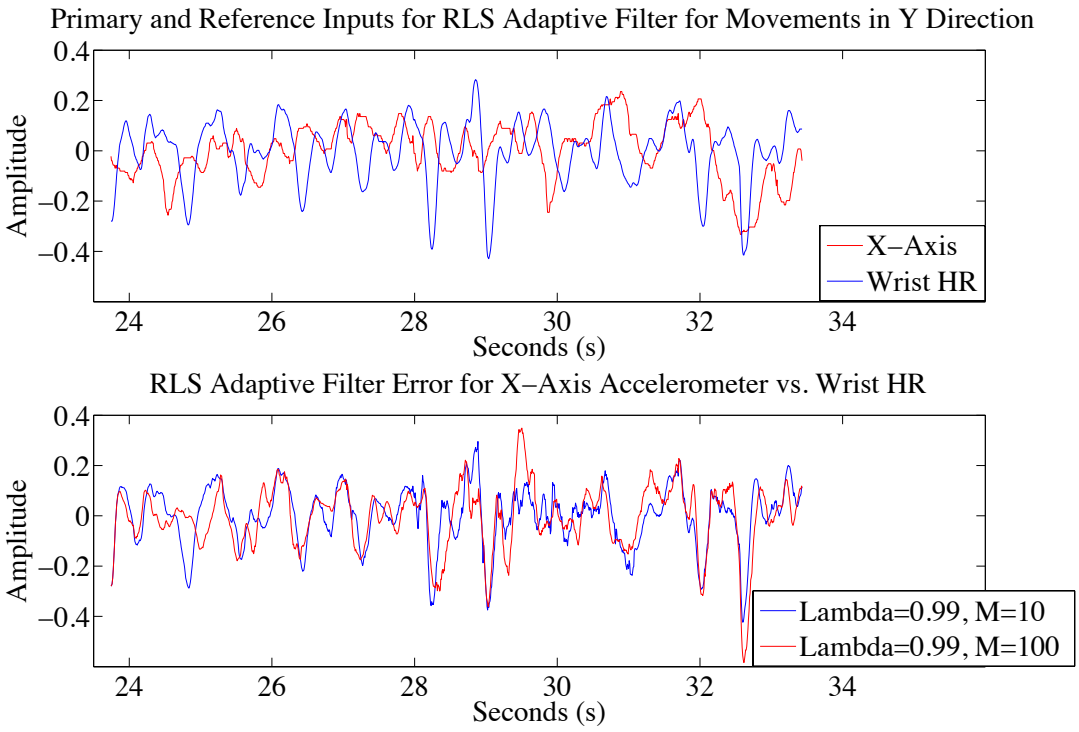

Figure 4.23: RLS Output for X-Axis in Y Direction

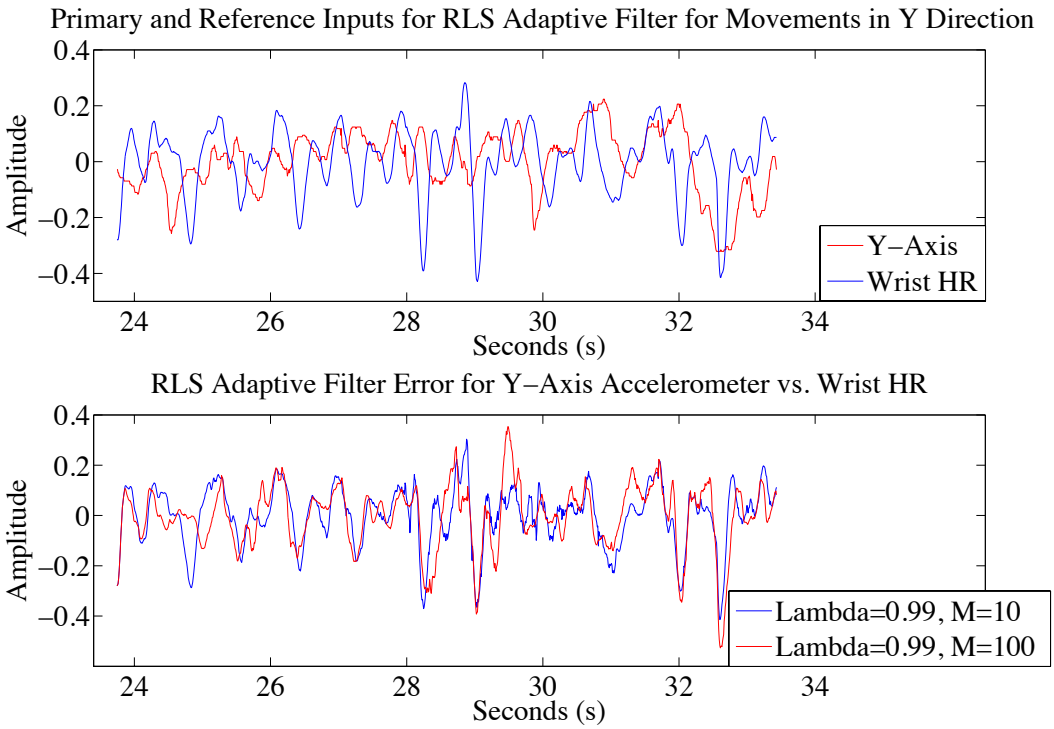

Figure 4.24: RLS Output for Y-Axis in Y Direction

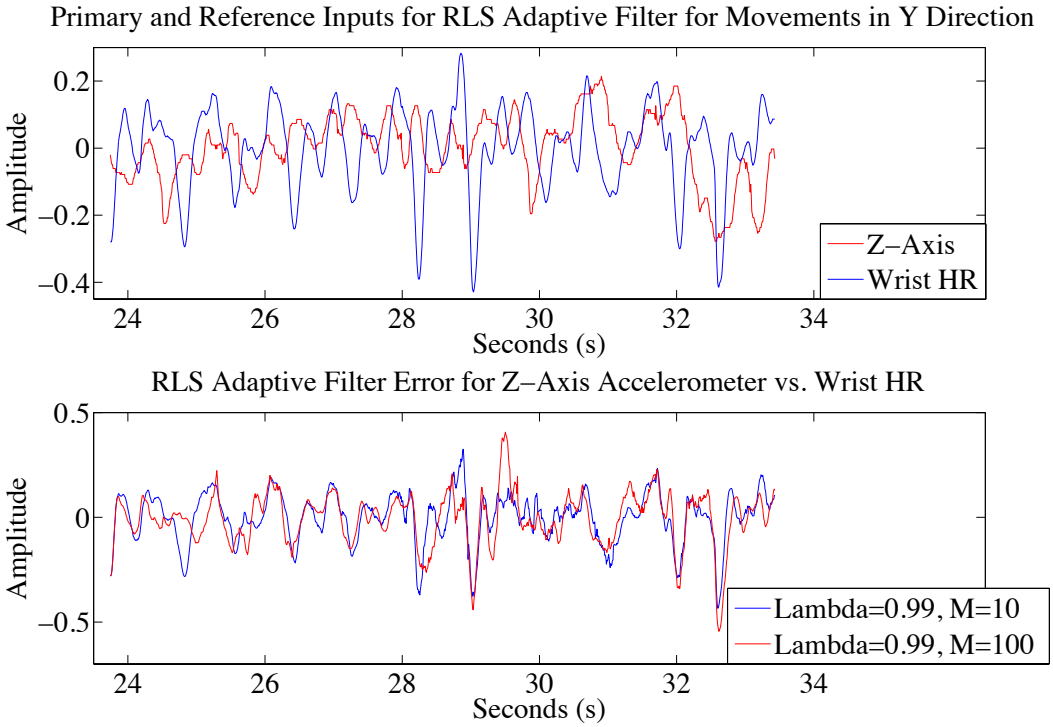

Figure 4.25: RLS Output for Z-Axis in Y Direction

The results for the filter outputs in the *Y* direction show that the filter partially reconstructed the heart rate signal. Additional signal processing is required to calculate the heart rate BPM. Because there was high correlation between the primary and reference inputs, the filter outputs for large forgetting factors produced acceptable results. Three data sets in the *Y* direction, in comparison to the first three data sets, provided results with more potential for signal recovery.

With movement in the Z direction, the filter outputs for the accelerometer *X-axis* are shown in Figure 4.26, the filter outputs for the accelerometer *Y-axis* are shown in Figure 4.27, and the filter outputs for the accelerometer *Z-axis* are shown in Figure 4.28.

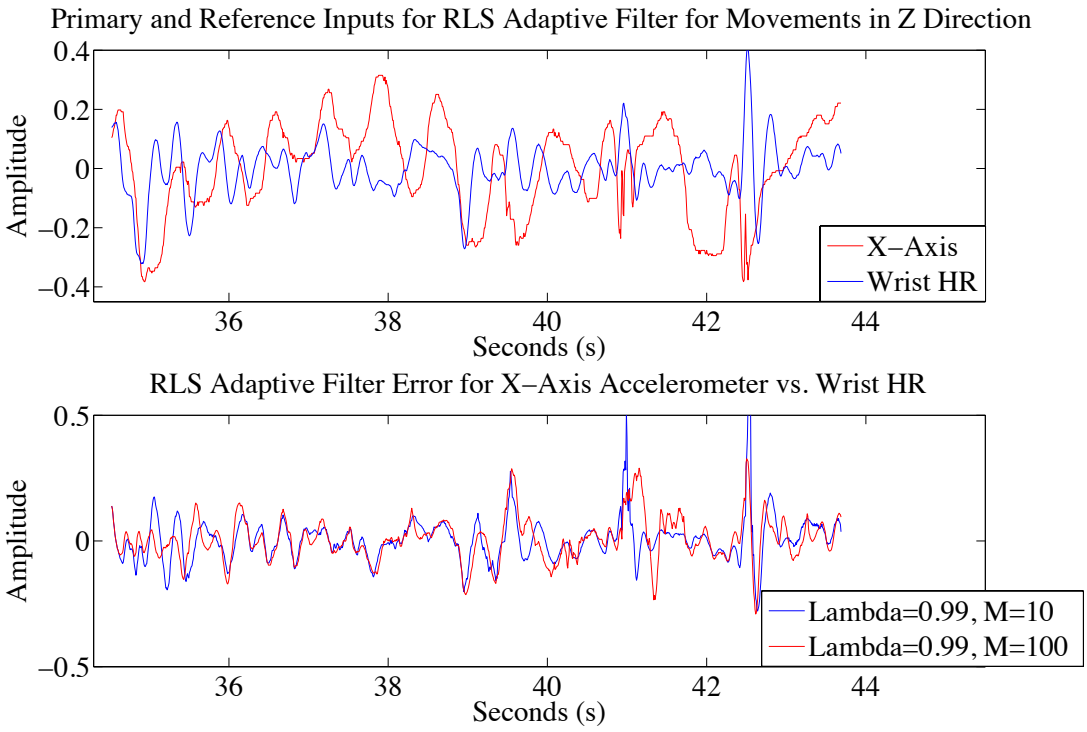

Figure 4.26: RLS Output for X-Axis in Z Direction

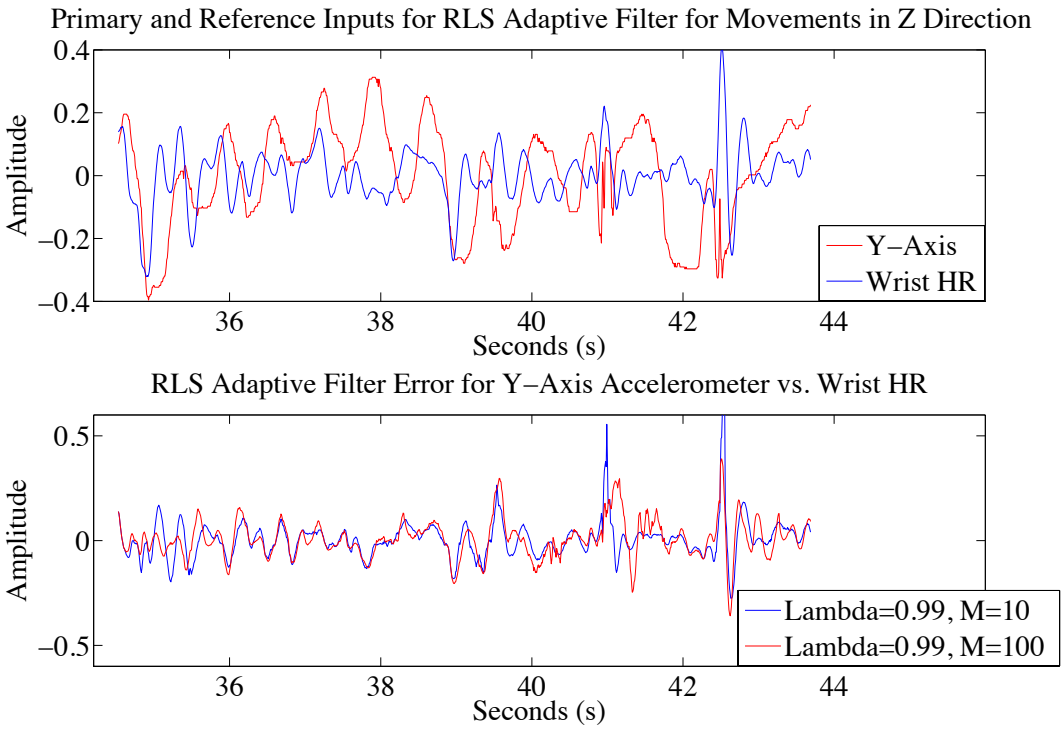

Figure 4.27: RLS Output for Y-Axis in Z Direction

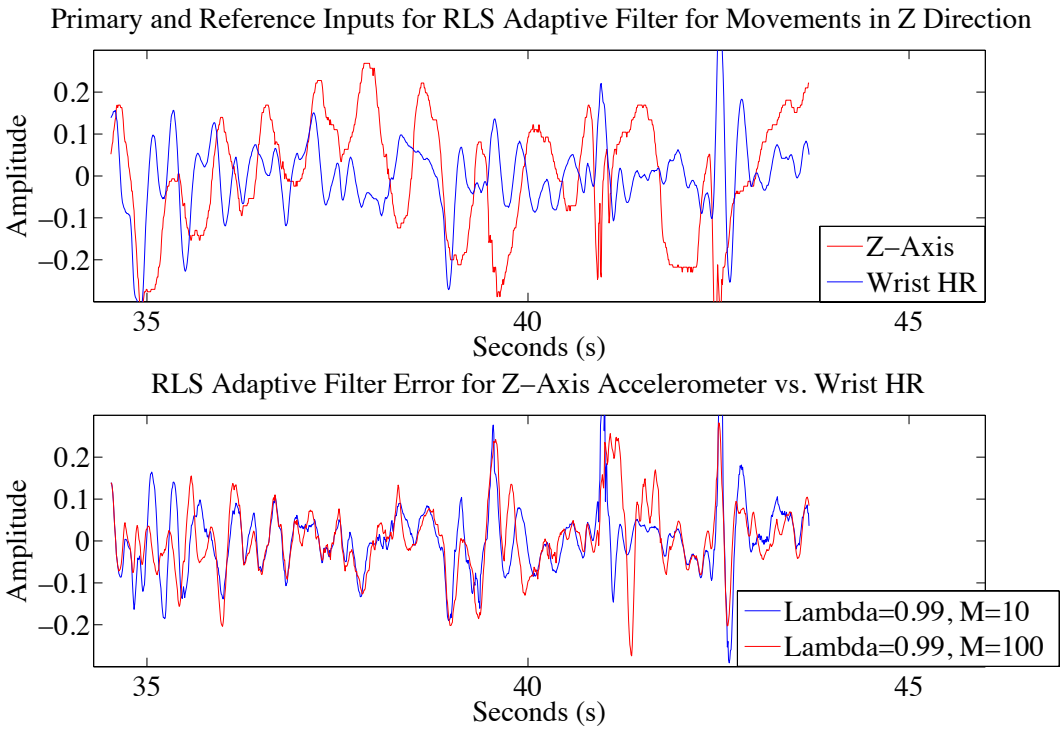

Figure 4.28: RLS Output for Z-Axis in Z Direction

The results for the RLS adaptive filter with movement in the *Z* direction were poor. The reason for this was because the primary and reference inputs were not correlated closely enough to each other. The signal integrity of the filter outputs was very poor. The filter outputs did not contain meaningful information and they lacked sufficient peaks in the time frame required to calculate the heart rate BPM. The heart rate could not be calculated from the filter outputs in any of the last three data sets with movement in the *Z* direction.

4.5 Summary

The LMS and RLS adaptive filters were applied to nine unique sets of data. Each set provided the filter outputs of the primary and the reference signals, which were the heart rate signal and the accelerometer signals, respectively. The previous sections presented the results of the adaptive filters for each of the nine data sets.

The LMS adaptive filter is computationally less expensive than the RLS adaptive filter. In contrast, the RLS adaptive filter minimizes the error quicker than the LMS adaptive filter. The two filters provided somewhat acceptable results in a few of the data sets, where acceptable results imply that the heart rate was partially reconstructed or could have been reconstructed with additional signal processing.

The data sets with movement in the *Y* direction provided the best results. The high correlation between the primary and reference inputs allowed the filter outputs to recover

the heart rate signal partially. The remaining data sets might have had the capabilities of reconstruction had inputs been more closely correlated with each other.

In comparison, the RLS filter outputs reconstructed the heart rate signal more accurately than the LMS filter outputs did. The reasons for this rely solely on the RLS algorithm and its complexity. Nonetheless, the RLS adaptive filter requires more processing power than the LMS adaptive filter to achieve the smaller error. The decision of which of the two adaptive filters to use is defined by the prioritization of the filter characteristics, as presented in Chapter 2.

5 Conclusions

Health and fitness awareness has continued to grow and expand with the new trend of wearable devices. One of the more important topics in this scope is the heart rate conditioning. Among other biological signals that can be monitored in the human body, the heart rate can reveal considerable insight regarding one's health.

The number of calories expended in one day can be calculated accurately based on the heart rate. Although it is possible to accurately obtain this number from sensors attached to fitness machines or chest straps, these methods can be cumbersome for long and continuous periods of time. Hence, the continuous and non-invasive method of PPG for acquiring the heart rate is attractive for applications in wearable devices.

In addition, short-range wireless technologies were explored for heart rate data communication. The transmission of heart rate data to mobile devices provides more versatility in what can be done with the data, such as being stored in an digital fitness journal to be reviewed at a later time. There are three ideal short-range wireless technologies for the heart rate data transmission. Bluetooth, ANT, and ZigBee provide strong features that would be considerable for the findings of this study.

Furthermore, wrist PPG is appealing because it can non-invasively measure one of the most critical biological signals in the human body from low-power LEDs, as opposed to sensors that have large form factors and are likely to consume more power. The only setback with the PPG method are motion artifacts as a result of body movement and how the heart rate signal becomes distorted.

The distorted heart rate signal can be processed digitally and passed through an adaptive filter with a reference input. The reconstruction of the heart rate signal directly depends on the correlation between the desired heart rate signal and the reference signal. The heart rate signal from the wrist PPG method suffers from low amplitude, which impacts the ability of the adaptive filters to reconstruct the distorted signal.

5.1 Future Work

The low amplitude heart rate pulses from this wrist PPG method significantly impair the ability of the adaptive filter to fully reconstruct the distorted signal. The low amplitude was a result of the combination of anatomy of the wrist and the consequences of light reflection and absorption in the tissue and skin during body movement.

The potential to amplify the wrist PPG signal is significant. There are various choices of LEDs that can be used to measure the PPG signal. The wavelengths of the LEDs may have different effects on the reflection and absorption of light by the hemoglobin, which can lead to higher peak-to-peak amplitudes and correlations with other reference signals.

The partial reconstruction of the heart rate signal with the most commonly used adaptive filters indicates that further testing can yield improved results. Other filters have been compared to the LMS and RLS adaptive filters and yield similar or better results. Kalman filters in [12], [13] proved to be successful and have potential to produce improvements on the experiment conducted in this study.

References

[1] P. T. Gibbs, L. B. Wood, and H. H. Asada, "Active Motion Artifact Cancellation for Wearable Health Monitoring Sensors Using Collocated MEMS Accelerometers," in *Proc. SPIE 5765*, pp. 811-819, 2005.

[2] M. Izzetoglu, A. Devaraj, S. Bunce, B. Onaral, "Motion Artifact Cancellation in NIR Spectroscopy Using Wiener Filtering," *Biomedical Engineering, IEEE Transactions*, vol. 52, no. 5, pp. 934-938, May 2005.

[3] W. Kist, "Comparison of Two Pulse Oximeters During Sub-maximal Exercise In Healthy Volunteers: Effects of Motion," *Journal of Exercise Physiology*, vol. 5, no. 1, pp. 42-48, Feb. 2002.

[4] J. E. Hall, "*Guyton and Hall Textbook of Medical Physiology,"* Philadelphia, PA: Saunders Elsevier, pp. 107, 2011.

[5] R. Dresher, "Wearable Forehead Pulse Oximetry: Minimization of Motion and Pressure Artifacts," M.S. Thesis, Worcester Polytechnic Institute, United States, 2006.

[6] L. J. Mengelkoch, D. Martin, and J. Lawler, "A Review of the Principles of Pulse Oximetry and Accuracy of Pulse Oximeter Estimates During Exercise," *Journal of American Physical Therapy Association*, vol. 74, no. 1, pp. 40-49, Jan. 1994.

[7] Y. Mendelson, "Pulse oximetry: theory and applications for noninvasive monitoring." *The American Association for Clinical Chemistry*, vol. 38, no. 9, pp. 1601-1607, Sept. 1992.

[8] S. Haykin, *Adaptive Filter Theory, 4th ed*. Englewood Cliffs, NJ: Prentice Hall Inc., 2002.

[9] R. D. Miller, *Miller's Anesthesia, 11th ed*. Philadelphia, PA: Churchill Livingston Elsevier, pp. 1215-1216, 2010.

[10] A. D. Poularikas, Z. M. Ramadan. *Adaptive Filtering Primer with MATLAB*. Boca Raton, Fl: CRC Press, 2006.

[11] J. Gnitecki, Z. Moussavi, H. Pasterkamp, “Recursive least squares adaptive noise cancellation filtering for heart sound reduction in lung sounds recordings," *Eng. in Medicine and Biology Soc., 2003. Proc. of the 25th Annu. Int. Conference of the IEEE*, vol. 3, pp. 2416-2419, Sept. 2003.

[12] S. Seyedtabaii, L. Seyedtabaii, “Kalman Filter Based Adaptive Reduction of Motion Artifact from Photoplethysmographic Signal,” *Int. Journal of Electronics, Circuits & Systems*, vol. 2, no. 1, pp. 31-34, 2008.

[13] B. Lee, J. Han, H.J. Baek, J. H. Shin, K. S. Park, W. J. Yi, “Improved elimination of motion artifacts from a photoplethysmographic signal using a Kalman smoother with simultaneous accelerometry,” *Institute of Physics and Engineering in Medicine*, vol. 31, no. 12, pp. 1585-1603, Dec. 2010.

[14] *Wireless LAN/MAN Working Group*, IEEE Standard 802.11, 1997.

[15] D. Tse, P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY: Cambridge University Press, 2005.

[16] *Wireless Personal Area Network (WPAN) Working Group*, IEEE Standard 802.15.1, 2002.

[17] *Body Area Networks*, IEEE Standard 802.15.6, 2011.

[18] C. Zhong, “A Sensor Network Architecture for Mobile Users,” Ph.D. Dissertation, Dept. Of Comp. Sci., Elect. and Space Eng., Lulea Univ. Of Tech., Sweden, 2011.

[19] W. Yao, C.H. Chu, Z. Li, “The Use of RFID in Healthcare: Benefits and Barriers,” *2010 IEEE International Conference on RFID - Technology and Applications (RFID-TA)*, pp. 128,134, June 2010.

[20] Near Field Communication and Protocol (NFCIP-1), ISO/IEC 18092, 2004.

[21] K. Curran, A. Miller, C. McGarvey, “Near Field Communication,” *Int. Journal of Elect. and Comp. Eng. (IJECE)*, vol. 2, no. 3, pp. 371-382, June 2012.

[22] D. Gangwar, “Biomedical Sensor Network for Cardiovascular Fitness and Activity Monitoring,” *Point-of-Care Healthcare Technologies (PHT) IEEE*, pp. 279-282, 2013.

[23] *Personal Health Device (PHD) Working Group*, IEEE Standard ISO/IEEE 11073, 2013.

[24] *Low Rate WPAN Working Group*, IEEE Standard 802.15.4, 2003.

[25] "Specifications," ZigBee Alliance. [Online]. Available:http://www.zigbee.org/.

[26] A. Sikora, V. F. Groza, "Co-Existence of IEEE 802.15.4 with other Systems in the 2.4 GHz-ISM-Band," *IEEE Instrumentation and Measurement Technology Conference, IMTC*, 2005.

[27] "What is ANT+," ANT+. [Online]. Available:http://www.thisisant.com/.

[28] "Bluetooth Basics," Bluetooth SIG. [Online]. Available: http://www.bluetooth.com/Pages/Basics.aspx.

[29] D. Crowe, "Short Range Wireless Comes of Age," *Wireless Telecom*, vol. 25, no. 1, pp. 28-30, 2007.

[30] "Bluetooth Specification Version 4.1," Bluetooth SIG, vol. 1, pp. 17, 2013.

[31] "Ant Message Protocol and Usage," ANT, vol. 5.0, June 2011.

[32] S. Miguel-Bilbao, J. Garcia, M. D. Marcos, V. Ramos, "Short Range Technologies for Ambient Assisted Living Systems in Telemedicine: New Healthcare Environments," in *Telemedicine*, Intech, pp. 141-162, Dec. 2013.

[33] "nRF24AP2 ultra-low power transceiver with ANT ultra-low power network protocol technical backgrounder," Nordic Semiconductor, 2010.

[34] "ZigBee Wireless Sensor Applications for Health, Wellness, and Fitness," ZigBee Alliance, Mar. 2009.

[35] "ZigBee and Wireless Radio Frequency Coexistence," ZigBee Alliance, 2007.

[36] "New ZigBee PRO Feature: Green Power," ZigBee Alliance, Dec. 2012.

[37] J. Murphy, "Pulse Sensor Amplified," TAPR Open Hardware License, 2012.

Appendix A: MATLAB Code

LMS

```
%LMS Adaptive Filter
close all
%
% ***Ranges***
% No Movement = (21, 1242)
% X Direction = (1284, 2527)
% Y Direction = (2696, 3907)
% Z Direction = (4044, 5189)


start= 1284; % Start index for Y direction
stop = 2527; % End index for Y direction

accel=time_arr((start:stop),3); %3 for X axis, 4 for Y axis and 5
for z axis
hr=time_arr((start:stop),2); %2 for HR

% Normalized Unit Amplitude
hr=(hr-mean(hr));
accel=(accel-mean(accel));
hr=hr.*(3/1024);
accel=accel.*(6/1024);

step=0.008;
order=10;

step1=0.08;
order1=10;

step2=0.8;
order2=10;

step3=0.008;
order3=100;

LMS=adaptfilt.lms(order, step);
[out, error]=filter(LMS,accel,hr);

LMS1=adaptfilt.lms(order1, step1);
[out1, error1]=filter(LMS1,accel,hr);
```

```
   LMS2=adaptfilt.lms(order2, step2);
   [out2, error2]=filter(LMS2,accel,hr);

   LMS3=adaptfilt.lms(order3, step3);
   [out3, error3]=filter(LMS3,accel,hr);

   % Accelerometer X axis
   figure
   subplot(2,1,1)
   plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('X Axis','Wrist HR'); legend
('Location','NorthWest'); title('Primary and Reference Inputs for LMS
Adaptive Filter for Movements in X Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),error,'b',...
        time_arr((start:stop),1),error1,'r');
    hold on;
       plot(time_arr((start:stop),1),error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),error3, 'Color',[1 0.5 0])
       hold on; legend('Error(System Output) for step=0.008,
M=10','Error(System Output) for step=0.08, M=10','Error(System Output)
for step=0.8, M=10','Error(System Output) for step=0.008, M=100');
       legend('Location','SouthWest');
       title('LMS Adaptive Filter Error for X-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);

    % Accelerometer Y axis
    accel=time_arr((start:stop),4); %3 for x axis, 4 for y axis and 5
for z axis
    accel=(accel-mean(accel));
    accel=accel.*(6/1024);

   LMS=adaptfilt.lms(order, step);
   [out, error]=filter(LMS,accel,hr);

   LMS1=adaptfilt.lms(order1, step1);
   [out1, error1]=filter(LMS1,accel,hr);

   LMS2=adaptfilt.lms(order2, step2);
   [out2, error2]=filter(LMS2,accel,hr);

   LMS3=adaptfilt.lms(order3, step3);
   [out3, error3]=filter(LMS3,accel,hr);

    figure
    subplot(2,1,1)
    plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('Y Axis','Wrist HR'); legend
```

```
('Location','NorthWest'); title('Primary and Reference Inputs for LMS
Adaptive Filter for Movements in X Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),error,'b',...
        time_arr((start:stop),1),error1,'r');
    hold on;
       plot(time_arr((start:stop),1),error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),error3, 'Color',[1 0.5 0])
       hold on; legend('Error(System Output) for step=0.008,
M=10','Error(System Output) for step=0.08, M=10','Error(System Output)
for step=0.8, M=10','Error(System Output) for step=0.008, M=100');
       legend('Location','SouthWest');
       title('LMS Adaptive Filter Error for Y-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);


    % Accelerometer Z axis
    accel=time_arr((start:stop),5); %3 for x axis, 4 for y axis and 5
for z axis
    accel=(accel-mean(accel));
    accel=accel.*(6/1024);

   LMS=adaptfilt.lms(order, step);
   [out, error]=filter(LMS,accel,hr);

   LMS1=adaptfilt.lms(order1, step1);
   [out1, error1]=filter(LMS1,accel,hr);

   LMS2=adaptfilt.lms(order2, step2);
   [out2, error2]=filter(LMS2,accel,hr);

   LMS3=adaptfilt.lms(order3, step3);
   [out3, error3]=filter(LMS3,accel,hr);

    figure
    subplot(2,1,1)
    plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('Z Axis','Wrist HR'); legend
('Location','NorthWest'); title('Primary and Reference Inputs for LMS
Adaptive Filter for Movements in X Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),error,'b',...
        time_arr((start:stop),1),error1,'r');
    hold on;
       plot(time_arr((start:stop),1),error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),error3, 'Color',[1 0.5 0])
```

```
       hold on; legend('Error(System Output) for step=0.008,
M=10','Error(System Output) for step=0.08, M=10','Error(System Output)
for step=0.8, M=10','Error(System Output) for step=0.008, M=100');
       legend('Location','SouthWest');
       title('LMS Adaptive Filter Error for Z-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
```

RLS

```
%RLS Adaptive Filter
close all
% ***Ranges***
% No Movement = (21, 1242)
% X Direction = (1284, 2527)
% Y Direction = (2696, 3907)
% Z Direction = (4044, 5189)


start= 4044; % Start index for X direction
stop = 5189; % End index for X direction

accel=time_arr((start:stop),3); %3 for X axis, 4 for Y axis and 5
for z axis
hr=time_arr((start:stop),2); %2 for desired HR

% Normalized Unit Amplitude
hr=(hr-mean(hr));
accel=(accel-mean(accel));
hr=hr.*(3/1024);
accel=accel.*(6/1024);

order=10;
ff=0.99;
incov=10*eye(order);

order1=10;
ff1=0.49;
incov1=10*eye(order1);

order2=10;
ff2=0.09;
incov2=10*eye(order2);

order3=100;
ff3=0.99;
incov3=10*eye(order3);

RLS=adaptfilt.rls(order,ff,incov);
[rls_out, rls_error]=filter(RLS,accel,hr);

RLS1=adaptfilt.rls(order1,ff1,incov1);
[rls_out1, rls_error1]=filter(RLS1,accel,hr);
```

```
   RLS2=adaptfilt.rls(order2, ff2,incov2);
   [rls_out2, rls_error2]=filter(RLS2,accel,hr);

   RLS3=adaptfilt.rls(order3, ff3,incov3);
   [rls_out3, rls_error3]=filter(RLS3,accel,hr);

   % Accelerometer X axis
   figure
   subplot(2,1,1)
   plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('X Axis','Wrist HR'); legend
('Location','NorthWest'); title('Primary and Reference Inputs for RLS
Adaptive Filter for Movements in Z Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),rls_error,'b',...
        time_arr((start:stop),1),rls_error1,'r');
    hold on;
       plot(time_arr((start:stop),1),rls_error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),rls_error3, 'Color',[1 0.5 0])
       hold on; legend('Error(System Output) for lambda=0.99,
M=10','Error(System Output) for lambda=0.49, M=10','Error(System
Output) for lambda=0.09, M=10','Error(System Output) for lambda=0.99,
M=100');
       legend('Location','SouthWest');
       title('RLS Adaptive Filter Error for X-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
       ylim([-0.6 0.4])

    % Accelerometer Y axis
    accel=time_arr((start:stop),4); %3 for x axis, 4 for y axis and 5
for z axis
    accel=(accel-mean(accel));
    accel=accel.*(6/1024);

   RLS=adaptfilt.rls(order,ff,incov);
   [rls_out, rls_error]=filter(RLS,accel,hr);

   RLS1=adaptfilt.rls(order1,ff1,incov1);
   [rls_out1, rls_error1]=filter(RLS1,accel,hr);

   RLS2=adaptfilt.rls(order2, ff2,incov2);
   [rls_out2, rls_error2]=filter(RLS2,accel,hr);

   RLS3=adaptfilt.rls(order3, ff3,incov3);
   [rls_out3, rls_error3]=filter(RLS3,accel,hr);

    figure
    subplot(2,1,1)
```

```
    plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('Y Axis','Wrist HR'); legend
('Location','NorthWest'); title('Primary and Reference Inputs for RLS
Adaptive Filter for Movements in Z Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),rls_error,'b',...
        time_arr((start:stop),1),rls_error1,'r');
    hold on;
       plot(time_arr((start:stop),1),rls_error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),rls_error3, 'Color',[1 0.5 0])
       hold on; legend('Error(System Output) for lambda=0.99,
M=10','Error(System Output) for lambda=0.49, M=10','Error(System
Output) for lambda=0.09, M=10','Error(System Output) for lambda=0.99,
M=100');
       legend('Location','SouthWest');
       title('RLS Adaptive Filter Error for Y-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
      ylim([-0.6 0.4])

    % Accelerometer Z axis
    accel=time_arr((start:stop),5); %3 for x axis, 4 for y axis and 5
for z axis
    accel=(accel-mean(accel));
    accel=accel.*(6/1024);

   RLS=adaptfilt.rls(order,ff,incov);
   [rls_out, rls_error]=filter(RLS,accel,hr);

   RLS1=adaptfilt.rls(order1,ff1,incov1);
   [rls_out1, rls_error1]=filter(RLS1,accel,hr);

   RLS2=adaptfilt.rls(order2, ff2,incov2);
   [rls_out2, rls_error2]=filter(RLS2,accel,hr);

   RLS3=adaptfilt.rls(order3, ff3,incov3);
   [rls_out3, rls_error3]=filter(RLS3,accel,hr);

    figure
    subplot(2,1,1)
    plot(time_arr((start:stop),1),accel,'r',...
        time_arr((start:stop),1),hr,'b')
       hold on; legend('Z Axis','Wrist HR'); legend
('Location','NorthWest'); title('Primary and Reference Inputs for RLS
Adaptive Filter for Movements in Z Direction','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   subplot(2,1,2)
   plot(time_arr((start:stop),1),rls_error,'b',...
        time_arr((start:stop),1),rls_error1,'r');
```

```
    hold on;
       plot(time_arr((start:stop),1),rls_error2, 'Color',[0 0.5 0])
       hold on;
       plot(time_arr((start:stop),1),rls_error3, 'Color',[1 0.5 0])
       hold on; legend('Error(System Output) for lambda=0.99,
M=10','Error(System Output) for lambda=0.49, M=10','Error(System
Output) for lambda=0.09, M=10','Error(System Output) for lambda=0.99,
M=100');
       legend('Location','SouthWest');
       title('RLS Adaptive Filter Error for Z-Axis Accelerometer vs.
Wrist HR ','Fontsize',16)
       xlabel('milliseconds mS','Fontsize',16);ylabel
('Amplitude','Fontsize',16);
   ylim([-0.6 0.4])
```

Appendix B: Pulse Sensor Schematic

The PPG sensor that was used in this work is depicted in Figure B.1. The sensor is made up of an LED, a photodiode, a diode, an operational amplifier, capacitors, and resistors. The Pulse Sensor can operate on 3 V or 5 V.

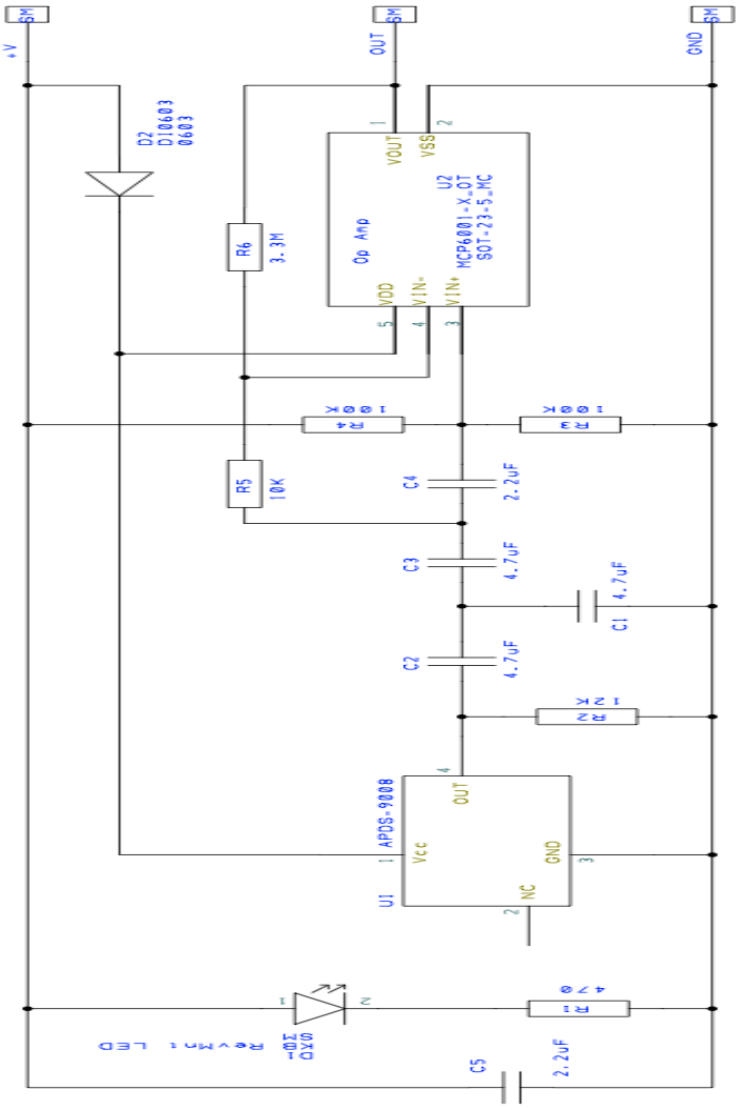

Figure B.1: Pulse Sensor Schematic [37] (Adapted from Pulse Sensor Amplified)