DESIGN AND VALIDATION OF WEARABLE WIRELESS SENSORS

A DISSERTATION IN

Electrical and Computer Engineering
and
Telecommunications and Computer Networking

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
FAHAD ABDUL MOIZ

BSEE, University of Missouri-Kansas City, 2004
MSEE, University of Missouri-Kansas City, 2005

Kansas City, Missouri
2012

DESIGN AND VALIDATION OF WEARABLE WIRELESS SENSORS

Fahad Abdul Moiz, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2012

ABSTRACT

Recent years have seen an increase in research and development efforts towards wearable and implantable health monitoring systems. Such systems are needed to provide real-time information about patients to physicians, care-givers, emergency personnel and relatives. The challenge lies in their designing as they need to satisfy a variety of criteria and constraints. These include small weight and size, low power consumption, easy to use, and should be aesthetically pleasing. Advances in semiconductor fabrication have made commercially available highly integrated systems-on-chip (SOC) which are being exploited to develop such systems. Use of these SOCs reduces cost and development time.

This dissertation presents system prototypes that can capture human body motion, measure strain on bones and perform electromyography (EMG). Design of these systems is centered on ultra-low power microcontrollers and other required circuit components. We present in detail their design, functionality and compare our results with present solutions.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled "Design and Validation of Wearable Wireless Sensors," presented by Fahad Abdul Moiz, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

W. Daniel Leon-Salas, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Ghulam Chaudhry, Ph.D.
Department of Computer Science & Electrical Engineering

Deep Medhi, Ph.D.
Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.
Department of Computer Science & Electrical Engineering

Reza Derakhshani, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude and love toward my family without whose endless support, patience and sacrifice, I would have never been able to accomplish this.

I would like to express my gratitude and appreciation toward my advisor and committee chairman Dr. Walter D. Leon-Salas for his encouragement, leadership and guidance. His tireless efforts and advice were essential toward my obtaining this personal and professional achievement.

Additionally, I would like to thank my doctoral committee members Dr. Ghulam M. Chaudhry, Dr. Deep Medhi, Dr. Yugyung Lee, and Dr. Reza Derakhshani for their advice and support through out this process.

In general, I also acknowledge the inspiration and contributions of the many teachers, educational administrators and scientific leaders, now living and many who have long since departed, each who has participated in my educational and professional journey.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Researchers today are are working towards making available wearable computing platforms for a variety of applications. These include fitness, healthcare, and entertainment. The vision is to have systems that are permanently present and active, virtually invisible, and act as intelligent personal assistants. These systems should enhance the users intelligence, expand his ability to communicate and interact with the environment and provide assistance in a variety of everyday situations. The envisioned system is called a Body Area Sensor Network, as the sensor nodes are carefully placed around the body to perform various tasks. These nodes can be employed to collect physiological data such as electrocardiogram (ECG), electromyogram (EMG) electroencephalogram (EEG), blood pressure, strain felt by the skeleton, heart rate, skin temperature, respiration frequency, sweat production, motion etc.

To achieve social acceptance, body area sensor network nodes must have many properties that make them significantly different from a conventional wireless sensor network. In terms of functionality, these sensor nodes should be able to collect near perfect data, stay connected to the central node at all times and have a sophisticated user interface that allows the system to be used while mobile.

Along with the functionality requirements, designing the hardware has its share

of challenges too. A body area sensor network needs to be unobtrusive to the degree that it does not interfere with the user's activity and does not change his appearance in any unacceptable way. For this the nodes must be extremely noninvasive, and should be smaller in size relative to a conventional wireless sensor network. Since the size should be as small as possible the nodes cannot operate from large batteries however be able to last as long as possible without the need to replace or recharge them. Most of the battery is used up by the sensors, frontend electronics, digital processing of data, and radio communication. Therefore, choosing the best electronics is very critical but the state of the art presents limitations.

System integration has provided commercially available off the shelf system-on-chips, however to choose the best among the many for a particular application is challenging. For example, decisions need to be undertaken to have better battery life over size or vice versa. The applications also dictate the employment of power management schemes. Therefore, designing a wearable or implantable body area sensor network can be categorized as an optimization problem.

Adding to the task of optimization, the design and implementation of such a system requires knowledge of a multitude of disciplines, such as: human physiology, sensor types, electronic control and processing units, wireless communication protocols and links, graphical interface for the user, software, and advanced algorithms for data extracting and decision making. Therefore, a collaborative effort between engineers, doctors, and computer scientists is inevitable.

1.2 Overview

The aim of this dissertation is to provide solutions for the mentioned body area sensor network for capturing human body motion, measure strain on bones and record muscle potentials. We do this by providing in detail the design and validation of wireless nodes for these applications. Each solution is housed in its own chapter, where we make available a survey of previously done work, theoretical background and the results achieved.

Sensor designed to capture human body motion and classify them is presented in chapter 2. In this chapter, we also propose an approach to increase battery life of a accelerometer based gesture collecting wireless sensor by using state of the art classifiers.

Successive versions of bone strain monitoring sensor are discussed in chapter 3. Our final version is a multichannel strain measuring sensor, that can also capture motion using acclerometers as well can be charged wireless. Wireless charging makes it suitable for implantable applications.

And finally, a wireless surface Electromyography (EMG) sensor is presented in chapter 4. We present the hardware design for the EMG node and propose a robust wireless network solution to employ four such sensors at once.

Conclusions are drawn out in chapter 5.

1.3 Contributions

This dissertation contributions includes the use of state of the art available to create engineering solutions for measuring physiological signals. The designed nodes are

centered around a highly-integrated microcontrollers. The design of PCB is carefully done to minimize noise. Other required circuit is chosen meticulously from the pool of available parts to reduce the overall power consumed by the final product.

A multi channel strain measuring sensor node is designed for measuring strain on bones wirelesly and can be used in implanted applications. A robust network is created to collect EMG information from four locations on the body and over comes the data rate limitations put forth by the hardware. The network uses TDMA technology and assign data transmission slots to nodes. Neural networks are employed to device a battery conserving technique for gesture capturing wireless nodes.

CHAPTER 2

A WEARABLE MOTION TRACKER

This chapter presents the development and testing of a wearable, multi-modality, motion capture platform. This platform can be used in a range of applications including virtual and augmented reality, biomechanics, sign language translation, gait analysis and graphics in movies and video games. Our platform includes inertial and ultrasonic motion sensing modalities. The combination of these modalities is expected to improve the overall accuracy of the captured motion data. An electronic board for this has been designed, fabricated and programmed. The board measures 3.2 cm x 4.8 cm and includes a low-power microcontroller, a radio unit, a three-axis accelerometer, a two axis gyroscope, an ultrasonic transmitter and an ultrasonic receiver. Results using the inertial and ultrasonic sensors to estimate position are presented.

Gestures captured using accelerometer data is classified using state of the art classification techniques, such as linear discriminant analysis, static neural network and time delay neural networks. A comparison of their accuracies in user-dependent and user-independent scenarios is also discussed. Radio communication is the most power hungry operation done by wireless sensor nodes. We show that by sending less bits of accelerometer data (more samples can be packed into one data packet) at lower than nyquist rates (reduced power by transmitting less often) still can achieve high classification accuracies. This can prove handy in applications where signal integrity is less important compared to

accurately classifying gestures being performed.

2.1 Body Motion Capture Literature Survey

Body motion capture is an essential task in many areas such as computational biomechanics, virtual and augmented reality, assisted living, sign language translation, camera tracking, and exercise and fitness among others. In computational biomechanics, limb motion and ground reaction forces are combined in an inverse dynamic method to calculate the net reaction forces and torques acting on body joints [76]. In virtual and augmented reality, body motion tracking is necessary for spatial consistency between the real and virtual objects [6]. Tracking the motion of elderly people at home may help them live safely and more independently as they can interact with their environment by means of arm or hand gestures [86]. Motion tracking is also required in automated sign language translation [68]. A sign language translation device would improve the quality of life of deaf people allowing them to communicate with people who do not understand sign language. Motion capture also allows training and exercise movements to be recorded for latter analysis to provide feedback to an athlete.

Nearly every physical principle has been explored to measure motion. Motion capture systems based on mechanical, inertial, acoustic, optical, magnetic, and electromagnetic sensing have been proposed [45, 74]. Optical-based motion tracking systems employ 1D or 2D photo-sensors conveniently fixed around a room. The subject wears markers whose locations are calculated from the acquired images by algorithms running on a dedicated computer. The markers could be passive or active. Passive markers are

retro-reflectors that have to be illuminated by infrared light synchronized to high-speed cameras. Active markers are usually composed of light-emitting diodes (LEDs) which are sequentially energized making the task of multiple-marker tracking easier. The Cartesian Optoelectronic Dynamic Anthropometer (CODA) system employs user-worn LEDs and multiple 1D CCD cameras [43]. The cameras do not use lenses but instead rely on a pseudo-random optical grating that combined with digital signal processing techniques allows the estimation of the angle to the LED. Multiple such measurements enable the calculation of the position of an LED.

Another approach used by Motion Analysis Corp. [52] and Vicon [71] employs high-speed, high-resolution digital cameras to simultaneously image several targets. The images proceeding from the different cameras are combined and interpreted on a computer. A powerful computer is required to process the large amount of digital imaging data in real time. Optical systems provide accurate position estimates but they suffer from occlusions. While these systems have good accuracies, the subject is constrained to move inside a controlled lab environment. In some applications, it is highly desirable to capture body motion outside the lab as the subject conducts his or her daily activities. Another drawback of the multi-camera approach is its high cost.

Ultrasound or acoustic sensors have been proposed for motion tracking. In these systems, ultrasound transmitters or markers are placed along the limbs. The distance between the markers and three ultrasound microphones placed on a treadmill at known positions is measured [34]. The absolute coordinates are calculated by triangulation. The system is commercially available from Noraxon Inc [57]. The system is not wearable and

the subject has to walk on a treadmill equipped with ultrasound sensors on each side. An effort to make such system wearable and mobile is reported in [15] where an e-textile garment equipped with ultrasound markers is proposed. Employing ultrasound to calculate the positions of body parts faces the problems of shadowing or blocking of the ultrasound signal by the body and clothing, directivity and size of the transmitters, multipath, and the dependence of the speed of sound on temperature and wind. Ultrasound has also been proposed to measure the gait velocigram or instantaneous horizontal velocity as well as medial lateral and vertical displacements using acoustic Doppler techniques [84]. From these measurements, velocities and accelerations can be computed [73]. Although simple, this technique does not directly compute the position of the limbs and requires the subject to move in front of an ultrasound sensor.

Inertial sensing, which involves the use of accelerometers and gyroscopes [5, 23, 59, 70, 85] has been widely applied to body motion capture. Due to the advancements in micro-electrical mechanical systems (MEMS), wearable inertial motion capture systems are possible. An accelerometer-based motion capture system called Motion Capture with Accelerometers (MOCA) is reported in [17]. The MOCA system is composed of accelerometers, an acquisition board, and a wearable computer for sensor data processing. Its main aim is to provide a low-cost motion capturing system. Another human motion capture system based on accelerometers and gyroscopes is reported in [3]. The system is used to detect activity in clinical settings. The system can correctly identify sitting, standing, walking, and lying positions. The advantages of inertial systems are small size sensors and their immunity to electromagnetic interference. However, their weakness is

drift. To estimate position, the output of an accelerometer has to be integrated twice. If an accelerometer has a bias error of just 1 mili-g, after only 30 seconds of integration, the position estimates would have diverged by 4.5 meters [74]. Techniques like periodic resetting have been proposed to alleviate this problem. Some inertial-based commercial motion tracking systems are available [79], but they require the user to wear a special suit with bulky sensors attached to it.

Magnetic-based systems have been also proposed. They detect position and orientation using a magnetic field that could be the Earth's magnetic field or a field generated by a coil. This approach usually requires three orthogonally oriented magnetic sensors and is affected by ferromagnetic and conductive materials in the environment.

To address the shortcomings of the different sensing methods described above, we propose to employ a multi-modality sensing approach. In particular, we combine inertial and acoustic sensing to obtain a wearable motion-sensing platform with increased accuracy and reliability. The combination of inertial and acoustic sensing has been used before but in the context of indoor tracking [21]. In indoor tracking, ultrasound emitters are placed on the ceiling constraining the user to move inside a covered area.

2.2 System Overview

Figure 1 shows a picture of the envisioned wearable motion capture system. It is composed of small sensor nodes that contain inertial and ultrasonic sensors, a low-power microcontroller and a radio unit. The sensor nodes can be placed on joints, limbs or any other part of the body whose motion needs to be tracked. A set of these sensor nodes

are worn on a belt around the waist. These are the reference nodes and form a reference frame upon which the position of the other nodes can be estimated using time-of-flight measurements of ultrasound pulses and multilateration algorithms. The ultrasonic sensors could also be employed to track the position of the user as he or she moves across a room, office space or nursing home [21, 27]. Also worn on the belt is a processing unit that communicates with the sensor nodes and processes the motion data.

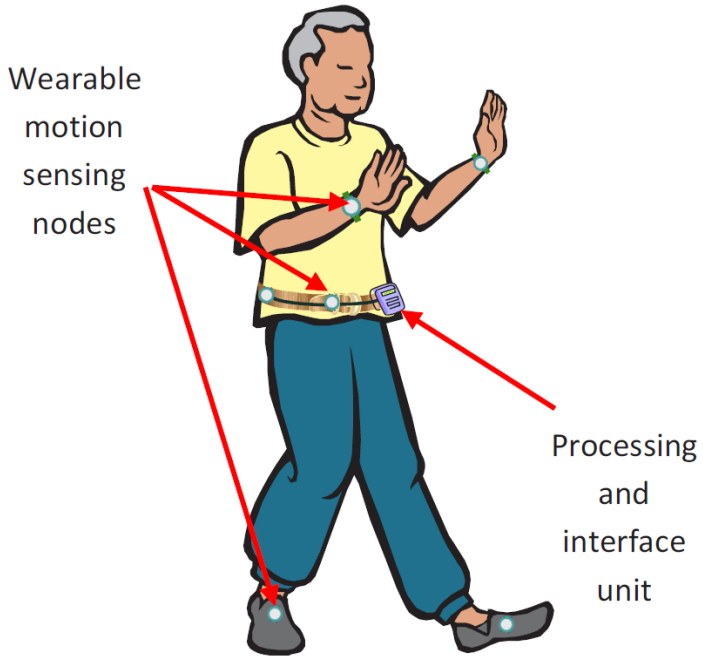


Figure 1: Wearable motion capture network.

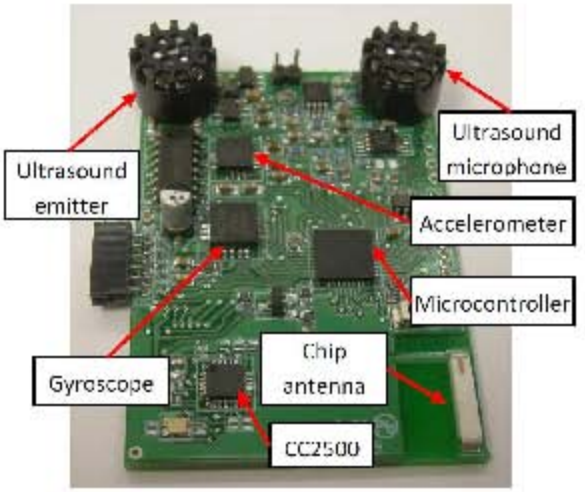
Other wearable multi-modal motion sensing platforms have been reported in the literature but require placing reference sensors on the chest or on the head [72]. This placement, although convenient in research settings, reduces the long-term wearability of the system. Therefore, we limit the placement of the sensors to locations where the user

would normally wear an accessory, i.e. wrist, waist, ring finger, necklace, etc. The overall goal for this system is to not only acquire motion data but also to classify and interpret the motion trajectories in a way the user can interact with the environment through motion.

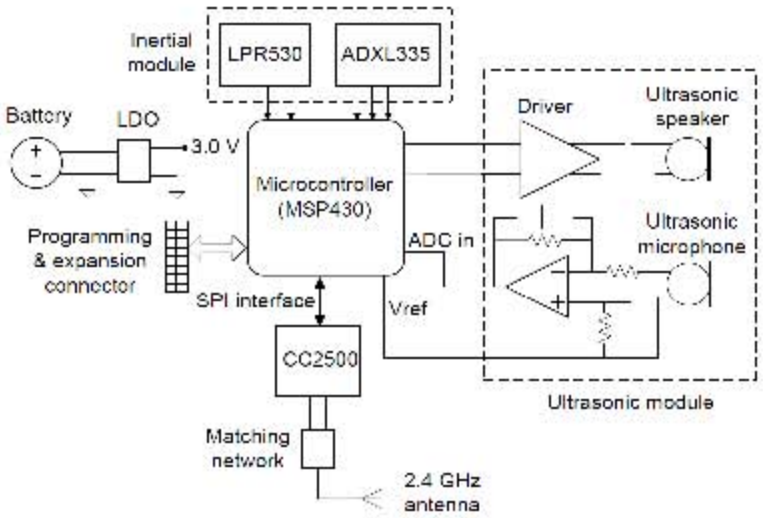
2.3 Hardware Design

A circuit board for the motion capture sensor nodes has been designed, built and tested. The board includes a three-axis accelerometer (Analog Devices ADXL335), a two-axis gyroscope (ST Microelectronics LPR530), a lowpower micro-controller (Texas Instruments MSP4302274), a 2.4 GHz radio transceiver (Chipcon CC2500), a chip antenna, an ultrasonic emitter, an ultrasonic microphone, an amplifier, and a LDO voltage regulator. It also contains a programming and expansion header. The board dimensions are 3.3 cm x 4.8 cm. Figure 2 shows the picture of the motion sensor node and its schematic diagram.

The microcontroller generates a 40 kHz square waveform that is applied to the ultrasonic emitter through a current driver. The current driver is made up of logic inverters connected in parallel. Short ultrasound bursts at intervals of 1 Hz to 20 Hz can be generated by the micro-controller using its internal timers. The output of the ultrasound microphone is amplified, filtered and fed to one of the channels of the micro-controller's internal ADC. The internal ADC is also employed to sample and digitize the three analog outputs of the accelerometer (X, Y, Z axes) and the two analog outputs of the gyroscope (pitch and roll). These samples are transmitted wirelessly to the reference nodes through the radio transceiver.



(a)



(b)

Figure 2: Motion seonsor node. (a) PCB . (b) Schematic.

The microcontroller communicates with the CC2500 radio transceiver through its SPI interface. Shortly after power up the micro-controller writes the registers of the CC2500 to select the channel, modulation, data rate, packet length and power transmission levels. Then the micro-controller programs its timers to generate interrupts at 80 kHz (to generate the ultrasound pulses) and 300 Hz (to sample the analog outputs of the accelerometer and gyroscope). The sampling rate can be easily varied by changing the period of the timer. Once the micro-controller has collected 10 sample points of the X, Y, Z and the pitch and roll values it creates a packet that is transferred to the CC2500 for its transmission. The packet has an address and a header to help synchronization with the receiver. The receiver is another motion sensing board that is programmed to receive radio packets and send them to a PC through the micro-controller's UART port.

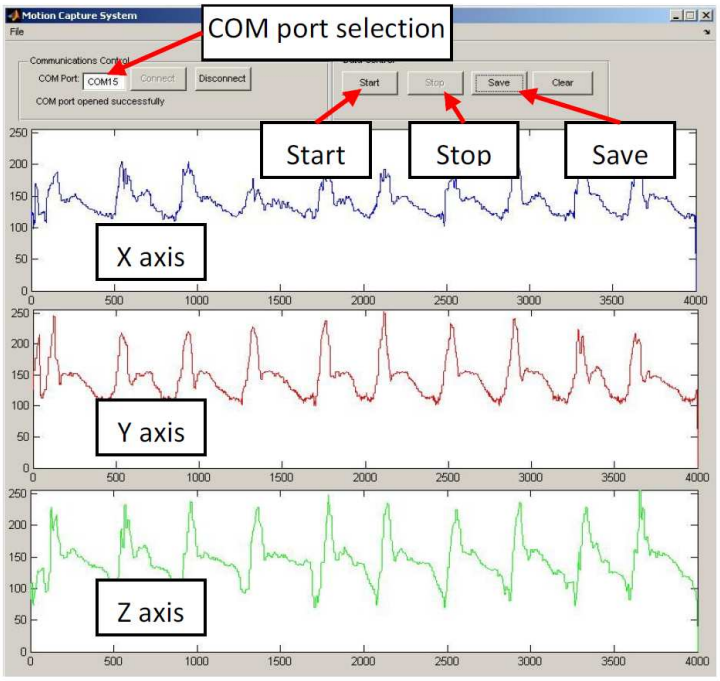


Figure 3: Matlab© graphical user interface for motion data collection.

On the PC side a Matlab© -based graphical user interface (GUI) was created to receive, plot and save the incoming motion data. Figure 3 shows a screen shot of the GUI. Start, Stop and Save controls were included to make the data collection simple and user friendly. The GUI plots in real time the X, Y and Z axis of the accelerometer to provide visual feedback. The pitch and roll are not displayed but are collected in memory and saved to disk when indicated by the user.

2.4 Position Estimation

This section describes the position estimation algorithms employed based on i) inertial sensors and ii) ultrasound ranging.

2.4.1 Inertial Based Positioning

In this technique, position is estimated by a double integration of acceleration. Consider first the case in which the board is perfectly horizontal. In this case, only the Z-axis of the accelerometer is affected by gravity and it will be manifested as a DC offset. Before integration, the DC offsets in the three axes need to be removed. This can be done by subtracting their corresponding mean values. Double integration of the acceleration data can be performed in the time domain or the frequency domains. In the time domain, the trapezoidal rule can be used or a simple sum of the last 50 samples is also an option. The double integration in the frequency domain is performed by calculating the FFT of the acceleration samples, multiplying the resulting FFT coefficients by $(j\omega)^2 = -\omega^2$ and then applying the inverse FFT.

When the board is not perfectly horizontal, the X, Y, and Z-axes of the accelerometer will be affected by gravity. Thus, a gravity compensation scheme is needed to remove the effects of gravity before integration. Figure 4 shows the gravity components for a pitch angle θ and a roll angle ϕ.

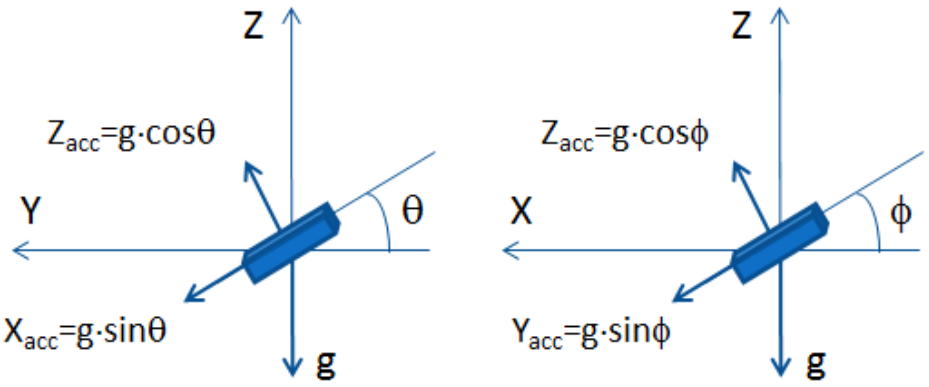


Figure 4: Gravity components along the accelerometer's X, Y, and Z axes for a pitch angle θ and a roll angle ϕ.

From the figure, the acceleration correction factors for the accelerometer's X, Y, and Z axes are: gsinθ, gsinϕ, and gcosθcosϕ. The outputs of the gyroscope are integrated once to obtain the pitch and roll angles.

2.4.2 Acoustic Based Positioning

The acoustic positioning is based on a time difference of arrival (TDOA) multilateration algorithm. The ultrasonic emitters periodically transmit pulses that are received by the reference sensors. Upon reception, the reference sensors record the time of arrival of each pulse and use this time information to compute the position of the moving sensors. Figure 5 shows a representation of the geometry of the problem.

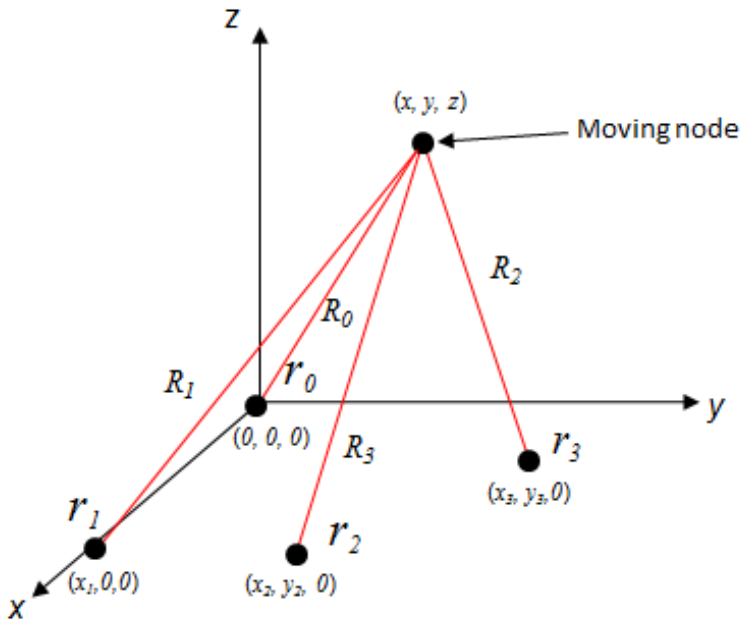


Figure 5: Geometry of the ultrasound-based positioning problem.

In the figure, r_i is the position of the i^{th} reference node. To simplify the equations, we place node 0 at the origin and aligned node 1 with the x-axis. Nodes 2 and 3 lie somewhere on the xy-plane. Let ρ_i be the distance between the moving node and the i^{th} reference node. Thus, we can write:

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = v(t_i - t)$$

where, *v* is the speed of sound, t_i is the arrival time of the pulse at the i^{th} reference node, *t* is the time at which the pulse was transmitted and *i* = 0,1,2,3 is the reference node index. We can write the following set of pseudo-range equations:

$$\rho_1 - \rho_0 = v(t_1 - t_0)$$

$$\rho_2 - \rho_0 = v(t_2 - t_0)$$

$$\rho_3 - \rho_0 = v(t_3 - t_0)$$

Thus, we have three equations and three unknowns (*x, y, z*). These equations are non-linear and their exact solution requires complex computations [28, 29].

Foy in [22] have proposed the use of Taylor series to linearize these equations and then use an iterative method to solve them. The iterative method begins with an initial guess and improve the estimate at each iteration by determining the local linear least-square solution. Although it can provide an accurate result, is robust and can make use of redundant measurements, it requires a good initial guess and can be computationally intensive. Chan [9] and Fang [16] provide non-iterative methods to reach a closed form exact solution. However, they both need prior information to solve an ambiguity in their calculations. We implemented Fangs method as its computational load is comparatively lower than Chan's. We start by rewriting the pseudo-range equations as follows:

$$\rho_{01} = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x_1 - x)^2 + y^2 + z^2} \tag{2.1}$$

$$\rho_{02} = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + z^2} \tag{2.2}$$

$$\rho_{03} = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x_3 - x)^2 + (y_3 - y)^2 + z^2} \tag{2.3}$$

where $\rho_{01} = \rho_1 - \rho_0$, $\rho_{02} = \rho_2 - \rho_0$ and $\rho_{03} = \rho_3 - \rho_0$.

Consider equations 2.1 and 2.2. After transposing the second terms to the left hand side, squaring, simplifying and equating them, we can solve for *y* in terms of *x*:

$$y = gx + h \tag{2.4}$$

and z in terms of x:

$$z = \pm\sqrt{dx^2 + ex + f} \tag{2.5}$$

where

$$g = \frac{\rho_{02}\left(\frac{x_1}{\rho_{01}}\right) - x_2}{y_2}$$

$$h = \frac{(x_2^2 + y_2^2) - \rho_{02}^2 + \rho_{02}\rho_{01}\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right)}{2y_2}$$

$$d = -1 + \left(\frac{x_1}{\rho_{01}}\right)^2 - g^2$$

$$e = x_1\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right) - 2gh$$

$$f = \frac{\rho_{01}^2}{4}\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right)^2 - h^2$$

Applying the same procedure to equations 2.1 and 2.3, we obtain:

$$y_1 = g_1 x + h_1 \tag{2.6}$$

and z in terms of x:

$$z = \pm\sqrt{d_1 x^2 + e_1 x + f_1} \tag{2.7}$$

where

$$g_1 = \frac{\rho_{03}\left(\frac{x_1}{\rho_{01}}\right) - x_3}{y_3}$$

$$h_1 = \frac{(x_3^2 + y_3^2) - \rho_{03}^2 + \rho_{03}\rho_{01}\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right)}{2y_3}$$

$$d_1 = -1 + \left(\frac{x_1}{\rho_{01}}\right)^2 - g_1^2$$

$$e_1 = x_1\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right) - 2g_1h_1$$

$$f_1 = \frac{\rho_{01}^2}{4}\left(1 - \left(\frac{x_1}{\rho_{01}}\right)^2\right)^2 - h_1^2$$

The solution presented above gives two position vectors (same *x*, different *y* and *z*) for the moving node. Equating them, squaring and simplifying yields the following quadratic equation:

$$px^2 + qx + r = 0$$

where

$$p = d - (g_1 - g)^2$$

$$q = e - 2(g_1 - g)(h_1 - h)$$

$$r = f - (h_1 - h)^2$$

Solve for *x* using the quadratic formula: $x = \frac{-q \pm \sqrt{q^2 - 4pr}}{2p}$, substitute it in 2.7 or 2.5 to solve for *z* and in 2.6 or 2.4 to solve for *y*.

2.5 Inertial based Position Estimation Results

The results of the two integration techniques, described in section 2.4.1, for the X-axis of the accelerometer are shown in figure 6 while figure 7(a) shows the 3D trajectories reconstructed from integrating the X, Y, and Z-axes. This data was captured while moving the board in a circle. Figure 7(b) shows captured circular motion after gravity compensation.

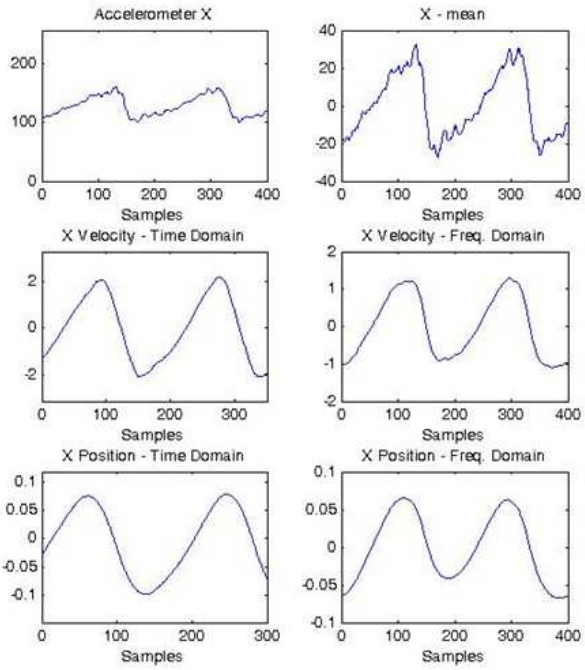
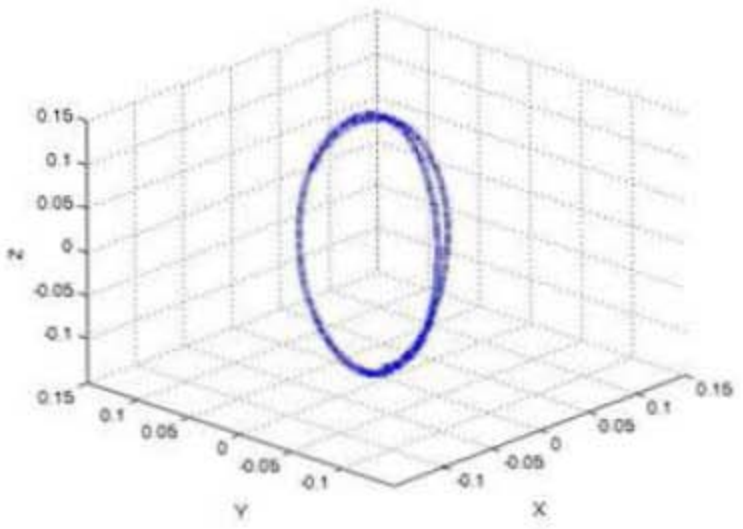


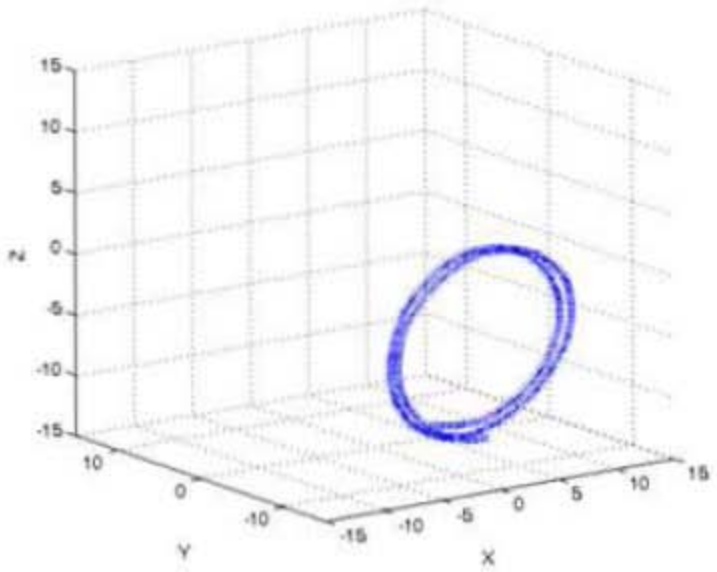
Figure 6: Integration of acceleration data.

2.6 Acoustic based Positioning Results

To acquire the time delay of arrival, our first setup used four designed board as the reference nodes and one designed board as the moving marker. The second setup consisted of CPLDs with ultrasound-microphones and other needed electronics as the reference nodes. Details of both setups and their results are discussed below.



(a)



(b)

Figure 7: (a) Estimated trajectory for a circle motion. (b) 3D Trajectory after gravity compensation for a circular motion.

2.6.1 Designed board based reference nodes

As described in section 2.3, The micro-controller of the moving marker generates a 40 kHz square waveform that is applied to the ultrasonic emitter through a current driver. The current driver is made up of logic inverters connected in parallel. Short ultrasound bursts at intervals of 20 Hz is generated by the micro-controller using its internal timers. At the reference nodes, these ultrasound bursts are received by their ultrasound microphone. The output of the microphone is amplified, filtered and fed to one of the channels of the micro-controller's internal ADC. Figure 8 shows the waveforms received at the four reference nodes. It was observed that between two ultrasound bursts received, a total of 578 ADC samples are produced. That is, we have 578 samples in 50 ms(20 Hz). Therefore, it takes 50/578 = 86.5 μs to generate one sample. Knowing this and looking at the figure we can calculate time difference of arrivals, $t_3 - t_0$, $t_2 - t_0$, and $t_1 - t_0$, between the reference nodes.

After calculating the time differences, we implemented Fang's algorithm (section 2.4.2) and found it to be very robust. The estimated position was within 1 inch of the actual position. This error could be attributed to the limited sampling rate (200 kS/s) of the microcontroller employed which precluded obtaining a better resolution in the measurement of the time differences.

2.6.2 CPLD based reference nodes

To over come the errors in positions calculations, the designed boards were replaced with CPLD based setup as the reference nodes. Figure 9 shows the schematic and

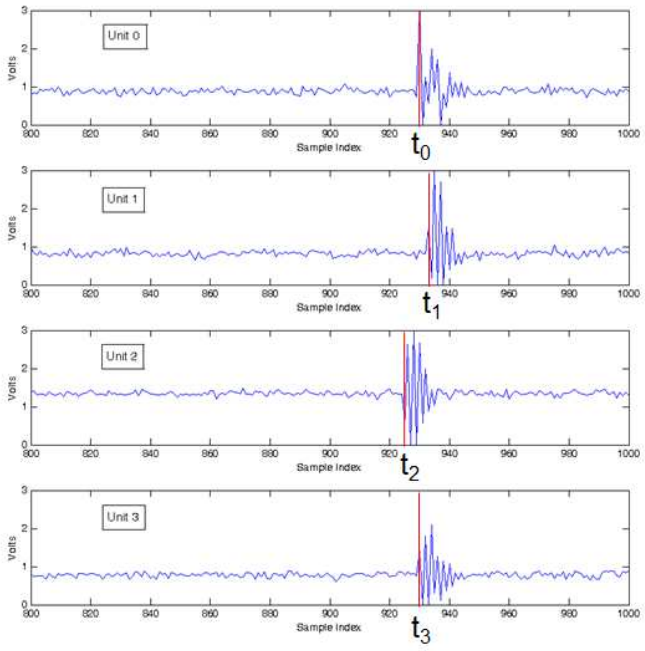


Figure 8: Waveforms at the output of the ultrasonic sensor amplifier of each reference node.

figure 10 shows the setup.

Each CPLD was programmed as a 16-bit counter to provide higher resolution compared to the micro-controller. The counting stopped when an ultrasound burst was received and the count value was stored. Each CPLD flagged the central micro-controller that it has received a burst and is ready to transfer its count value. When the controller received flags from all four CPLDs, it provides clocks to serially receive the four count values. The CPLDs are arranged in a daisy chain setup, therefore a total of 64 clock pulses are needed to receive all four count values. After receiving the count values the micro-controller sends out a reset to the counters, so that they can start listening to the next burst.

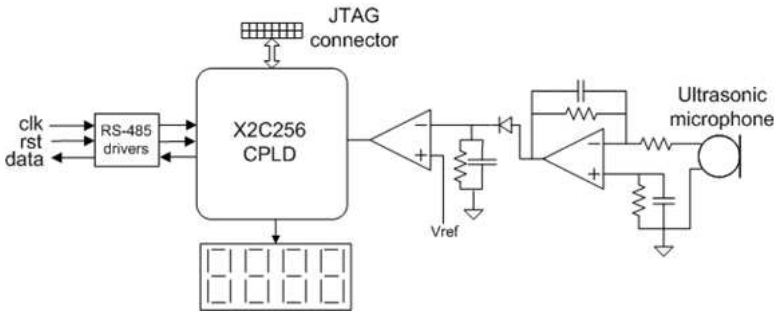


Figure 9: Schematic of the CPLD based reference node.

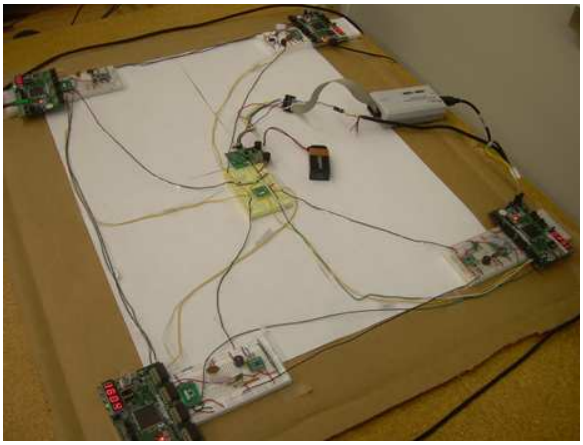


Figure 10: Setup of the CPLD based reference nodes.

In the mean time the micro-controller transfers the count values to Matlab©, running on a PC, to calculate the position of the moving unit. Using this automated setup we were able to move the marker unit over the setup and continously calculate its position. Table 1 shows the results from one such readings.

Position calculated by CPLD based reference nodes faired better than the micro-controller based setup as it had higher resolution and did not suffer from internal delays caused by sequential processing architecture.

The main error source are the spatially and temporally changing room temperature

Table 1: Ultrasound positioning using CPLD based reference nodes.

Counter 1	Counter 2	Counter 3	Counter 4	X (in)	Y (in)	Z (in)
4C59	583D	55E1	5958	4.87	3.88	18.58
8868	9378	9055	92C0	3.41	4.18	16.66
C743	D4C0	CE7E	D202	3.22	5.62	20.96
F43C	FD38	F80F	FA7E	2.15	4.01	12.12
8109	8D1F	8C29	9021	6.27	3.15	19.21
F2BC	FD74	FC91	002F	5.42	3.03	16.56

and air movement because they determine the local speed of sound. Unfortunately, these error sources which are caused by air turbulences and convection currents in the room cannot easily be eliminated.

Medical imaging and blood flow measurements use ultrasound to produce accurate assessment of the direction of blood flow and the velocity of blood and cardiac tissue at any arbitrary point using the Doppler effect, but with limitations. However, doppler effect depends on the speed of the wave in a medium, given by the equation:

$$f = (c + v_r)/(c + v_s)f_0$$

where c is the velocity of the waves in the medium. Experimenting with hot air at different wind speed we saw varying results for position of the marker node when stationary.

2.7 Gesture Recognition Literature Survey

Gesture recognition (GR) is an increasingly important tool in various applications, from hand held devices and video game controllers to biomedical equipment [51]. Gestures can generally be categorized as culture agnostic or culture specific, depending on the application. Gesture recognition uses computational techniques to accurately identify and

interpret human bodily movements allowing for interaction between man and machine at a qualitatively different level compared to conventional methods such as keyboard and mouse. Gestures are commonly thought as originating from the movement of limbs, especially the hands, leading to the performance of specific discernable actions [54]. Facial gestures are also significant sources of nonverbal information.

Hand gestures have been grouped to fall under several categories [78], namely controlling gestures, conversational gestures, communicative gestures, and manipulative gestures. The interpretation of digitized data from captured motions and the accuracy of the interpretation remains a challenge [58]. Hand gestures are used in many modern applications such as video game controllers, automated sign language systems, urban traffic systems, navigational systems, and medical rehabilitation [39, 48, 78], among other human-computer interaction applications.

Another application for accelerometer-based human motion capture and classification is in the monitoring of elderly at home for detection of falls or other abnormal ambulation patterns, and without a need for more complicated video-based systems [86]. Motion capture also finds applications in training and exercise science as it will allow athletes and trainers to study motion in detail and observe motion features that would otherwise be undetectable to the naked eye [8].

Machine learning algorithms and models have been widely used for the human motion and gesture recognition. Classification algorithms for gesture recognition include Hidden Markov Model (HMM), Naïve Bayes classifier, Artificial Neural Networks (ANN) and decision tree [48]. Many efforts in this field have been made mainly for

motion/gesture detection in a continuous video stream. Recent research focuses on gesture recognition employing accelerometer data obtained from devices, such as Wiimote, which has built-in accelerometer. [66] presented the WiiGee library which enables gesture training and recognition through Wiimote. They achieved an accuracy of 90% using HMM model and Bayes classifier for gesture models, such as square, circle, roll, etc. [38] used Dynamic Time Warping(DTW) equipped with HMM model for the similar task and improved the accuracy up to 94% for user dependent and about 75% for user independent models. [42] achieved 97.2% recognition rate with the service of HMM for VCR control gestures including play, stop, next, previous, increase, decrease, fast forward, and fast rewind. [63] employed the WiiGLE library and Weka, which are reflecting on various cultural specific interactions, and reported a classification results near 100% for the similar tasks using HMM, Nearest Neighbor, Naïve Bayes for both user independent and dependent models.

Several researches have demonstrated the strength of ANN in recognizing accelerometer based gestures as well. [81] compare recognition rate of neural networks algorithm to *k*-nearest neighbour algorithm and show user-dependent classification of upto 95.24% by neural networks as apposed to 87.18% by *k*NN. [56] achieve 92% recognition rate in controlling a robot arm by performing gestures while wearing a 3 axis acclerometer. In our previous work [49], we have shown near perfect classification using time delay neural networks in recognizing simple and easy to use gestures like circle, triangle etc.

There are also examples of hardware implementation of HMMs and ANNs. [46] present a HMM-based system that can be implemented on an 8-bit micro-controller.

FPGA based speech recognition using HMM is demonstrated by [44]. [47] provide a comprehensive survey of ANNs being implemented in hardware (analog, digital, mixed and FPGA). Both, HMM and ANN have shown great accuracies in classification, however, the regular computing [33] and parallel processing [47] architecture of ANN prove advantageous for hardware implementation. Therefore, in this work, we choose to analyze the performance of ANNs to provide a technique in reducing power consumption of resource-limited wireless sensors.

2.8 Accelerometer based Gesture Recognition

Although much has been achieved in the area of gesture recognition research, a comparative evaluation of static, dynamic, linear and nonlinear techniques has the potential of providing better understandings of the efficacy of one technique over another. Such studies can also provide insights into the drawbacks and advantages of specific gesture recognition techniques, without associating the technique to a particular class of application. Prior to using our designed board we used Texas Instruments Chronos sports watch and its accelerometer data to capture and classify gestures. After achieving near perfect classifications, we extend the study using acceleration data from our board, to provide a solution for reducing power consumption of wireless sensors. Here we present the classifiers used, gestures performed and the static features extracted from the collected acceleration data .

2.8.1 Fisher Linear Discriminant Classifier

Linear Discriminant Analysis or LDA [14] is a linear supervised classification and dimensionality reduction method. LDA casts multidimensional features into a single dimension using a linear mixture so that a classification criterion of interest, Fisher's in this study [7], is optimized. More specifically, assume that the desired linear projection is given by $\vec{W}$, casting input $\vec{X_n}$ from classes C_1 and C_2 onto $y_n = \vec{W} \cdot \vec{X_n}$. Fisher criterion for class separability, J, is then defined as the interclass variance to intraclass variance ratio [7]:

$$J = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

where m_1 and m_2 are projections of the means, $\vec{M_1}$ and $\vec{M_2}$, of C_1 and C_2. They are given by $m_1 = \vec{W} \cdot \vec{M_1}$ and $m_2 = \vec{W} \cdot \vec{M_2}$. It can be shown that the direction of desired $\vec{W}$ that maximizes J is given by

$$S_{\vec{w}}^{-1} \cdot (\vec{M_1} - \vec{M_2})$$

where $S_{\vec{w}}$ is the total intraclass covariance matrix given by

$$S_{\vec{w}} = \sum_{i=C_1} (\vec{X_i} - \vec{M_1})(\vec{X_i} - \vec{M_1}) + \sum_{j=C_2} (\vec{X_j} - \vec{M_2})(\vec{X_j} - \vec{M_2})$$

2.8.2 Neural Networks

We used static and dynamic neural networks for nonlinear signal classification. The static artificial neural networks (NN) were in the form of feed-forward one-hidden

layer perceptrons with hyperbolic-tangent nonlinearities, and have been shown to be universal function approximators. However, similar to LDA, NN need static features characterizing incoming signals to be derived during the pre-processing stage (Section 2.8.5). Dynamic or time delay neural networks (TDNN) are feed forward focused neural networks with input tapped delay lines and can be used as dynamic function approximators and signal classifiers. In contrast to NNs, TDNNs can directly operate on the signals of interest and thus forego the feature extraction, albeit at the expense of added complexity.

TDNNs use tapped delay lines (TDL) at its input as simple memory structures to keep the last n_d samples of the input signal and present them to their feed-forward multilayer perceptron section to generate the desired class label [25]. In that sense, TDNNs are generative classifiers [14] and thus their performance can be measured with receiver operating curves (ROC) as explained later on (Section 2.8.3). Our choice of TDNN for nonlinear classification is based on that fact that given adequate TDL depth, it can approximate any nonlinear dynamic shift-invariant myopic mapping between its input and output domains [65], and thus can provide better performance when target classes are not linearly separable. Moreover, we are interested in classifiers that can perform well with low resolution and noisy data.

It has been shown that neural networks are especially robust under the aforementioned detrimental circumstances, where the network can learn in-variance to input noise and distortions while training with low quality data [61]. As any other learning machine, the learning capacity and performance of NNs and TDNNs is not only dependent on their training data set and learning algorithm, but also their architecture and size [7, 11, 61].

The defining structural parameter of NN is the number of nodes in the hidden layer. The defining structural parameter of TDNN is the number of nodes in the hidden layer, and the length of input tapped delay lines.

To keep the model as simple as possible, we used one hidden layer with hyperbolic tangent nonlinear nodes. Four hyperbolic tangent functions were used at the output layer as indicator functions to deliver the four binary decisions on detected class. NNs and TDNNs were then trained using Bayesian Regularization [7] in conjunction with MSE cost function. Given the effective reduction of extra degrees of freedom under Bayesian Regularization, the generalization capability of the trained network is usually preserved without a need for early-stopping validation. This in turn increases the quality of training, as one does not have to set aside a portion of the training data for validation. Reported results are from an average of 5 training runs, as randomly initialized neural networks converge to a different local solution during each gradient descent [61]. Thus, the expected merit of each configuration was estimated by taking the results of multiple runs into account during each ROC analysis.

2.8.3 Receiver Operating Curves as Classifier Quality Measurer

Receiver Operating Characteristic (ROC) curves are powerful tools in characterizing the overall quality of classifiers with continuous outputs, such as neural networks and LDAs [18]. Thus, along with their area under the curve and equal error rates, we chose them as classification metrics for this study. ROC curves are essentially the plot of Genuine Accept Ratio or GAR, also known as sensitivity; versus False Accept Ratio or

FAR (1-specificity); across different output decision thresholds. The area under the curve of ROC plots, or ROC AUC (the higher the better), along with the operating point where sensitivity equals specificity (equal error rate or EER, the lower the better), are two other important scalar metrics describing a system's overall classification performance. ROC analysis is especially important when dealing with unbalanced or multiclass datasets with hypothesized unknown distributions, such as the problem at hand.

Though ROCs are traditionally defined for two-class problems, they may also be utilized for multi-class scenarios such as our four-class GR. Since M-class confusion matrices include M^2 cells [18, 26], they result in complicated multi-dimensional ROCs (polytypes) for M $>$ 2. However, it can be shown that if the classes are equiprobable, an M-class ROC AUC from a single M x M contingency table will provide the same results as a more-involved pair-wise comparisons of M contingency tables [18]. Such multi-class ROC can be garnered from dichotomizing M-dimensional decision profile (DP) matrix [1, 35–37] across different thresholds. DP(x) is calculated from a bank of M classifiers, 4 here, with shared input x. Each elements $dp_ij(x)$ represents the (continuous-valued) output of classifier i when x belongs to class j, also known as the support of classifier i to case j. The counts of diagonal vs. off-diagonal elements of DP(x) across different thresholds provides the GARs and FARs needed for plotting a multi-class ROC curve similar to the two-class case, which is the method used for generating ROCs for this study.

2.8.4 Gesture Set

We employed a subset of a gesture vocabulary that was identified by a Nokia research study, as it is simple to use and preferred by users when interacting with home appliances such as TV, VCR and lights [31]. The gestures that were used in our study are shown in figure 11. The dot denotes the starting point and the arrow the direction of the hand movement in the plane of motion.

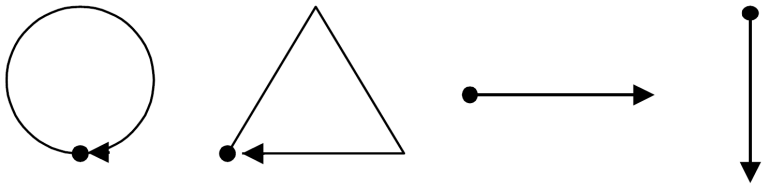


Figure 11: Set of gestures employed in this work.

2.8.5 Feature Extraction

In this study, using three axes of acceleration signals, we derived the following static features for LDA and NN:

- mean of X-axis signals
- mean of Z-axis signals
- correlation between X and Z-axis signals
- standard deviation of X-axis signals
- standard deviation of Z-axis signals

These features were chosen given their reported success in the literature [32, 62]. With the Y-axis being normal to the plane containing the hand motion paths, we did not include its signals because the gestures were mainly in the X-Z plane. As stated earlier, these features are not needed for TDNNs as they operate directly on the accelerometer signals

2.9 Gesture Recognition Results

EZ430-Chronos Sports Watch from Texas Instruments was employed to capture hand motion data. The EZ430-Chronos is based on the CC430F6137 chip, which integrates a MSP430 microcontroller core with CC1101 868/915 MHz wireless transceiver. The watch also includes a three-axis accelerometer, CMA3000-D01 from VTI Technologies. The watch comes pre-programmed to collect acceleration measurements in all three axes and transmits these measurements as time signals to a USB dongle or Access Point (AP). The main component of the AP is the C1111, which is a chip that integrates the RF transceiver CC1101 with an industry-standard enhanced 8051 microcontroller core. Figure 12 shows the setup for collecting accelerometer data using the sports watch.

A MATLAB© script was written to communicate with the AP. Figure 13 shows the flowchart of the script's algorithm. First the script opens the AP's COM port and sets the required parameters (Baud Rate=115200, Data Bits=8 and Stop Bits=1). After opening the COM port, the program sends the start AP command and pauses for the user to switch the watch to accelerometer data sending mode. Once the user is ready, he/she hits any key on the keyboard for the data acquisition to begin. The script now requests the

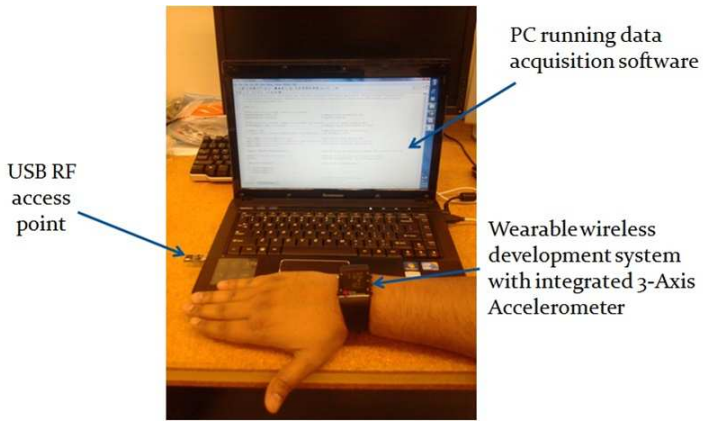


Figure 12: Setup for collecting accelerometer data using the sports watch.

acceleration measurements received by the AP and stores it. Once the desired number of data samples is stored, AP is stopped and the COM port is closed.

We collected three runs of each of these gestures for each of six users. For Run1, the first gesture was performed by all the six users. Then they performed the second gesture and so on, until everyone had executed all four gestures. Second and third runs were carried out in the same manner. While performing a gesture, the user was instructed to repeat that gesture for 10 seconds at a speed of about one gesture per second. During the mentioned 10 seconds, we collected a total of 300 samples of acceleration data at a sampling rate of 30 Hz. While performing the gesture, the users wore the sports watch on the wrist of their dominant hand. The data was captured in a quiet room while the users were comfortably standing upright.

The k-fold cross-validation technique was used and the results from all folds were utilized to produce the training and validation ROC curves. We assessed how the classifiers performed for user-dependent and user-independent cases.

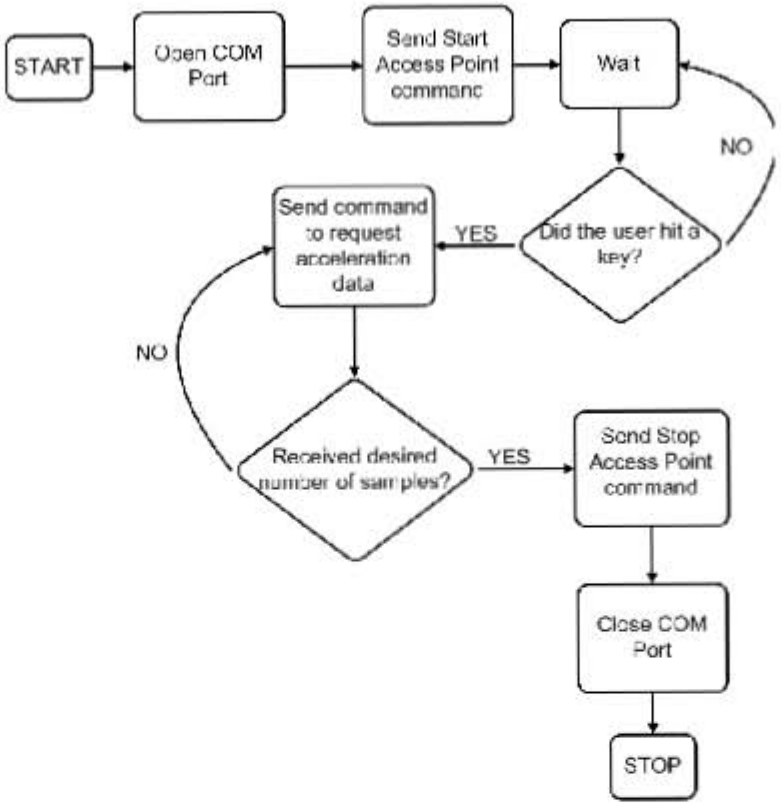


Figure 13: Flowchart for collecting accelerometer data using sports watch.

For the user-dependent case, given the three rounds of collected data for each user, we performed a 3-fold cross-validation using data from one user at a time for training (2/3 of data) and validation partitions (the remaining 1/3). Next, all the training and validation results were used to produce their corresponding ROCs.

For user-independent case, a 6-fold cross validation was performed, where during each fold all the three datasets from one user was withheld for validation and the rest of the datasets from the remaining 5 users was used for training. Since the previously unseen validation data during each fold is from a user not included in the training partition, the results are deemed as user-independent.

A total of four LDAs were trained in parallel. Each one was trained to classify one gesture from the rest, i.e., circular from non-circular, triangular from non-triangular, and so forth. Table 2 summarizes results for the user-dependent case, where LDAs were trained and tested using data from one individual at a time. Table 3 shows result for user-independent case, where LDAs were trained using data from five users and tested on the sixth. All results reflect the k-fold cross validation in addition to Monte Carlo averages.

Table 2: User-Dependent Fisher LDA classifier results.

User	Training		Validation	
	EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
User 1	8.3333	97.917	38.8890	62.269
User 2	0	100	19.444	92.245
User 3	0	100	25	84.954
User 4	0.69444	99.71	9.7222	95.949
User 5	6.25	99.132	22.222	89.352
User 6	0	100	15.278	92.245

Table 3: User-Independent Fisher LDA classifier results.

Training		Validation	
EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
13.38	93.72	18.056	89.905

To keep the NN model as simple as possible, we used one hidden layer with hyperbolic tangent nonlinear neurons. Four hyperbolic tangent functions were used at the output layer as indicator functions to deliver the four binary decisions on detected class. To avoid over-parameterization, training was performed using Bayesian regularization [40]. Given the effective reduction of extra degrees of freedom under this regularization, the generalization capability of the trained network is usually preserved without a need for early-stopping validation. This in turn increases the quality of training, as one does not

have to set aside a portion of the training data for validation. For static NNs, we first decided on number of neurons in the hidden layer that would provide the best result. This was determined by exploring NN performance using validation ROC AUC from a subset of user data. Table 4 shows the results for three different counts of hidden layer neurons. Consequently, we used 5 neurons for the hidden layer. Training was stopped when either minimum gradient (10^{-10}) or 1000 epochs was reached. For each case the network was initialized 15 times as randomly initialized neural networks converge to a different local solution during each gradient descent [61].

Table 4: Validation ROC AUCs from a pilot subset to decide the best NN architecture.

		ROC AUC (%)
Number of Hidden Layer Neurons	10	95.343
	5	96.875
	3	94.88

Next, user-dependent 3-fold cross validation and user-independent 6-fold cross validations were carried out. Using Bayesian regularization and maximum of 1000 epochs of training or adaptive step size of 10^{10} as training exit condition, each NN was initialized 5 times and retrained for each fold. Tables 5 and 6 show the result for user-dependent and independent cases, respectively.

Similar to the static NNs, we utilized TDNNs with one hidden layer and hyperbolic tangent nonlinear neurons. Four hyperbolic tangent functions were used at the output layer as indicator functions to deliver the four binary decisions on detected class. The area under each output signal was taken as the scalar classifier output for each gesture. Bayesian regularization was used for training.

Table 5: User-Dependent NN classifier results.

User	Training		Validation	
	EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
User 1	0	100	18.61	84.694
User 2	0	100	6.6667	99.176
User 3	0	100	5	99.144
User 4	0	100	0	100
User 5	0	100	3.0556	99.694
User 6	0	100	8.3333	97.958

Table 6: User-Independent NN classifier results.

Training		Validation	
EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
0.055556	99.976	18.194	84.067

As the users performed each gesture at the speed of one gesture per second at a sampling frequency of 30 Hz, the size of the input tapped delay line was set to 30. We trained and tested the network for three different hidden layer sizes; 10, 15 and 20, to find a suitable size for the hidden layer. User-dependent 3-fold cross validation and user-independent 6-fold cross validations were performed. Networks for each fold were initialized 5 times. Tables 7 to 9 show the user-dependent result across different hidden layer sizes. Tables 10 to 12 repeat the same results but for user-independent case.

Table 7: User-Dependent TDNN classifier results. 10 Hidden Layer Neurons

User	Training		Validation	
	EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
User 1	0	100	5	99.338
User 2	0	100	0.2777	99.954
User 3	0	100	3.6111	99.546
User 4	0	100	0	100
User 5	0	100	1.9444	99.958
User 6	0	100	3.3333	97.116

Table 8: User-Dependent TDNN classifier results. 15 Hidden Layer Neurons

User	Training		Validation	
	EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
User 1	0	100	3.0556	99.685
User 2	0	100	1.6667	99.944
User 3	0	100	1.9444	99.894
User 4	0	100	0	100
User 5	0	100	0	100
User 6	0	100	3.0556	99.889

Table 9: User-Dependent TDNN classifier results. 20 Hidden Layer Neurons

User	Training		Validation	
	EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
User 1	0	100	3.0556	99.829
User 2	0	100	0.2777	99.981
User 3	0	100	3.3333	99.852
User 4	0	100	0	100
User 5	0	100	0	100
User 6	0	100	3.0556	99.769

Table 10: User-Independent TDNN classifier results. 10 Hidden Layer Neurons

Training		Validation	
EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
0.4722	99.988	4.8611	98.102

Table 11: User-Independent TDNN classifier results. 15 Hidden Layer Neurons

Training		Validation	
EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
0.1667	100	3.889	99.054

Table 12: User-Independent TDNN classifier results. 20 Hidden Layer Neurons

Training		Validation	
EER (%)	ROC AUC (%)	EER (%)	ROC AUC (%)
0	100	3.3333	99.182

It is concluded from the tabulated results that, using statistical (static) features of the acceleration signals, Fisher LDA and NN provided about the same user-independent classification rates for validation, though neural networks provided a better user-dependent performance on the average. However, the advantage of using neural networks was more striking with TDNNs, where gesture classification rates for validation were almost perfect (ROC AUC ≈ 1) for both user-dependent and user-independent modes. More specifically, from the tabulated results we can see that the average user-dependent validation ROC AUC and EER (Section III.C) for Fisher LDA are 86.17% and 21.76%, respectively. For NN these figures stand at 96.78% and 6.94%, while for TDNN they jump to 99.81% and 1.96% (average over the three different architectures).

For the user-independent case, LDA validation ROC AUC is 89.9% and EER is 18.05%. These figures for NN are 84.06% and 18.194% respectively. Finally for TDNN the validation ROC AUC is 98.78% and EER is 4.03% (averaged over the three different architectures).

Results were generally improved when moving from static linear classifiers to nonlinear dynamic architectures. Improved results were obtained by using NNs as they have the ability to learn invariance to input noise and distortions through training, a hallmark of non-ideal accelerometer data collected from highly variable human user behavior. TDNNs, though computationally more complex, provided almost perfect results. The fact that they do not require any feature extraction makes them an attractive technique for gesture recognition. It was also observed that better accuracies were achieved in NN and TDNN when the number of hidden layer nodes were half or more of the sampling

frequency, 30 Hz in this case.

2.10 Increasing battery life of a gesture recognition wireless network using Neural Networks

Radio transmissions are probably the most energy-demanding operation that a sensor node undertakes. For instance, a typical low-power integrated radio transceiver module would consume between 15 mA to 20 mA in reception mode and 11 mA to 21 mA during transmission mode at V_{DD} = 3.0 V (actual levels depend on the speed and output power levels selected). Meanwhile a low-power micro-controller such as the MSP430 consumes around 6 mA at V_{DD} = 3.0 V (while running at 16 MHz). The power consumption of commercially available MEMS inertial sensors is typically low (350 μA for a three-axis accelerometer such as the ADXL335 and 6.8 mA for the LPR530AL gyroscope). Thus, reducing the energy spent during radio transmissions will have the most impact on the power consumption on the sensor node.

Our approach to reduce the radio transmission power consumption relies on scaling down the transmission rate (number of bits per second). We accomplish this by varying the sampling rate and the bits per sample of the motion sensor. This reduction inevitably leads to a loss of information. However, by using state-of-the-art classification algorithms like linear discriminant analysis and neural networks at the receiver side, a number of motion gestures can still be correctly classified but at a fraction of the original transmission rate. The main advantage of this method is its low complexity requiring simple bit shift and down-sampling operations. Data compression techniques could also

be employed but they require transforming the sensor signal to another signal space and encoding the resulting coefficients. These operations might be too costly or too difficult to implement in a low-cost sensor node with limited computational and memory resources. Another feature of the proposed technique is that it shifts the computational burden to the receiver/decoder side which usually has access to more computational and memory resources.In this work we study the impact of reducing the bit transmission rate on gesture classification. The sampling rate was varied from 100 Hz down to 5 Hz while the quantization bits were varied from 8 bits down to 4 bits. Classifiers employed are Fisher's linear discriminant classifier, static neural networks and time delay neural networks (TDNN).

Gesture vocabulary employed were the same used for the sports watch study, however, the number of users were now four instead of six. Static features for LDA and NN were the same as used by the sports watch study. We collected three sets of each of these gestures for each of four users. Users were instructed to repeat each gesture at a speed of about one gesture per second. Motion data was collected for 10 seconds and then the user switched to the next gesture. Four fold cross validation and only subject-independent classification was explored.

Initially we collected inertial data at a sampling rate of 300 Hz and estimated the power spectral density using the Burg method. We then measured the -10 dB frequency of each data set. We found that the maximum -10 dB frequency across gestures and subjects was 50 Hz. Therefore, we started our study with a maximum sampling of f_s = 100 Hz and progressively decreased it to 5 Hz. This was accomplished using decimation. To avoid aliasing, the data was low-pass filtered using an 8th order Chebyshev digital filter. In

practice, a lower-order filter could be implemented on the micro-controller or externally in the analog domain.

A total of four LDAs were trained in parallel. Each one was trained to classify one gesture from the rest, i.e., circle and non-circle, triangle and non-triangle, etc. Table 13 summarizes these results. The average accuracy obtained for each combination was in the range of 80% to 89%. As the quantization bit rate was decreased to 4 bits, the correlation between the X and Z axes was not reliable due to low amplitude, thus, it was discarded.

Table 13: Results for Fisher's LDA classification

Quantization bits	Performance measure	Sampling frequency f_s (Hz) 100	50	25	10	5
8	EER	0.256	0.250	0.267	0.267	0.270
	AUC	0.829	0.830	0.827	0.819	0.807
6	EER	0.256	0.270	0.270	0.270	0.267
	AUC	0.831	0.827	0.818	0.806	0.808
4	EER	0.274	0.277	0.274	0.315	0.291
	AUC	0.802	0.803	0.795	0.783	0.794

For the static neural network we used the same features used for the LDA. A network with one hidden layer was employed. The size of the hidden layer was determined from the sports watch study to be half the sampling rate plus two. The output layer has four neurons. Each of these neurons signals the classification result of the network. The networks were trained using Bayesian regularization and the number of epochs was set to 100. We initialized the network five times for each fold (4-fold cross validation, leaving out one subject at a time). Table 14 summarizes the accuracies achieved when unseen data was fed to the trained network. For sampling rates of 100 Hz to 10 Hz and number of bits 8, 6 and 4, the accuracy was in the range of 78% to 84%. For lower sampling rates the

accuracy was about chance level. As the sampling rate was reduced to 5 Hz, the number of samples were down to 34, therefore we used the raw data instead of extracted features as it resulted in more accurate classification rates.

Table 14: Results for static neural network classification

Quantization bits	Performance measure	Sampling frequency f_s (Hz)				
		100	50	25	10	5
8	EER	0.241	0.228	0.218	0.211	0.534
	AUC	0.804	0.816	0.801	0.815	0.478
6	EER	0.227	0.256	0.243	0.247	0.518
	AUC	0.811	0.787	0.800	0.783	0.494
4	EER	0.228	0.207	0.243	0.211	0.455
	AUC	0.822	0.845	0.816	0.812	0.548

Finally, we used Time Delay Neural Networks (TDNN). The size of input tap delay line was set equal to sampling rate over two and the number of nodes in the hidden layer was half the sampling rate plus two. Both hidden layer and the four-output layer neurons used the hyperbolic tangent function as their output function. Just as in static neural network case, we initialized the network five times for each fold. Table 15 summarizes the results. The accuracy achieved were 90% and over, for each combination of sampling rate and number of bits. For a change in the quantization bits from 8 to 4 bits, the AUC decreases by about 3%. Thus, TDNN turned out to be the best among the three methods used in this study, based on EER and AUC of the ROC curves. This suggests that the motion data is nonlinear in the given feature space. Besides accuracy another advantage of TDNNs is that they do not need feature extraction. Our results indicate that, in the extreme case of f_s = 5 Hz, a total 10 samples/s (X and Z axes) need to be transmitted. Considering the quantization rate of 4 bits/sample we obtain a transmission rate of

40 bits/s. Compared with the initial rate of 1600 bits/s (f_s=100 Hz and 8 bits/samples), transmission savings of 40 times could be achieved.

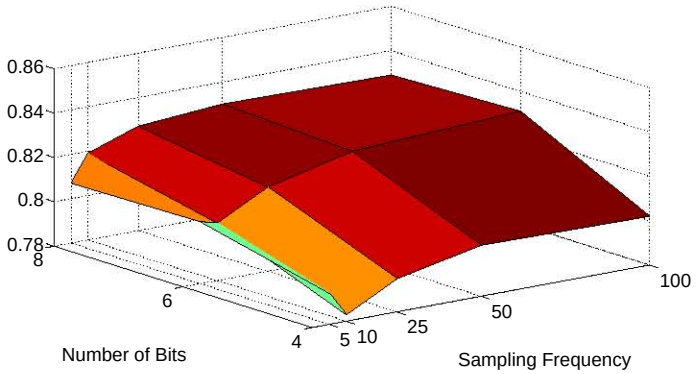
Table 15: Results for time delay neural network classification

Quantization bits	Performance measure	Sampling frequency f_s (Hz) 100	50	25	10	5
8	EER	0.122	0.118	0.125	0.152	0.202
	AUC	0.906	0.944	0.967	0.926	0.904
6	EER	0.089	0.097	0.123	0.168	0.181
	AUC	0.857	0.960	0.961	0.923	0.900
4	EER	0.225	0.212	0.187	0.229	0.231
	AUC	0.869	0.889	0.903	0.888	0.900

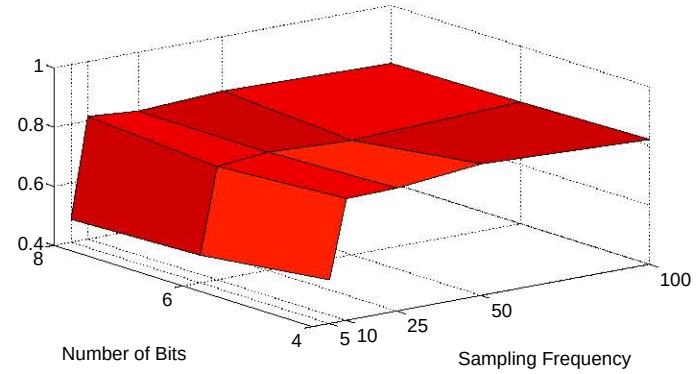
Plotting AUCs for all combinations shows that each have the same level of merit. Therefore, it is imperative for low power consumption that we choose 4 bits of data, 5 Hz of sampling frequency and TDNN as the classifier for its highest accuracy rates.

Preliminary current consumption analysis was performed. The radio transceiver of the designed motion tracker is programmed to transmit 50 byte data as part of the transmitted packet. Each packet consists of a 32-bit preamble, 32-bit sync word, 1-byte address indicator, 1-byte length of packet indicator and a 16-bit CRC. All this along with the 50 byte data adds upto 496-bits as the packet size. The transmission rate is programmed to be 250 kbps. Therefore, time taken to transmit a single packet, T_1, is equal to 2 ms. The transmission power is set to be 0 dBm and according to the datasheet of the transceiver current, I_{TX}, consumed in transmitting a single packet is 21.2 mA. Time between two consecutive packets, T_2, is given by the equation:

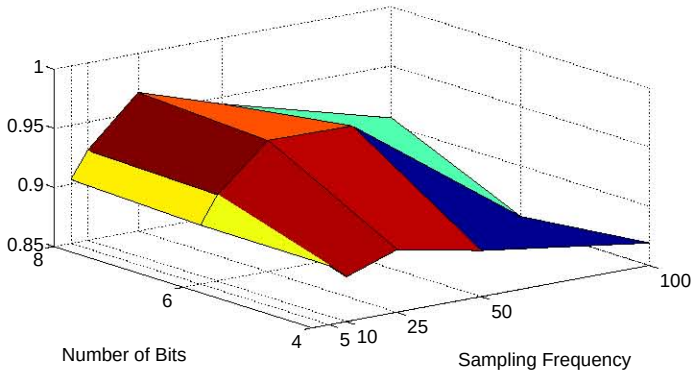
$$T_2 = (packet\ size)/(transmission\ rate)$$



(a)



(b)



(c)

Figure 14: Area under the curve (AUC). (a) LDA AUC. (b) Static neural networks AUC. (c) Time delay neural networks AUC.

Finally, the average current consumed by the motion tracker is calculated by:

$$I_{avg} = (I_{TX} \times T_1)/(T_1 + T_2)$$

Figure 15 shows the timing diagram for packet transmission.

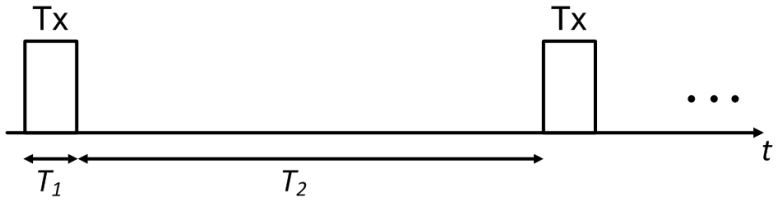


Figure 15: Packet transmission timing diagram

Transmission of packet is controlled by the sampling frequency because a transmission happens only when desired number of samples are available. Thereby reducing the total current consumed. Additionally, decreasing the number of bits allows transmission of more samples, which requires more time. Table 16 below shows the amount of current consumed for the combination of sampling frequency and quantization bits employed above for gesture classification. From the table it can be seen that the current consumed decreases for 0.0841 mA for sampling frequency of 100 Hz and 8-bit samples to 0.0021 mA for 5 Hz and 4-bit samples. This is because of the reduced transmission time between two packets. Table 17 shows how the time changes for the different sampling frequency and bits per sample.

In conclusion, with the designed board, we explored reducing the sampling rate and the bits per sample in order to reduce the energy spent on radio transmissions with the aim of extending battery life. Therefore, to compensate for the loss of information due to reduced sampling rate and number of bits, we used time delay neural networks, an

Table 16: Current consumption (mA) for the combination of sampling frequency and quantization bits

Bits	Sampling frequency f_s (Hz)				
	100	50	25	10	5
8	0.0838	0.0420	0.0210	0.0084	0.0042
6	0.0629	0.0315	0.0158	0.0063	0.0032
4	0.0420	0.0210	0.0105	0.0042	0.0021

Table 17: Time (s) between two packets

Bits	Sampling frequency f_s (Hz)				
	100	50	25	10	5
8	0.5000	1.0000	2.0000	5.0000	10.0000
6	0.6667	1.3333	2.6667	6.6667	13.3333
4	1.0000	2.0000	4.0000	10.0000	20.0000

advanced signal classifiers and achieved user-independent classification rates of 90% and above even when the transmission rate is as low as 40 bits/s. These classification rates were maintained even without drift and gravity compensation and at sampling rates that are below the Nyquist rate. The proprietary platform designed for this study consists of wearable sensors nodes that include inertial and acoustic motion sensors as well as a low-power micro-controller, a low-cost 2.4 GHz radio transceiver and supporting circuitry.

One way to increase battery life of a context-aware wireless sensor is by using the programmability of the micro-controller and regulate the activity of the sensor. In addition to that, results of this study highlights the fact that in applications where regeneration of the signal is not needed, such as gesture classification, transmitting less information at reduced rates can deem advantageous too.

CHAPTER 3

BONE STRAIN MEASURING TELEMETRY UNITS

This chapter presents the design and validation of telemetry units used to measure strain on bones. These prototypes are created using commercially available off-the-shelf components and vary in design and sizes. Several researchers have proposed designing custom integrated circuits to measure strain on bone. Their approach does reduce size of the device, however, the development time is considerably larger than using off-the-shelf components. Also, recent advances in semiconductor fabrication have made commercially available highly integrated systems on-chip which we exploit to develop a small strain-monitoring telemetry units.

3.1 Bone Strain Measurement

The skeleton is an organ that is constantly adapting its mass and architecture in order to meet the demands resulting from its three primary functions of protecting the internal soft tissue organs, provide structural support and be a reservoir of calcium. It is known that the mass and structural properties of the skeleton adjust in proportion to changes in mechanical load, but the molecular basis of how this is accomplished is only partially understood. In order to unravel the mechanisms of bone formation, scientists need to determine the mechanical load levels that trigger bone mass increase. To accomplish this task, localized bone strain levels need to be measured as a load is applied to the

bone. This kind of bone biology studies might lead to advances in musculoskeletal diagnostics and in the development of pharmaceutical targets enabling new paradigms and treatments for bone diseases such as osteoporosis.

Besides its importance to bone biology studies, bone strain measurement is also of interest in orthopedic implant development and monitoring. The design of orthopedic implants requires information about the range of acting loads and resulting implant and bone deformations. Knowledge of bone strain also facilitates rehabilitation monitoring and feedback as well as improved data collection in clinical studies [77]. In comparison to diagnosis using X-ray images which only show bone callus geometry, strain monitoring is more effective in guiding the rehabilitation exercises as well as to predict implant malfunction and to continuously monitor the healing process [10] .

Strain gauges are normally employed to monitor bone strain due to their small size, robustness and good sensitivity [80]. Strain gauges are transducers that convert strain into electrical resistance. Hence, an electronic circuit that measures resistance is needed in strain monitoring. Moreover, the measuring device should have a wireless means of transferring the strain information as the usage of wire leads will preclude full implantation and constrain motion range.

Many researchers have reported telemetry units for strain measurements on prosthetic implants. These studies gather information regarding forces that need to be sustained by the prosthetic implant and offer confirmation of recovery. In [24] an ASIC for strain monitoring in prosthetic implants is presented. This unit is inductively powered using an AC-DC circuit and strain data is wirelessly transmitted using an on-chip radio

transmitter. The power-receiving coil dimensions depend on the type of prosthetic. Due to space constraints this antenna cannot be easily tuned limiting the range of the whole setup to 50 cm. Although the designed ASIC is very small, the entire unit including the power coil and the antenna measures a few centimeters long.

In [82] the authors successfully placed a small cylindrical transducer (3.2 mm diameter and 4.6 mm height) into an incision made between the tibia and the femoral. This transducer is connected to a transmitter (4.5 cm x 2.2 cm) and transmitter powering unit (3.04 cm x 1.6 cm) placed outside the incision. This approach is not only large in size but it is bone-invasive and compromises the architecture of the bone.

Another solution presented in [12] uses a commercially available telemetry unit from MicroStrain®. It has a large size (9.0 cm x 9.0 cm), weighs 100 g and is powered by two 3.7 V batteries. The large size of this unit precludes its implantation in small animals. Furthermore, when it runs out of battery or the gage fails, the subject has to be sacrificed.

3.2 Target Application

The target application for the telemetry unit presented here is bone biology studies where localized bone strain needs to be monitored under different load conditions. Bone biology scientists use this information to understand the mechanisms that regulate bone formation. Currently our collaborators use the setup shown in figure 16 to carry out bone strain measurements. The setup consists of a specialized bench-top data acquisition unit for strain gauge measurement (Vishay Micro-Measurement System 7000). In this setup the mouse or subject is fully immobilized while a known force is applied to its ulna

bone and strain readings are collected. The force is applied with the Bose ElectroForce 3200 load instrument. This procedure is repeated for a period of several days after which the mouse is sacrificed and the ulna bone is studied for changes in the bone matrix and expression of certain genes.

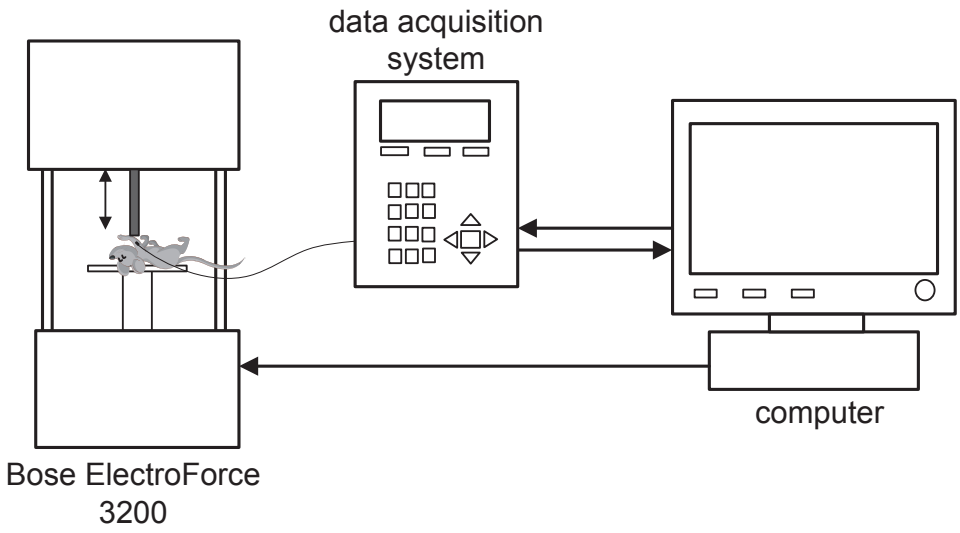


Figure 16: Current measurement setup consisting of a Bose ElectroForce 3200 load test instrument and a Vishay Micro-Measurement 7000 data acquisition system. Both systems are controlled by a dedicated computer.

A related question that bone researchers would like to answer is to what degree exercise impacts bone formation. To this end, researchers will like to monitor bone strain as the mouse performs a set of cage exercises. The current strain acquisition system is bulky and requires wires to be connected from the data acquisition unit to the bone. Hence, it is not suitable for this type of experiments. To provide a solution to this need we have developed a multichannel telemetry unit for bone strain monitoring. Figure 17 shows

a conceptual diagram of the bone monitoring setup with a telemetry unit. The telemetry unit is small enough to be mounted on the back of a mouse. It can also be implanted in larger animals for bone biology studies or to monitor orthopedic implants.

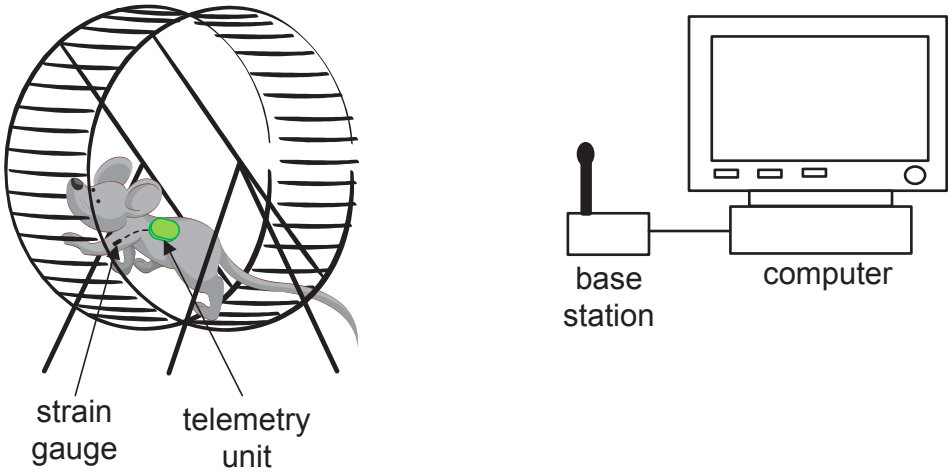


Figure 17: Conceptual diagram of a wireless system for real-time bone strain monitoring. The subject is free to move and perform bone-growth stimulating exercises.

3.3 Measuring Strain

Strain gauges are piezoresistive sensors, i.e., its resistance changes when it is deformed due to applied strain. The most common type of strain gauge is the metallic strain gauge which consists of a very fine wire arranged in a grid pattern. This wire is bonded to a thin and flexible substrate which is attached directly to the test specimen. As the test specimen is deformed, the thin wire in the gauge is stretched or compressed changing its electrical resistance [55]. Other types of strain gauges are based on semiconductor

materials, like silicon. Silicon based strain gauges are usually more sensitive than metallic gauges. However, metallic gauges tend to have better linearity [30]. The change in resistance ΔR and the strain are related by the following equation:

$$G = \frac{\Delta R}{R \times \varepsilon} \tag{3.1}$$

where, G is the gauge factor, R is the nominal gauge resistance and ε is the strain experienced by the gauge in units of micro-strain ($\mu\epsilon$). Figure 18 shows a typical strain gauge pattern which has a zigzagged conductor path. This pattern is commonly used to effectively increase the length of the resistor and the amount of total resistance under a given area.

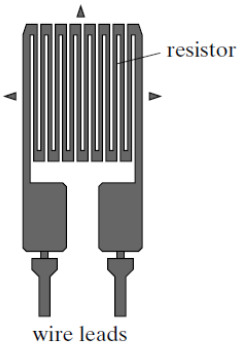


Figure 18: Strain gauge.

The current measurement setup uses uni-axial metallic strain gauges of nominal resistance of 120 ohms and a gauge factor of $G = 2.07$ (Vishay EA-06-015DJ-120). These strain gauges were chosen due to their small size. The maximum bone strain that is expected in the experiments is 3000 $\mu\epsilon$. Therefore, the maximum expected change

in resistance is 0.75 ohms or 0.625%. The traditional approach to measure such small resistance changes is to use a Wheatstone bridge in combination with an amplifier as shown in Figure 19(a).

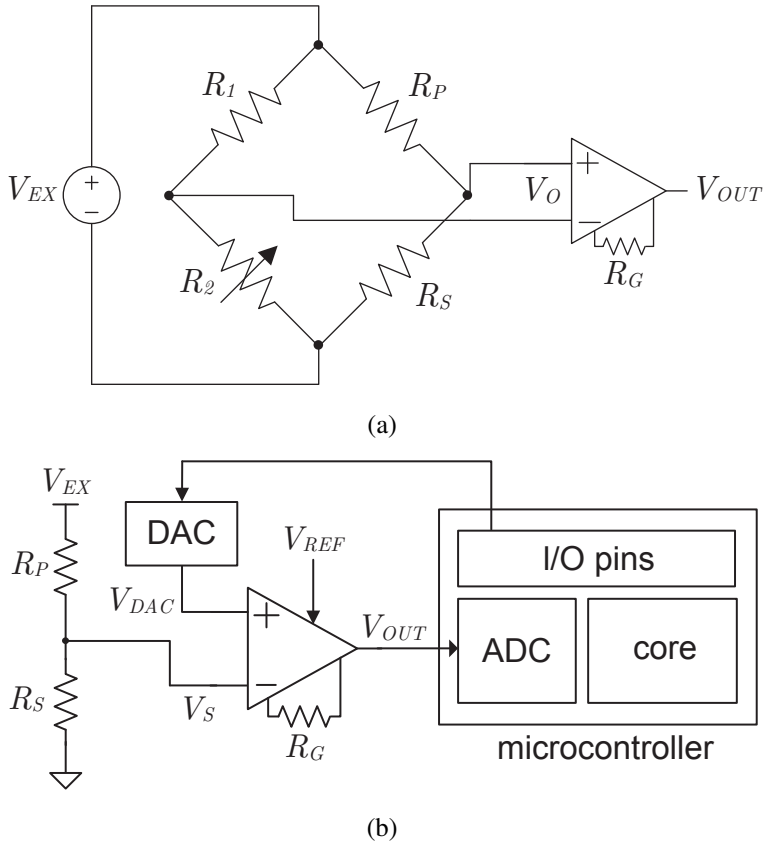


Figure 19: Resistance measurement and calibration circuit. (a) Wheatstone bridge. (b) DAC-based approach.

In the figure, R_S is the strain gauge resistance. An instrumentation amplifier is needed to amplify the small bridge voltage V_O. The variable resistor R_2 is used to calibrate the bridge such that $V_O = 0$ when no strain is applied. In the targeted application a mechanical potentiometer to implement R_2 was ruled out to avoid vibration-induced changes

in its resistance. A digital potentiometer is not affected by vibrations but commercially-available digital potentiometers do not have enough resolution to match the expected resistance change in R_S.

To address this problem we employed a calibration approach that is based on a high-resolution digital-to-analog converter (DAC) instead of a variable resistor. The branch of the Wheatstone bridge composed by R_1 and R_2 was replaced with a DAC controlled by a microcontroller as shown in figure 19(b). The resistance R_G sets the gain, A, of the amplifier. The calibration procedure is depicted in figure 20. The basic idea of the calibration procedure is to generate a voltage ramp and monitor the output of the instrumentation amplifier when no load is applied to the strain gauge. Calibration is complete when the output of the amplifier, V_{OUT} , equals the reference voltage V_{REF}. The DAC value at the end of calibration is stored and applied in subsequent readings of the amplifier. From figure 19(b) the output of the instrumentation amplifier is given by:

$$V_{OUT} = V_{REF} + A(V_{DAC} - V_S) \tag{3.2}$$

Ideally, we would like to set $V_{DAC} = V_S$ so that $V_{OUT} = V_{REF}$. To that end, the microcontroller is employed to generate a ramp at the output of the DAC. As the ramp is generated the microcontroller monitors the amplifier's output voltage by means of its internal analog-to-digital converter (ADC). Calibration is achieved when the amplifier's output equals V_{REF}. At that point the ramp is stopped and the DAC input value is stored.

In practice, the calibration procedure described above is limited by the resolution of the DAC. At the end of calibration the maximum value of the difference $V_{DAC} - V_S$ is

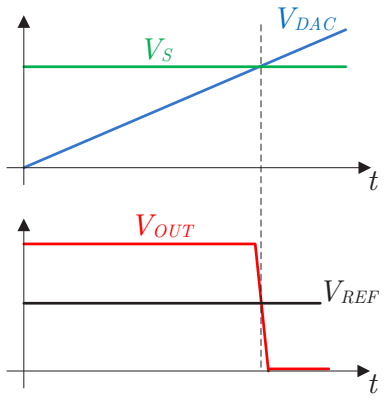


Figure 20: Employed calibration procedure based on a DAC and a microcontroller. The microcontroller generates a ramp using the DAC output until the amplifier's output equals the reference voltage V_{REF}.

$V_{LSB}/2$, where $V_{LSB} = V_{DD}/2^n$ and n is the DAC resolution. This difference produces a maximum difference of $A \cdot V_{DDA}/2^{n+1}$ at the amplifier's output from the ideal value of V_{REF}. Considering a target gain of $A = 330$, a supply voltage of $V_{DDA} = 3.0$ V and a 12-bit DAC resolution, the maximum output offset is 120 mV. This offset is much smaller than the supply voltage and does not have a major impact on the dynamic range of the amplifier's output. Thus, a DAC resolution of 12 bits is sufficient for the intended application. Since the offset due to finite DAC resolution remains constant throughout the measurement process, it can be canceled out digitally.

3.4 Telemetry Unit 1.0

This unit has been designed around an ultra low-power microcontroller (MSP430). The microcontroller makes the design highly flexible and programmable. The telemetry unit also includes a high-performance instrumentation amplifier to amplify the strain

gauge output. The gain and offset of the amplifier are digitally set by the microcontroller eliminating the use of manual potentiometers. The board has an expansion connector that allows up to 16 additional strain gauges to be connected to the unit and incorporates a low power radio transceiver operating in the 2.4 GHz ISM band. Figure 21 shows the block diagram of the unit and figure 22 shows the designed two layer PCB with components.

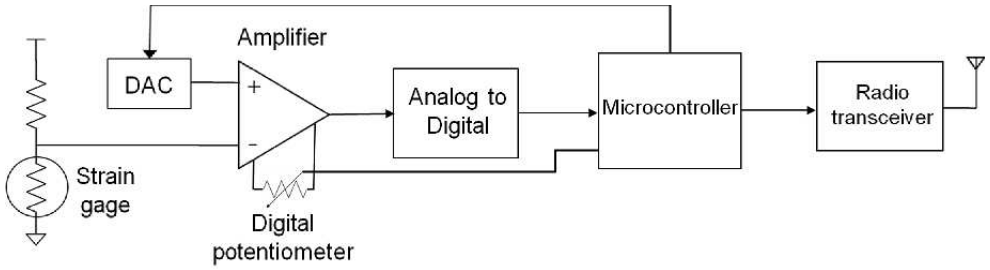


Figure 21: Telemetry unit block diagram.

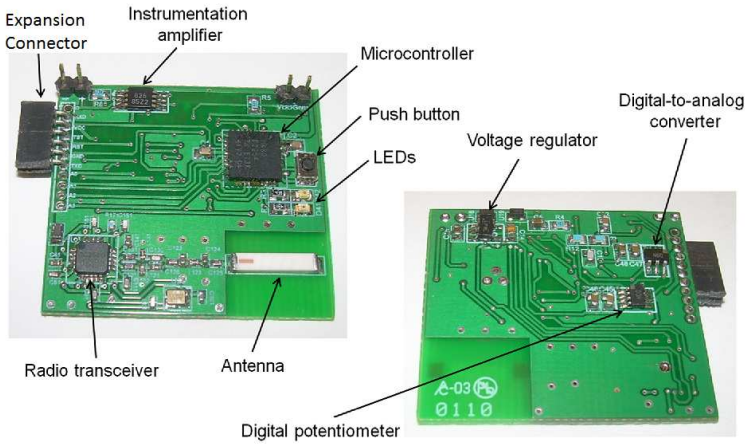
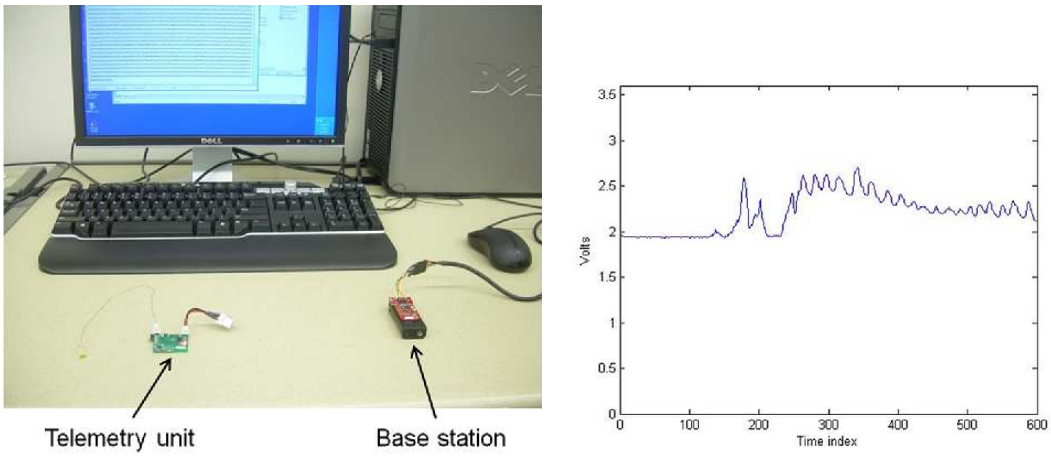


Figure 22: Telemetry unit PCB with components.

The telemetry unit has been tested in a lab setting and is able to transmit the strain

data at distances greater than 20 m while consuming less than 30 mW of power. This low power consumption allows the unit to be powered by a micro-battery weighting less than 3 grams. The telemetry unit can be used in other biomedical applications such as in the monitoring of orthopedic implants and can be easily configured to use other type of sensors. Figures 23 show the testing of the unit and real time strain plot.



(a) (b)

Figure 23: (a) Telemetry unit under lab test. (b) Real time strain data received from the telemetry unit.

3.5 Telemetry Unit 2.0

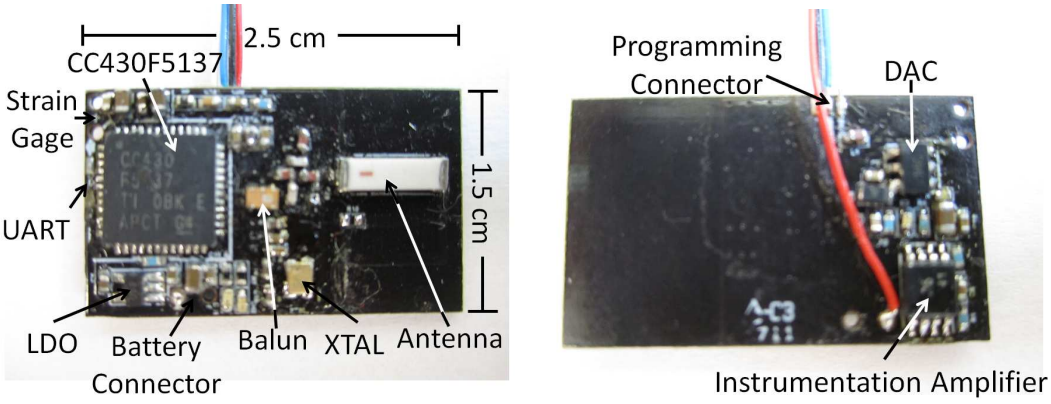
This telemetry unit has been designed around an ultra-low-power microcontroller CC430F5137 from Texas Instruments. The CC430F5137 integrates a sub1-GHz RF transceiver with a 16-bit RISC CPU, a 12-bit analog-to-digital converter (ADC) and other peripherals. The microcontroller measures 8 mm x 8 mm and along with rest of the surface mount components satisfy the design challenges of the telemetry unit: small size

and low power consumption. The unit has a small size of 2.5 cm x 1.5 cm and operates from a 3.7 V Li-Pol battery that weighs less than 3 grams. A low dropout (LDO) voltage regulator TLV70033 was employed to provide a steady 3.3 V voltage supply.

The RF transceiver requires an antenna impedance matching network. To reduce the number of components and the size of the matching network, we employed an impedance matching balun from Johanson Technology (0896BM15A0001). A 915 MHz chip antenna from Johanson Technology (0915AT43A0026) and a small 26 MHz crystal oscillator from Nihon Dempa Kogyo Co., LTD. (NX2016AB) are also needed by the RF transceiver and were included on the board.

Zeroing is performed by the microcontroller by generating a voltage ramp at the output of the DAC (AD5320). The ramp is stopped when the output of the amplifier reaches the desired zero-level voltage. The telemetry unit employs the precision instrumentation amplifier INA326 from Texas Instruments. The INA326 is a low-power amplifier that features rail-to-rail input common-mode voltages, has low offset voltage and very low 1/f noise. Figure 24 shows the top and bottom of the unit with soldered components and figure 25 depicts its block diagram.

The telemetry unit was tested on an *ex-vivo* setting. *In-vivo* tests could not be performed at this time because the placement of the gauges requires survivable surgery and IACUC approval of the protocol. For the ex-vivo testing, a 120 ω strain gauge (Vishay, EA-06-015DJ-120) was first cut into a size of 2.54 mm length and 0.51 mm width and then was glued to a dissected ulna of a mouse. The adhesive used was M-bond 2000 from Vishay-Micro-Measurements. The ulna with the attached strain gauge was then placed on



(a) (b)

Figure 24: Telemetry Unit. (a) Top side. (b) Bottom side.

the bench top setup.

Readings were first collected using data acquisition system. The bone was loaded with a sinusoidal force of 2 Hz and 3 N peak-to-peak for few cycles. The StrainSmart® software recorded the strain data and saved it to a text file. Figure 26(a) shows 5 cycles of such reading. Strain was measured to be around -3000 to 0 μstrain.

Next the bone-attached strain gauge was connected to the telemetry unit. The process of applying force was repeated in the same manner as in the case of data acquisition system. Figure 26(b) shows the strain acquired by the telemetry unit. It also shows a digitally filtered version that removes most of the noise. It can be seen that the telemetry unit has accuracy comparable to the benchtop system. Figure 27(a) shows this telemetry unit connected to the strain gauge attached to the bone that on the load force system.

The sampling frequency of the telemetry unit can be modified by changing the timer settings of the microcontroller. We measured the current consumed by the unit by varying the sampling frequency. It can be seen from the plot of figure 27(b) that the unit

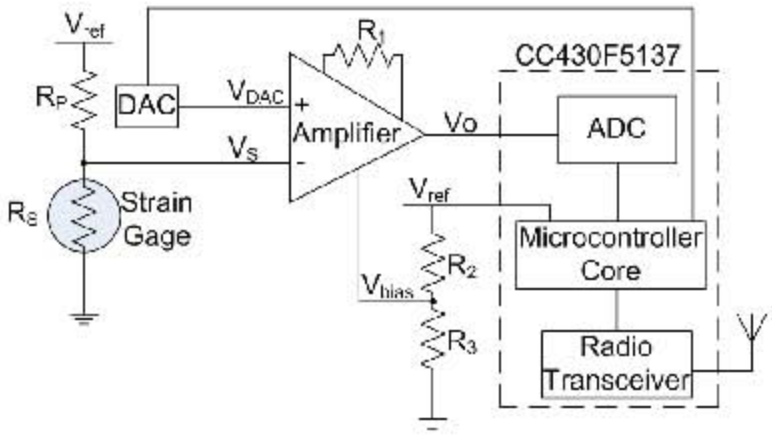


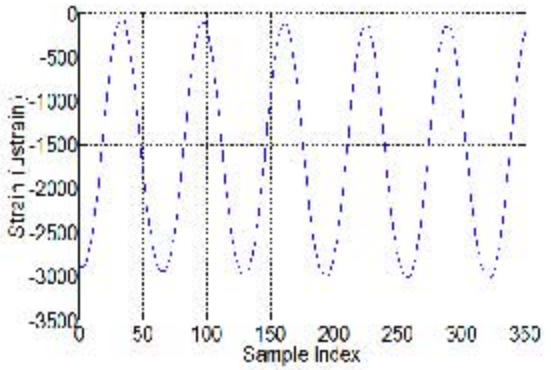
Figure 25: Telemetry unit block diagram.

consumes 1 mA at 15 Hz to 7 mA at 300 Hz. The plot also shows that when the unit is asleep, current consume is on.y 0.5 mA.

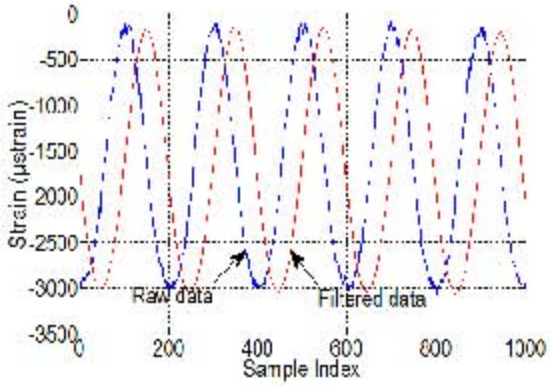
3.6 Telemetry Unit 2.1

In addition to the features of the previous version, this telemetry unit incorporates a 3-axis digital accelerometer and an expansion connector to connect more than one strain gauge at once. This unit was also tested in an *ex-vivo* setting and a user interface of the previous version was modified to capture strain data along with accelerometer data. The user interface is designed in Matlab and is connected to the base station via a virtual COM port. Figures 28 and 29 show the units PCB and its block diagram respectively.

Like version 2.0, this user interface can be used to wirelessly change the sampling frequency, put the unit to sleep, change transmission power level and run the calibration routine. The user can also save the data on the screen, start and stop the data collection.



(a)

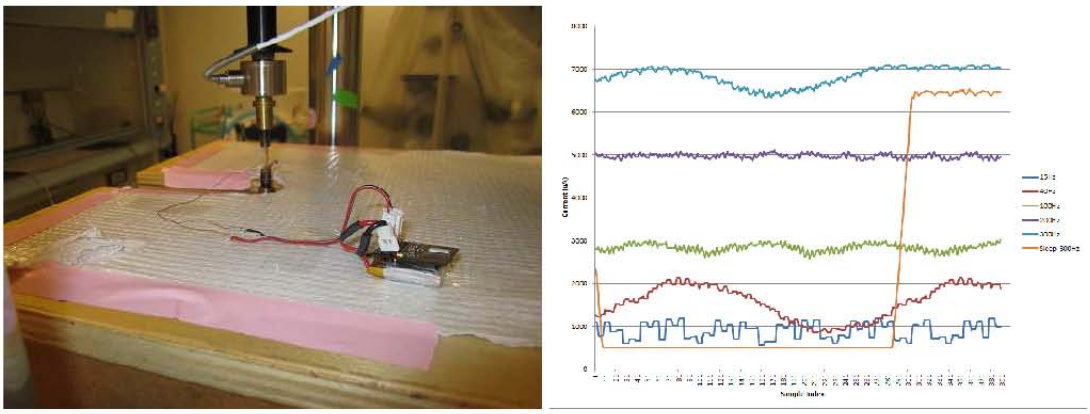


(b)

Figure 26: Strain readings. (a) StrainSmart® strain reading. (b) Filtered and raw strain readings from the telemetry unit.

3.7 Telemetry Unit 3.0

A smaller unit measuring only 2.4 cm $\times$ 1.3 cm was designed and tested. This unit considerably adds to the previous versions. Figure 30 shows a simplified schematic diagram of the telemetry unit. The telemetry unit is designed around the CC430 microcontroller from Texas Instruments. This microcontroller was chosen because it integrates a range of peripherals such as a 12-bit ADC, a 16-bit timer and a 915 MHz radio transceiver.



(a) (b)

Figure 27: Strain readings. (a) Telemetry unit connected to the strain gauge attached to the bone. (b) Telemetry Unit current consumption at different sampling frequencies.

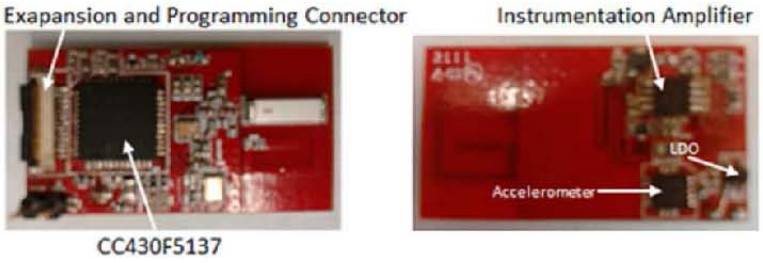


Figure 28: Telemetry Unit PCB.

The integration of these peripherals along with a low-power 16-bit CPU on a single chip that measures 8 mm $\times$ 8 mm enables the development of a compact wireless strain sensor.

An 8-channel multiplexer (MUX) is employed to allow up to 8 different strain gauges to be connected to the instrumentation amplifier. A precision instrumentation amplifier A_1 (INA333) is employed to amplify the voltage difference $V_{DAC} - V_S$. The gain of A_1 is set by a single resistor R_G as follows:

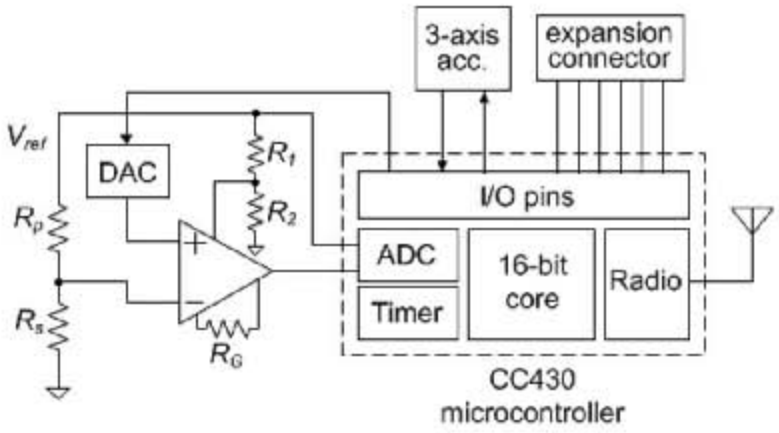


Figure 29: Telemetry unit block diagram.

$$A_1 = 1 + \frac{100\ \text{k}\Omega}{R_G} \tag{3.3}$$

and was set to 334 by choosing $R_G = 300$ ohms.

The voltage V_S is a function of the strain gauge resistance through the following voltage resistive divider relationship:

$$V_S = V_{DDA}\frac{R_S}{R_S + R_P} \tag{3.4}$$

The resistance R_P is a precision resistor with a value matching the nominal resistance of the strain gauges. Using (3.3) and (3.4) yields the following expression for the output of the instrumentation amplifier:

$$V_{OUT} = V_{REF} + A(V_{DAC} - V_S)$$

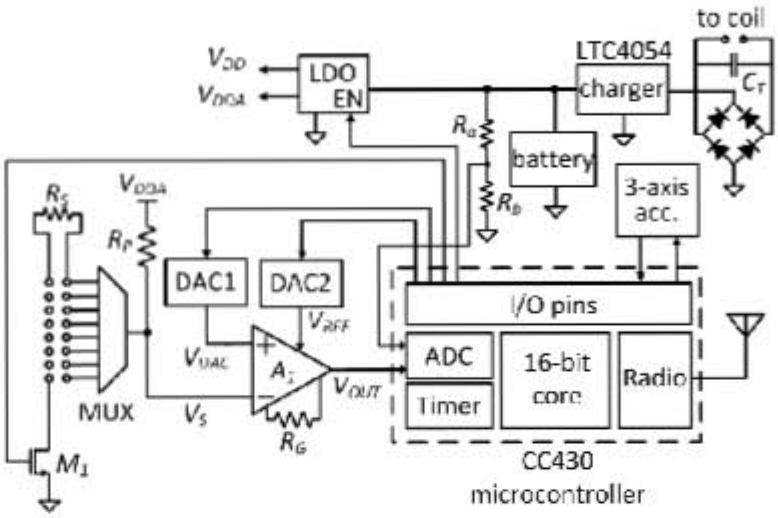


Figure 30: Schematic diagram of the telemetry unit.

$$= V_{REF} + \left(1 + \frac{100\ \text{k}\Omega}{R_G}\right)\left(V_{DAC} - V_{DDA}\frac{R_S}{R_S + R_P}\right) \tag{3.5}$$

Thus, the amplifier's output is a function of the strain gauge resistance R_S which in turn is a function of the strain applied to the gauge through (3.1). Hence, the strain experienced by the gauge can be calculated from the voltage output of the amplifier. The current that flows through R_S and R_P is given by:

$$I_R = \frac{V_{DDA}}{R_S + R_P} \tag{3.6}$$

Considering $R_S = R_P = 120$ ohms and $V_{DDA} = 3$ V results in a current of 12.5 mA flowing through the strain gauge. This current is quite large for a low-power sensor that is expected to run for long periods of time from a small battery. To reduce

this current, R_P could be increased. However, an increase in R_P will result in a reduced voltage across the strain gauge ultimately affecting the signal-to-noise ratio (SNR). To reduce current consumption without sacrificing SNR, the MOSFET M_1 (PMV16) was added in series with R_P and R_S. The MOSFET works as a switch allowing current to flow through the resistors only when a reading is being taken. Otherwise, the MOSFET is turned off to save current consumption.

A second DAC was added to provide a programmable voltage reference V_{REF} to the instrumentation amplifier. A programmable reference level gives the flexibility of moving the amplifier's output baseline up or down to match the range of certain test signals such as haversines which are unidirectional. A 12-bit DAC (DAC7311) with low-power consumption and small footprint was used to implement both DAC1 and DAC2.

A 3-axis accelerometer was included in the telemetry unit to capture motion information. The motion information will be used to estimate the degree of exercise performed by the subject. The MMA8453Q accelerometer was employed due to its small size (3 mm $\times$ 3 mm $\times$ 1 mm) and very low power consumption.

A wireless inductive battery charger was also included on the design to enable full implantation of the unit. The charger is composed by a coil and a capacitor C_T, a full-wave rectifier and the LTC4054 battery charger. A small and rechargeable lithium-polymer battery with a capacity of 45 mAh is used to power up the telemetry unit. The voltage level of the battery is monitored by the microcontroller by means of the $R_a - R_b$ resistive voltage divider. The battery voltage level is sent to the base station in every transmitted radio packet. Thus, the end user can be alerted when the battery is running

low and can recharge it.

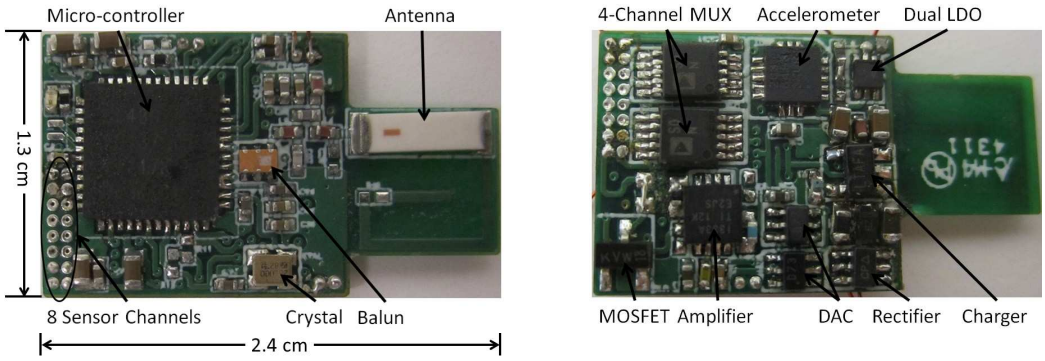
The inductive charger works at a frequency of 13.5 MHz. This frequency was chosen for two reasons: i) it is low enough to penetrate tissue [4] and ii) it is used in the ISO-15693 RFID standard and commercially available RFID readers can be used to charge the unit [19]. The frequency of the charger is tuned by setting the value of capacitor C_T.

A dual output low-dropout voltage regulator (LDO) was employed to provide a stable supply voltage to the analog and digital components of the telemetry unit. The dual output LDO allows to power down portions of the telemetry unit hardware to reduce power consumption when the unit is forced to enter into a *deep-sleep* power down mode or when the battery voltage has dropped below 2.9 V.

In the deep-sleep mode the analog front-end (amplifier, DACs and MUX) of the sensor as well as the accelerometer are turned off, the microcontroller is put into a low-power mode and the radio is turned off. Every three minutes the microcontroller wakes up, turns its radio on, transmits a status packet and listens for possible response from the base station. If no response is received it goes back to deep-sleep mode. On the other hand, if a response from the base station is received the unit exits the deep-sleep mode and proceeds to read and transmit data from its input channels. The deep-sleep mode is designed to minimize power consumption when the unit is not being used to collect strain or motion information.

A 4-layer printed circuit board (PCB) to host all the electronic components was designed and fabricated. The PCB with mounted components is shown in Figure 31.

Special effort was made in the PCB design to minimize noise coupling into the analog signal chain. Likewise, special efforts were made to minimize the size of the board. The PCB measures 2.4 cm $\times$ 1.3 cm. To reduce the number of discrete components needed by the radio, a balun from Johanson Technology (0896BM15A0001) was employed in the impedance matching network. A 915 MHz chip antenna from Johanson Technology (0915AT43A0026) and a small 26 MHz crystal oscillator were also employed to reduce board space.



(a) (b)

Figure 31: Telemetry Unit. (a) Top side. (b) Bottom side.

3.7.1 Radio Communications

The program running on the microcontroller was written in C and has an interrupt-driven architecture. The ADC conversion rate is set by an internal timer and it can be changed according to the application requirements by reprogramming the timer. The conversion rate of the ADC is given by:

$$f_{conv} = \frac{f_{clk}}{\text{TA}} \tag{3.7}$$

where, f_{clk} is the microcontroller's clock frequency and is set to 500 kHz and TA is the time period. Once enough samples have been collected a radio packet is transmitted. The packets have a payload fixed length of 60 bytes. The payload format of packets transmitted by the telemetry unit is shown in figure 32.

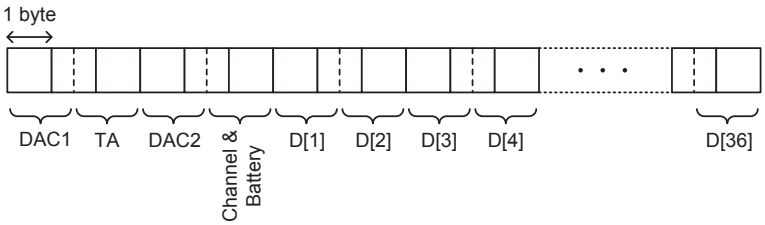


Figure 32: Format of the radio packets' payload.

Since the integrated ADC's resolution is 12-bits, the packet payload is divided into 12-bit-long units of information. Each packet contains the values of the two DACs, the timer period TA, the channel number being sampled, the battery voltage level and 36 data points. Each data point is equal to the digital conversion of the voltage V_{OUT}. Hence, the packet transmission period is equal to:

$$T_p = \frac{36}{f_{conv}} = \frac{36 \times \text{TA}}{f_{clk}} \tag{3.8}$$

Besides the payload a radio packet includes other fields such as preamble, synchronization, address, length and CRC yielding a total packet length of 576 bits. Considering that the radio transmission rate is set ot 75 kbps,transmitting a radio packet takes 7.7 ms. After the transmission of every packet the radio transceiver is programmed to switch to reception mode and listens for a arriving packet from the base station for 31.2

ms. Therefore, the minimum packet transmission period, $T_{p_{min}}$ is 7.7 ms + 31.2 ms = 38.9 ms. Considering the minimum packet transmission period and using (3.8) yields a maximum ADC sampling rate of 925 Hz which is divided among the 8 channels giving a maximum sampling rate of 115 Hz/channel. This sampling rate is more than enough for the target application. If higher sampling rates are needed less number of channels would have to be read. The minimum sampling rate is set by Nyquist rate. We conducted *ex vivo* tests with a 3 Hz haversine force applied to a mouse bone. Thus, the sampling rate in the *ex vivo* tests can be as low as 6 Hz.

Radio transmissions are the most power-expensive operation performed by the telemetry unit. During transmission the radio transceiver consumes 18 mA for a power output of 0 dBm. In reception mode the transceiver consumes 16 mA of current [69]. To reduce power consumption due to radio communications, the radio transceiver is turned off when it is not being used as illustrated in figure 33.

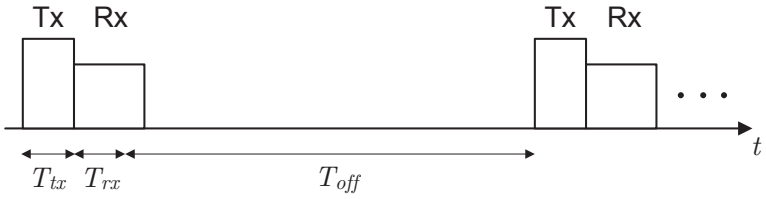


Figure 33: Transmission (TX) and reception (RX) timing diagram.

Thus, the average current consumption due to radio communications is given by:

$$I_{avg} = \frac{T_{tx}I_{tx} + T_{rx}I_{rx}}{T_{tx} + T_{rx} + T_{off}} \tag{3.9}$$

where, T_{off} is the time the radio transceiver remains off, $T_{tx} = 7.7$ ms, $T_{rx} = 31.2$ ms, I_{tx} is the current consumption during transmission and I_{rx} is the current consumption during reception. Notice that $T_{tx} + T_{rx} + T_{off} = T_p$. Combining this result with (3.8) yields the following relationship:

$$\begin{aligned} I_{avg} &= f_{conv}\left(\frac{T_{tx}I_{tx} + T_{rx}I_{rx}}{36}\right) \\ &= (Nf_s)\left(\frac{T_{tx}I_{tx} + T_{rx}I_{rx}}{36}\right) \end{aligned} \tag{3.10}$$

where, N is the number of channels being scanned and f_s is the sampling rate per channel.

Thus, the average power consumption due to radio communications is directly proportional to the conversion rate. Figure 34 shows the average power consumption predicted by the model in (3.10) in which an additional 1.4 mA has been added to account for the current consumption due to the micro-controller, voltage regulation and the analog signal chain. The figure shows the average current consumption as a function of the sampling rate per channel (f_s) for different number of active channels (N) when the radio transmission rate is set to 75 kbps and the transmission output power is 0 dBm.

According to the battery's manufacturer if 4.0 mA of current are continuously drawn from the battery, its voltage will drop to 3.0 V after approximately 12 hours. Thus, from figure 34 we conclude that, when transmitting at 75 kbps and 0 dBm, to collect 12 hours of continuous data using all 8 channels, the sampling rate needs to be about 18 Hz each. Alternatively, if only one channel is in use, sampling at a rate of 33 Hz allows the battery to last a whole day without recharging. A typical sampling rate per channel is

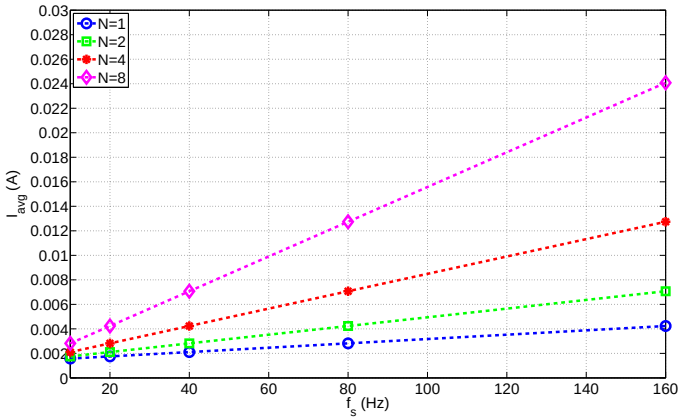


Figure 34: Average current consumption of the telemetry unit for different number of channels being read (N) and the sampling rate per channel (f_s). Radio transmission rate is set to 75 kbps and the transmission output power is 0 dBm.

between 3 to 5 Hz. Hence, the telemetry unit can run for 24 hours of continuous operation.

Reducing the transmission rate to 38 kbps further decreased packet loss. At this rate it was observed from figure 35 that for the battery to lasts 12 hours with all 8 channels in use, sampling rate needs to be at 15 Hz each, and for 24 hours, one channel needs to be sampled at 28 Hz.

3.7.2 Acquired Micro-Strain Data

We tested the unit the same way as the previous versions. This setup was tested by gluing the strain gauge to a surgically removed bone of a mouse and placing it on the Bose ElectroForce 3200 load test instrument. A Vishay EA-06-015DJ-120 strain gauge of nominal resistance of 120Ω and gauge factor(GF) of $2.07 \pm 2\%$ is used. The adhesive used was M-bond 2000 from Vishay-Micro-Measurements. The strain sensed by the strain

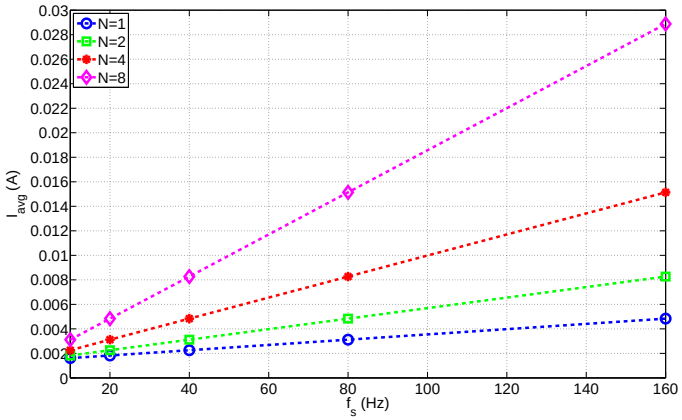


Figure 35: Average current consumption of the telemetry unit for different number of channels being read (N) and the sampling rate per channel (f_s). Radio transmission rate is set to 38 kbps and the transmission output power is 0 dBm.

gauge is acquired by the Vishay Micro-Measurement 7000 data acquisition system. Data is collected by applying a sinusoidal force to the bone. The magnitude of the force is varied from 1 N to 3 N and its frequency is varied from 0.5 Hz to 3 Hz.

The strain gauge was then disconnected from the data acquisition system and connected to the telemetry unit. The same sinusoidal force was reapplied to the bone of the mouse. The strain data was transmitted by the telemetry unit to the base station. The development board for CC430F5137 is employed as the base station which is connected to the PC via FTDI serial-to-USB cable. A graphical user interface(GUI) was designed to allow easy data collection and system configuration. The GUI can be used to remotely change telemetry unit settings such as sampling rate, transmission power, the baseline of the amplifier output, and select sensor channel. The calibration routine can also be triggered from the GUI. The GUI connects to the base station via a COM port and has the

options to start and stop readings and to save data into a file. The GUI is equipped with a digital filter to filter the incoming raw data and plots them together in real-time.

Figure 36 shows readings from both the acquisition system and the telemetry unit. The figure shows the strain reading when a 3 N peak-to-peak force at 2 Hz was applied. For both, the recorded strain ranged from -3000 to 0 $\mu\epsilon$ making the telemetry unit a efficacious replacement of the bulky bench top setup.

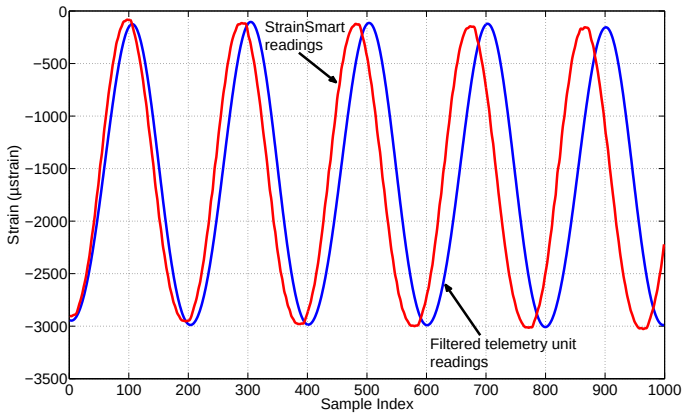


Figure 36: StrainSmart® readings vs Telemetry unit filtered readings.

3.7.3 Current Consumption

The on-chip timer of the micro-controller triggers the analog-to-digital conversion of the strain data. Varying the period of the timer sets the sampling frequency of the strain data. With the master clock of the microcontroller operating at 500 kHz, the radio transmission baud rate 75 kbps, and radio transmission power of 0 dBm, we sampled one channel at frequency ranging from 10 Hz to 160 Hz. Figure 37 shows that at 10 Hz, current consumed by the telemetry unit is only 1.75 mA which increases to 4 mA for

sampling frequency of 160 Hz. The figure also shows that during sleep the unit consumes only 0.4 mA of current. During sleep the micro-controller puts the radio core into power down mode, turns off the MOSFET switch, and disables the LDO output shutting down the amplifier, the two DACs, the two MUXs and the accelerometer.

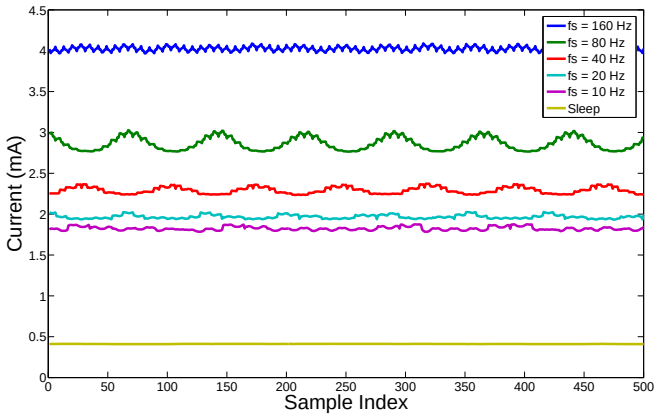


Figure 37: Telemetry unit current consumption.

3.7.4 Performance Through Tissue Phantom

Inductive coupling theory is the underlying principle for wireless power charging which is an important feature for any implantable telemetry device. Every moving charges (electrons in wires or in a vacuum), i.e. flow of current, is associated with a magnetic field (figure 38) in accordance to the shape of the conductor. The generic equation:

$$\sum I = \oint \vec{H}\vec{ds}$$

for the magnetic field strength H is independent of material properties of space and hence could be used for different types of conductor.

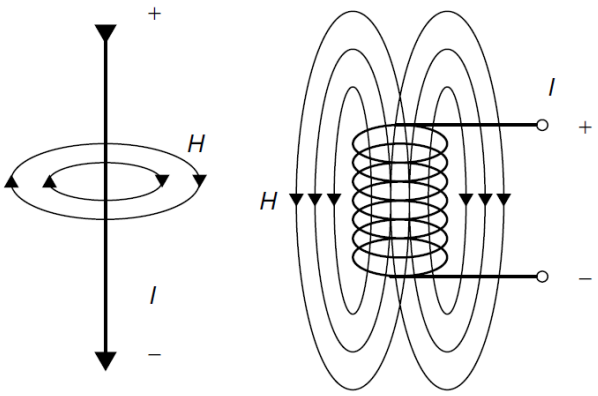


Figure 38: Lines of magnetic flux around a current-carrying conductor and a current-carrying cylindrical coil.

The magnetic field strength H for a short cylindrical coil antenna shown in figure 39 similar to the type used in the antennas of inductively coupled RFID systems is given as equation:

$$H = \frac{INr^2}{2\sqrt{r^2 + x^2}}$$

Where N is the number of winding, R is the circle radius r and x is the distance from the center of the coil in the x direction. It could be inferred from above equation that as the distance x increases the magnetic field strength H decreases.

The changing magnetic field created by one circuit (the primary) can induce a changing voltage and/or current in a second circuit (the secondary). The mutual inductance of two circuits describes the size of the voltage in the secondary induced by changes in the current of the primary and is the basic principle behind most inductively coupled passive RFID systems.

Telemetry unit, with the coil was place on top of the small battery and the whole

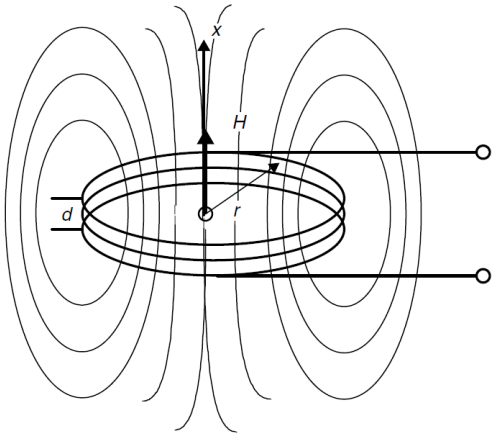


Figure 39: The path of the lines of magnetic flux around a short cylindrical coil.

setup was wrapped with polymide film tape as they are proven to be biocompatible [64]. The diameter of the coil is 2 cm and it has 20 turns. The length of the coil from the edge of the PCB is 3 cm. With the battery, the dimensions of the setup is 2.4 cm $\times$ 4.3 cm $\times$ 0.5 cm. The setup was then placed in a tissue phantom created by using gelatin and NaCl. Gelatin gels were normally used as a model for the soft tissue because of the presence of gel in the human tissue and ease of implementation [53], [83]. The artificial skin model using the gelatin gel is easy to deal with and care. Moreover, it is an alternative measurement method of an human test. In 500 ml of water, 15 grams of gelatin was added and mixed with 1.2 grams of NaCl and brought to a boil. After pouring the mixture into a container, the telemetry unit was placed 1 cm below the surface of the phantom. Then the setup was refrigerated over night to solidify. The setup is depicted in figure 40.

We compared the current being delivered to the 3.7 V, 45 mAh battery while charging it through air and 1 cm between the coils and when the unit was 1 cm deep into the tissue phantom. This was done by measuring the voltage drop across a 1 Ω current sense

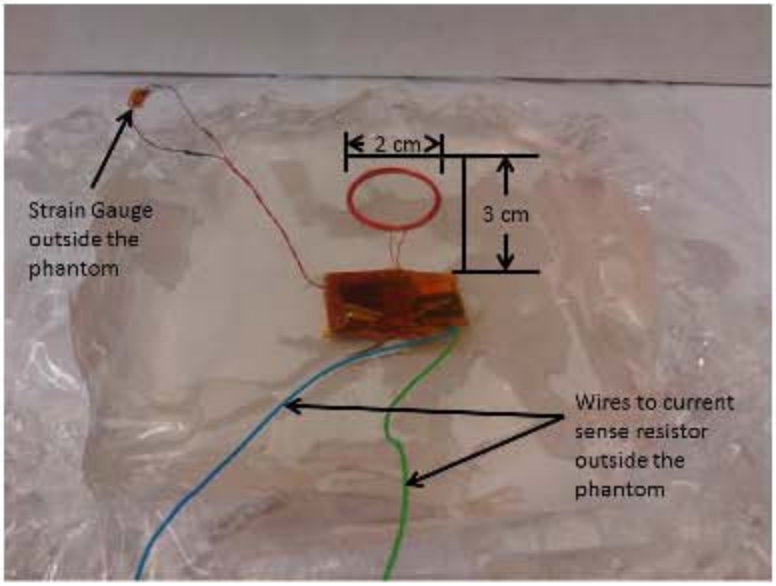


Figure 40: Telemetry unit in tissue phantom.

resistor (shunt resistor [20]) connected between the charger and the battery. Figure 41 shows the schematic diagram of the circuit designed to measure the current to the battery. During charging the unit was put into deep-sleep mode. Voltage, V_1 and V_2, at each node of the resistor was fed into an amplifier (AD620) with gain equal to 100 and a reference voltage of 1 V. Therefore, the current being delivered to the battery is calculated by:

$$I_{bat} = ((V_1 - V_2) \times 100 - V_{ref})/1\Omega$$

Figure 42 shows that when implanted, the battery receives 1.3 mA of current compared to 3.4 mA when the medium is air between the coils.

The voltage level of the battery is monitored by the microcontroller by means of the $R_a - R_b$ resistive voltage divider. The battery voltage level is sent to the base station in every transmitted radio packet. Thus, the end user can be alerted when the battery is running low (2.9 V) and can recharge it. When the battery is at 2.9 V, it is 20%

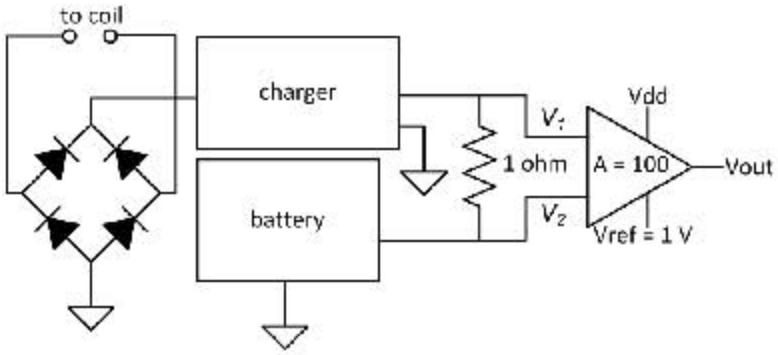


Figure 41: Circuit to measure current delivered to battery while charging.

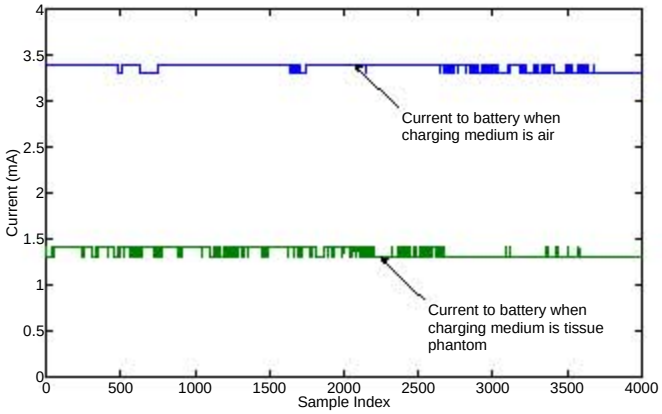


Figure 42: Current delivered to telemetry unit during wireless charging.

discharged, which means its capacity is lowered to about 33 mAh. In order to charge it to full capacity, about 12 mAh needs to be put back in to the battery. Considering this and the amount of current received, it takes 3.5 hrs to charge through air and 10 hrs to charge when implanted. This can easily be done over night when the unit is not being used and is in deep-sleep.

Packet loss at transmission rates of 75 kbps and 38 kbps and transmission power 0 dBm, -12 dBm and -30 dBm were examined when the setup was placed at a distance of 50 cm, 1 m and 2 m away from the base station. The experiment was repeated after removing the unit from the phantom, with air as the medium. Table 18 shows the results through phantom and Table 19 shows the results through air. It can be seen that the tissue phantom increases packet loss when we go from 38 kbps, 0 dBm and 50 cm to 75 kbps, -30 dBm and 2 m.

Table 18: Tissue phantom readings. Percentage of packets received for different transmission power and distance.

Tx power	38 kbps			75 kbps		
	50 cm	1 m	2 m	50 cm	1 m	2 m
0 dBm	100	100	100	92.60	90.94	85.44
-12 dBm	100	100	98.6	63.96	60.90	48.34
-30 dBm	98.96	96.04	84.50	54.04	24.66	10.90

Table 19: Air readings. Percentage of packets received for different transmission power and distance.

Tx power	38 kbps			75 kbps		
	50 cm	1 m	2 m	50 cm	1 m	2 m
0 dBm	100	100	100	94.54	92.92	92.28
-12 dBm	100	100	98.6	92.18	90.68	90.20
-30 dBm	99.24	99.66	98.86	69.16	68.64	57.22

Future work involves using the telemetry unit in real live subjects in labs.

CHAPTER 4

WIRELESS SURFACE ELECTROMYOGRAPHY (EMG) SENSOR

This chapter presents the design and validation of EMG sensor node. The prototype is created using commercially available off-the-shelf components. Extra care was taken to make the node seamlessly wearable. We also propose a robust propriety wireless network protocol that allows the use four such nodes at once. Details of the algorithm running on the nodes and the base station are presented. Performance comparison with available industry EMG sensor is also drawn.

4.1 Literature Survey

Electromyography is a method of detecting muscle activity. In particular, EMG is applied to the study of skeletal muscle. The skeletal muscle tissue is attached to the bone and its contraction is responsible for supporting and moving the skeleton. The contraction of skeletal muscle is initiated by impulses in the neurons to the muscle and is usually under voluntary control. The methods relies on the change of membrane potential of the muscle cells with muscle activity. This can occur both in spikes when the muscle is stimulated or constantly when the muscle contraction is spasmodic.

There are many applications for the use of EMG. EMG is used clinically for the diagnosis of neurological and neuromuscular problems. It is used diagnostically by gait

laboratories and by clinicians. EMG is also used in many types of research laboratories, including those involved in biomechanics, motor control, neuromuscular physiology, movement disorders, postural control, and physical therapy [2]. EMG can also be used to sense muscular activity that does not translate into movement. This feature allows capturing motionless gestures without being noticed and sees its use in hands free applications [67]. [60] have shown that in interactive computer gaming, EMG along with other sensors can be used to replace hand held joypad and joystick. In this EMG can provide more intuitive human movement as compared to traditional controllers. At the NASA Arms Research Center at Moffett Field, California, the extension of the Human Senses Group uses bio-control systems interfaces. These NASA researchers have used EMG signal to substitute for mechanical joysticks and keyboards. As an example, they developed a method for flying a high-fidelity flight simulator of a transport aircraft using EMG based joystick [75]. EMG has also been demonstrated to be useful in non-voice communication [41]. This serves well for people without or damaged vocal chords. [50] employed EMG signals of shoulders to control electric-power wheelchair to assist people with spinal chord injury.

In order to measure and record potentials and, hence, currents in the body, it is necessary to provide some interface between the body and the electronic measuring apparatus. Biopotential electrodes carry out this interface function. In any practical measurement of potentials, current flows in the measuring circuit for at least a fraction of the period of time over which the measurement is made. Ideally this current should be very small. However, in practical situations, it is never zero. Biopotential electrodes must

therefore have the capability of conducting a current across the interface between the body and the electronic measuring circuit. The silver/silver chloride (Ag/AgCl) electrode is a practical electrode that approaches the characteristics of a perfectly nonpolarizable electrode and can be easily fabricated in the laboratory.

EMG can be measured both non-invasively on the skin surface above the muscle or invasively by needles. Table 20 below lists the advantages and disadvantages of using needle or surface electrodes.

Table 20: EMG electrode types

Inserted	Surface
Advantages	
- Extremely sensitive - Record single muscle activity - Access to deep musculature - Little cross-talk concern	- Quick, easy to apply - No medical supervision or required certification - Minimal discomfort
Disadvantages	
- Extremely sensitive - Requires medical personnel and certification - Repositioning nearly impossible - Detection area may not be representative of entire muscle	- Generally used only for superficial muscles - Cross-talk concerns - No standard electrode placement - May affect movement patterns of subject - Limitations with recording dynamic muscle activity

During muscle activity the membrane potential change to approximately 10 mV. Since EMG signal suffers from electrical noise, a differential amplifier with high input impedance is typically employed. One source of noise is the other surrounding electronics

and unfortunately can not be removed. It can only be reduced by carefully selecting high quality components. Other sources of noise include electromagnetic radiation from radio stations, electrical wires etc and is also not easy to remove. Motion artifacts also effect EMG signals. They are generated from using wires and cable and not preparing the skin properly before placing them on to the skin. The bandwidth of EMG signal is 0 to 500 Hz, however, noise from mentioned sources range between 0 - 20 Hz, and dominant line frequency of 60 Hz [13]. A high pass filter after amplification along with an anti-aliasing low pass filter is recommended as part of the analog front end for EMG devices. A notch filter can be employed to remove the 60 Hz line noise but is not advised as it falls in the dominant EMG bandwidth.

4.2 System Design

The EMG node is designed around the CC430 microcontroller from Texas Instruments. This microcontroller was chosen because it integrates a range of peripherals such as a 12-bit ADC, a 16-bit timer and a 915 MHz radio transceiver. The integration of these peripherals along with a low-power 16-bit CPU on a single chip that measures 8 mm $\times$ 8 mm enables the development of a compact wireless sensors. Figure 43 shows the block diagram of the EMG node.

Separate LDOs are employed to isolate the analog front-end amplifier from the rest of the circuit. LDO1 can be turned on or off from the microcontroller. The unit also features a battery charger, therefore it can be cased in with a rechargeable battery.

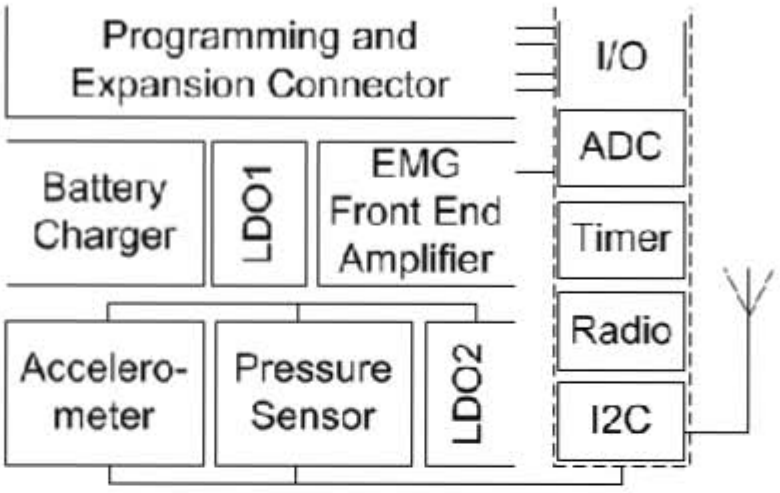
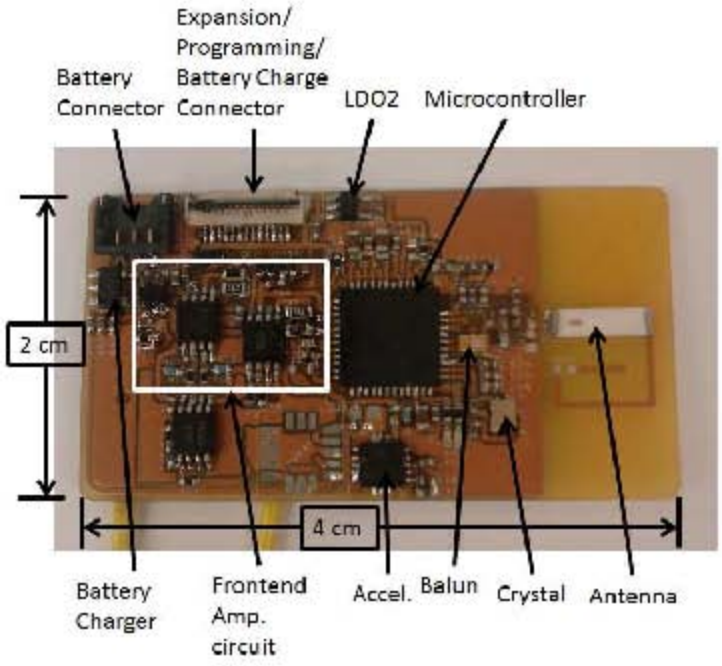


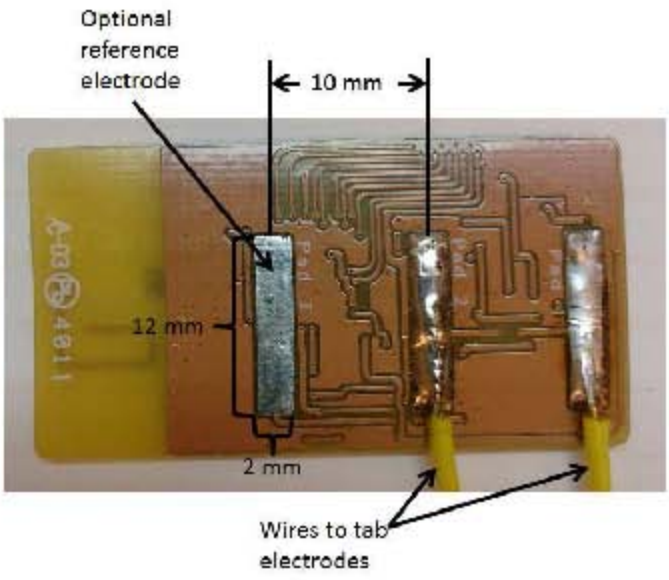
Figure 43: Block diagram of the EMG node.

A 3-axis accelerometer is included to capture motion information. The MMA8453Q accelerometer was employed due to its small size (3 mm $\times$ 3 mm $\times$ 1 mm) and very low power consumption. Place for another sensor, for example atmospheric pressure, that can be programmed by I2C is also provided. A two layer PCB is designed to house all the components and measures 4 cm $\times$ 2 cm. All the electronic components are place on the top side of the PCB leaving room for just the electrodes at the bottom. Each electrode measures 2 mm $\times$ 12 mm and are 10 mm apart. Figure 44 shows PCB with the soldered components.

The front end amplifier circuit comprises of a precision differential amplifier (INA321) chosen for its high input impedance and very low power consumption. The output of the differential amplifier is fed into a unity gain anti-aliasing low pass filter with cutoff frequency of 500 Hz. It is a Sallen-Key filter with a Butterworth response



(a)



(b)

Figure 44: EMG node. (a) Top side. (b) Bottom side.

characteristics. Figure 45 depicts the schematic diagram of the front end amplifier. In the figure resistors R1 and R2 sets the gain of the differential amplifier to 1000 according to equation:

$$Gain = 5 + 5(R2/R1)$$

and R3 and R4 sets the gain of the filter.

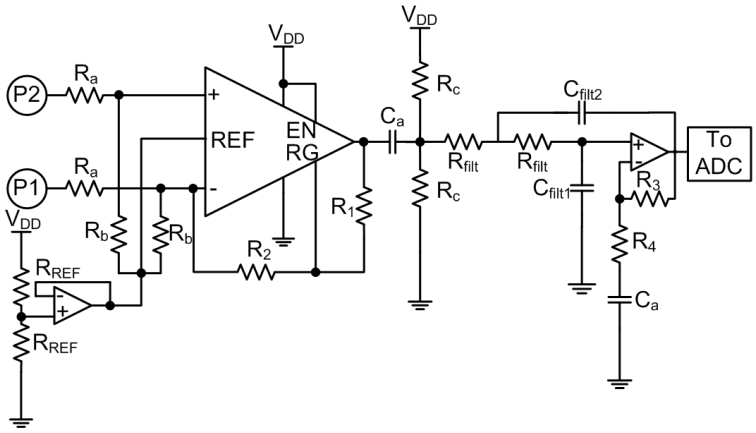


Figure 45: Schematic diagram of the EMG front end amplifier.

The development board for CC430F5137 is employed as the base station which is connected to the PC via FTDI serial-to-USB cable. A graphical user interface(GUI) was designed to allow easy data collection. The GUI connects to the base station via a COM port and has the options to start and stop readings and to save data into a file.

4.3 Base station and EMG node program algorithms

A network is designed to collected EMG data from four nodes at once. The base station acts as the central unit and allocates time slots to the nodes when they join the

network. During this time slot is only when that specific node is allowed to transmit its data after a SYNC command is received from the base station. Each command sent or received in the network has its first byte a ”C” and is checked to make sure a command is received. Figure 46 shows the format of the commands sent from the base station and the replies from the nodes.

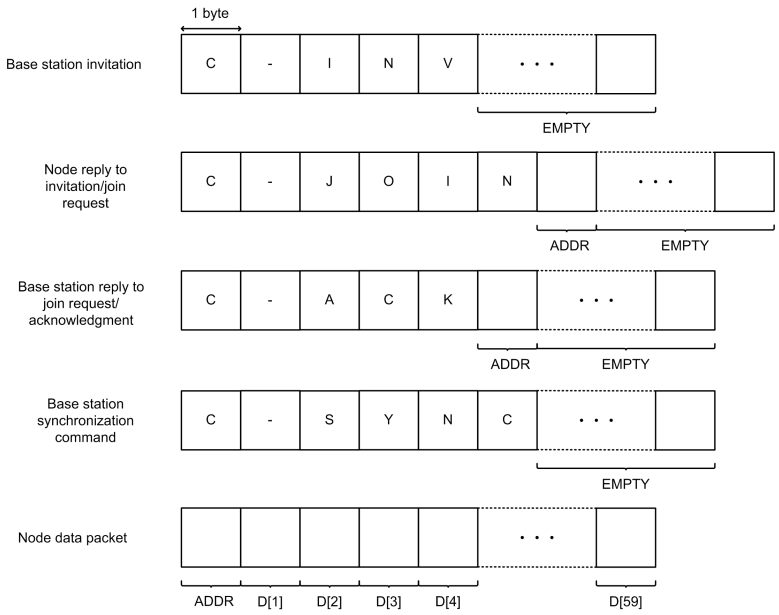


Figure 46: Packet format of the base station commands and node replies.

The program running on the base station is written in C language and has an interrupt driven architecture. The internal timer controls the transmission of commands to invite, respond to join requests and synchronize the timers of each connected node. Only after when four nodes have joined the network, the base station stop inviting more nodes and transmits SYNC commands. In the radio interrupt the base station checks for join requests, however after all four nodes have joined the radio interrupt processes the

data received from the nodes. The flow charts in figures 47 and 48 describe what the base station processes during its timer and radio interrupts respectively.

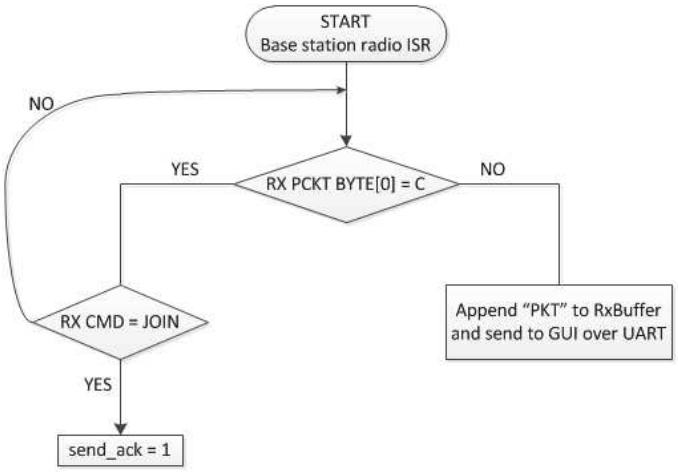


Figure 47: Base station radio interrupt.

The architecture of the program running on each EMG node is also interrupt driven. Just like the base station, the timer and the radio are the main interrupts. The flow charts in figures 49 and 50 describe what the node processes during each interrupts. In the radio interrupt the node looks for the three commands sent from the base station and acts accordingly. The timer interrupt triggers the sampling of the ADC at 1 KHz and also facilitates the delays needed when SYNC is received, so that the node transmits data only in its own time slot and prevent collision.

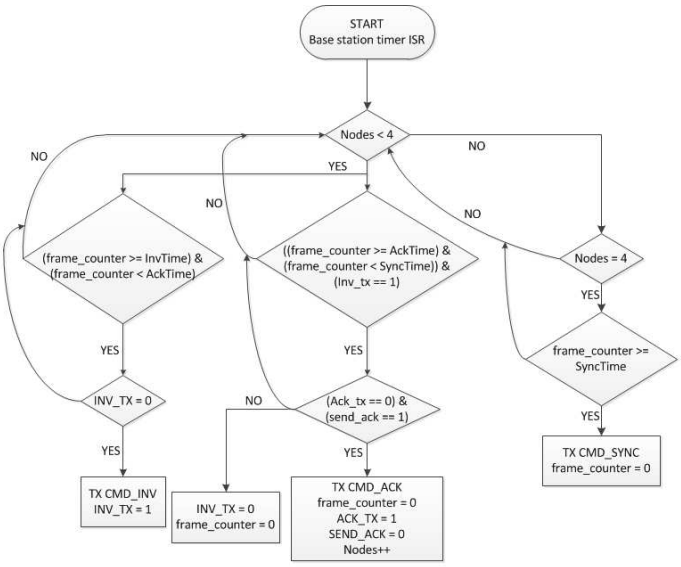


Figure 48: Base station timer interrupt.

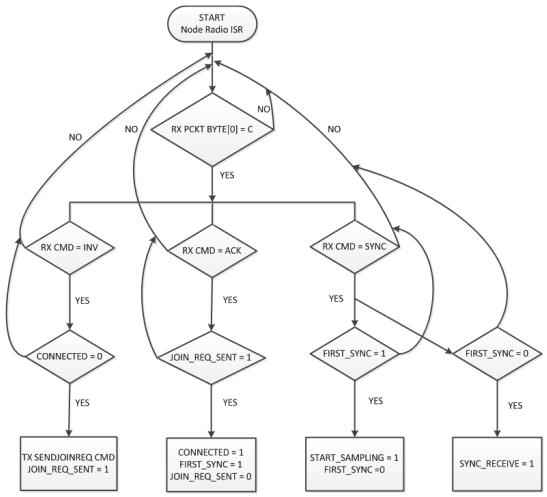


Figure 49: Node radio interrupt.

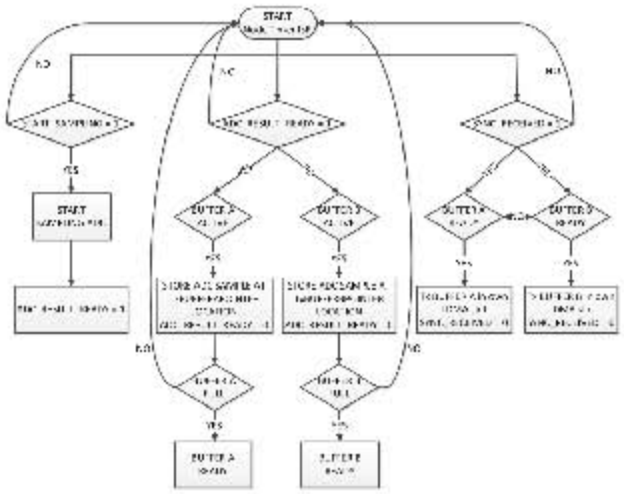


Figure 50: Node timer interrupt.

4.4 EMG Network

Figure 51 shows the activity of the base station when a node tries to join its network. In the figure t_{TX}, time to transmit a packet is 9 ms (discussed below), time between invitation and acknowledgment, t_{ACK} is 18 ms and time between two invites, t_{INV} is 36 ms. During this process, the base station keeps a count of how many nodes have joined its network. Once that number reaches four, only SYNC command is transmitted.

The throughput of the network needs to to 1 KHz as the bandwidth of EMG signal is 0 - 500 Hz. To match this, each node samples data at 1 KHz as well. The internal timer interrupt is programmed to trigger the ADC sampling. Two buffers of 60 bytes are used to store the sampled data to be transmitted alternatively. Each transmitted packet contains these 60 byte sized samples of EMG data, plus preamble, sync word, CRC bits and RSSI,

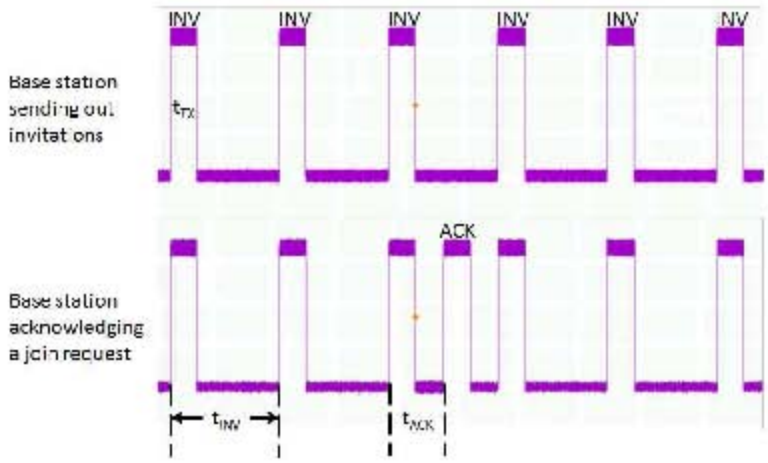


Figure 51: Base station replying to join request of a node.

for a total of 576 bits. Considering radio transmission baud rate of 64.8 kbps, it takes about 9 ms to transmit a single packet.

Figure 52 depicts the working of the network. In the figure t_6 is 9 ms, and the time it takes the base station to process a received packet, t_7 is 77 ms. t_1 is the time Node1 waits after receiving SYNC from base station and is 3 ms, t_2 is the wait time for NODE2 and is 15 ms, t_3 is the wait time for NODE3 and is 27 ms, and t_4 is the wait time for NODE4 and is 39 ms. t_5 is the time between two SYNCs 60 ms as well as the time between data transmitted from one node. 60 samples are transmitted every 60 ms making the throughput of the network to be 1 KHz, which is ideal for EMG signal whose bandwidth is 0 - 500 Hz.

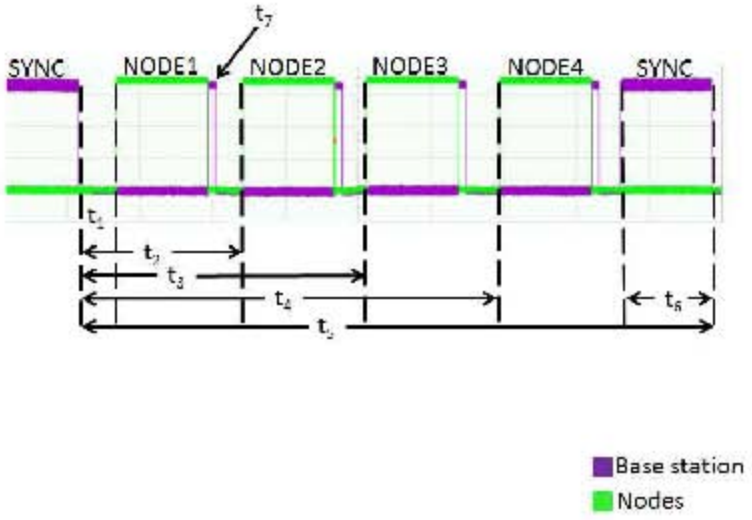


Figure 52: EMG network showing SYNC from base station and data from the nodes.

4.5 EMG data collection

We tested the performance of the EMG sensor against Delsys Inc. EMG sensors. Tab electrodes connect to our nodes were pasted on to the forearm of the user. Nodes of Delsys system was attached close to the tab electrodes on the same muscle. This setup was prepared to simultaneously collect data using both systems. Figure 53 shows the two connected to the forearm of the user.

The user was asked to move their wrist up and down, there by flexing the forearm muscles. This movement was repeated for few cycles at a slow pace to be able to capture EMG data. It can be seen from figure 54 that EMG collected by the designed board while operating in the network is comparable to the EMG collected by the Delsys system.

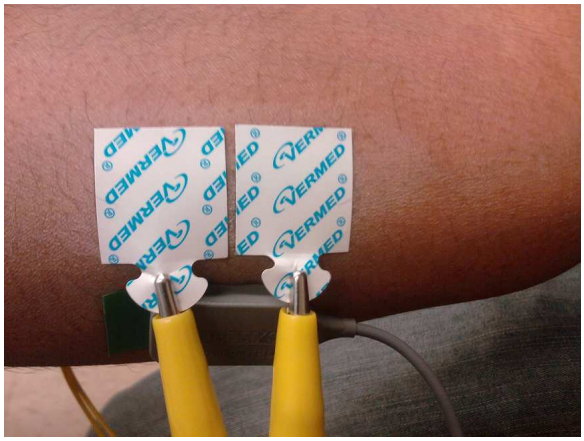


Figure 53: Tab electrodes connected to designed EMG node and Delsys Inc. Node attached to the forearm of the user.

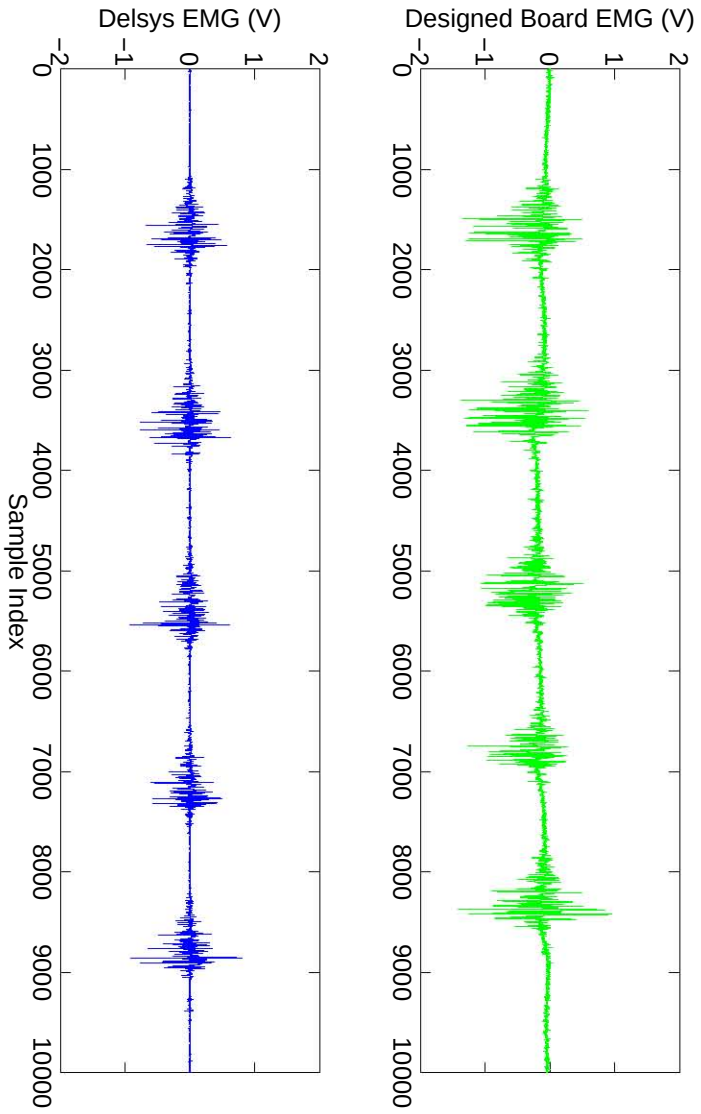


Figure 54: EMG data using designed board vs Delsys system.

CHAPTER 5

CONCLUSIONS

Body area sensor networks are being researched extensively. These comprise of intelligent nodes that can be seamlessly worn or implanted while taking physiological data of the user. These nodes communicate with base stations connected to PCs for further processing of the collected data. This research field is challenging but the results envisioned are not impossible. Choosing from available state of art hardware and signal processing algorithms are the areas that are being explored. This dissertation delivers solutions that are comparable to industry standards and does this by providing detailed design and validation techniques of such wearable and implantable sensors. These sensors can operate by themselves or can be integrated into a network of several such sensors.

We first presented a prototype to capture body motion and then classify the gestures performed. Two techniques were used to do so. Namely inertial position and acoustic positioning. For inertial based position we used 3-axis accelerometers and 2-axis gyroscopes. For acoustic based positions, ultrasound speakers and microphones were employed. It was seen that results gathered by ultrasound positioning are erroneous as they are effected by spatial and temporal changes in room temperature and air movements. Unfortunately, these errors can not be rectified as they are caused by air turbulence. Therefore, we chose to use inertial based gesture recognition.

Several algorithms have been proposed to classify gestures. In this dissertation

we compared linear and non-linear classifiers. Fisher Linear Discriminant Analysis was employed as the linear classifier and artificial neural networks were the non-linear classifiers. It was found that neural networks perform the best. In them too, the Time Delay neural networks (TDNN) faired to the be the most successful, classifying gesture at near perfection. We also proposed using them to reduce transmission power in a wireless sensors, by reducing the number of bits of the acceleration data being sent and transmitting less often. In the most detrimental scenario of transmitting 4 bits at 5 Hz, we achieved 90% classifying rates with TDNNs.

Second prototype presented in this dissertation are telemetry units for measuring strain on bones. We went through several versions to reduce size and power consumption of the telemetry units. Each version was shown to be able to replace the existing bulky bench top load force system, which requires the test subject to be sedated and to be immobile. Our systems were tested in *ex-vivo* setup but have been shown to be suitable for *in-vivo* use too. For *in-vivo* testing, the unit was placed in a tissue phantom. We showed that using inductive coupling we can charge the battery of the unit through the phantom in 10 hrs. This can easily be done over night when the unit is not being used and is in deep-sleep.

Finally, we present a network of four EMG collecting sensors. This network is robust to body motion and the EMG data collected is comparable to industry standards. The base station used for the network orchestrates the collection of data from the four nodes. Upon turning on, the base station invites nodes to join its network. When a total of four nodes have joined, the base switches to sending synchronization commands to all of it

connected nodes. Individual nodes are assigned time slots when they join the network and are made to transmit their collected data only in that time slot to avoid collisions. Upon receiving the sync commands, the nodes use their internal timer to wait for their turn. Since the EMG bandwidth is 0 - 500 Hz, the throughput of the network was programmed to have a sampling frequency of 1 KHz.

REFERENCE LIST

[1] Abdel Hady, M., and Schwenker, F. Decision templates based RBF network for tree-structured multiple classifier fusion. Springer, pp. 92–101.

[2] Airaksinen, M., Kankaanpää, M., Aranko, O., Leinonen, V., Arokoski, J., and Airaksinen, O. Wireless on-line electromyography in recording neck muscle function: a pilot study. *Pathophysiology 12*, 4 (2005), 303–306.

[3] Aminian, K., and Najafi, B. Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications. *Computer Animation and Virtual Worlds 15*, 2 (2004), 79–94.

[4] Astrin, A., Huan-Bang, L., and Kohno, R. Standardization for body area networks. *IEICE Transactions On Communications 92*, 2 (2009), 366–372.

[5] Aylward, R., and Paradiso, J. Sensemble: a wireless, compact, multi-user sensor system for interactive dance. In *Proceedings of the 2006 conference on New interfaces for musical expression* (2006), IRCAM Centre Pompidou, pp. 134–139.

[6] Baillot, Y., Davis, L., and Rolland, J. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augumented reality* (2001), 67.

[7] Bishop, C. *Pattern recognition and machine learning (information science and statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[8] Chambers, G., Venkatesh, S., West, G., and Bui, H. Hierarchical recognition of intentional human gestures for sports video annotation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (2002), vol. 2, IEEE, pp. 1082–1085.

[9] Chan, Y., and Ho, K. A simple and efficient estimator for hyperbolic location. *Signal Processing, IEEE Transactions on 42*, 8 (1994), 1905–1915.

[10] Claes, L., and Cunningham, J. Monitoring the mechanical properties of healing bone. *Clinical Orthopaedics and Related Research® 467*, 8 (2009), 1964–1971.

[11] Cover, T. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, 3 (1965), 326–334.

[12] De Jong, W., Koolstra, J., Van Ruijven, L., Korfage, J., and Langenbach, G. A fully implantable telemetry system for the long-term measurement of habitual bone strain. *Journal of biomechanics 43*, 3 (2010), 587–591.

[13] De Luca, C. The use of surface electromyography in biomechanics. *Journal of applied biomechanics 13* (1997), 135–163.

[14] Duda, R., Hart, P., and Stork, D. *Pattern classification*. New York [etc.]: Wiley, 2001.

[15] Einsmann, C., Quirk, M., Muzal, B., Venkatramani, B., Martin, T., and Jones, M. Modeling a wearable full-body motion capture system. In *Wearable Computers,*

2005. Proceedings. Ninth IEEE International Symposium on (2005), IEEE, pp. 144–151.

[16] Fang, B. Simple solutions for hyperbolic and related position fixes. *Aerospace and Electronic Systems, IEEE Transactions on 26*, 5 (1990), 748–753.

[17] Farella, E., Benini, L., Riccò, B., and Acquaviva, A. Moca: A low-power, low-cost motion capture system based on integrated accelerometers. *Advances in Multimedia 2007*, 1 (2007), 1–1.

[18] Fawcett, T. An introduction to ROC analysis. *Pattern recognition letters 27*, 8 (2006), 861–874.

[19] Finkenzeller, K. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Wiley, 2010.

[20] Forghani-zadeh, H., and Rincon-Mora, G. Current-sensing techniques for DC-DC converters. In *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on* (2002), vol. 2, IEEE, pp. II–577.

[21] Foxlin, E., Harrington, M., and Pfeifer, G. Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 371–378.

[22] Foy, W. Position-location solutions by Taylor-series estimation. *Aerospace and Electronic Systems, IEEE Transactions on*, 2 (1976), 187–194.

[23] Giansanti, D., Maccioni, G., and Macellari, V. The development and test of a device for the reconstruction of 3-D position and orientation by means of a kinematic sensor assembly with rate gyroscopes and accelerometers. *Biomedical Engineering, IEEE Transactions on 52*, 7 (2005), 1271–1277.

[24] Graichen, F., Arnold, R., Rohlmann, A., and Bergmann, G. Implantable 9-Channel Telemetry System for In Vivo Load Measurements With Orthopedic Implants. *Biomedical Engineering, IEEE Transactions on 54*, 2 (2007), 253–261.

[25] Hampshire, J., Waibel, A., et al. A novel objective function for improved phoneme recognition using time-delay neural networks. *Neural Networks, IEEE Transactions on 1*, 2 (1990), 216–228.

[26] Hand, D., and Till, R. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning 45*, 2 (2001), 171–186.

[27] Hazas, M., and Ward, A. A novel broadband ultrasonic location system. Springer, pp. 299–305.

[28] Ho, K., and Chan, Y. Solution and performance analysis of geolocation by TDOA. *Aerospace and Electronic Systems, IEEE Transactions on 29*, 4 (1993), 1311–1322.

[29] Hoshen, J. The GPS equations and the problem of Apollonius. *Aerospace and Electronic Systems, IEEE Transactions on 32*, 3 (1996), 1116–1124.

[30] Johnson, C. *Process control instrumentation technology*. Prentice Hall, 2006.

[31] Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing 10*, 5 (2006), 285–299.

[32] Khan, A., Lee, Y., Lee, S., and Kim, T. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *Information Technology in Biomedicine, IEEE Transactions on 14*, 5 (2010), 1166–1172.

[33] Kim, C., and Lee, S. A digital chip for robust speech recognition in noisy environment. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on* (2001), vol. 2, IEEE, pp. 1089–1092.

[34] Kiss, R., Kocsis, L., Knoll, Z., et al. Joint kinematics and spatial–temporal parameters of gait measured by an ultrasound-based system. *Medical engineering & physics 26*, 7 (2004), 611–620.

[35] Kuncheva, L. Combining Pattern Classifiers: Methods and Algorithms (Kuncheva, LI; 2004)[book review]. *Neural Networks, IEEE Transactions on 18*, 3 (2007), 964–964.

[36] Kuncheva, L., Bezdek, J., and Duin, R. *Pattern Recognition 34*, 2 (2001), 299–314.

[37] Kuncheva, L., and Jain, L. Designing classifier fusion systems by genetic algorithms. *Evolutionary Computation, IEEE Transactions on 4*, 4 (2000), 327–336.

[38] Leong, T., Lai, J., Panza, J., Pong, P., and Hong, J. Wii want to write: An accelerometer based gesture recognition system. In *International Conference on Recent and Emerging Advanced Technologies in Engineering* (2009), pp. 4–7.

[39] Liu, Y., Gan, Z., and Sun, Y. Static hand gesture recognition and its application based on support vector machines. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on* (2008), IEEE, pp. 517–521.

[40] MacKay, D. A practical Bayesian framework for backpropagation networks. *Neural computation 4*, 3 (1992), 448–472.

[41] Manabe, H., Hiraiwa, A., and Sugimura, T. Unvoiced speech recognition using EMG-mime speech recognition. In *CHI'03 extended abstracts on Human factors in computing systems* (2003), ACM, pp. 794–795.

[42] Mäntyjärvi, J., Kela, J., Korpipää, P., and Kallio, S. Enabling fast and effortless customisation in accelerometer based gesture interaction. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia* (2004), ACM, pp. 25–31.

[43] Maynard, V., Bakheit, A., Oldham, J., and Freeman, J. Intra-rater and inter-rater reliability of gait measurements with CODA mpx30 motion analysis system. *Gait & posture 17*, 1 (2003), 59–67.

[44] Melnikoff, S., Quigley, S., and Russell, M. Speech recognition on an FPGA using discrete and continuous hidden Markov models. *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream* (2002), 89–114.

[45] Meyer, K., Applewhite, H., and Biocca, F. A survey of position trackers. 173–200.

[46] Milosevic, B., Farella, E., and Benini, L. Continuous Gesture Recognition for Resource Constrained Smart Objects. In *UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (2010), pp. 391–396.

[47] Misra, J., and Saha, I. Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomputing 74*, 1 (2010), 239–255.

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

[49] Moiz, F., Natoo, P., Derakhshani, R., and Leon-Salas, W. A comparative study of classification methods for gesture recognition using a 3-axis accelerometer. In *Neural Networks (IJCNN), The 2011 International Joint Conference on* (2011), IEEE, pp. 2479–2486.

[50] Moon, I., Lee, M., Chu, J., and Mun, M. Wearable EMG-based HCI for electric-powered wheelchair users with motor disabilities. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), IEEE, pp. 2649–2654.

[51] Morency, L., Sidner, C., Lee, C., and Darrell, T. The role of context in head gesture recognition. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE* (2006), vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1650.

[52] Motion. Motion Analysis Corp. http://www.motionanalysis.com, 2009.

[53] Mridha, M., and Ödman, S. Characterization of subcutaneous edema by mechanical impedance measurements. *Journal of investigative dermatology 85*, 6 (1985), 575–578.

[54] Murakami, K., and Taguchi, H. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (1991), ACM, pp. 237–242.

[55] NATIONAL INSTRUMENTS. *Tutorial: Measuring Strain with Strain Gauges*, 2012.

[56] Neto, P., Pires, J., and Moreira, A. Accelerometer-based control of an industrial robotic arm. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on* (2009), IEEE, pp. 1192–1197.

[57] Noraxon. Zebris Motion Analysis. http://www.noraxon.com/zebris/index.php3.

[58] Pantic, M., and Bartlett, M. Machine analysis of facial expressions. I-Tech Education and Publishing, 2007.

[59] Park, C., Liu, J., and Chou, P. Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* (2005), IEEE, pp. 398–403.

[60] Park, D., and Kim, H. Muscleman: Wireless input device for a fighting action game based on the EMG signal and acceleration of the human forearm.

[61] Principe, J., Euliano, N., and Lefebvre, W. *Neural and adaptive systems: fundamentals through simulations*. Wiley New York, 2000.

[62] Ravi, N., Dandekar, N., Mysore, P., and Littman, M. Activity recognition from accelerometer data. In *Proceedings of the national conference on artificial intelligence* (2005), vol. 20, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1541.

[63] Rehm, M., Bee, N., and André, E. Wave like an Egyptian: accelerometer based gesture recognition for culture specific interactions. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1* (2008), British Computer Society, pp. 13–22.

[64] Rousche, P., Pellinen, D., Pivin Jr, D., Williams, J., Vetter, R., et al. Flexible polyimide-based intracortical electrode arrays with bioactive capability. *Biomedical Engineering, IEEE Transactions on 48*, 3 (2001), 361–371.

[65] Sandberg, I., Lo, J., Fancourt, C., Principe, J., Katagiri, S., and Haykin, S. *Nonlinear dynamical systems: Feedforward neural network perspectives*, vol. 21. Wiley-Interscience, 2001.

[66] Schlömer, T., Poppinga, B., Henze, N., and Boll, S. Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (2008), ACM, pp. 11–14.

[67] Stanford, V. Biosignals offer potential for direct interfaces and health monitoring. *Pervasive Computing, IEEE 3*, 1 (2004), 99–103.

[68] Starner, T., Weaver, J., and Pentland, A. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 12 (1998), 1371–1375.

[69] TEXAS INSTRUMENTS. *Datasheet: CC430F613x, CC430F612x, CC430F513x MSP430 SoC With RF Core (Rev. F)*, 2011.

[70] Tong, K., Granat, M., et al. A practical gait analysis system using gyroscopes. *Medical engineering & physics 21*, 2 (1999), 87.

[71] Vicon. http://www.vicon.com, 2009.

[72] Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., and Popović, J. Practical motion capture in everyday surroundings. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 35.

[73] Weir, R., and Childress, D. Portable devices for the clinical measurement of gait performance and outcomes. In *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE* (2000), vol. 3, IEEE, pp. 1873–1875.

[74] Welch, G., and Foxlin, E. Motion tracking: No silver bullet, but a respectable arsenal. *Computer Graphics and Applications, IEEE 22*, 6 (2002), 24–38.

[75] Wheeler, K., and Jorgensen, C. Gestures as input: Neuroelectric joysticks and keyboards. *IEEE pervasive computing 2*, 2 (2003), 56–61.

[76] Winter, D. A. *Biomechanics and motor control of human movement*. Wiley, 2005.

[77] Wong, L., Chiu, W., Russ, M., and Liew, S. Review of techniques for monitoring the healing fracture of bones for implementation in an internally fixated pelvis. *Medical engineering & physics 34*, 2 (2012), 140–152.

[78] Wu, Y., and Huang, T. Hand modeling, analysis and recognition. *Signal Processing Magazine, IEEE 18*, 3 (2001), 51–60.

[79] Xsens. 3D motion tracking. http://www.xsens.com, 2010.

[80] Yang, G., Johnson, G., Tang, W., and Keyak, J. Parylene-based strain sensors for bone. *Sensors Journal, IEEE 7*, 12 (2007), 1693–1697.

[81] Yang, J., Wang, J., and Chen, Y. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters 29*, 16 (2008), 2213–2220.

[82] Young, D., Howard, W., and Orne, D. In-vivo bone strain telemetry in monkeys/M. nemestrina. *ASME, Transactions, Journal of Biomechanical Engineering 99* (1977), 104–109.

[83] Zell, K., Sperl, J., Vogel, M., Niessner, R., and Haisch, C. Acoustical properties of selected tissue phantom materials for ultrasound imaging. *Physics in medicine and biology 52*, 20 (2007), N475.

[84] Zhang, Z., Pouliquen, P., Waxman, A., and Andreou, A. Acoustic micro-Doppler radar for human gait imaging. *The Journal of the Acoustical Society of America 121*, 3 (2007), EL110–EL113.

[85] Zhou, H., Stone, T., Hu, H., and Harris, N. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical Engineering and Physics 30*, 1 (2008), 123.

[86] Zouba, N., Boulay, B., Bremond, F., and Thonnat, M. Monitoring activities of daily living (ADLs) of elderly based on 3D key human postures. *Cognitive Vision* (2008), 37–50.

VITA

Fahad Abdul Moiz was born on September 11, 1981 in Kuwait. He was educated in Indian School Kuwait and graduated from there in 1999. He attended the Metropolitan Community Colleges in the Kansas City Metro area and received his Associates in Engineering in December 2001. He received his Bachelor of Science and Master of Science in Electrical Engineering from University of Missouri - Kansas City in May 2004 and December 2005, respectively.

During his Masters he interned at General Electric Global Signalling in Grain Valley, Missouri. Later he was an adjunct instructor of Math at American University in Dubai.

In August of 2007, Mr. Fahad returned to University of Missouri - Kansas City to pursue a PhD in Electrical and Computer Engineering with a minor in Telecommunication and Network. During his PhD work, he was a research assistant in the Integrated Circuits and Systems Lab and also a graduate teaching assistant for electrical engineering laboratories.

Currenly, Mr. Fahad is based in Canada where he continues to pursue his career as an Electrical Engineer.