Wireless Sensors and their Applications in Controlling Vibrations

by

Ehsan Emami

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Ehsan Emami

Abstract

As wireless devices are becoming more powerful, more flexible and less costly to produce, they are often being applied in new ways. Combining wireless technology with new types of sensors results in the ability to monitor and control the environment in ways not previously possible. For example, an intelligent wireless sensor system that consists of a sensor, digital processor and a transceiver can be mounted on a board the size of a coin. The data collected by these devices are then transmitted to a central unit which is able to thoroughly process and store this data. Not only can the central processing station provide reports about certain physical parameters in the environment, it can also control the environment and other parameters of interest. The design process of these wireless sensor platforms is a well-developed area of research that covers concepts like networking, circuit design, Radio-Frequency (RF) circuits and antenna design. The design of a wireless sensor can be as simple as putting together a microcontroller, a transceiver and a sensor chip or as complicated as implementing all the necessary circuitry into a single integrated circuit.

One of the main applications of the sensors is in a control loop which controls physical characteristics in an environment. Specifically, if the objective of a control system is to limit the amount of vibrations in a structure, vibration sensors such as accelerometers are usually used. In environments where the use of wires is costly or impossible, it makes sense to use wireless accelerometers instead. Among the numerous applications that can use such devices are the automotive and medical vibration control systems. In the automotive industry it is desirable to reduce the amount of vibrations in the vehicle felt by the passengers. These vibrations can originate from the engine or the uneven road, but they are damped using passive mechanical elements like rubber, springs and shocks. It is possible however, to have a more effective vibration suppression using active sensor-actuator systems. Since adding and maintaining wires in a vehicle is costly, a wireless accelerometer can be put to good use there. A medical application for wireless accelerometers can be used with a procedure called Deep Brain Stimulation (DBS). DBS is a relatively new and very effective treatment for advanced Parkinson's disease. The purpose of DBS is to reduce tremors in the patients. In DBS a set of voltages is applied to the brain of the patient as some optimum combinations of voltages will

have a very positive effect on the tremors. Those optimum voltages are currently found by trial and error while a doctor is observing the patient for tremors. Wireless accelerometers with the use of a computer algorithm can assist in this process by finding the optimum voltages using the feedback provided by the accelerometers. The algorithm will assist the doctor in making decisions and has the potential of finding the optimums completely on its own.

Acknowledgements

I am grateful to those who have chosen to be supportive of others in times they need it. Luckily my supervisor Dr. R. Mansour is a great example of that type of people.

I am also thankful to my friends and family as they are always the main support for me and my ideas.

Table of Contents

AUTHOR'S DECLARATION....ii

Abstract....iii

Acknowledgements....v

Table of Contents....vi

List of Figures....viii

List of Tables....x

Chapter 1 Introduction....1

  1.1 Objectives....2

  1.2 Outline of the Thesis....3

Chapter 2 Wireless Sensors and Their Applications....5

  2.1 Wireless Accelerometers for Automotive Applications....7

  2.2 Wireless Accelerometers for Medical Applications....12

    2.2.1 Parkinson's Disease....14

Chapter 3 Wireless Transceiver Design....22

  3.1 Previous Work....22

  3.2 Design Concepts of a Miniature Active RFID System....28

    3.2.1 System and Design Process....29

    3.2.2 Protocol....33

    3.2.3 Circuits and Components....39

Chapter 4 Minimizing Vibrations Using Wearable Wireless Sensors....51

  4.1 System Overview....51

  4.2 Hardware....52

    4.2.1 Vibrating Structure....52

    4.2.2 Analog Output....55

    4.2.3 Wireless Sensor....55

  4.3 Software....57

  4.4 Date Analysis....58

  4.5 Optimization....64

  4.6 Results....72

Chapter 5 Conclusion and Future Work....78

Appendices....81

Appendix A Main MatLab Functions Written for Optimization 81
Appendix B Key Functions for Reading Wireless Sensor Data 83
References.......... 88

List of Figures

Figure 2.1: Categorization of automotive sensors - Courtesy of Fleming [3]. 8
Figure 2.2: Capacitive fingers for sensing acceleration along a single axis - Courtesy of Sadat [4]. 9
Figure 2.3: Block diagram of adaptive feedforward control - Courtesy of Fuller [5]. 10
Figure 2.4: Schematic representation of an AVC system - Courtesy of Bohn [9]. 11
Figure 2.5: Network of body sensors - Courtesy of P L Lo [11]. 14
Figure 2.6: Brain regions important in PD - Courtesy of Speak Campaigns [13]. 15
Figure 2.7: Dyskinesias Acceleration Signal and its Spectrum – Courtesy of Burkhard [19]. 16
Figure 2.8: Accelerometer Data Score VS the Clinically Observed Score – Courtesy of Hoff [15]. 17
Figure 2.9: The Algorithm for Management of Parkinson's Disease – Courtesy of Olanow [18]. 19
Figure 2.10: Deep Brain Stimulation - Courtesy of SpeekCampaigns [13]. 20
Figure 3.1: The SHIMMER wearable sensor platform - Courtesy of Lorincz [20]. 22
Figure 3.2: Block Diagram of a Generic Reader. 25
Figure 3.3: The Reader Designed by Dr. Wu. 26
Figure 3.4: Block Diagram of a Generic Tag. 27
Figure 3.5: The Tag Designed by Dr. Wu. 28
Figure 3.6: General Block Diagram of a Wireless Sensor Network. 30
Figure 3.7: Tag Schematic. 40
Figure 3.8: Tag Layout. 46
Figure 3.9: Manufactured Tag board and the required components. 47
Figure 4.1: Block Diagram of the System. 52
Figure 4.2: DC Motor with the shaking head attached to the shaft. 53
Figure 4.3: The driver circuit for the shaker. 54
Figure 4.4: The shaking "hand". 54
Figure 4.5: NI-USB-6800 Digital/Analog Input/Output. 55
Figure 4.6: ZStar3 wireless accelerometer and reader – Courtesy of Freescale [30]. 56
Figure 4.7: ZStar3 system block diagram - courtesy of Freescale [30]. 57
Figure 4.8: Sensor Read Application Window. 58
Figure 4.9: Raw signal from maixmum and minimum vibrations. 59
Figure 4.10: Maximum and minimum vibration signals after filtering. 60
Figure 4.11: Intensity of the vibration signal VS motor power for 6 attempts. 61
Figure 4.12: Modulation of the signal VS motor power for 6 attempts. 61

Figure 4.13: Periodicity of the signal VS voltage for 6 attempts. 62
Figure 4.14: Rate of movement of the signal VS motor power for 6 attempts. 62
Figure 4.15: Maximum and minimum vibrations (left) and their frequency spectrum (right). 63
Figure 4.16: Original function (top), function after slope-limiting (middle) and after quantization (bottom). 64
Figure 4.17: Guessing the direction of next step by approximating the gradient. 65
Figure 4.18: Test1 evaluated at 100x100 points. 67
Figure 4.19: The four optimizers that were attempted with Test1. 69
Figure 4.20: The process of finding optimum voltages to stop the vibrations. 71
Figure 4.21: Schematics of the variable optimum circuit. 72
Figure 4.22: Input and output of the variable optimum circuit. 73
Figure 4.23: A 20x20 set of actual data collected from the shaker. 74
Figure 4.24: Test2 evaluated at 100x100 points 74
Figure 4.25: Intensity of vibrations introduced by each motor and the threshold. 75
Figure 4.26: A round of real-time vibration optimization. 76
Figure 4.27: Converging of the optimizer to the optimum point in the first case. 77
Figure 4.28: Converging of the optimizer to the optimum point in the second case. 77

List of Tables

Table 1: Main Components of a Generic Reader. 24
Table 2: Main Components of a Generic Tag. 26
Table 3: Reader to Tag Command Packet. 36
Table 4: Tag Response Command Packet. 36
Table 5: Results of the real-time optimization. 75

Chapter 1

Introduction

The purpose of this work is to take advantage of the new developments in wireless technology and signal processing by exploring the new ways the technology can be applied. Circuit designers are improving design parameters such as cost, size, power consumption and flexibility on a daily basis for processors, radios and sensors. It is now possible for a device as small as a few millimeters to sense the physical environment around it, and then process and transmit the data gathered wirelessly. Digital processors have already become so fast that the dynamics of a vehicle can be calculated in real-time on a single digital processor. One of the possibilities from this development is the embedding of intelligent systems into structures and machines. These intelligent systems combine the use of sensors, actuators and digital processors to collect and process data from the system and in turn modify the system's behaviour in a desirable way.

Wireless sensor networks can track and monitor certain physical parameters in its surrounding environment. The data collected by the sensor nodes can be processed and analyzed either directly onboard the nodes, or in a base station, or both. Then the processed results can be used to improve the environment itself, to manage the surrounding conditions and perform other desirable tasks. For example, wireless vibration sensors can be used in applications where vibrations are an unwanted effect and needs to be reduced, such as the discomfort of passengers in a car from the vibration of a car engine and uneven road, or the shaking of the limbs for individuals with Parkinson's disease. In the case of automobile vibrations, much time is spent in every design to reduce the vibrations felt by the passenger employing passive damping, which uses springs, shocks and rubber dampers in the suspension and engine mounts. Recently some work has been done on an idea called active vibration control. The concept is to use vibration sensor input and digital processing to control the actuators used to cancel the remaining vibrations passengers are likely to feel. In the case of patients with Parkinson's disease, pharmacological methods have been tried to reduce the symptoms of vibrations or tremors. These methods have their limitations and new surgical methods are being utilized to help the patients in advanced stages of the disease. One of the innovative treatments for Parkinson's disease is Deep Brain Stimulation (DBS). An electrode is inserted into patient's brain to stimulate neurons with electrical signals. Commercially available, electrodes have four channels. It has been found that if certain voltages are applied to each channel, the shaking in different parts of patient's body is greatly reduced. These are called optimum voltages, but those voltages vary from time to time and from patient to patient. Therefore, frequent monitoring of the shaking as well as trial and error with different stimulation voltages is required. It is here, an intelligent system employing sensors can help adjust the signals automatically.

In terms of applications for such systems, the demand of the automotive market for more comfortable cars has already resulted in research for active vibration control in vehicles. Feed-forward methods are already being developed for better performance. As new types of sensors are invented and more powerful microprocessors are made, advancements in this field is foreseeable. Specifically, the use of wireless sensors in the engine could be the next step, because of the ease in their deployment and maintenance. There is also a great opportunity to use this type of vibration control in a medical application, as more than one million people suffer from Parkinson's disease and many of them are troubled by unwanted vibrations from their limbs.

1.1 Objectives

The work presented in this thesis suggests a hardware and software platform for controlling vibrations. The assumption is that in a vibration system the input that affects the vibrations can be controlled or modified to reduce the vibration level. Although this work will focus on Parkinson's disease the concepts can be applied to any problem that fits the criteria mentioned.

A very important component in the automatic control of vibrations is the sensor. For many applications wireless sensors are preferred because the location of sensors in some systems does not easily allow wiring. One of the objectives of this project was to identify the challenges in designing suitable wireless sensors and clarifying the design steps by making a prototype sensor board. Small-sized and flexible wireless sensors already exist on the market for different applications, but some of the applications in mind here may need a custom sensor design which is why the design of wireless sensors was explored.

One approach is to employing feed back from the wireless sensors together with an optimization algorithm to adjust the parameters in a vibrating system. This algorithm if chosen and configured properly should be able to find the right values for the parameters to stop the vibrations. To demonstrate the proposed idea, a model was constructed. The model consists of a vibrating structure and the associated hardware attached to it. The voltages that control the structure are generated by a computer application. The vibrations are measured by a sensor attached to the structure and can be read by the same computer application. The ultimate objective was to successfully run an optimization algorithm that stops the vibrations using the data coming from the wireless sensors. Different algorithms and different ways of treating the raw vibration data were investigated to see which combination produces acceptable results.

1.2 Outline of the Thesis

The general idea and what motivated this work is discussed in Chapter 1 as an introduction chapter. Chapter 2 is a technology survey that covers wireless sensors and their applications in different fields. The focus is to build an understanding around the idea of using wireless sensors in applications where the data from the sensors is used to control specific unwanted effects. The vibrations in Parkinson's disease and the ways to measure and quantize them are also surveyed in that chapter.

Chapter 3 presents the works done regarding the design and implementation of wireless sensors and wireless sensor networks. Some of the sensor boards designed previously by others are examined. This will lay the foundation to define the important concepts and features of the wireless sensor design needed. One of those concepts is the network protocol used, in which several choices are available and has a significant effect on the performance of the system. Other activities include component selection, circuit and RF design and board layout. There are many commercial radios available to be used in a wireless sensor design and an even wider selection of controllers and processors. The understanding of the role of each component in the circuit is necessary for proper selections. The RF and antenna parts of the circuit should also be designed carefully.

Given that the proper wireless sensor is available, Chapter 4 focuses on its application in a real system. Several techniques to quantify the vibrations have been investigated along with several optimization algorithms that can make use of this data. The challenges in signal processing and algorithm design of view are identified and examined. The hardware and configurations as well as the circuitry necessary to make the optimization loop work are explained. The way the applications in the computer interact with the hardware is also explained, along with the solutions to overcome the challenges presented.

The author thinks there are still many possibilities unexplored in this framework. For those who are interested in taking this idea further than this, there are some pointers in the conclusion which might be useful.

Chapter 2

Wireless Sensors and Their Applications

The advancements in hardware and software technology are in some cases developing faster than the innovators can imagine their applications. Electrical and electronic circuits are becoming smaller and faster, and formerly separate components are being integrated onto single chips while new methods and technologies are being introduced. More and more computing power can be implemented onto small boards and the boards are consuming much less power than ever before. Making use of these advancements in the real world is now one of the main challenges and a promising field for research.

Sensors have been used in the industry for many years. Sensors are classically defined as [1] "*devices that transform (or transducer) physical quantities such as pressure or acceleration (called measurands) into output signals (usually electrical) that serve as inputs for control systems.*" We need sensors in a factory to know the temperature of an operating process or the air flow through a duct. Sensors are used in cars to monitor and gauge speed and fluid levels. Sensors usually convert a physical quantity into an electrical signal utilizing various electromechanical, electromagnetic or thermo-electrical processes. New technology can now make the sensor devices much smaller, like the way Micro Electro-Mechanical Systems (MEMS) enables the integration of electromechanical devices onto tiny chips. As a result we can condition, process and communicate the signals obtained using the same sensor board.

Wireless technology has also been advancing at a rapid rate. High-frequency transceivers can now be implemented on a tiny bit of silicon and consume very little power. Networking protocols have been or are being developed that enable wireless communication to replace wires almost everywhere. Control and monitoring systems that previously used wired sensors can now be wireless. Arampatzis has conducted a useful survey on applications of wireless sensors [2] and categorizes them as follows:

A. Military Applications
B. Environmental Monitoring
C. Support for Logistics
D. Human-Centric
E. Robotics

In short, it is now possible to implement a sensor along with the necessary logic and a transceiver on a single chip or a coin-sized circuit board. These wireless sensor nodes can communicate to a station or even network together while consuming micro-power. The two major applications are explored here.

An area to apply wireless sensors is the automotive industry, since cars today already have several types of sensors in them for critical operations, safety and comfort. Wiring is always difficult and sometimes almost impossible, such as sensors on tires to measure air pressure, vibrations and tilt. With the right wireless sensors in the right places, the assembly and maintenance cost can be greatly reduced. Furthermore, at times new features can be added to cars using wireless devices, when a wired solution was not feasible.

Vibration sensors are already used in active vibration reduction in cars. Many active mechanical systems have been developed to sense the unwanted vibrations caused by the engine or the road, and cancel them through intelligent actuators to increase comfort and safety [5]. With current technology it is possible to have the sensors communicate with the central processing units running software such as DSP or MatLab through a reliable wireless link. Note that a wireless sensor for this application needs to meet strict specifications like temperature tolerance and data error rates.

Another growing application field for wireless technology is patient monitoring in hospitals and at homes. Both patients and doctors would be happier if the doctor could have an indication of a patient's condition without having them come over to the hospital. That is now possible for some diseases.

Wearable sensors are already being used in many medical applications for monitoring patient's vital signs and for research purposes but they are mostly wired. To monitor patients sometimes required connecting patients to a messy about of wires. On the other hand, for some diseases if the patients are constantly monitored for certain changes in vital signals, the doctor can be informed in time and lives can be saved. This is only possible if patients are wearing wireless sensors throughout daily life. Advancements in the devices' processing power and communication, as well as reduction in power consumption make this type of application more attractive.

Recent research has introduced the different types of sensors in medical applications which were not previously used. It is now possible to get a solid indication of the condition of a patient with Parkinson's disease through acceleration sensors (accelerometers) attached to their body [19]. Doctors need these indications to adjust the medication and make other decisions. Wearable wireless sensors can make a reading, pre-process the data and transmit the results to a computer. It is even possible to get the results to the doctor through the internet without having the patient or the doctor leave their homes.

Vibration and acceleration sensors can be implemented in various ways. MEMS accelerometers are so compact that they can be packaged in the same chip along with a processing and communication core. In this survey, two different applications are explored which both share the concept of wireless accelerometers. Firstly, is the current technology in active vibration control of vehicles is examined while assessing the possibility of replacing existing vibration sensors with wireless ones. Secondly, is the way wireless accelerometers can help with care of patients with Parkinson's disease.

2.1 Wireless Accelerometers for Automotive Applications

A modern car is constantly processing information from various types of sensors installed inside the engine, suspension, passenger space and other areas. The information is essential for the monitoring, gauging, diagnosis, comfort, safety and control of vehicle functions. Automotive sensors have been around for several years; according to an overview by Fleming [3] there have been tens of sensors used as of 2001 in a vehicle. Figure 2.1 shows the classification of sensor types used in automobiles.

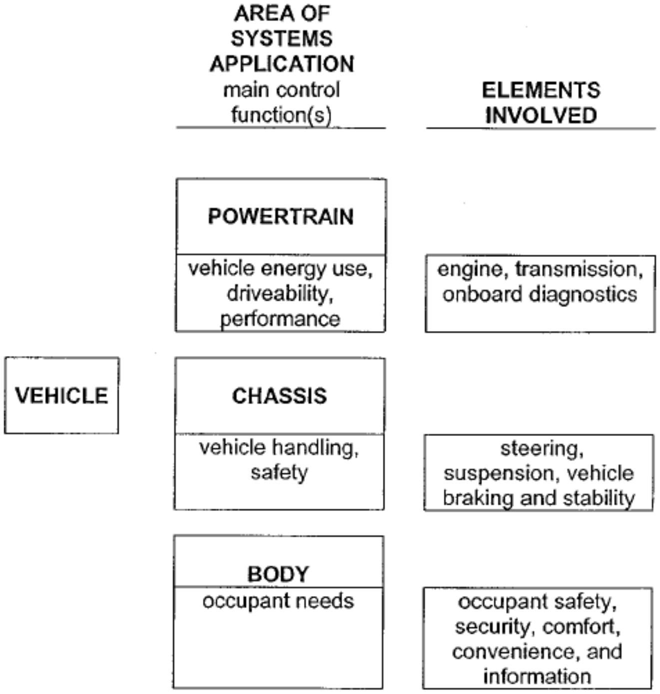

Figure 2.1: Categorization of automotive sensors - Courtesy of Fleming [3].

There are several types of sensors used in vehicles to measure the angular and linear position, pressure and such related physical quantities. Speed, acceleration, vibration level and frequency can also be related to position through sensor signal differentiation and/or integration. Examples of these sensors in cars are, potentiometric sensors used for fuel levels, Hall Effect sensors for wheel-to-chassis height, anisotropic magnetoresistive sensors for valve positions, optical position sensors for some applications and various micromachined sensors for pressure of different fluids [3].

A typical MEMS accelerometer can produce 3 signals, each indicating acceleration along an axis. In theory, the position information can be determined by integrating the acceleration signal twice although in practice challenges exist regarding that matter. A 3-axis accelerometer with an appropriate range, resolution and sampling rate can in theory measure all linear and angular displacement-related quantities if used together with a processor. The measurement of acceleration in these sensors is performed by variable capacitive fingers as shown in Figure 2.2. A few sets of those fingers along with the necessary amplifiers and other circuitry along with processing and communication capabilities on the same chip, make up an amazing sensor. Such sensors can be used in vehicles wherever vibration, speed, angular rate or similar quantities need to be measured.

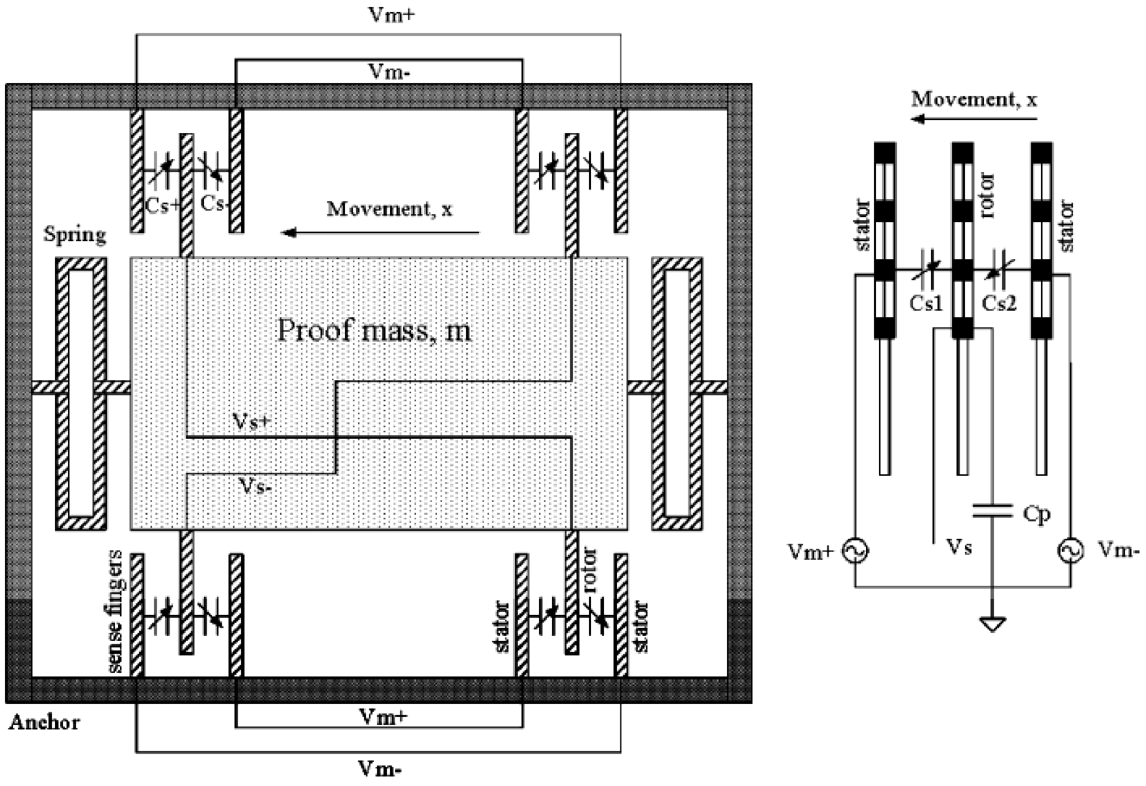

Figure 2.2: Capacitive fingers for sensing acceleration along a single axis - Courtesy of Sadat [4].

Increasingly popular automotive application for accelerometers is the active control of vehicle vibrations. The vibrations of interest in vehicles are periodical mechanical movements at frequencies above 50 Hz. The vibrations below 20 kHz can also be propagated through air as sound. Fortunately, passive elements conveniently eliminate frequencies higher than 1 kHz. Passive damping in cars consists of the traditional rubber dampers, coils and shocks which are part of a separate field in mechanical engineering. Therefore the target band in active vibration control is 50 Hz to 1 kHz [5]. The source of the vibrations is usually either the rotating parts of the machinery or the uneven surface of the road.

Simply put, to actively control vibrations, vibration data is fed to a microprocessor that is used to control the mechanical actuators near the vibrating source to cancel the unwanted vibrations. The actuators might be electromagnetic dampers (instead of rubber ones) [6], or shakers or even audio loudspeakers. Depending on the control model used, the signals fed to the microprocessor could be sensor data along with some reference signal (such as a speed signal from gas pedal) or disturbance signal fed forward from the source of vibrations.

Based on the classic control theory, a control configuration popular for this purpose is the adaptive disturbance feedforward. Linear Time-Invariant (LTI) feedback and adaptive feedback are other possible methods but they are not as popular for various reasons including limits on bandwidth, noise levels and dynamic range [5]. Adaptive feedforward uses a reference signal from the system that is convenient to get, and with the knowledge of the transfer function from that signal to the output, the processor can control the actuator (whose transfer function to the output is also known).

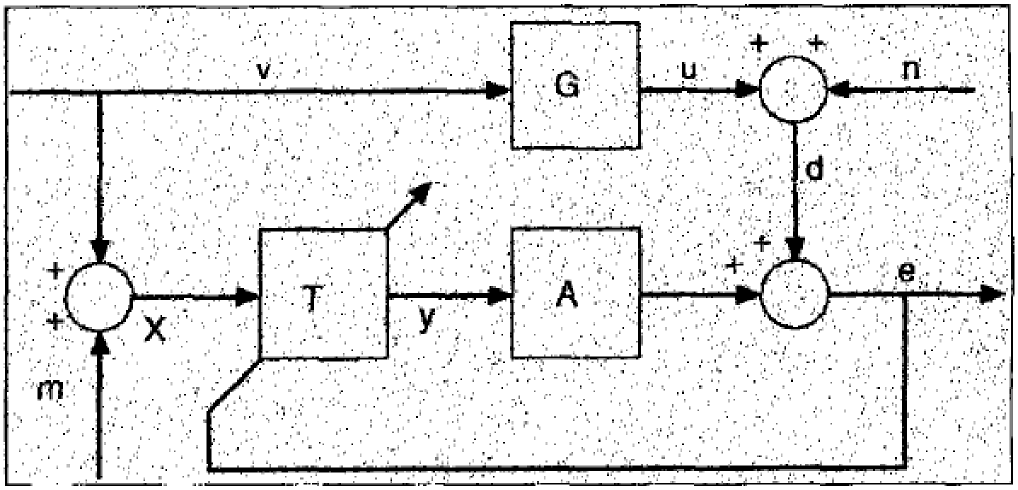

Figure 2.3: Block diagram of adaptive feedforward control - Courtesy of Fuller [5].

Figure 2.3 shows the mentioned feedforward system where *v* is the reference signal, *u* is the disturbance (to be cancelled), *y* is the control signal to the actuator and *e* is the error signal (which is desired to be zero). Noises are formulated as *m* and *n*. Ignoring noise for now, if the controller filter T is $-GA^{-1}$ then the disturbance is obviously cancelled [5]. In reality though, the noise is always present and the transfer function of the mechanical system is neither easily nor accurately obtained. Therefore in order for the system to function, the error is fed back to the controller so that it can adjust and adapt itself as it goes. References [7] and [8] are good papers explaining the feedforward method and the challenges involved. So with this context in mind, accelerometers have two different uses in these systems:

- Calculation of system transfer function *G* (in the development phase)
- Continuously sensing the error signal *e* required for the system to function

An example of accelerometers used in such systems is Bohn's design [9] of an Active Vibration Control (AVC) system which is represented in Figure 2.4. Here, the feedforward reference signal is the engine speed signal and the error signal comes from the accelerometers mounted right beside the actuators on the shafts.

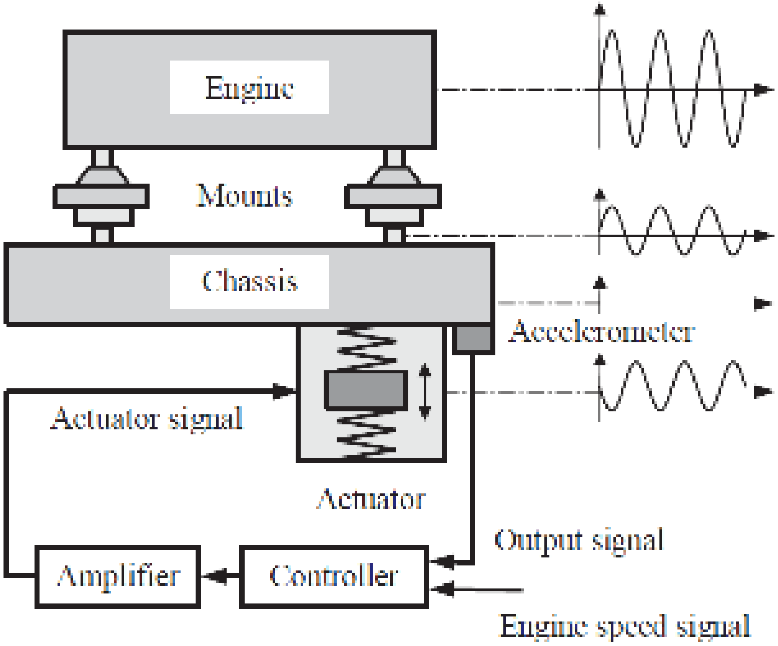

Figure 2.4: Schematic representation of an AVC system - Courtesy of Bohn [9].

Wireless accelerometer sensors can be conveniently attached on the suspension system, on the shafts, inside the tires or wherever necessary. At the development phase when the transfer function is needed to be calculated for the control filter design, the accelerometers send their data to a reader which can be attached to a computer for analysis. The accelerometers could act as error sensors while the system is operating by transmitting the vibration signals to a central processing unit installed somewhere inside the vehicle.

If the accelerometers are battery operated, their battery life should be carefully taken into account so that they do not fail while driving. Even if the sensors are powered through wires but communicate wirelessly, it is still simpler than wiring the sensor all the way to the processing unit, because a single power wire is usually available everywhere. Other specifications of sensors to be taken into account are the sampling rate, the resolution, the range, the communication rate and the endurance of the sensor with respect to the electromagnetic, the temperature and other physical conditions.

2.2 Wireless Accelerometers for Medical Applications

Sensors are already a part of many healthcare-related procedures and examinations. Sensing electrical signals related to the activity of the heart and the brain is not a very recent concept. Sensors can be attached to a patient in an operation room to monitor vital signs or can be worn by them as they are undergoing an examination. More recently, sensors can be implanted inside the body or could be constantly worn by patients or the elderly who need continuous monitoring as they are living their normal life [2].

Miniature sensors are now being implanted inside different parts of the body, collecting data and communicating with a processing unit outside the body. Although there are still many challenges to overcome [10], this technique is currently being used. Obvious requirements for these sensors are the need for limited to no maintenance, infrequent change of batteries, wireless communication and harmless to biological organs. Another type of sensor can be used externally, like those that are worn around the wrist, in the form of a glove, or attached on the surface of other parts of the body.

Schweibert [10] provides a list of the challenges in designing wireless biomedical sensor nodes:

- Low power
- Limited computation
- Material constraints
- Continuous operation
- Robustness and fault tolerance
- Scalability
- Security and interference
- Regulatory requirements

Since the small size and low power consumption are critical requirements for wearable and implantable sensors, it is usually necessary to have a central unit to further process the data acquired from the sensor nodes. This unit could be attached to the belt or somewhere else close to the person. Networking challenges are introduced as all the sensors need to report to and be controlled by the central unit. P L Lo [11] addresses the networking and protocol design challenges involved. Figure 2.5 depicts a network of sensors attached to the body and their communication with a central unit. The unit can act as a link between the sensors and a computer that stores all the data or presents them to the doctor.

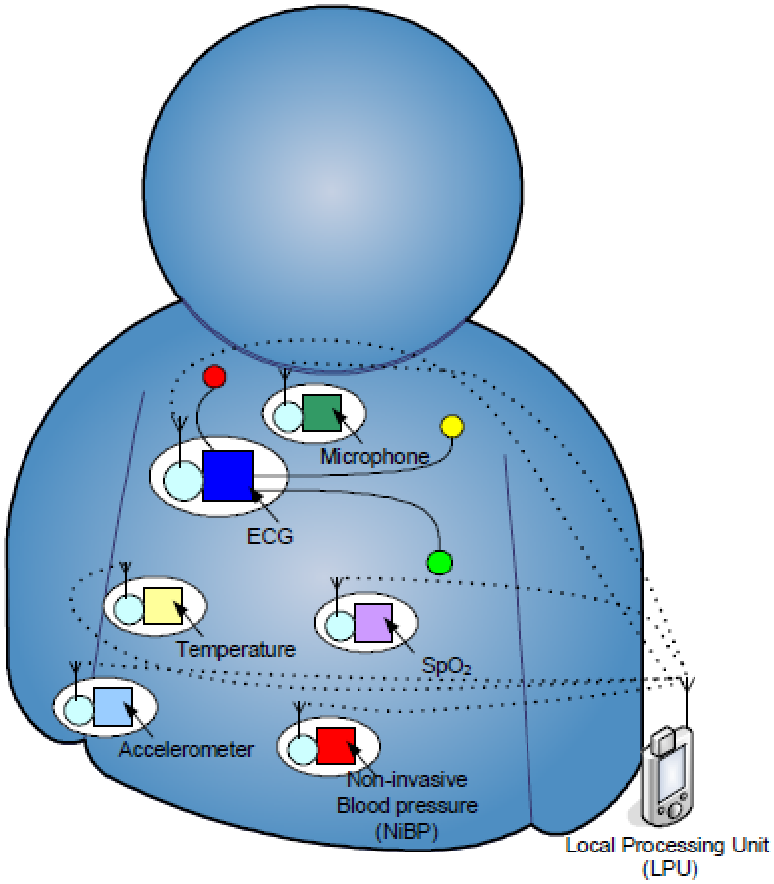

Figure 2.5: Network of body sensors - Courtesy of P L Lo [11].

2.2.1 Parkinson's Disease

Parkinson's Disease (PD) is a disorder which targets mostly the older individuals. Over one million people currently suffer from this disorder in the United States and each year 60000 more are diagnosed [18]. The disease is named after James Parkinson who provided the first complete clinical description of this disorder. PD is a disease that is caused by the malfunction of certain parts of the brain which produce a chemical called dopamine. This part of the brain is called *substantia nigra* [12]. The resulting lack of dopamine causes various disabilities and dysfunctions for the patient as dopamine is the agent carrying messages between brain cells. There is a standard scale based on which doctors can rate the symptoms of PD patients between 0 and 4 (0 being the mildest and 4 the most severe). The ratings are based upon mood, behavior, daily activities and other functions of the patient. One of the symptoms that the patients experience is problems with the movement of the body. Figure 2.6 shows the central brain regions which are of interest in PD.

One common way to treat PD is to replace the missing dopamine. Since dopamine cannot cross the blood-brain barrier, a chemical that produces it is injected called levodopa. While the patient is under the levodopa treatment the symptoms are significantly improved. The only problem is that after a few years of treatment the problems with movement (dyskinesia) comes back and treating it with levodopa is no longer effective. For an overview of PD and its methods of treatment refer to [13].

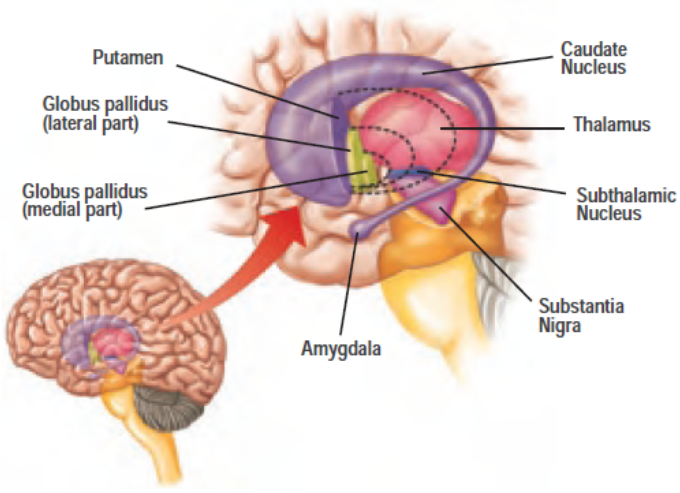

Figure 2.6: Brain regions important in PD - Courtesy of Speak Campaigns [13].

These movements or dyskinesia is the shaking of hands and other body parts. The difficulty in controlling this shaking is one of the most visible symptoms of PD. A doctor can rate a patient by looking at the shaking of their limbs while they are performing a series of tasks. The rating is used to adjust the dosage of levodopa and other medications. Instead of rating dyskinesias visually, the rating process can be automated wearable vibration sensor nodes that can read and evaluate shaking signals using software.

According to Burkhard [19] the acceleration signal of a patient with dyskinesia will look something like Figure 2.7. In their experiment they used rotation-sensitive sensors attached to patients and recorded the signals. They are using angular rate which can be converted to linear velocity given that the radius of the movement is known. The angular rate in Rad/s should be multiplied by the radius in meters to give linear velocity. Linear velocity can be differentiated to give linear acceleration in m/s^2. For example if the peak in the spectrum is 120 deg/sec at 2 Hz and the radius of movement is 10 cm, it is converted to around 50 m/s^2. The purpose of that experiment was to decrease the reliance on the visual scoring on the intensity of the disorder and provide a quantitative measure of the dyskinesias.

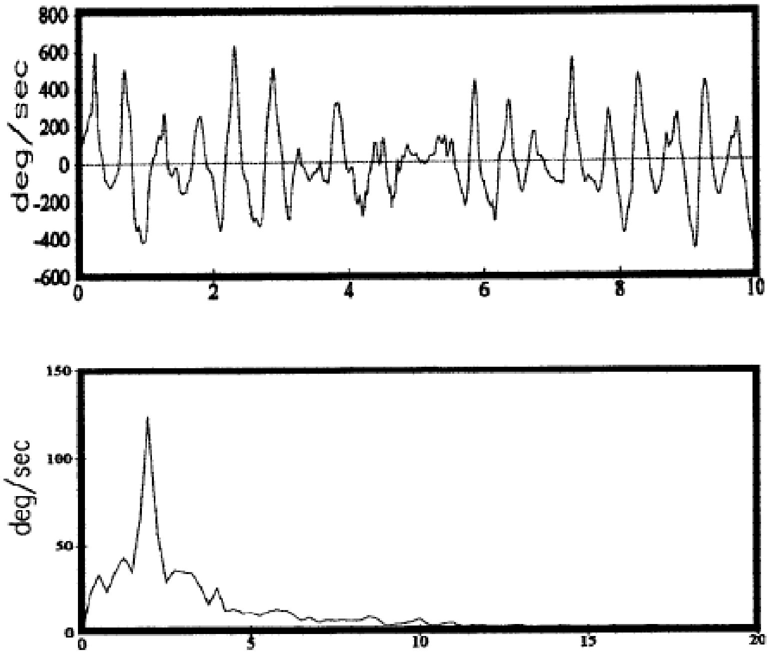

Figure 2.7: Dyskinesias Acceleration Signal and its Spectrum – Courtesy of Burkhard [19].

Hoff [15] used wired piezoresistive uni-axial accelerometers to measure the patient's vibrations. They analyzed the dominant frequency amplitude in 1 Hz to 4 Hz and 4 Hz to 8 Hz bands, and showed the correlation with a clinical score called modified Abnormal Involuntary Movement Scales (m-AIMS). Eight sensors were attached to different parts of the body of each patient while the patient was asked to perform certain tasks. The data from these sensors were transferred to a computer using wires to be processed and recorded. Some of the results that were obtained in that study are presented in Figure 2.8.

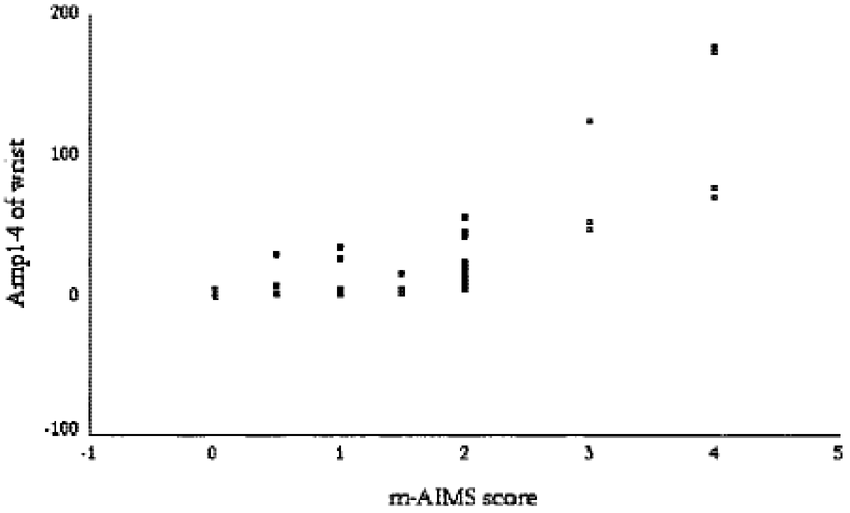

Figure 2.8: Accelerometer Data Score VS the Clinically Observed Score – Courtesy of Hoff [15].

Patel [14] shows that several seconds of the vibration signal from different parts of the body can rate the level of the patient's dyskinesia accurately. They used MEMS based wearable wireless accelerometers to acquire body vibration signals and used different features of the signals such as RMS and dominant frequency to rate the patients. They then compared the results with how the doctor had rated the patients and decided that the vibration signals can be used reliably. Patients are first rated according to the UPDRS by a clinician. Then accelerometers are used to record the movements of their body. Each recording is 30 seconds of vibration signals. Then 5 seconds from the 30 seconds of signal is randomly selected and analyzed.

The 5-second vibration signal is filtered to remove any frequency above 15 Hz and below 1 Hz as in this application the high frequencies are only noise and low frequencies are voluntary movements. Then the signal is de-trended. De-trending is the subtracting the linear trend of the signal from the original signal. A linear trend is a straight line best fitting to the signal. De-trending helps reduce the effects of time-dependence. The following features of the filtered signal are calculated:

Intensity: To have a measurement of how intense the movements are, the RMS of the de-trended signal is calculated. Let *intensity* = I, *detrended signal* = d_i , *number of samples* = n and *mean* = M:

$M = (1/n)\sum d_i$

$I = \sqrt{(1/n)\sum (d_i - M)^2}$

Modulation: The auto-covariance of the signal is calculated. The range of the auto-covariance values is used as an indication of how often the nature of the movements varies which is called modulation here. Let *auto-covariance* = A_m and *modulation* = V:

$A_m = \sum (d_{i+m} - M).(d_i - M)$

$V = max(A_m) – min(A_m)$

Rate of movement: simply the dominant frequency of the signal is considered the rate of movement which is an indication of how fast the shaking is. Let rate of *movement* = f_m and *spectrum* = S_f:

$S_f = FFT(d_i)$

$f_m = f$ *such that* $S_f = max(S_f)$

Periodicity: first the dominant frequency is calculated. Periodicity is the ratio of the energy of the signal at that frequency divided by the total energy. That would indicate to what extent the signal can be considered periodical: Let *periodicity* = P:

$P = S_{fm}/I$

Each of the calculated features provides some indication of the movements of the patient. As mentioned, along with the accelerometer measurements, a neurologist also scores the patients visually. The features are mapped onto a scale using different mapping methods. In the resulting mappings the data points related to patients with the same clinical score form clusters which are visible without knowledge of the score. It means that it is possible to predict the score with good approximation by using only the accelerometer data. Further research shows that among the mentioned features, intensity can provide the best predictions [16].

The classic treatments for PD start losing effectiveness on the patient after several years while new methods rely heavily on technology. The ability to have accurate and constant measurements of the symptoms helps with the effectiveness of the treatments. According to the studies of Olanow [18], at some stage of PD pharmacological treatments should be accompanied by other kinds of therapies to control the disease. The decision tree (Figure 2.9) clearly states that at some point surgical methods are the best option.

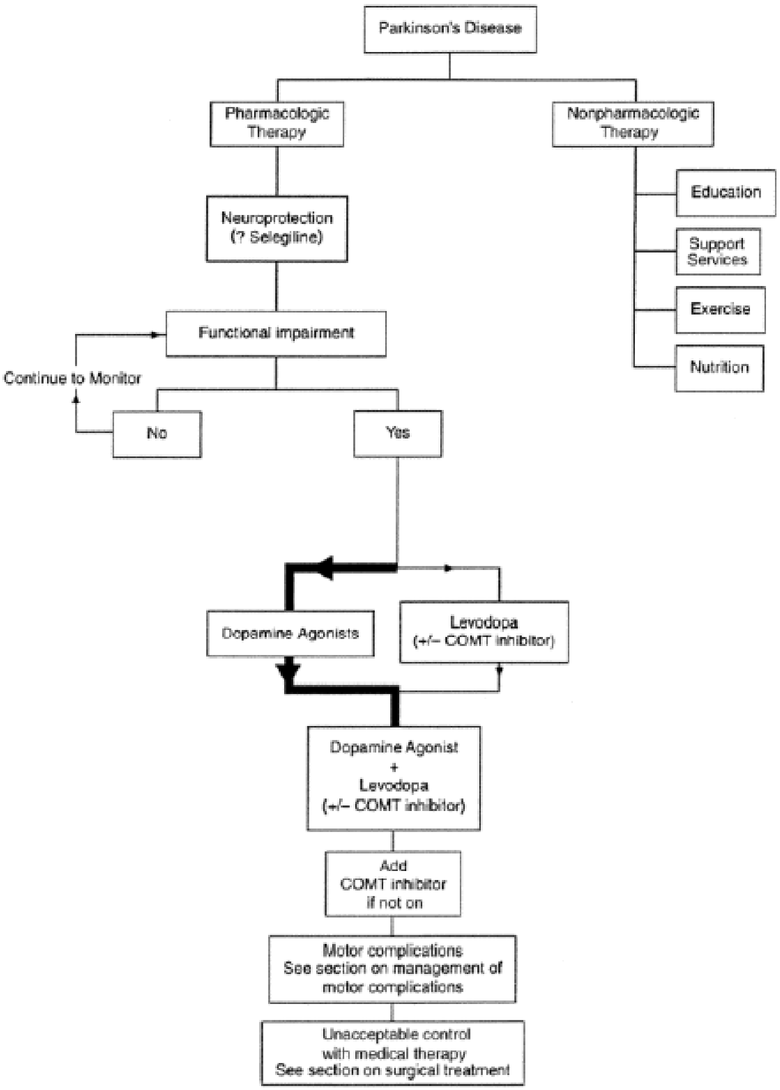

Figure 2.9: The Algorithm for Management of Parkinson's Disease – Courtesy of Olanow [18].

A relatively recent method for treating PD is Deep Brain Stimulation (DBS). It turns out that stimulating the *globus padius* and the *subthalamic nucleus* (Figure 2.6) with external electrical signals can be used to reduce dyskinesia. The electrical signals are a series of pulses with certain amplitudes and frequencies. A clear relationship has not been yet found between the signals applied to the brain and the severity of the dyskinesias. Therefore what is done is mostly trial and error. This means the doctor applies some set of voltages, rates the patient's dyskinesia and if not satisfied, applies a different set of voltages. As seen in Figure 2.10 the system consists of an electrode inside the brain and a signal generating device that controls the probe from outside the brain. The probe is implanted all the way to the central parts of the brain by surgery. For more details on DBS refer to Perlmutter's paper [17] on the subject.

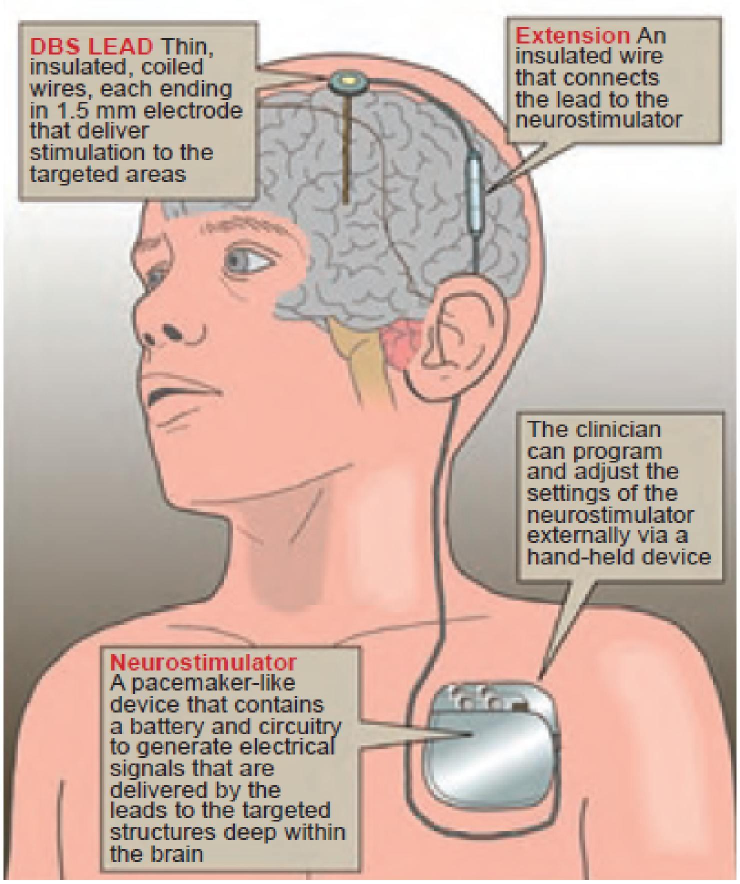

Figure 2.10: Deep Brain Stimulation - Courtesy of SpeekCampaigns [13].

Once a set of DBS signal parameters are found that minimizes the dyskinesia for a specific patient, the DBS control unit is programmed accordingly and the patient is sent home. The optimum signals however are subject to change over time and the patient will need to come back for re-adjustment of the system. The way wearable accelerometers can help is to track the dyskinesia by sensing the vibrations. The system can then use that data to inform the doctor that it is time to reprogram the DBS. On the other hand it is possible for the DBS and sensor systems to record their data on a computer so that over time extensive data becomes available relating to the signals applied with the amount of dyskinesia observed in different patients. The data could be processed further and become a very useful tool to help the doctor identify the correct DBS signal parameters that minimize the symptoms.

The signals applied to the brain in DBS are simply pulses of voltages. Signal variables which have effects in the management of the symptoms, are frequency, pulse width, amplitude and the location of the probe in the brain [18]. These four variables offer a huge set of possible combinations and some of those combinations are what the patient needs. The exact mechanism that causes the symptoms (like vibrations) to decrease is unknown at this time. If there was a way to systematically find the optimum combination with the help of computers it would make the life of many people more pleasant. One idea is to use wireless sensors in a closed loop configuration along with an optimization algorithm to try different parameters and find the optimum settings.

Chapter 3

Wireless Transceiver Design

This chapter aims to provide a complete understanding of wireless transceivers. All the circuits designed and the software written will be explained in the following sections. By doing so, it will explain the challenges, and decisions made in the design process. The hardware and software tools that are required for all the design steps and testing are explained here.

3.1 Previous Work

A study has been done by Lorincz [20] on a wireless sensor platform for medical applications. It is a small wearable sensor node made by Intel called SHIMMER (Figure 3.1). Equipped with an acceleration sensor, it also provides on-board processing power. One of the major advantages of SHIMMER is that it has a memory card slot for storing long intervals of data. The device is capable of transmitting the stored or real-time data to a reader which is connected to a computer for more advanced signal processing and recording.

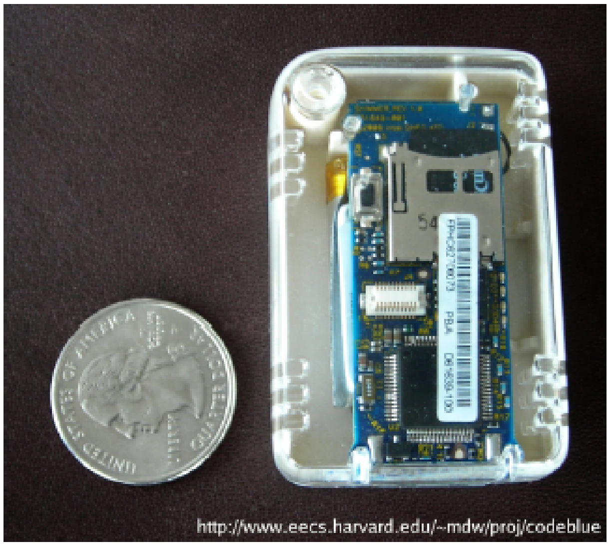
http://www.eecs.harvard.edu/~mdw/proj/codeblue

Figure 3.1: The SHIMMER wearable sensor platform - Courtesy of Lorincz [20].

Nino Zahirovic, currently a PhD student in the CIRFE group at University of Waterloo, designed and constructed a board which demonstrates wireless communication [34]. A digital sensor signal can be connected to the board and transmit the sensor data wirelessly at 2.4 GHz. To test the board, a development tool is available that can receive and display the wireless signal.

The main component on the board is the CC2430 chip from Texas Instruments. The CC2430 has a RF IC and a microcontroller in the same package. With the addition of a matching network, an antenna and a few passive components the CC2430 becomes a complete wireless transceiver. Any digital sensor signal can be connected to the digital I/O's of the microcontroller and transmit the sensor data via the RF link, becoming a wireless sensor.

The main advantage of the CC2430 is that it eliminates the need for an additional microcontroller on the board and provides a complete wireless solution with very few external components. Another advantage is the design support that the manufacturer provides for the application. Specifically, a development kit is available to help program the CC2430 in the circuit and then test and measure the wireless signal.

Zahirovic has a working prototype of such a wireless system. His circuit board makes avaiable all the unused pins of the CC2430 to the outside of the board. One advantage of his board is its very small. He has tested the board and documented the design with the test results [34].

Dr. Ke-Li Wu and his group have developed an a Radio-Frequency IDentification (RFID) system [34]. The design concepts of that system are very useful for the kind of applications this thesis covers. They have designed a 433 MHz system which includes readers and tags and a user interface. They have also made improvements on the standard protocol to make it more efficient. The tags are battery-operated while the reader should be connected to a PC and gets power from the port.

The design is a general purpose active RFID platform. Dr. Wu's tags and readers communicate with each other based on a given protocol. The reader presents the results to and gets commands from a PC through the parallel port. An application runs on the PC which allows the user to interact with and control the system. The user collect the ID's of all the tags in range, collect any sensor data from the tags, put the tags to sleep, or write data on the memory of the tags.

The tags and readers have a similar design for the wireless communication task. That part of both circuits consists of a CC1000 RF IC and an MSP430 microcontroller [25] from Texas Instruments. On the tags, there is a socket that can be connected to a sensor board. To demonstrate the capabilities, Dr. Wu has provided a sample battery powered sensor board with a temperature and vibration sensor. It has an efficient application specific antenna for the tags, but the readers use regular whip (dipole) antennas.

One advantage of this system is the large memory of the microcontroller on the tags. Other advantages include a long battery life and compatibility with most kinds of sensors. One of the disadvantages is the large size of tags [34].

The system consists of one or more wireless modules (tags) which are battery operated and could be equipped with various sensors. The reader module (or modules) connects to a PC and communicates with the wireless tag modules. In addition a software application is used on the PC to manage and display the data. The communication protocol between the reader and tags is in compliance with the ISO 18000/7 which is the standard 433 MHz active RFID protocol.

The reader board has a port for connecting an RS-232 cable which connects the board to a PC. It is also equipped with an antenna that sends and receives signals over the air to and from the tags. Power is provided to the board by an external 5.2 V 1 A power supply. Main components on the board are listed in Table 1.

Table 1: Main Components of a Generic Reader.

Component	Description
MSP430F449	Main Processor
CC1000	RF Transceiver
MAX3221E	RS-232 Transceiver
SP6201	Voltage Supply

The SP6201 is a voltage regulator that provides a constant regulated 3.3 volts for the other IC’s using the 5.2V input voltage. The heart of the circuit is the MSP430F449 a powerful microcontroller that is equipped with multiple ports and timers and an Analog to Digital Converter (ADC). The serial port of MSP430F449 is connected to MAX3221E. MSP430F449 is also connected to CC1000 through another port and also the ADC. In turn the CC1000 is connected to an antenna through a matching network of passive components. At the front end, the MAX3221E is connected to the RS-232 cable that is connected to the PC.

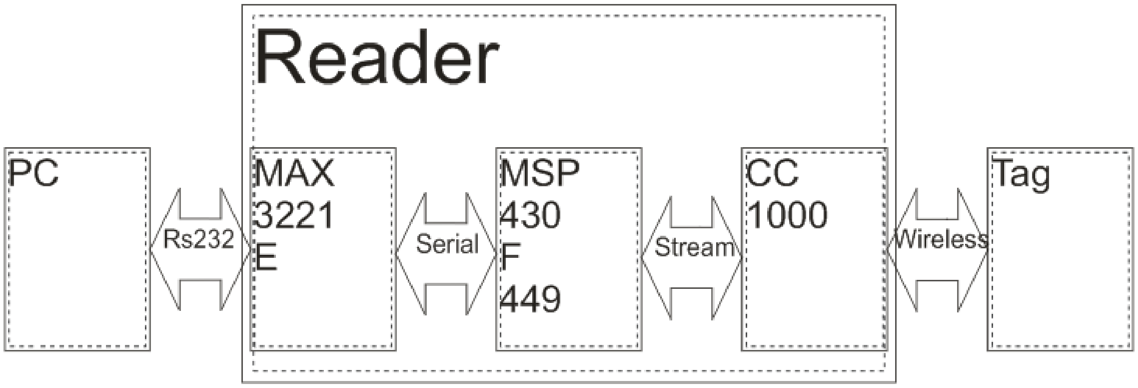

Figure 3.2: Block Diagram of a Generic Reader.

The MSP430F449 and the CC1000 each require a crystal for their internal oscillators. The rest of the components on the board are passive components used for completing the IC's function, and provides simple adjustments, coupling and filtering.

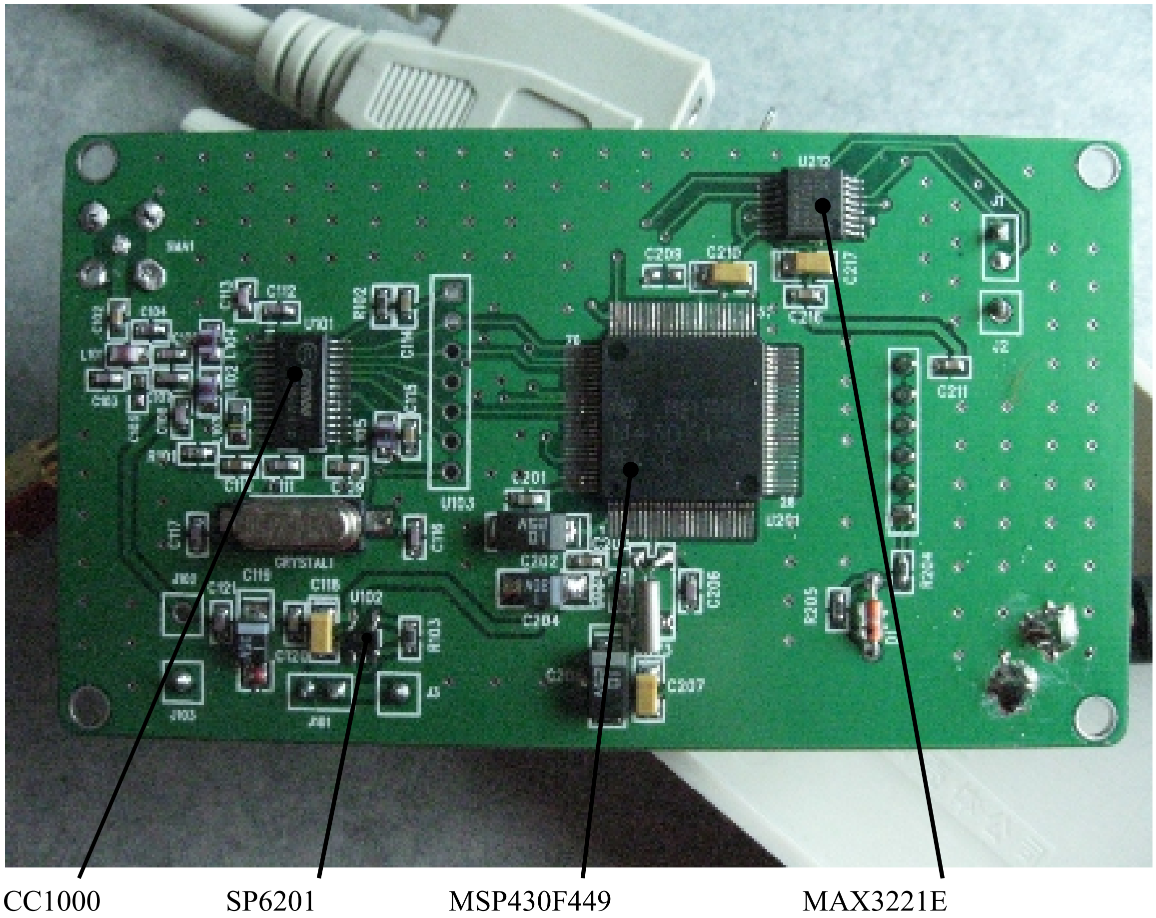

Figure 3.3: The Reader Designed by Dr. Wu.

The tag board is powered by a 3.6V battery. There is a connection header on the main board for connecting to a sensor board. The main board also includes an antenna that sends and receives signals to and from the reader. Main components on the board are listed in Table 2.

Table 2: Main Components of a Generic Tag.

Component	Description
MSP430F449	Main Processor
CC1000	RF Transceiver
SP6201	Voltage Supply

The SP6201 is a voltage regulator that provides a constant regulated 3.3 volts for the other IC's using the 3.6V battery voltage. The MSP430F449 is the same kind of microcontroller used for the reader. It connects to the CC1000 the same way it does in the reader. It also connects to the sensor board through the serial port and a few pins from other ports. The CC1000 is connected to the antenna the same way it does in the reader.

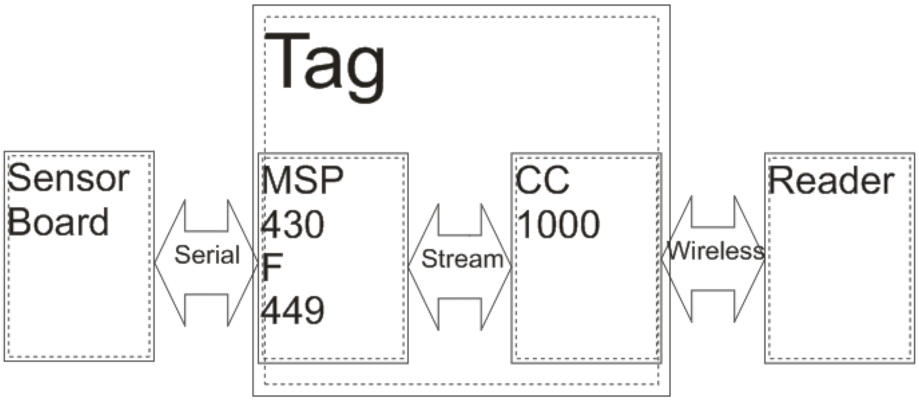

Figure 3.4: Block Diagram of a Generic Tag.

The MSP430F449 and the CC1000 each require a crystal for their internal oscillators. The rest of the components on the board other than the mentioned IC's and crystals are passive components used for completing the IC's function and provides simple adjustments, coupling and filtering.

As mentioned, an extra sensor board could be connected to the MSP430F449 on the main board. In this case, the sensor board contains a temperature sensor and a vibration sensor that get power from the main board.

It is also possible to store digital data inside the memory of the MSP430F449 that could be uploaded and/or downloaded to the tag.

A tag could also be used without any sensor board. In that case the only data (such as the tag ID) retrieved from the tag will be the data stored inside the memory of the MSP430F449.

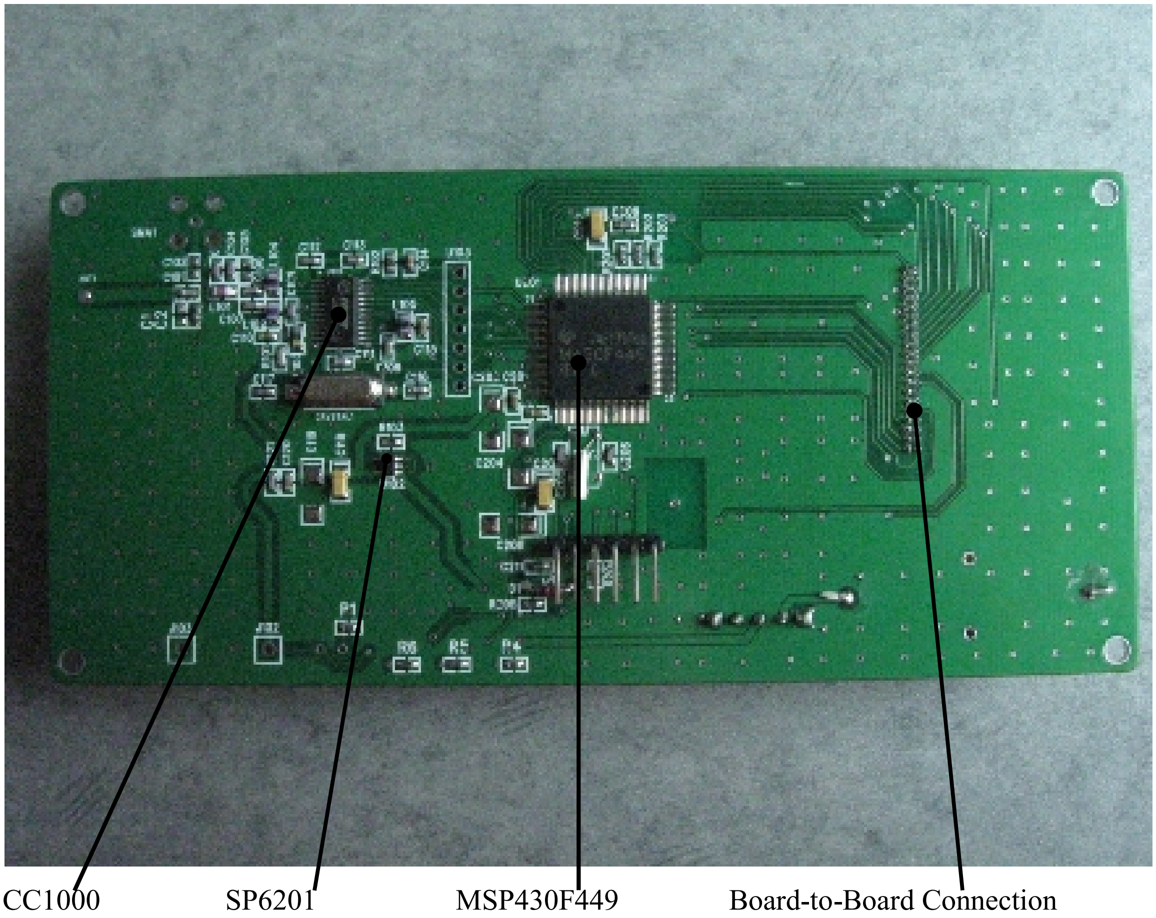

Figure 3.5: The Tag Designed by Dr. Wu.

3.2 Design Concepts of a Miniature Active RFID System

The purpose of this section is to provide the framework of an active RFID system that can be used later for different applications in the CIRFE group. The main focus is to design a RFID tag that is small, flexible, efficient and powerful. The next important feature is a user friendly interface application. The readers are relatively less critical because the size and power consumption are not a big issue and any working design that meets the basic requirements will do the job.

The main application planned for the active RFID framework developed here are wireless sensors. Active RFID is one of the possible solutions for a low-cost wireless sensor network. This is a very interesting and new area for research. Many possibilities are still unexplored and many questions are not yet answered.

There are different hardware architectures that can be used for implementing a wireless sensor node [27]. In theory, the whole system can be integrated on a single application-specific IC. Another approach is to use a general-purpose IC that will work together on a board to fulfill the requirements. We will be focused on the second approach because it is a relatively low-cost and simple approach. Moreover this work can be used as a starting point for future system integration of different functional modules onto an IC.

From a software point of view, the main concern is implementing the protocol. The software will be embedded in the microcontroller. In this section, the building blocks of the system are explained as well as their interactions. Also, the steps taken to achieve a design and a prototype system are explained.

3.2.1 System and Design Process

It is useful to look at the design from a system point of view at the conceptual level. The topic of wireless sensor networks and its applications are a popular research topic.. Many applications and solutions have been proposed and tried, but there is still much room for improvement. The network design can either be very simple or be very complicated with the interconnection of different types of devices. The network type required in this project is a simple central network which is easily implemented under the active RFID standard.

The system is categorized under the general title of wireless sensor network [28]. Different approaches are taken for each application based on the system requirements. The main parameters are:

- Data rate
- Delay
- Size

In a specific application different degrees of these parameters are required. For this project high data rate is not required but very short delay is desirable. Also the number of devices is small. The number of devices in a network is called the size of the network. The solutions to these network problems are mainly addressed by the protocol.

A wireless sensor network is a system that consists of multiple sensor nodes. The purpose of these nodes is to measure the physical parameter from the environment and forward the results to a central location for processing. The central processor running a user interface application is used to manage and process the data from the wireless sensors. There might also be routers used for a more effective distribution of data among the sensor nodes and the central processor. In this design, the system does not include routers but the other components mentioned are present.

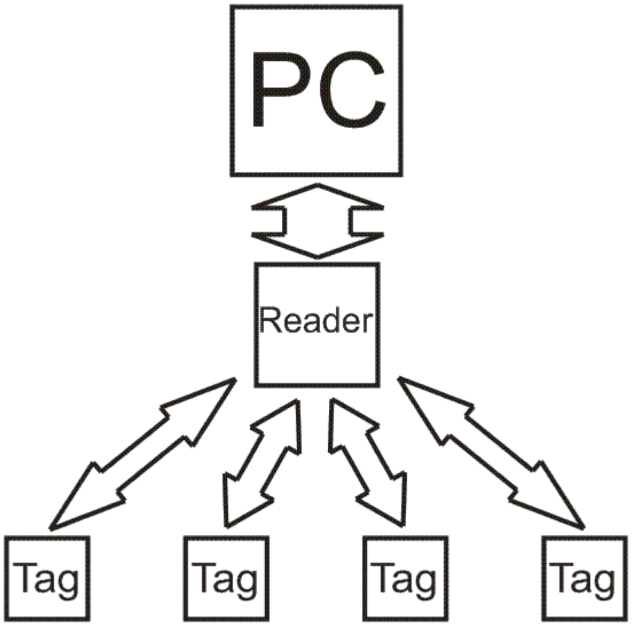

Figure 3.6: General Block Diagram of a Wireless Sensor Network.

In this system using the active RFID terminology, the nodes are called tags and the central wireless device is called the reader. Communication between the PC application and the reader is a wired serial connection which is provided through the USB port. The reader communicates with tags by means of RF signals according to the active RFID protocol. The reader is in charge of managing the tags and organizing the communication. All the basic functions are embedded in the hardware of the reader. The PC application stands at a higher level where it is responsible for organizing and displaying the incoming data and transferring the high-level commands from the user to the reader.

The complete process of designing an active RFID system consists of several steps. Any design approach should be able to achieve the following goals:

- Determining an appropriate communication protocol
- Selecting the hardware components required and making the proper system configuration
- Ensuring the hardware and software configure together properly and function as an integrated system
- Performing system and functional tests and measurements against the design specifications

The process depends on the technology being used as well as the experience level of the designer. The various steps taken in this project to achieve a complete designed is explained below.

To make an active RFID system it is necessary to understand the concept first. After reading some of the literature and the protocol, it was very useful to examine the existing designs, and to study the circuits and components developed by others for this purpose.

The first task was to read about the standard protocol which explains how the system functions. Once the basic understanding was formed it was then possible to cover other available technical literature. At that point, a general outline of the system was formed. With the help of that outline it is possible to conceptualize the different parts of the system and the devices necessary for its implementation.

The rapid advancement in semiconductor technology means a wider range of off –the-shelf components and devices that could be used in this project. It was necessary however, to go through the components' datasheets to get an understanding on what capabilities are offered at the present for RF technology, microcontrollers and other key components. The active RFID circuit technology that had a direct effect on the performance, size and cost of the circuits were the following:

- RF matching components and antenna
- Microcontroller or system-on-chip with multiple capabilities

There are other components required which can be chosen more freely but are still essential for the circuit and the system to work effectively. A careful selection of components that will result in an easier, faster and better design are:

- Voltage regulators and batteries
- PC interface such as USB
- Passive components and LED's
- Mechanical switches, battery holders, sockets and headers

Some manufacturers provide components with integrated functions and capabilities that can make the design process easier. An example is the CC2430 which is a single IC that has a microcontroller and an RF transceiver. When selecting different components, consideration should be given to those that come with extensive documentation, application notes, supporting hardware and software. For example, Texas Instruments provides many application notes for its microcontrollers in addition to easy-to-use development tools and programming interfaces. Another example is the USB transceivers from FTDI which provides a PC software framework to easily communicate with the transceiver. It is very useful for the designer to read these manufacturer application notes and guides to get a hands-on view on the design.

After deciding on the system and finalizing the technology and the components to be used, it is time to design the circuit boards. The low-frequency boards can be initially built with breadboards and tested before making a final Printed Circuit Board (PCB). Since radio-frequency boards are sensitive to parasitic capacitances and inductances they should be implemented on PCB's and soldered from the beginning.

The datasheets and design guides that might come with the main components are of great help with designing the circuit boards. Today the circuits are easily designed using different computer programs such as Cadence and Altium.

The components should be soldered onto the boards and different parts of the circuits should be tested. It is sometimes useful to solder different parts one-by-one and test them individually. When testing the circuit an oscilloscope is normally used to probe the different components to see if they are powered or producing the right signal and a multimeter may be used to measure voltages and to determine if there are any shorts or faults.

After the circuit is ready, the microcontrollers in the circuits are ready for programming. The embedded software that goes into the microcontrollers is developed using computer tools such as IAR Embedded Workbench. Additional hardware is required to connect the circuits for programming. Many cycles of programming and testing may be performed until the desired functionality is achieved by the circuits.

The active RFID reader will need to communicate with the user by means of a PC user interface application program. This application program will need to be developed and written.

The connection to the reader circuit is provided through one of the ports of the computer, preferably the USB port. A driver routine should be written which is compatible with the USB transceiver used in the reader. Sometimes the USB transceiver's manufacturer will provide the software driver the transceiver. The driver software provides the basic send and receive functions. The application will run inside the operating system. To develop such an application any high-level programming language is suitable. A basic user interface application with the ability to call the external functions from the driver software will do the job.

3.2.2 Protocol

Transmission of wireless data has a number of issues. A number of standard protocols have been developed to address some of the issues and provide solutions. The main questions that a wireless communication protocol is expected to answer are:

- What is the frequency band to be used and what kind of modulation is suitable?
- How is the data addressed to the different devices and what is the data packet structure?
- How can the algorithm deal with the collisions that occur in the shared communication medium?
- With how much data transfer delay and bit-rate is required from a source to the destination?
- What is the cost of implementation in terms of memory needed, system complexity and electrical power?

One may need to perform a trade-off between some of the properties of the protocol. For example, one might achieve a higher bit-rate with the cost of a greater complexity and power consumption. Therefore different applications may require a different protocol for the best performance efficiency.

To simplify the problem of communication and networking, the issues and solutions are generally organized in groups called layers [21]. The first layer is called the Physical (PHY) layer which addresses the frequency band, modulation and other physical properties of communication. The layer that deals with issues such as addressing collisions is called the Medium Access Control (MAC) layer. Depending on the scope and application of a protocol it might cover these or some other layers.

Wireless sensor networks are usually low bit-rate and need to be implemented with low cost solution. In most cases the sensor nodes are battery powered. Two of international standards that satisfy these conditions are:

- IEEE 802.15: low-rate wireless personal area networks (LR-WPAN)
- ISO/IEC 18000, Radio frequency identification (RFID) for item management [22]

These standards each clearly define a Physical and a MAC layer but leave other layers rather open to be tailored to specific applications. For example ZigBee Alliance [23] is a group that expands the IEEE 802.15 and provides additional resources and layers for the system. Another example is Real-Time Locating Systems (RTLS) [24] which provides a protocol based on the ISO/IEC 18000 for location sensing.

The protocol that is of most interest for this project is Active RFID. Therefore, the Active RFID protocol and the selected properties of the RTLS are explained. An overview of ZigBee is also provided for comparsion because of its popularity.

The active RFID protocol is aimed for applications where a reader is present and sends commands where a number of battery operated tags respond with their ID's. Also it is possible to include a small amount of data in each ID response. The commands and responses are transmitted wirelessly and in the format of packets. When a reader sends a command that multiple tags need to respond to, the responses are subject to colliding with each other [29]. In that case the checksum bits in the response packets show failure and the reader sends the command again. An algorithm is present in the protocol for the purpose of minimizing the number of collisions thus decreasing the average delay.

Data is transmitted wirelessly at f_c = 433.92 MHz using FSK. The HIGH and LOW frequencies are f_c+50 kHz and f_c-50 kHz respectively. Data is encoded using the Manchester format and sent.

Data is organized into bytes of 8 bits each, and the bytes are organized into packets. When transmitting bytes, most significant bit goes first and when transmitting packets, most significant byte goes first.

Each packet consists of a preamble signal, then the Manchester-coded data followed by a final signal. The preamble is a sequence of twenty pulses which are 30 µs HIGH and 30 µs LOW, followed by 42 µs HIGH (from tag to reader) or 54 µs HIGH (from reader to tag) and then followed by 54 µs LOW. The final signal is 36 µs of LOW. Each data byte consists of 8 data bits, least significant first, and a one stop bit which is a zero. A zero is 18 µs HIGH then 18 µs LOW. A one is 18 µs LOW then 18 µs HIGH.

In network studies, the structure of packets is included in another layer called the data link layer which is in close interaction with the MAC layer. Data is transmitted in packets. Packets are a certain sequence of bytes. Each group of bytes in a packet serves a purpose and is called a field. There is a mandatory field as well as an optional field in each packet. The optional fields are in square brackets in the following tables. Below is a list of possible fields in active RFID:

- Command prefix: The number 31 which is 00011111 in binary which indicates active RFID
- Command Type: Indicates the presence or absence of a tag ID and an owner ID in the packet which determines which tags should respond to this command
- Owner ID: A manufacturer of RFID systems can assign an owner ID for recognition among other systems
- Tag ID: A globally controlled unique number assigned to each RFID tag
- Reader ID: A number assigned to the reader which does not need to be globally unique

- User ID: A field for whatever purpose the user defines for it
- Command Code: Indicates the command itself
- Data: The optional data or parameters included in responses or commands
- CRC: Checksum for error detection
- Tag status
- Message length

Table 3: Reader to Tag Command Packet.

Command Prefix	Command Type	[Owner ID]	[Tag ID]	Reader ID	Command Code	[Data]	CRC
1 byte	1 byte	3 bytes	4 bytes	2 bytes	1 byte	P bytes	2 bytes

Table 4: Tag Response Command Packet.

Tag Status	Message Length	Reader ID	Tag ID	[Owner ID]	[User ID]	[Data]	CRC
2 bytes	1 byte	2 bytes	4 bytes	3 bytes	U bytes	D bytes	2 bytes

Active RFID uses a simple method called slotted ALOHA [21] for managing collisions in a shared medium. When a reader sends a command which does not include any tag ID, the tags receive the command, and all are required to respond, If they transmit their responses at the same time, collision occurs and the reader will not get any valid data. Therefore some kind of collision control is required.

After the command has been received by the tags, they need to transmit their answers within a specific length of time following the reception. That time is divided into a number of slots (N). Each slot is long enough for a packet. If more than one tag transmits an answer in the same time slot, collision occurs and data is destroyed in that slot. Therefore the tags need to take turns and send answers in a certain order. To make things simpler, the tags send answers in a random order.

In the slotted ALOHA protocol, each tag generates a random number R, where $R<N$, and sends its answer packet in the R_{th} slot. Still, there is the possibility that two or more tags could

choose the same slot and are cause a collision. To solve this problem, once the ID and CRC of a tag is successfully read, the reader will send a command to this tag and put it to sleep. Then the reader sends the initial command again and the tags that are not asleep will go through the same response process again. That process is called a contention round.

The reader keeps starting new rounds until it does not receive any reply attempt in three subsequent rounds. At this time, the reader stops sending anymore commands, and assumes that all the tags have been successfully read and is now all in sleep mode.

The RTLS is a variation of an active RFID. As the name suggests, this protocol has been developed for the purpose of locating objects. It however, has some properties that make it useful for other applications. An example is given to illustrate what can be built using the standard active RFID together with the RTLS protocol.

The RTLS uses only the commands included in the standard active RFID with one exception. A new command is introduced into the RTLS called the Blink command. Otherwise all the other properties such as frequency, modulation and packet remain the same.

A Blink command may be transmitted to a specific tag at a time by the reader. When a tag receives the command, it checks if it is intended for them. If so, they will respond with their ID and keep responding periodically in pre-defined time intervals. To avoid collisions a beacon may be provided by the reader. The tags may also randomize the timing when they transmit their Blink responses to some extent.

Blinking is originally intended for calculating the position of a tag when multiple readers receive the signals with different strengths and delays. There are several algorithms presented by the RTLS to calculate the location which are beyond the scope of this document.

After a tag is woken up and receives the Blink command, it does not need to be woken up repeatedly by the reader. It can wake itself up at the right times, blink and then go back to sleep. This response scheme saves battery power because the tags will no longer need to constantly check for a wake-up signal. This is because in the life cycle of a tag most of the battery energy is wasted in waking up and looking for a reacher command only to find no command was sent by the reader.

As explained before, the slotted ALOHA protocol can solve the communication collision problem; however, the penalty is an increase in communication delay. In some applications the blinking method can help reduce the delays caused by the numerous contention rounds of the slotted ALOHA broadcast commands. The Blink command opens opportunities to use collision avoidance methods that are more advanced and might be more suitable for certain applications.

The IEEE 802.15 standard was developed for low-data rate and low-cost wireless communication. The physical and MAC layers are defined in the standard. ZigBee uses this standard with a further developed higher layer. Although the technical details of ZigBee and IEEE 802.15 are beyond the scope of this document. Some of the most important differences between a typical ZigBee system and a typical active RFID system are simply explained as follows.

Three device types can be present in a ZigBee system as opposed to the two in active RFID. The first one is called the ZigBee coordinator. This device is similar to an active RFID reader. It organizes the communication and forwards the data to a PC or wherever the data is consumed. The second device type is called the ZigBee end-device. This device is similar to an active RFID tag. It is a low cost, low power device that is only capable of receiving commands and transmitting data back to other devices. The end-devices are not capable of communicating between themselves. The third device type is called the ZigBee router. This device is not presented in an active RFID system. It has all the capabilities of an end-device with the additional capability of forwarding data. That makes it possible to have advanced networking and routing in a ZigBee system.

The range of a low-power ZigBee device is designed to be around 10 meters whereas an active RFID device can be designed to have a range of 150 meters with the same power but at a lower bit-rate. This performance parameter is subject to variation but is given here to compare the two protocols. For both ZigBee and active RFID much longer ranges are possible but power consumption and costs need to be considered.

The main factors limiting the range are the transmitted power and receiver sensitivity which are standardized in ZigBee. Another limiting factor to consider is that the longer the range, the more end devices there are in the network that lead to more potential interference and can make the network unstable. On the other hand, ZigBee can support an average bit-rate (throughput) of up to 250 kbps which is generally higher than an active RFID.

ZigBee devices are capable of forming an advanced mesh network together without any further design required. Many IC manufacturers have IC's available with all the ZigBee protocol already implemented in them and that makes the design of the network devices much easier. A ZigBee network can expand up to 65000 devices. Whereas when a simpler network is required an active RFID may become a more attractive option.

3.2.3 Circuits and Components

The design of a tag circuit was completely done for this project including the component selections, board layout, assembly and test. Other circuit boards required for a fully functional solution are also explained here but they were not developed as that was not the main focus of the project.

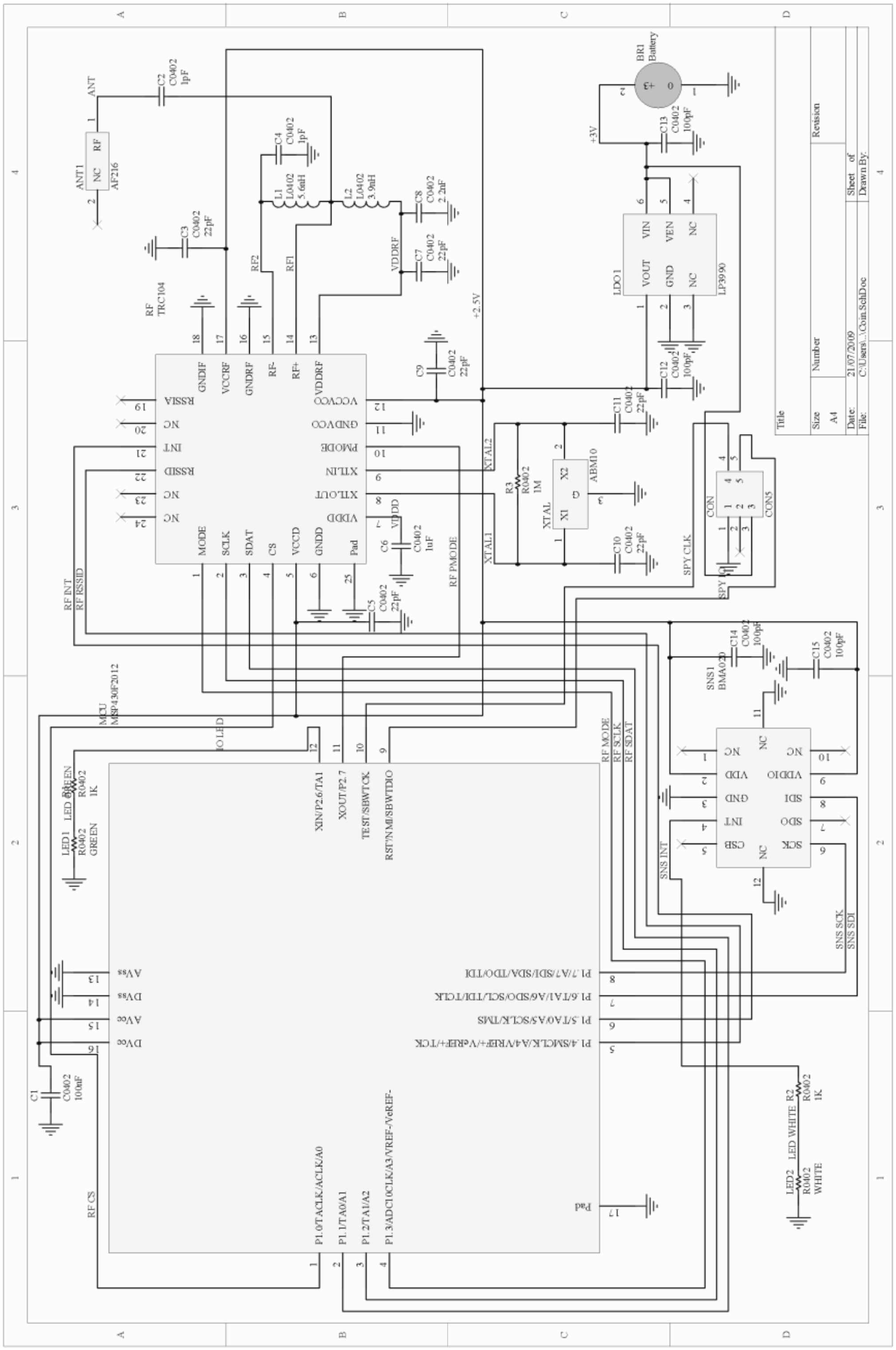

Figure 3.7: Tag Schematic.

An active RFID tag circuit should perform the following functions:

- Receive RF signal and determine if it is a command it needs to respond to.
- Transmit its unique ID number along with other required data according to the protocol in response to the commands.

It can send small packets of data along with its ID each time it transmits. The data can be from a sensor that is included in the tag circuit. For the mentioned functionality the following main components are included in the circuit:

- Microcontroller
- RF transceiver
- Sensor

Each of these components is available off the shelf with various options. There is a wide range to select from and many details to pay attention to. Details about the main components are explained in this section based on the functions that they are required to perform.

The microcontroller holds all the necessary data and logic, and manages the other components. Communication with a sensor can be performed over a serial or parallel port. Furthermore, it can be either analog or digital. One example is the SPI interface which is a serial digital interface provided in the hardware of many microcontrollers. Another example is a built-in analog to digital converter that comes with many microcontrollers. It can be used to read data from sensors that do not have digital circuitry. Other available controls such as putting the sensor into sleep mode or getting interrupts may also be considered. 2 pins to 4 pins and 300 bytes to 800 bytes of code memory are usually dedicated to interfacing with a sensor. The data rate does not need to be more than a few kbps.

The microcontroller must send data to and from the RF transceiver. Usually both are done via the same digital link. In most cases this link is a standard half duplex serial interface like a UART or SPI, but some transceivers do not provide such an interface instead they simply transmit or receive every single bit they are getting from or giving to the microcontroller. In that case, instead of a standard serial interface, the microcontroller needs to synchronize and communicate the data, and clock signal in the software.

The microcontroller needs to put the transceiver into various modes such as transmitting, receiving or sleep. Standard transceivers usually provide dedicated pins for these functions and need to be connected to the digital I/O pins of the microcontroller. Transceivers may also provide an interrupt pin for data interrupt when the data is ready. They may also provide a digital or analog Received Signal Strength Indicator (RSSI) pin which should be connected accordingly if needed. The RSSI indicates the strength of the signal at the input of the receiver without requiring the receiver to be on thus saving power.

Interfacing with the RF transceiver requires 3 pins to 8 pins and 500 bytes to 2000 bytes of code depending on how much of the protocol and packet forming is done inside the transceiver. The flash memory of the microcontroller is the place where permanent data is stored a tag. Almost all of the functions of an RFID tag are performed periodically and in sequence. The microcontroller needs to run an algorithm that triggers every event in its appropriate time.

Every communication needs a protocol and every protocol has an algorithm which may also need specific calculations. That is done by software inside the microcontroller. Depending on the complexity of the protocol, up to a few kilo bytes of required code is not unusual.

In most cases, an RFID tag is drawing power from a battery. This makes it critical to save as much power as possible. The logic in the microcontroller will turn on every component including the microcontroller itself only when required. Other times, the components are offline.

An RFID tag may need to have some kind of visual or audio indicator. The simplest form is having one or more LEDs connected to the I/O pins of the microcontroller. They can be turned on, off or blink to show certain events happening in the circuit. For example, an LED can be on every time the sensor is sensing a strong enough signal or every time the RF transceiver is receiving valid RFID signals.

When choosing a microcontroller for an RFID tag, there are certain specifications in the datasheets that are important. At the very least 2 kilobytes of code memory is required. It is safer to choose a microcontroller with 8 kilobytes or 16 kilobytes of code memory because in order to run an efficient protocol there are many little issues that the software needs to take care of. The microcontroller needs at least 4 to 8 general purpose I/O pins. It is easier if it also has two standard configurable serial ports. One 16-bit timer with capture/compare capabilities can be enough but the more the better. Power consumption is very important when comparing different microcontrollers especially in battery powered applications. Although the microcontroller does not consume the biggest portion of power in the circuit where an RF power amplifier is present, it is still important because every bit counts. Another feature to look at is the number of sleep/low-power modes and wake up times. In applications where the RFID tag needs to be very small, physical dimensions of the microcontroller become important.

The microcontroller finally chosen for our tag was the MSP430F2012 by Texas Instruments [25]. It has a very small footprint in its QFN package, 2 kB of programming memory, serial ports, adequate IO pins and ultra-low power consumption.

The transceiver is an IC which has all the RF circuitry required to transmit and receive data bytes in the form of RF signals. In most cases the oscillator requires an external crystal but most of the other parts of the oscillator are built into the IC. A matching network also needs to be connected to the RF input/output and to an antenna. Other than that, everything else is internal aside from decoupling capacitors.

The matching network and antenna can be implemented using PCB transmission lines or using lumped elements depending on considerations such as the application, size and efficiency. Where the IC provides a differential RF input/output, a balun may also be required.

Half-duplex transceivers usually share the same antenna and matching network to transmit and receive, and use an internal switch to change modes.

When laying out the circuit, the matching network and crystal must be mounted as close as possible to the IC because of the high-frequency nature of the signals there.

The transceiver has a PLL, a local oscillator, a power amplifier and all the other elements for a complete transceiver architecture (usually superheterodyne). In one end is the digital circuitry including the registers and a state machine to communicate data and configuration the bits and bytes with a microcontroller. In the other end is the RF circuitry which operates in a much higher frequency and processes RF signals.

Different RF transceivers come with different modulation types. For example, to be able to support the RFID standard, a simple FSK modulation is required where the IC needs to send or receive two fixed frequencies, keying between those frequencies for digital zeros and ones.

Sleep capabilities are very important in battery operated applications because if the receiver and transmitter are always on, they consume considerable power. Therefore they should be only turned on when required.

When an in-band RF signal is present at the RF input, a measure of its strength is indicated on the RSSI pin from which the microcontroller can decide to turn on the full receiver. When the receiver is awake, it will start demodulating the signal and pass the resulting digital signal to the digital circuitry. The digital end of an RF transceiver may have the ability to perform additional tasks like detecting packets and providing interrupts for a microcontroller. In any case the output is presented at a few pins which is then processed by a microcontroller.

In order to transmit, first the microcontroller turns on the transmitter and presents it with digital data on the appropriate pins. The transceiver IC might have the ability to organize the data into packets or even add an address and other data to it. If not, the microcontroller should take care of that. The result is a stream of synchronized bits at the input of the built-in RF transmitter which is modulated and transmitted through the RF output.

It is easier to choose the same transceiver for both the reader and the tag to ensure they are compatible. When choosing an RF transceiver for an RFID tag, there are certain specifications in the datasheets that are of importance. Power consumption of the transmitter and the receiver is one of the most important issues when choosing the transceiver especially for battery operated applications. Typical for this application is 10 mA to 30 mA at 3 V while actively transmitting or receiving. Also sleep modes and wake-up times should be taken into consideration. Output power of the transmitter is the key factor in determining the range of the wireless device along with antenna and matching network loss. Around 0 dBm is typical for small tag applications. The receiver should be sensitive enough to pick up the RF signal sent to the tag from far away. Around -80 dBm is typical for small tag applications. Some RF transceiver IC’s have certain digital functions built in. For example, some have the ability to put data into packets automatically and add the header and footer. Some even have a certain communication protocol built into their hardware like WLAN or ZigBee. Fewer external components mean less time to design. It is easier to have more of the circuit already built into the IC. Although some components like crystal and antenna need to be external. In applications where the RFID tag needs to be very small, physical dimensions of the RF transceiver becomes important. In active RFID applications data is communicated in bursts. If the bit rate is high enough, the bursts can be very short and reduce the possibility of collisions.

The transceiver chosen was the TRC104 [26] by RFM which is a 2.4 GHz FSK full transceiver in a very small QFN package with -95 dBm receive sensitivity, low power consumption, various sleep modes, multiple channels, various built-in functions and a bit-rate of up to 1000 kb/s.

The critical component in the tag is the sensor. A sensor is a device that produces a digital or analog signal according to a measurement in the environment such as temperature or acceleration. For each type of measurement there is a wide selection available with different technologies and functionalities.

The sensor is expected to sense the environmental variable and present the result as an electrical signal that is understandable by a microcontroller. It may also receive commands and configurations from the microcontroller. The two main functions of a sensor may be implemented on a single IC with 1 or 2 dies, or in separate IC's or any other form.

Different technologies have been used to convert various physical quantities into an electrical signal. One example is the MEMS technology which can be used to sense acceleration using capacitance. The sensing part of the sensor may be from a few millimetres up to a few metres depending on the application. The resulting electrical signal is the same.

Once an electrical signal is available from the sensor, it may be passed on to the microcontroller as-is, or it may be further processed or conditioned. Some sensors have an internal Analog to Digital Converter (ADC) and provide the result in digital form. Digital or analog filtering may also occur in the same IC. 2 pins to 5 pins are usually used to present the final data to the microcontroller.

There are a wide range of sensors available in the market for any given application. To find the best fit, it is important that the actual performance range is close to the specification indicated in the datasheet of the sensor. Sometimes it is easier if the sensor provides a standard digital interface instead of presenting raw analog data. Depending on the application, the physical dimensions of the sensor might be important. The sensor can be the major consumer of power in the circuit. Specifically in battery operated applications, it is better to choose the lower power and more efficient sensor. Since a sensor is meant to be in direct contact with a physical environment, it must be able to survive the conditions of that environment such as mechanical or electrical shocks or high temperatures.

After all the components were chosen, the schematic of the tag circuit was drawn in Orcad as it is seen in Figure 3.7. Then the layout was developed in Protel DXP which is shown in Figure 3.8. This circuit was fabricated and assembled with some of the functions tested. Figure 3.9 shows the fabricated circuit and the components.

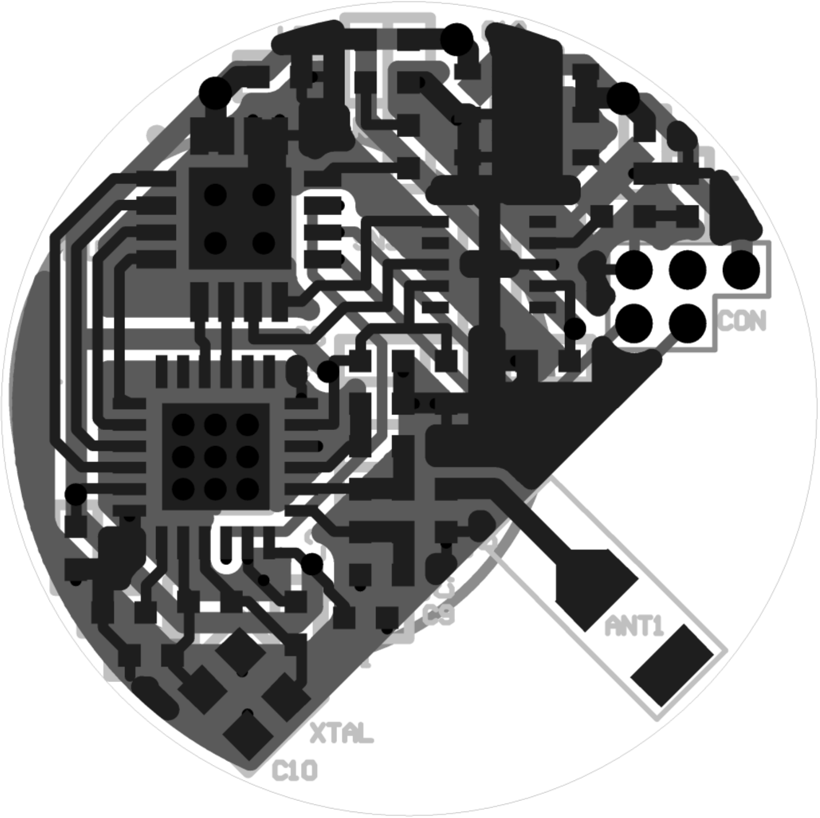

Figure 3.8: Tag Layout.

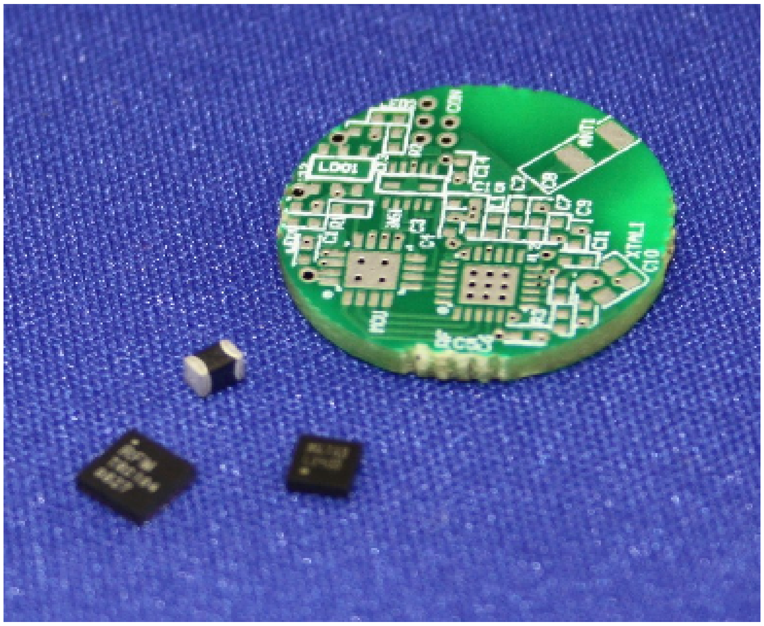

Figure 3.9: Manufactured Tag board and the required components.

An active RFID reader circuit should perform the following functions:

- Broadcast commands and data for tags via the RF link.
- Receive and process responses from tags, putting them to sleep and waking them up.
- Communicating with a computer application via a computer port.

It sends out commands along with the tag ID that the commands are meant for and gathers responses in return. The reader should also include additional logic to make sure collisions in tag responses are dealt with and the communication is timely and synchronized. For the mentioned functionality the following main components are included in the circuit:

- Microcontroller
- RF transceiver
- USB transceiver

Each of these is available off the shelf with different options. There is a wide range to select from and many details to pay attention to. The RF transceiver used in the reader can be identical to the transceiver used in the tag for this application and that makes the design easier. The microcontroller used can also be similar to the and one used for the tag with the exception of more code memory that is needed because it will be running more important parts of the protocol with less restriction on power consumption because it can get power from the USB port instead of a battery.

With USB transceivers readily available, all the microcontroller needs to do is to communicate the date to the transceiver over a standard serial protocol like an UART or SPI. No more than 200 bytes of code and 3 pins of a serial port are usually required for this task.

When choosing a microcontroller for an RFID reader, aside from what was mentioned about the tag, there are certain specifications in the datasheets that are of importance. To run the reader protocol, a 64 kilobyte microcontroller with 4 to 8 general purpose I/O pins and two or three 16-bit timers with capture/compare capabilities should be adequate. A reader is required to run certain algorithms embedded in the microcontroller at high speed. Therefore a fast and capable microcontroller is required. An example would be a 16 MHz CPU with features like hardware multiplier will be a good choice.

The USB IC transceiver provides the communication interface between the reader circuit and a PC through the USB port. At one end it has the built-in ability to connect to a USB port. At the other end there are a number of pins provided for a standard serial or parallel communication with a microcontroller with an optional supply voltage pin that the microcontroller and other circuit components can use.

A PC that is connected to the USB transceiver needs to run certain applications to be able to communicate with the transceiver. The port driver software must take care of the USB details and provide the necessary functions for the user interface programming. The USB port includes 4 wires:

- VCC
- Ground
- USBDM
- USBDP

The VCC pin can provide 3.3 V or 5 V that can be used to power the USB transceiver and even the rest of the circuit. The USBDM and the USBDP are used for serial communication.

The USB transceiver provides the serial interface pins to be connected to appropriate USB port pins and may also provide an internal voltage regulator to regulate the USB voltage. A USB transceiver must provide either a serial or a parallel interface to be connected to the microcontroller. 2 pins to 8 pins can be dedicated to this purpose. Standard serial protocols like the SPI or UART, or an 8-wire parallel register transfer may be used.

Since there is a wide range of manufacturers that provide USB transceiver IC's with various capabilities, some consideration is required for choosing the right one. A transceiver should be chosen that supports the type of communication that is intended in the circuit. The USB communication requires an oscillator, resistors and capacitors at various places in the circuit. Some USB transceivers provide these internally on the same IC. If the manufacturer of the USB transceiver also provides the PC USB port driver it would be much easier to work with. A good choice regarding all these issues would be the FT232R or something similar from FTDI.

In order for the circuits to operate, the microcontrollers must be programmed by the designer. The required programming interface can be divided into two parts:

- The manufacturers of microcontrollers usually provide a device which is connected to a computer and provides a programming interface with the microcontroller. That is called the programmer or debugger.
- A circuit board is required just to provide the proper connections and the few passive elements that may be required. That is called the programming board.

Where a programmer is purchased along with the microcontroller, a programming board is also needed to connect the debugger to the circuit boards containing the microcontrollers.

Standard microcontroller programmers provide a standard header with the appropriate wires containing the signals. One popular example is the JTAG connection which comes in a 14-pin header and the appropriate socket for the programming board. The programmer may require some capacitors and resistors to be connected to the wires.

The signal wires coming from the programming socket need to be connected to specific pins on the microcontroller. It is up to the designer to choose how to make the connection. Any header and socket or even direct wire connection can be used.

It is optional to include additional useful features on the programming board to help with design and debugging. One example is to provide power for the programming board. Another possibility is to have a current mirror on the programming board for measuring the power consumption of the circuit.

For the MSP430F2012 which was used in this project, TI provides a USB to the JTAG programming interface which can be used with the IAR Embedded Workbench to program the microcontroller. The output of this interface will need a few external components before it can be connected to the pins of the microcontroller. These extra components were mounted on a board and a connection was soldered for the tag board to be programmed directly.

Chapter 4

Minimizing Vibrations Using Wearable Wireless Sensors

4.1 System Overview

The objective of the system proposed here is to minimize the physical vibrations which are controlled by a set of voltages. The application is targeted at patients with Parkinson's disease who have a DBS electrode brain implant. The problem has been simplified down to a shaking hand which vibrates a certain way depending on the voltages applied to the DBS electrode. It is assumed that the effect of each voltage on the vibrations is unknown to the minimizing algorithm. In other words, the transfer function from the voltages to the resulting vibrations, although important is not calculated or approximated. Instead, the proposed system should be able to deal with any transfer function there might be, in the presence of other factors that cause variations with time. It is further assumed that as the minimizing algorithm is running, the system does not change significantly. It typically takes a few minutes for the minimizing algorithm to converge.

The system is simply a closed-loop circuit containing an optimizer as it is seen in Figure 4.1. To emulate the vibrating limb of a patient, a vibrating "hand" was designed with two motors causing the vibrations. The "hand" will vibrate differently depending on the voltages applied to the motors. A vector of two voltages input into the motors replaces the vector of the voltages applied to the implanted DBS electrode. To achieve the minimum, the optimizer which is an algorithm in MatLab tries different voltages and each time measures the output in terms of the vibrations. The vibrations are measured with a wearable wireless accelerometer and the result is presented to MatLab by means of a wireless reader and appropriate software. Then the optimizer intelligently guesses new voltages that might produce less vibration and continues trying until it decides that minimum vibration has been reached.

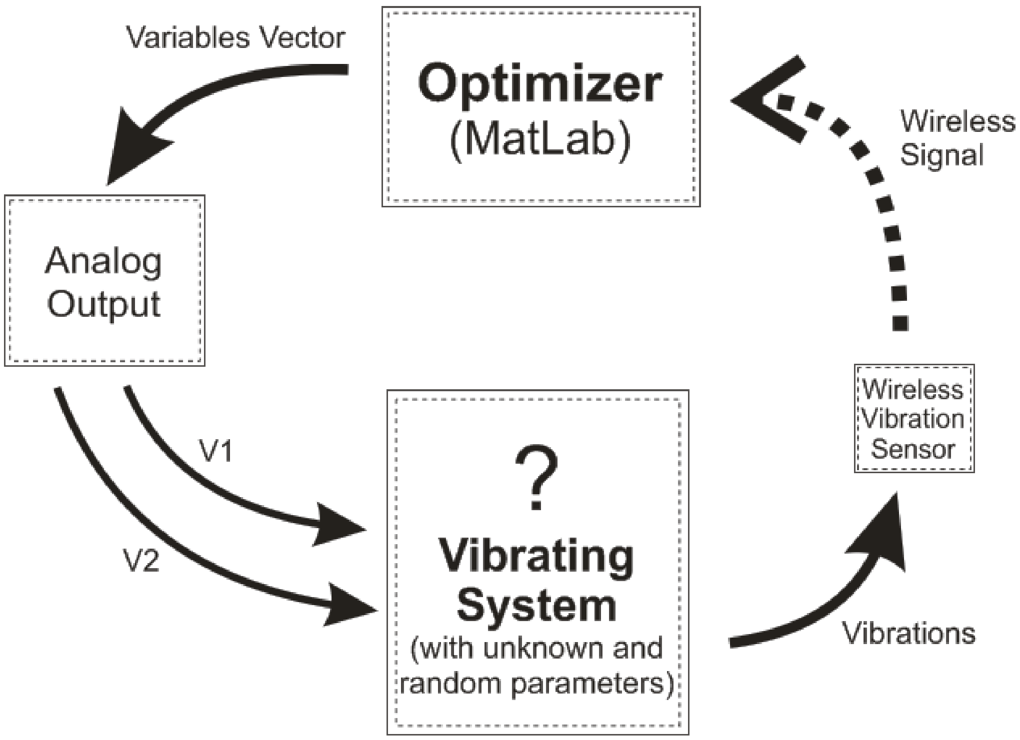

Figure 4.1: Block Diagram of the System.

4.2 Hardware

One of the main challenges in this project was to integrate all the pieces of hardware and software to realize the optimization loop. The hardware provides a structure in which everything can be measured and observed. The hardware used in the optimization loop is explained in this section.

4.2.1 Vibrating Structure

A structure was built to hold two shaking units which are small DC motors with unbalanced loads attached to their shafts. The motors used are Solarbotics GM 18 geared motors that has a maximum of 556 Rounds Per Minute (RPM) at 6 V, and a 388 gm·cm stall torque at 528 mA. A plastic structure is attached to the shaft with some weight on one end so that when the motor rotates the structure shakes. Figure 4.2 shows the motor with the attachment. The two vibrating motors are positioned at a 90° angle with respect to each other so that we can have vibrations in two different directions. Different voltages (0 Volts to 6 Volts) can be applied to the two motors independently.

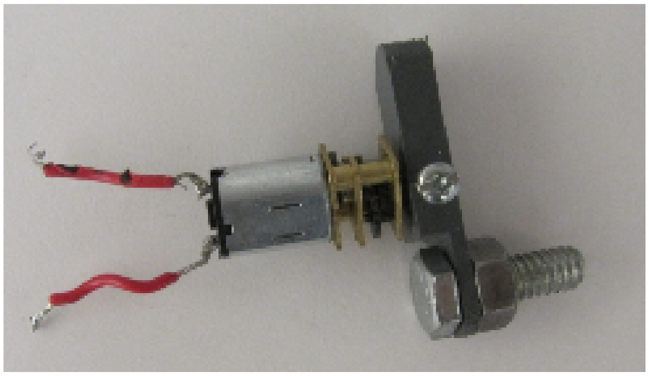

Figure 4.2: DC Motor with the shaking head attached to the shaft.

The amount of shaking of the structure as a whole is determined by the voltages applied to the motors. A given property of the vibration can be taken as the output which is a function of the input voltages. The function is most likely to be non-linear and partially dependent on uncontrolled conditions. In any case, when 0 Volt is applied to both motors it will result in no shaking of the structure. Figure 4.4 shows the assembled structure.

A driver circuit was designed to drive the motors in a very specific manner. The driver circuit takes two voltages as inputs in the range of 0 Volts to 5 Volts and provides the necessary current and voltage to the motors. To better simulate the conditions of a real patient, the driver circuit is designed to have a transfer function with one minimum value. Certain input voltages result in 0 Volts being applied to motors. Any other input voltage will result in a non-zero voltage on the motors and therefore vibrations. Furthermore, the input voltages that stop the shakings are adjustable by changing the position of two sliders in the driver circuit. These sliders are introduced in the circuit to simulate the random nature of the actual DBS problem. Figure 4.3 shows the constructed driver circuit.

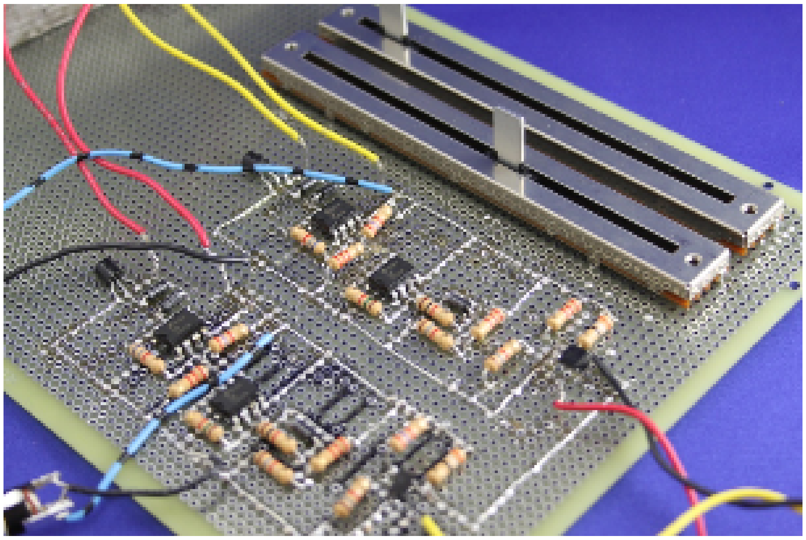

Figure 4.3: The driver circuit for the shaker.

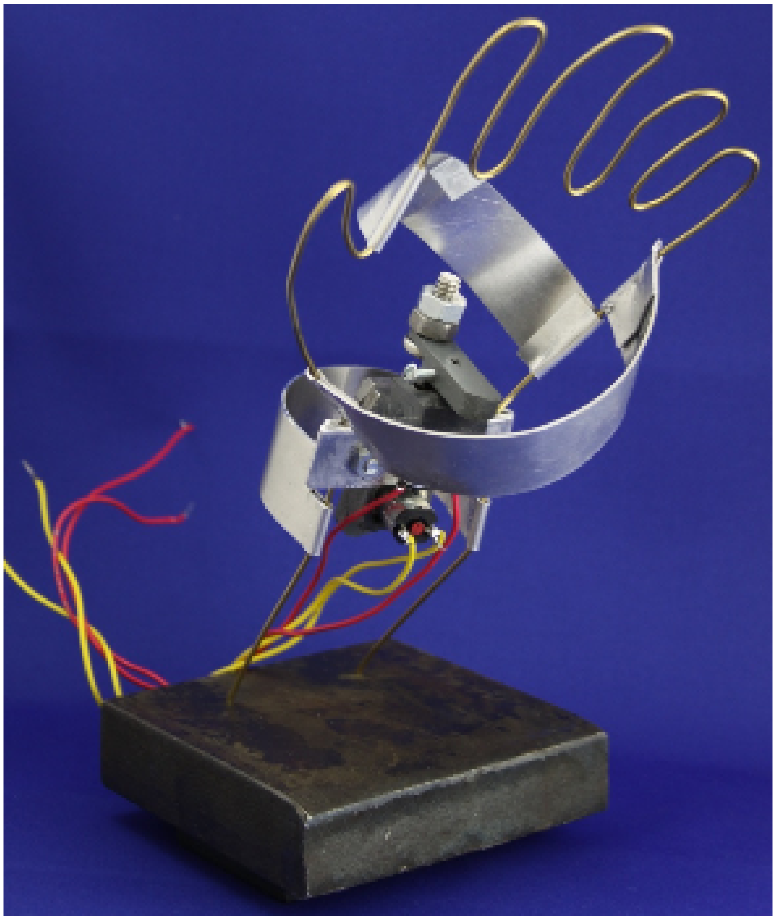

Figure 4.4: The shaking "hand".

4.2.2 Analog Output

A multi-purpose IO device from National Instruments (NI-USB-6008) was used to apply analog voltages to the target system outside the computer. The device provides 16 digital IO channels, 4 analog inputs and 2 analog outputs. The analog outputs provide a voltage range of 0 Volts to 5 Volts (12 bits resolution) and can drive 5 mA each. Extra circuitry is required to provide the current required by the motors. The device is attached to the computer through a USB port and supported in MatLab. We can apply any voltage in the range 0 Volts to 5 Volts to the external world from within the MatLab code if the driver software provided by the vendor is installed properly. NI-USB-6008 is shown in Figure 4.5.

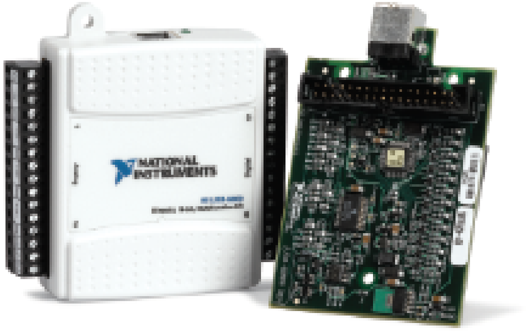

Figure 4.5: NI-USB-6800 Digital/Analog Input/Output.

4.2.3 Wireless Sensor

One of the key components in this system is the wireless sensor. In order for this application to be feasible a suitable sensor should be used. The sensor should sense the vibrations accurately enough, be inexpensive, light-weight and consume little power so that it could be conveniently attached to the system and be powered from its own battery. The design and specifications of sensors for these applications is a topic covered in the project and is explained in more details in Chapter 3 of this thesis.

The sensor used here is the ZStar3 from Freescale [30] (Figure 4.6). The sensing system consists of a coin-shaped wireless board which incorporates a transceiver, a microcontroller and an acceleration sensor chip to measure the vibrations and send the data to the reader. The reader collects the signal from the wireless board(s) and feeds the data to a computer through a USB port. Freescale also provides all the necessary software and drivers to easily access the functions of the sensor board from the computer.

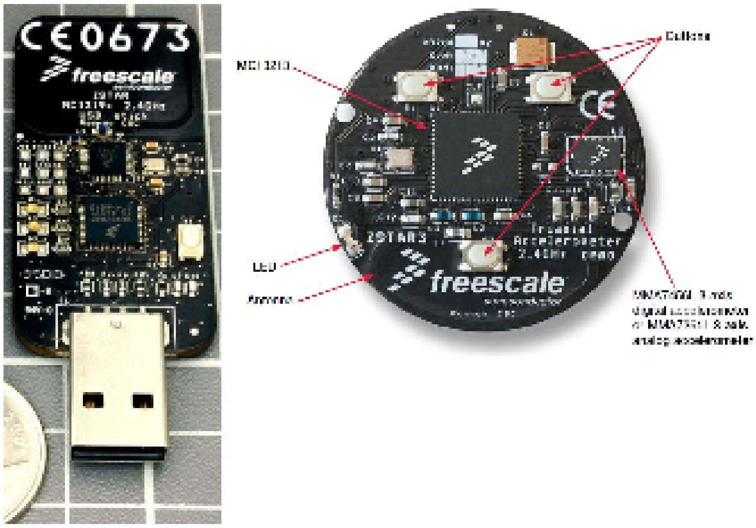

Figure 4.6: ZStar3 wireless accelerometer and reader – Courtesy of Freescale [30].

The sensor integrates a Freescale MMA7361L accelerometer and a Freescale MC1321 transceiver on the board. The protocol for the wireless communication between the sensor board and the reader is ZigBee. The sensor board is battery powered while the reader is powered by the USB port. The accelerometer can sample acceleration along 3 axes at up to 120 samples per second within ±4 g. The block diagram and other specifications of the boards are provided in the datasheet of the product. The schematics are in the appendix. Figure 4.7 depicts the block diagram of the boards.

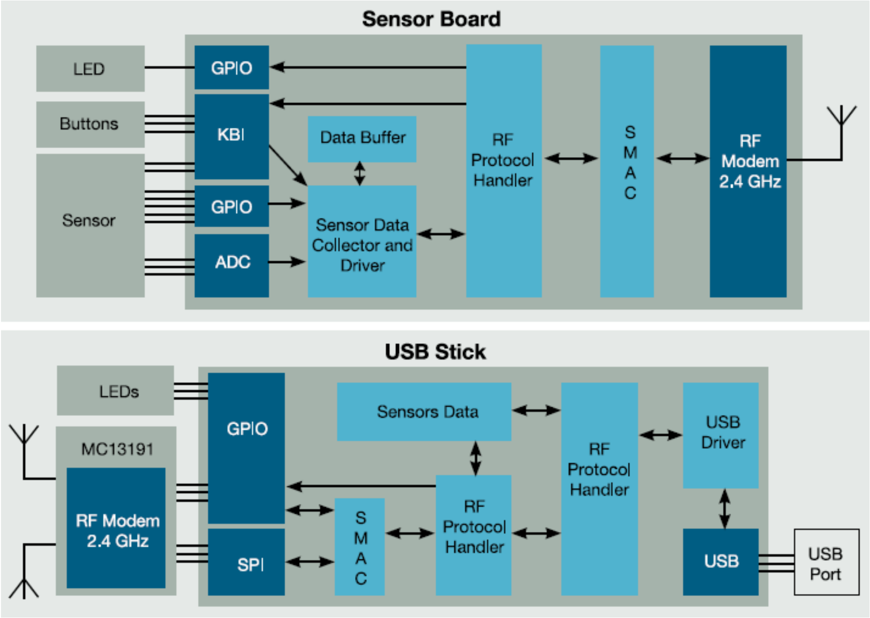

Figure 4.7: ZStar3 system block diagram - courtesy of Freescale [30].

4.3 Software

In this system we have used the Optimization Toolbox of MatLab as the optimizer. To complete the optimization loop, a program needs to get the data from the sensor reader and forward it to MatLab. For this purpose, an application was developed using C#. The libraries provided by Freescale for the sensor product were a great help in the development. The application starts up the sensor and sets the sampling rate. Then by clicking a button (Figure 4.8) it starts writing the incoming stream of acceleration data into a file. The file can contain an arbitrary number of samples which is set by the application. Each new sample is appended to the end of the file as a line of text and the first line which is the oldest sample is deleted instead.

Figure 4.8: Sensor Read Application Window.

The way the system works is that MatLab looks at the whole system as a function with 2 numerical inputs in the range of 0 to 5 (or a vector of length 2) and a numerical output which is some indication of a vibration. The vibration is represented by $f(x)$ where $x = (x_1, x_2)$, and x_1 and x_2 represent the voltages applied to the motors. The function is written in MatLab code and controls all the necessary tasks to deliver the voltages to the external system and provide the resulting vibration measurement. In the function, first a command is sent to the analog output device to set the voltages to the given numbers. Then the function holds the execution to wait for the system to respond. Then it starts reading the file which contains the vibration data and performs the necessary calculations to come up with a numerical indication of the vibrations output. As MatLab reads the file of raw data, all the necessary processing of the signal such as filtering and averaging is done within the MatLab code. The source codes are available in the appendix.

4.4 Date Analysis

The raw data streamed in from the vibration sensor should be interpreted and processed to make it useful for the optimization process. It is helpful to know what needs to be minimized and try to measure exactly that. The first thing was to acquire a sample of the raw signal from the two states: one when the model hand is shaking very hard and the other when it has minimum vibrations as in Figure 4.9. The signal with a higher amplitude shows the strong vibrations from the undesirable state. The other signal shows the vibrations from the desirable state when the person is not experiencing vibrations.

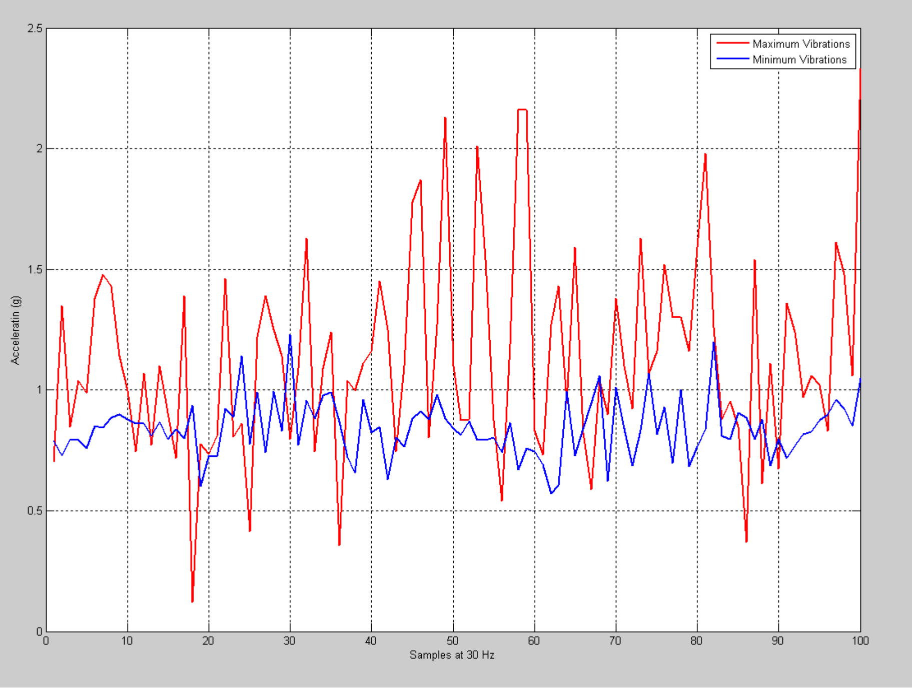

Figure 4.9: Raw signal from maixmum and minimum vibrations.

The first obersvation about the signals is that they show absolute "acceleration" and the unit is g. 1 g is equivalent to the acceleration of a free-falling object due to the gravity of earth which is equal to 9.80665 m/s^2**.** Acceleration is a 3-dimensional vector and what is important here is the length of that vector. Another obersvation is that our model hand (pretty much like the actual hand of a person with PD) may vibrate in a limited frequency band. Moreover, the vibrations which are important in such applications fall into the 0.5 Hz to 15 Hz range. Thus, a band-pass filter can be used to capture and eliminate the unwanted DC and high-frequency noise without loss of useful information.

Figure 4.10 shows 6 seconds of vibration data sampled with a rate of 120 Hz with any frequency component below 1 Hz and above 10 Hz removed. The signals have a good resolution because the sample rate is 12 times the highest frequency present. A higher voltage applied to both motors results in a vibration signal with higher amplitude and lower voltage results in lower amplitude. The RMS of the strong signal is around 2.5 g and that of the weak signal is around 0.7 g.

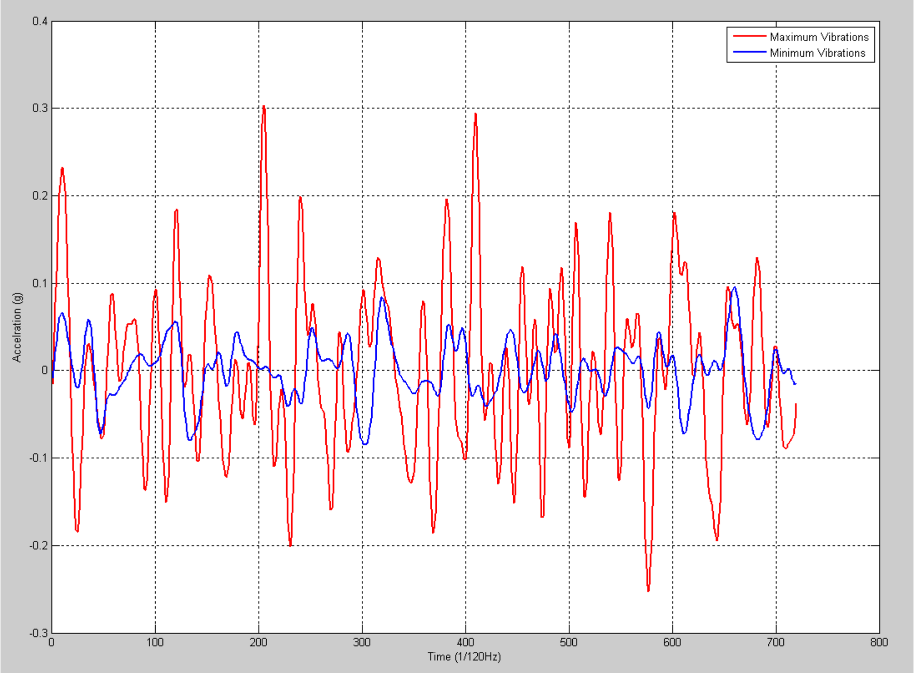

Figure 4.10: Maximum and minimum vibration signals after filtering.

The voltages applied to the motors were swept simultaneously in the range 1 Volt to 5 Volts over 21 points and at each point the vibration signal was recorded for later processing and comparison. This process was repeated 6 times and at the end, 6 sets of signals were available, each set showing the response of the shaker to the 21 different voltages. The goal of this analysis is to find the feature of the vibration output signal to be optimized. The function to be optimized should be a consistent indication of the shaking and should be preferably varying smoothly with the voltages and exclude as much noise as possible. A few features of each signal was calculated in all recorded sets and plotted. Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14 show the different features of the signal when the voltage applied to motors is swept in its range. These features are intensity, modulation, periodicity and the rate of movement. Intensity is calculated by taking the RMS of the time signal. Modulation of a signal is the range the auto-covariance of the signal covers. Periodicity is the energy in the dominant frequency over the total energy of the signal. The rate of movement is simply the dominant frequency in the spectrum of the signal.

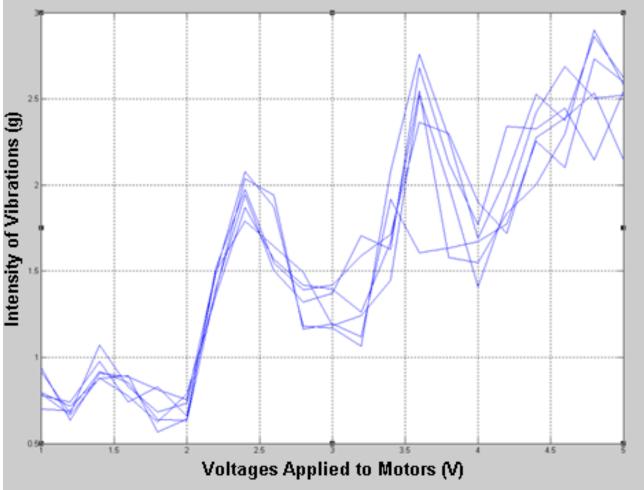

Figure 4.11: Intensity of the vibration signal VS motor power for 6 attempts.

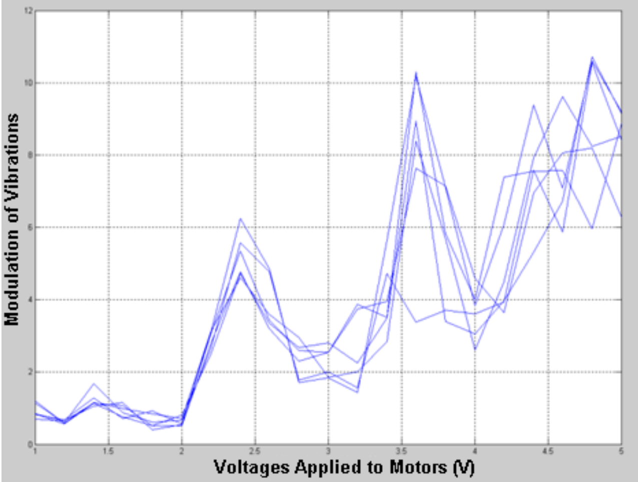

Figure 4.12: Modulation of the signal VS motor power for 6 attempts.

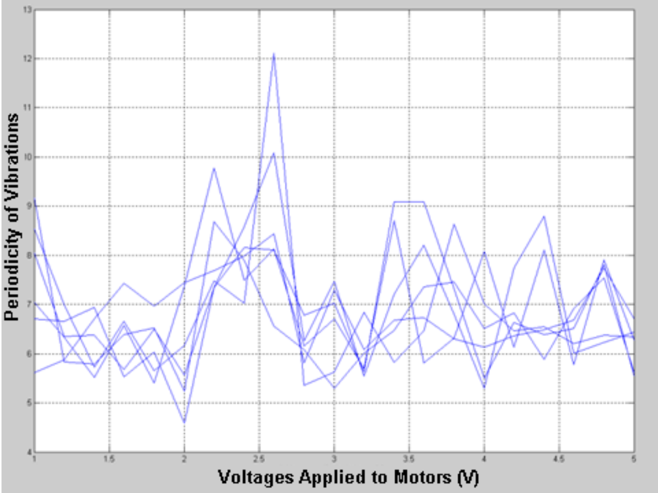

Figure 4.13: Periodicity of the signal VS voltage for 6 attempts.

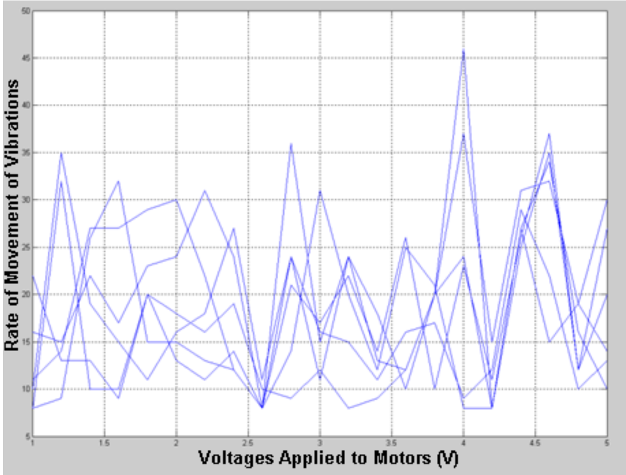

Figure 4.14: Rate of movement of the signal VS motor power for 6 attempts.

It was immediately concluded that the intensity and modulation of the signal provided a better indication of the vibrations compared to the periodicity and rate of movement. The reason that measurements relying on the frequency content do not provide consistent indications was because the spectrum is too noisy. Figure 4.15 shows the maximum and minimum signals along with their Fast Fourier Transforms (FFT). It is observed that with motors running more slowly the frequency content is shifted to the overall left as expected, but because there is too much noise in the spectrum, the dominant frequency does not directly correspond to the rotation of the motors.

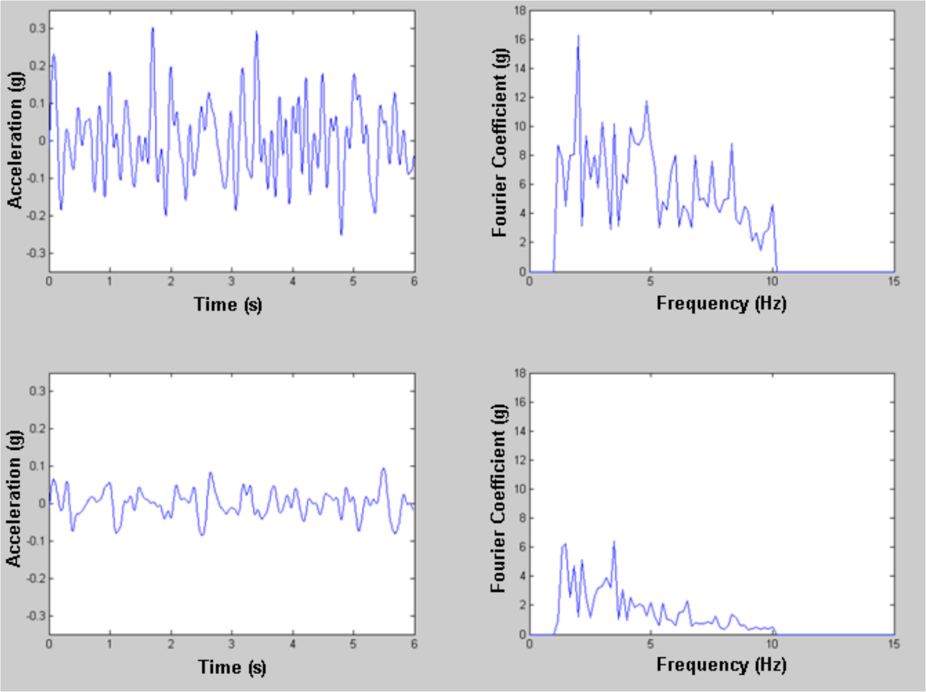

Figure 4.15: Maximum and minimum vibrations (left) and their frequency spectrum (right).

It was decided that either the intensity or modulation of the signal or a combination of the two should be used as the quantity to be minimized. If the intensity is selected as the objective function for optimization, it is likely to reach good results because of the consistency that it shows. There was a further attempted to construct a smoother and less noisy function out of the measurements using the following techniques.

One technique that was attempted was to save the last measurement and allowing only up to a certain rate of change to be made in the next measurement. In other words, every time the samples are acquired and the RMS is calculated, that number is saved onto the hard disk. Then, when new voltages are applied and the associated RMS is calculated again, the new RMS is compared with the saved one. If the slope of the change is more than a set limit; the value is trimmed to fit within those limits. After trying that, it was found that it would not help the situation as much as expected.

Another attempt was to quantize the measured values into fewer levels. The motivation for using this technique was that the accuracy of the measurement is not as important as having a smooth and less noisy function, since the goal is to find a point near the minimum. It was observed however, that quantization even with very coarse levels would not eliminate enough of the unwanted variations. Figure 4.16 demonstrates the effect of these attempts on the intensity function.

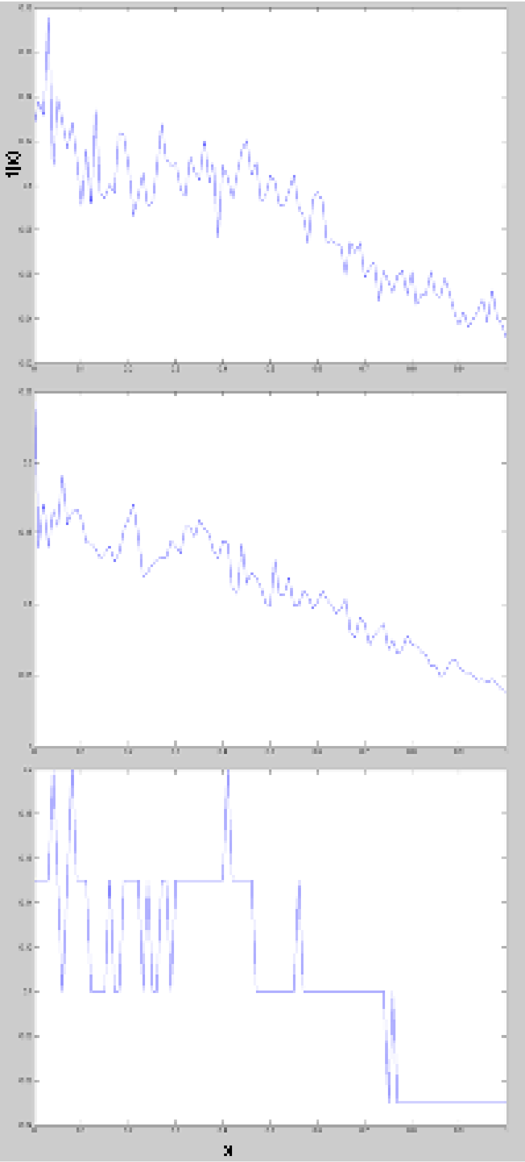

Figure 4.16: Original function (top), function after slope-limiting (middle) and after quantization (bottom).

4.5 Optimization

An optimization problem can be formulated as follows [33]:

$\min f(x)$ *subject to:* $ci(x) = 0,\ i \in E,\ ci(x) \geq 0,\ i \in I.$

$x \in IRn$

Where c_i*,* $i = 1, 2, \ldots$ *are the equality and inequality constraint functions, E is the set of indices for equality constraints and I is the set of indices for inequality constraints.*

The function *f* is called the objective function. The general case finds the minimum of *f* and in case the maximum is desired *–f* can be used as the objective function. The basic theory of optimization starts with optimizing smooth functions under no constraints, meaning the function is differentiable to the first and second degree and all the entries in the input vector can take any value.

Unconstrained optimization of a single variable function is performed by starting at an initial point at which the function is evaluated. Then the derivative is used to guess a suitable direction for the next point which causes a decrease in function value. The move from one point to the next is called a step. The length of each step can be set to a constant number or be changed each time to deal with variations in the function. The second derivative can also be used to get better convergence. The algorithm can keep iterating this way until an acceptable minimum is reached.

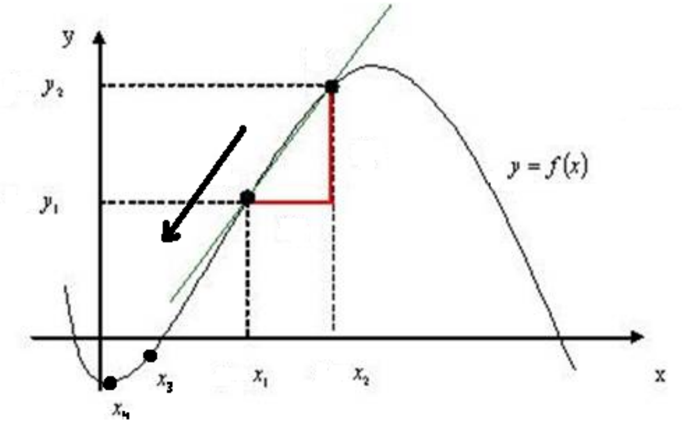

Figure 4.17: Guessing the direction of next step by approximating the gradient.

If the function takes vectors as an input instead of a number, the same concept applies except that instead of a derivative we have the gradient $\nabla f(x)$ and instead of a second derivative we have the Hessian $\nabla^2 f(x)$. Note that if explicit functions for gradients are not available some algorithms can approximate them using function values. There are different optimization algorithms and they vary in many ways including the way they find gradients and calculate the step direction and length. The basis of most of these calculations is found in Taylor's Theorem [33]:

> *Suppose that* $f : \mathrm{IR}n \rightarrow \mathrm{IR}$ *is continuously differentiable and that* $p \in \mathrm{IR}n$. *Then we have that*
>
> $f(x + p) = f(x) + \nabla f(x + tp)T p$
>
> *for some* $t \in (0, 1)$. *Moreover, if* f *is twice continuously differentiable, we have that*
>
> $\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp)p\, dt$
>
> *And that*
>
> $f(x + p) = f(x) + \nabla f(x)T p + 1/2\, pT\nabla^2 f(x + tp)p$
>
> *For some* $t \in (0, 1)$.

MatLab includes a number of optimization algorithms and has a user friendly interface. Normally it takes a function and a starting vector with the use of specified algorithm and settings to find the minimum of that function.

The first challenge is to select an optimization algorithm which is suitable for this problem. Different algorithms are suitable for different problem types and the smart choice of an algorithm determines if it converges and at what speed convergence occurs. The most obvious way to choose an algorithm for a certain problem is to try all algorithms a few times and see which ones converge. If we use our actual vibrating system for the trial and error process it will take an unnecessarily long time. Instead, to initially narrow down the algorithm selection, an example test function was constructed with properties similar to the output of the actual system. The test function for the optimizers needs to be a positive 2-variable function with a minimum value and some randomness. Such function was defined as follows:

$Test1(x_1, x_2) = (a.x_1^4+b.x_1^3+c.x_1^2+d.x_1+e).\ (a.x_2^4+b.x_2^3+c.x_2^2+d.x_2+e)$

Where $a = 0.0019196$, $b = 0.22837$, $c = 3.0209$, $d = 7.2808$, $e = 28.811$

And x_1 and x_2 are the variables and r is a random number in the (-10,10) interval.

Test1 evaluated at 100x100 points is plotted in Figure 4.18. The minimum occurs somewhere near the point (7, 7) with a value of close to 100. This function was passed to the solvers in the optimization toolbox to see which one deals with it better.

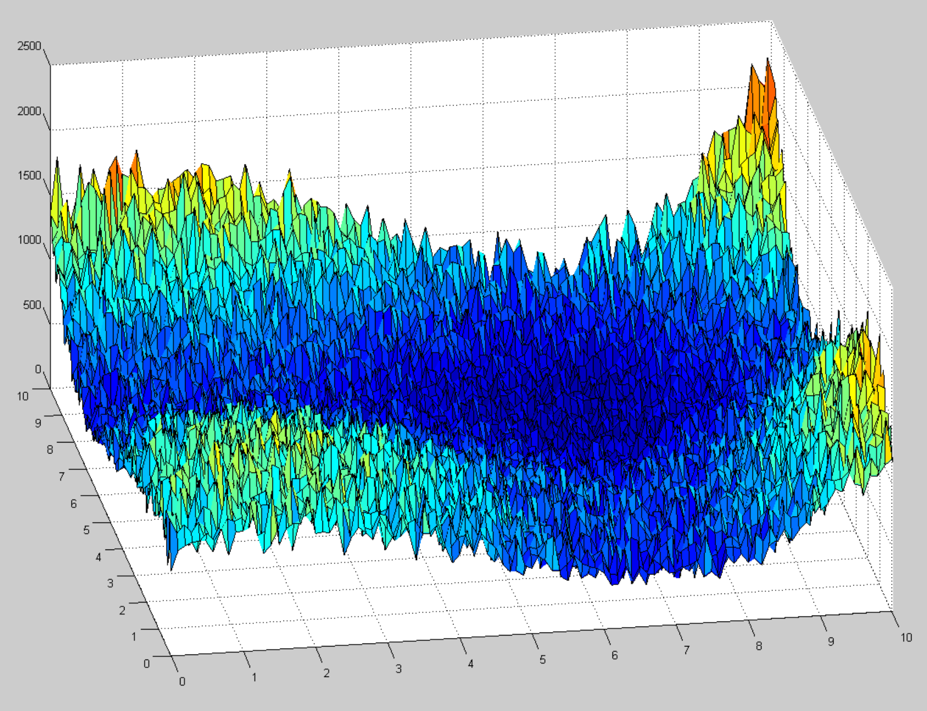

Figure 4.18: Test1 evaluated at 100x100 points.

Among the solvers, “fmincon” which is a standard constrained nonlinear optimization algorithm tried to solve the problem but showed no chance of converging to the answer. The reason that these kinds of solvers can not find the minimum of this test function is that they all use approximated gradients to find the direction. At each evaluation, the function returns a value which contains considerable noise and as a result, any attempt by the algorithm to use gradients to find the right directions is doomed to fail. One obvious way, is to pass the resulting function through a filter or fit it to a smoother curve. However, for that we need the whole function where the enitre point of optimization is to achieve the minimum with as few evaluations of the function as possible.

Aside from using filtering and similar techniques to reduce the noise in the objective function, another way to improve convergence is to use a different class of solvers called

Derivative-Free Optimizers (DFO). “Pattern Search”, “Genetic Algorithm” and “Simulated Annealing” are examples of DFO which are also available in the optimization toolbox of MatLab. They were tried on the *Test1* function and the results were observed. “Pattern Search” converged to a low value most of the time but the value was not close enough to the expected minimum. It converged to around 600 when the function can return values as low as 100. “Genetic Algorithm” is a powerful algorithm with many adjustable parameters to solve different problems but with default settings it was very far from a good convergence. “Simulated Annealing” on the other hand showed excellent convergence every time. After less than 200 iterations it finds the true minimum area. Figure 4.19 shows the convergence plots of the mentioned solvers on *Test1*.

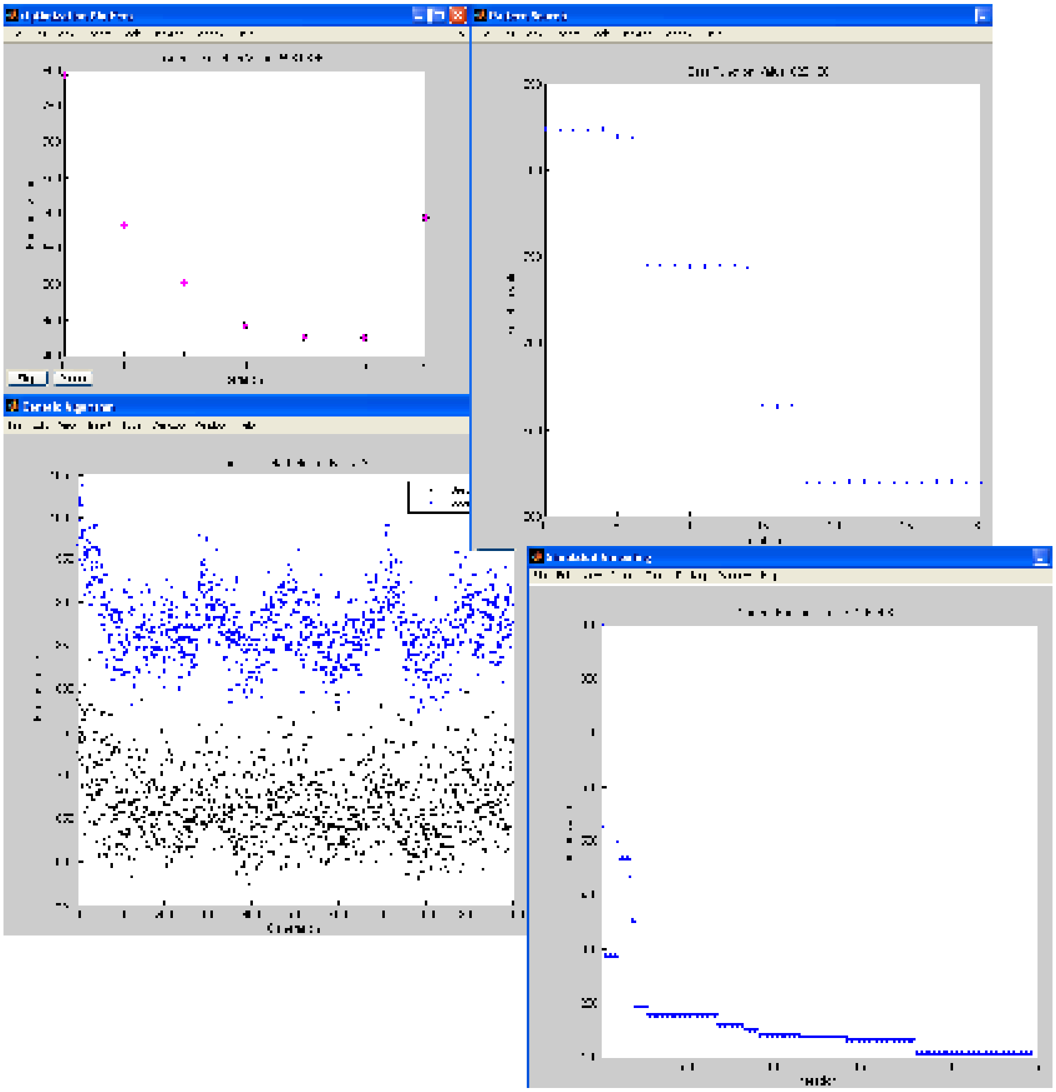

Figure 4.19: The four optimizers that were attempted with Test1.

Simulated Annealing was chosen as the optimization algorithm to be used in this application. It provides a very good combination of ease of use and robustness [32]. This algorithm which is inspired by thermodynamics has properties that make it suitable for dealing with randomness in the objective function. It mimics the behavior of a slowly cooling system by introducing a parameter called the temperature (T) in the algorithm. T turns out to be the most important parameter to adjust to improve the convergence. Suppose that the algorithm is trying to find the minimum of the function *OBJ(x)*. It starts by evaluating the function at an initial point and recording the value. Then it moves on to a second point taking a step:

$x_{new} = x_{old} + r_1.V$

Where r_1 is a random number between 0 and 1 and V is the step length.

The function is evaluated at this new point. If the new value is less than the old one it is accepted meaning that the algorithm chooses it as the current point and repeats the process from there. That is called a downhill move. But if the new value is not less than the old one, it still could be accepted given that a certain condition is satisfied. In other word, an uphill move is also possible under some conditions. Usually optimization algorithms always move downhill because the goal is to find a minimum but Simulated Annealing provides the probability of some uphill moves which makes it very flexible. To decide if an uphill move should be done, a value is calculated called the "acceptance function". The standard acceptance function p is defined as follows:

$p = exp(-(OBJ(x_{new})-OBJ(x_{old}))/T)$

If p is greater than r_2, a random number between 0 and 1, then the uphill move is accepted. Obviously a lower value of T will result in less chance of an uphill move and less acceptances (meaning more rejections). The algorithm is designed in such a way that if too many rejections occur, the step length is decreased. Also T is decreased every time using an exponential decay:

$T_{new} = r_T.T_{old},\ r_T<1$

That means more rejections and further reduction in the step length occur as the algorithm moves on. That simulates a process of "cooling" down to the minimum points slowly but surely. If the initial T is too low, the algorithm might never get to the minimums and will freeze around a wrong point. If the initial temperature is high enough, with a proper cooling schedule the algorithm converges, but very high "temperatures" may result in too many function evaluations which might be unnecessary and costly. Figure 4.20 shows the flowchart of the complete optimization process.

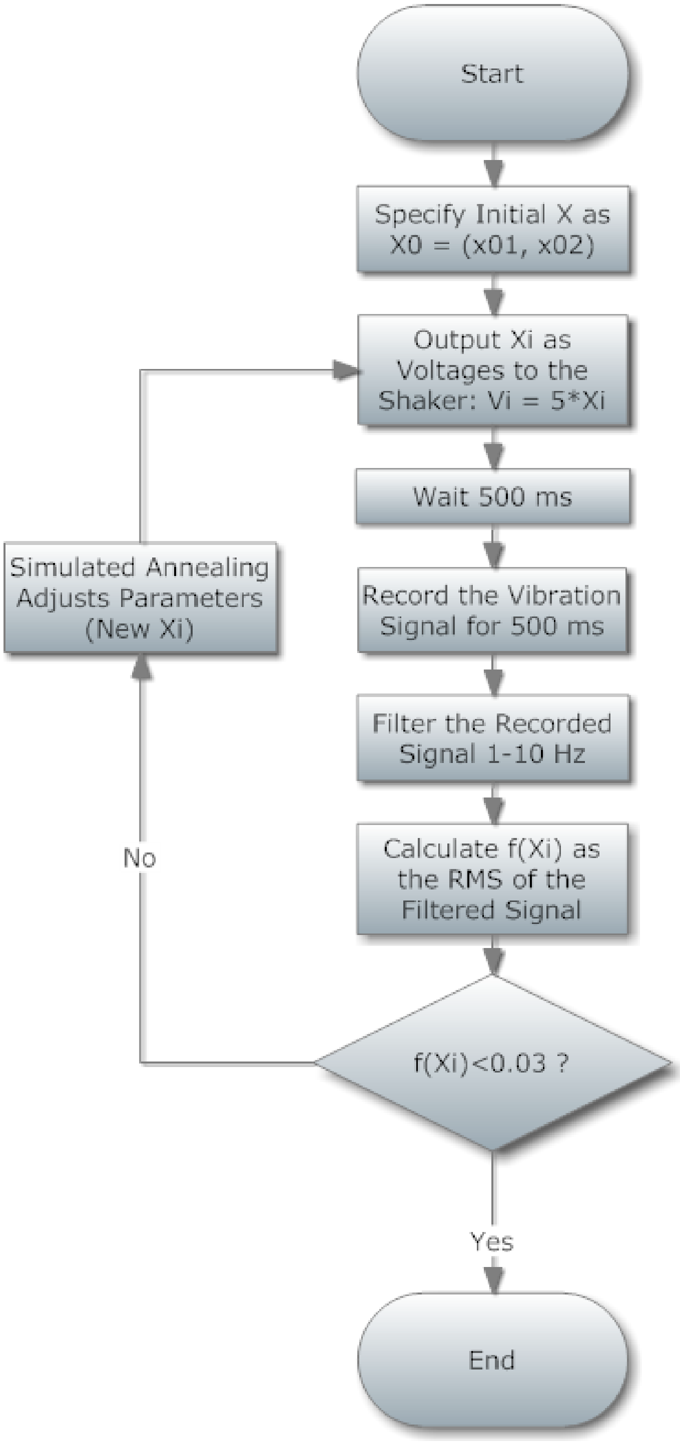

Figure 4.20: The process of finding optimum voltages to stop the vibrations.

4.6 Results

To test if the closed loop optimization is successful in finding the voltages that minimize the vibrations an additional circuit was designed for driving the vibrators in a more manageable way. The driving circuit takes the optimization variables as input voltages, combines them in a certain way with other voltages and puts the results on the motors. The way the driver circuit combines the variables with other voltages is such that only a certain set of input voltages can make the motors stop while all other values cause intense vibrations. Furthermore, the optimum point required to stop the vibrations can be adjusted in the circuit. The circuit in Figure 4.21 realizes this functionality as shown in the circuit simulation in Figure 4.22. At the first stage, a variable voltage is added to the input voltages. The variable voltage is set by a variable resistor. The added voltages are applied to an OpAmp-Diode circuit with a V-shaped characteristic meaning that the output of it is minimum for a value somewhere in the middle range of the input. The output of this circuit is passed to an amplifier to increase the "sharpness" of the dip.

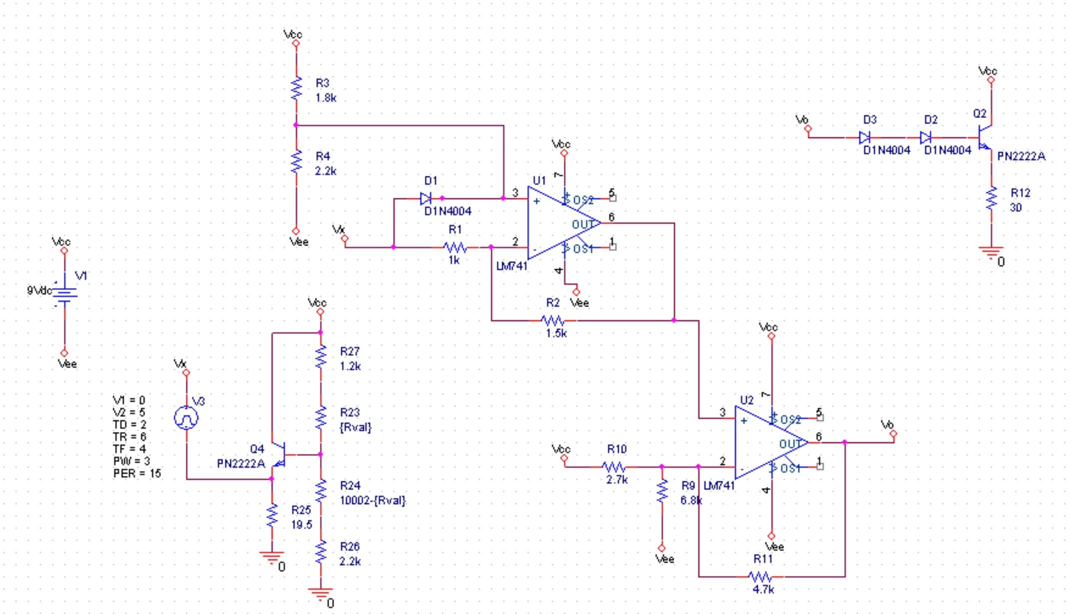

Figure 4.21: Schematics of the variable optimum circuit.

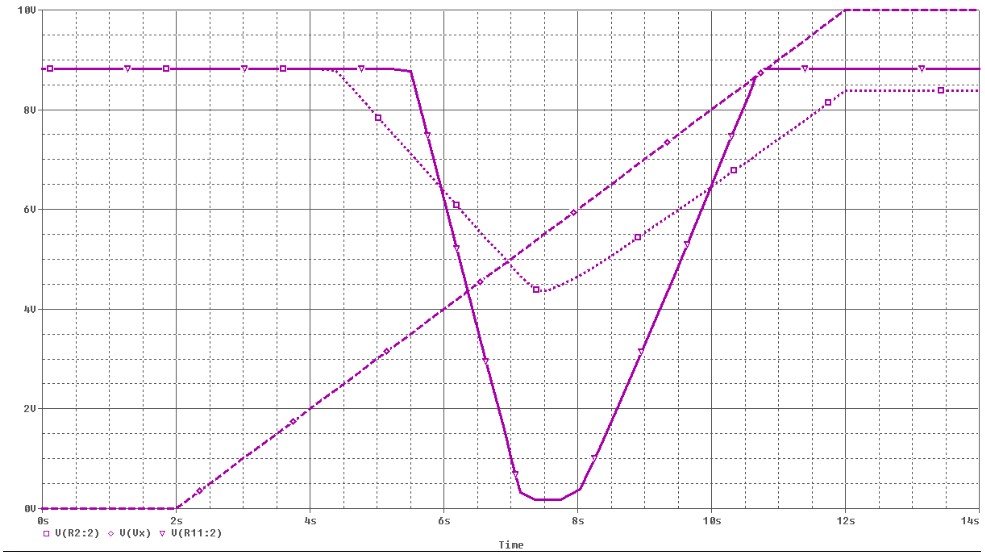

Figure 4.22: Input and output of the variable optimum circuit.

Two of these circuits are used on the two motors. So that we can change the position of the two sliders to emulate the fact that in PD patients who use DBS electrodes, the stimulation parameters that stop tremor may vary over time. For example, if both sliders are set somewhere in the middle, a point near (0.5, 0.5) given to the objective function will minimize the vibrations, and even if the sliders are set near different ends of the range, a point like (0.1, 0.9) vibrations will still be minimized.

To adjust the parameters of Simulated Annealing, since experimenting with real-time optimization takes a long time, a set of sample points were acquired first. The sliders were set in the middle and the vibrations were sampled when the inputs where varied over 10x10 points in the (0, 1)x(0, 1) range. The interval (0, 1) for each variable, results from the 0 V to 5 V range of the actual output voltage after normalization. The resulting level of vibrations at each point is plotted in Figure 4.23. From this data, a function was constructed that is expected to closely follow the actual vibrations called *Test2*. The way Test2 retunes a value given two inputs is to interpolate between the real data points using the "spline" interpolant, add a random number and return the resulting value. Figure 4.24 shows Test2 evaluated at 100x100 points and it is apparent that it is almost the same as the real data only interpolated with some added noise to mimic the noise and uncertainty in the actual system.

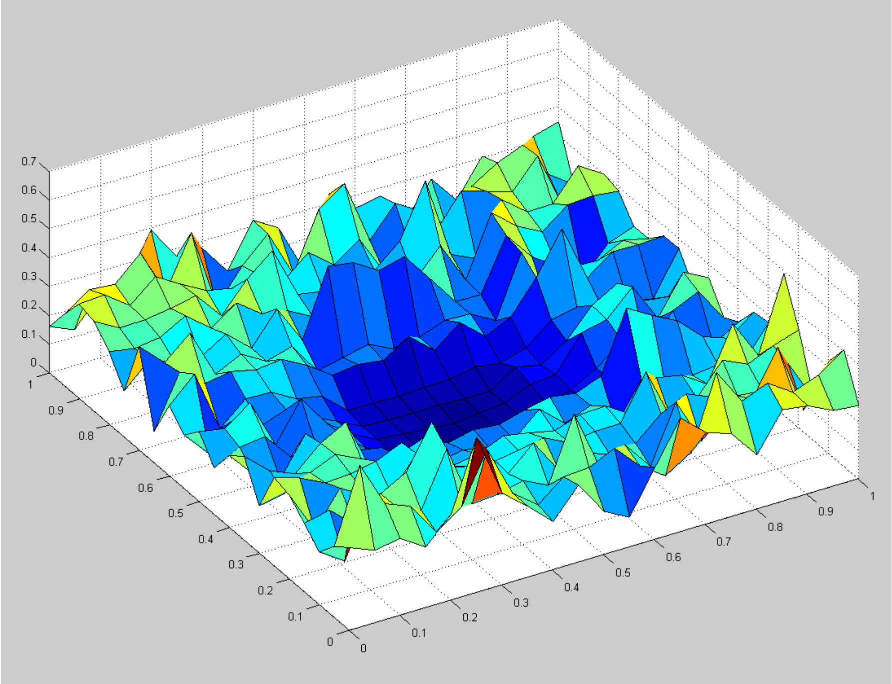

Figure 4.23: A 20x20 set of actual data collected from the shaker.

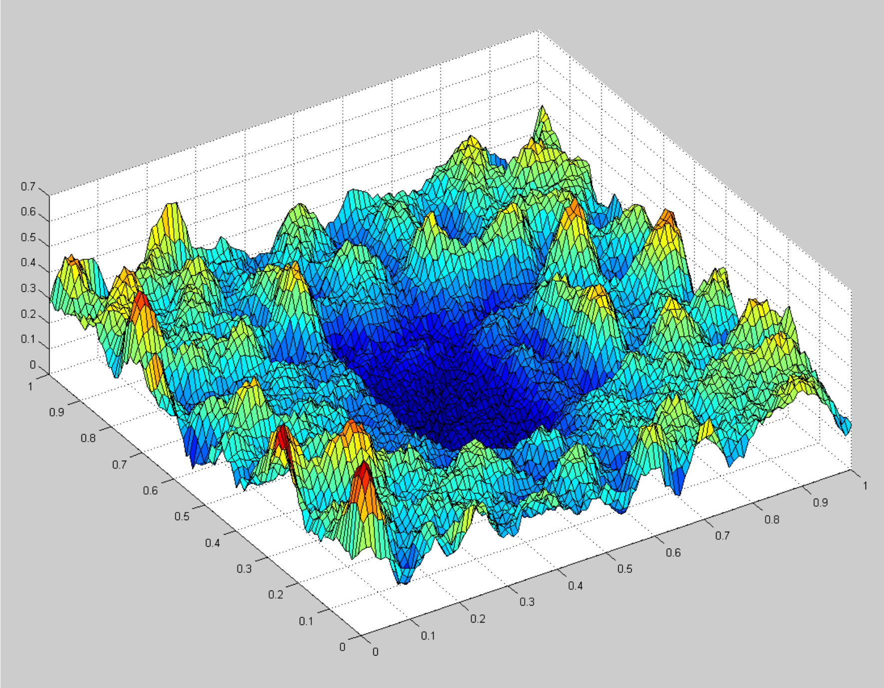

Figure 4.24: Test2 evaluated at 100x100 points

A very strict threshold of 0.03 g was chosen as the goal of the optimization. Figure 4.25 shows the acceleration that each motor produces over the range of its corresponding variable and the threshold is compared to those functions. The function was passed to the optimizer and after several rounds of trial and error it was observed that setting the "Initial Temperature" and "Re-Annealing Interval" parameters in Simulated Annealing to around 500 will ensure acceptable convergence.

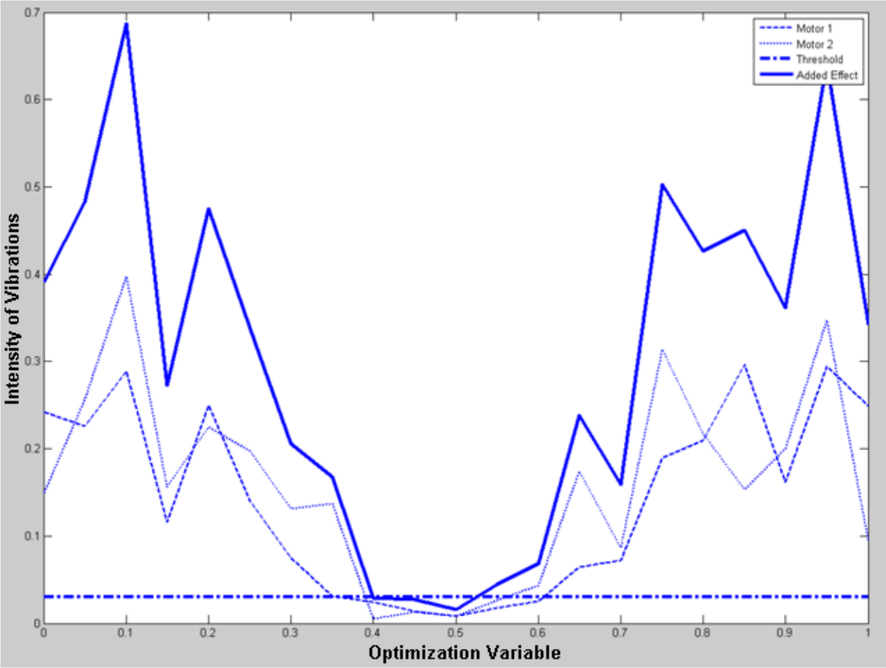

Figure 4.25: Intensity of vibrations introduced by each motor and the threshold.

The same settings were then used on the actual system and the result was that no matter what voltages cause the minimum vibrations,, the optimization loop finds them accurately within 200 iterations. Figure 4.26 shows the optimization toolbox working along with the sensor reading application that has resulted in finding the correct voltages successfully.

Table 5: Results of the real-time optimization.

Attempt	Final Vibration Level	Final Values	Number of Iterations
1	0.029866	(0.51, 0.68)	167
2	0.027366	(0.06, 0.92)	182

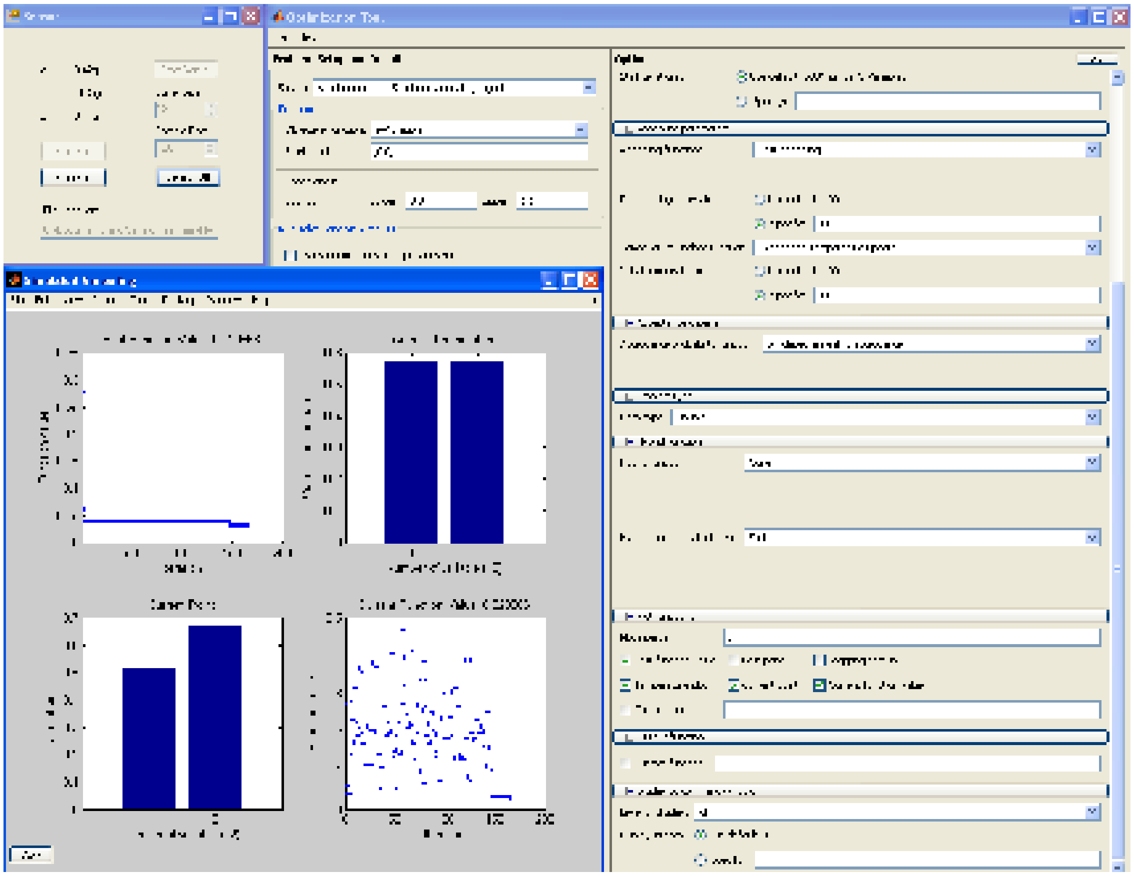

Figure 4.26: A round of real-time vibration optimization.

Figure 4.27 and Figure 4.28 show the convergence of the optimizer for two different configurations of the sliders. It is worth mentioning that after moving the sliders, even the user does not know what exact voltages make the motors stop but the optimizer finds them automatically. In all cases, the optimizer keeps trying different voltages on the system while the hand is shaking with different intensities until the voltages that stop the vibrations are found and applied. Table 5 shows the statistics of the optimizations.

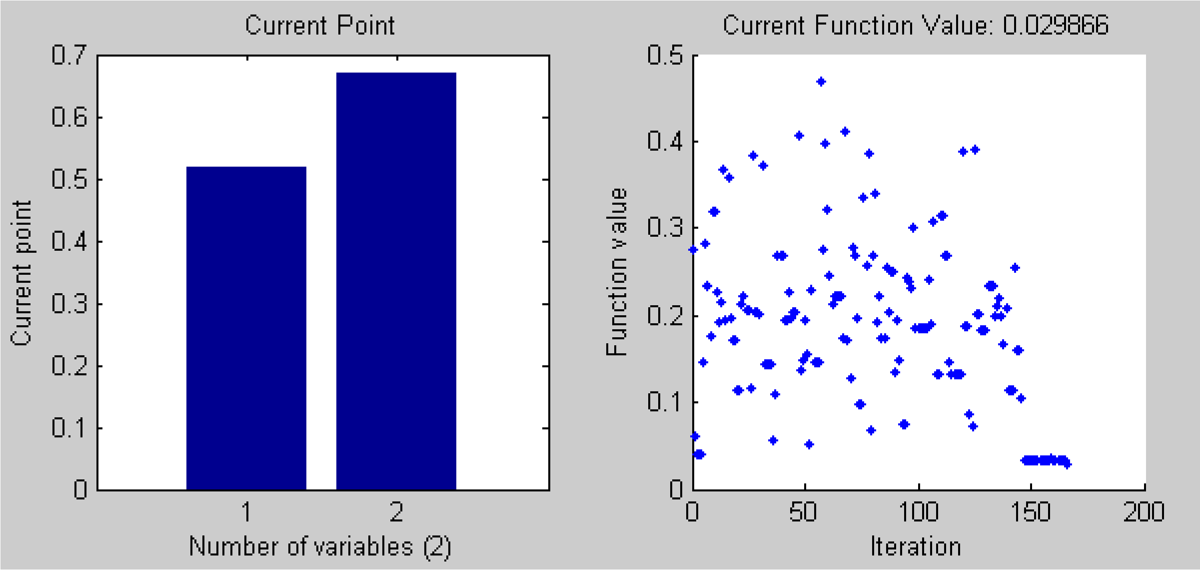

Figure 4.27: Converging of the optimizer to the optimum point in the first case.

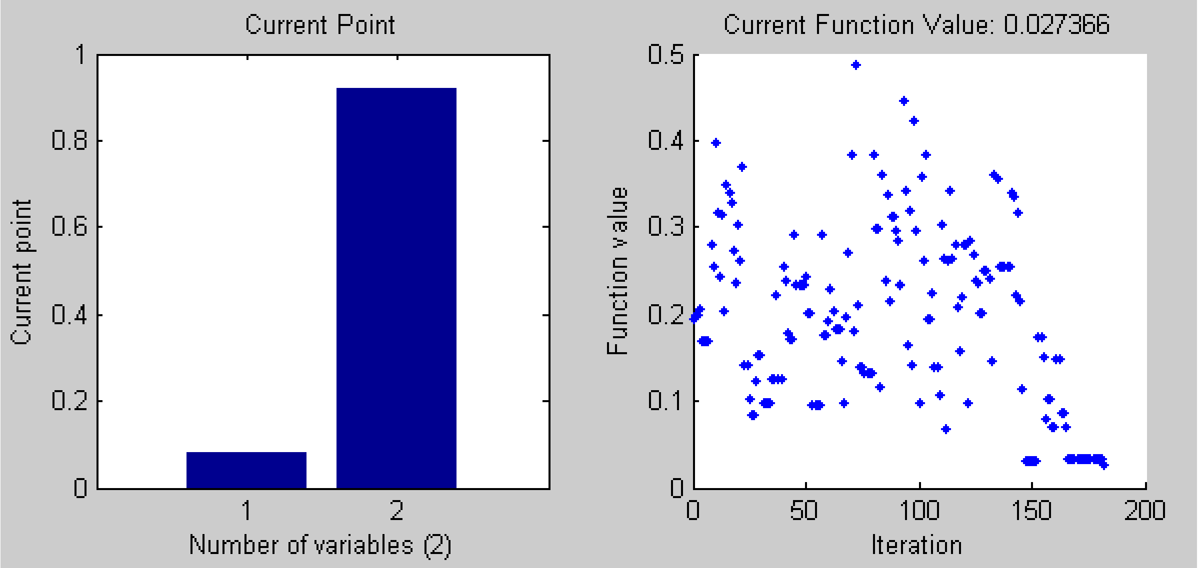

Figure 4.28: Converging of the optimizer to the optimum point in the second case.

Chapter 5

Conclusion and Future Work

A wireless sensor that communicates using the active RFID protocol can be very cost effective, have a long range and low battery consumption. Those are desirable characteristics that open new possibilities in many fields. Aside from what was done for this project, there are huge possibilities regarding active RFID sensors. In RFID tags it is critical for the RF communication to have low power consumption as well as a long range. A commercially manufactured chip antenna along with the necessary matching circuit made up of discreet components has a high loss and also has the problem of tolerances in the components. It is possible to use a distributed matching circuit (PCB transmission line) and to design an application-specific to a PCB antenna to optimize the efficiency. At 2.4 GHz the wavelength is around 10 cm therefore it is possible to design a PCB antenna or a matching circuit occupying a few centimeters in the tags.

The active RFID standard protocol has been designed to cover a range of applications. Therefore it is not optimized for every single project. More specifically, there are trade-offs in communication with the existing technology between different parameters and the protocol could choose the optimum point in between. Please note though that the optimum point is different for different applications. This means sometimes it is necessary to tweak the standard protocol a little or even consider re-optimizing it for the application at hand. For example, given a constant bit-rate, there is a trade-off between the number of tags that is supportable, and the average delay of every packet of data from the tag to the user interface. Another important example is how frequently each tag, which sends its packet of data periodically, can update the reader. This will be a trade-off with the average delay and obviously also affected by the number of tags present. Power consumption and processing power are also part of the trade-offs with other parameters. Applications for this project mainly focus on very few tags which require very short delay and the bit-rate required will not be high. It will useful in the future to rethink the protocol to find an even better fit for this application. Specifically the MAC layer has room for improvement. Depending on how much time is worth spending on this issue, minor tweaks or complete changes could be made in the protocol. Care should be taken however to avoid falling outside public RF regulations and to keep inside the RFID regulation boundaries.

Some applications like medical implants require the active RFID tags to be very small. Fortunately it is becoming possible to integrate more and more parts of the circuit into IC's. One way to make the tags smaller is to use multilayer circuit boards and smaller and fewer components. That is possible and rather easy because many suppliers are offering capable and tiny components off the shelf. Another approach is to design an IC which includes all the circuitry or combines a number of the required functions. For example, the microcontroller, the RF transceiver, the matching network and any required analog-to-digital converters can all be implemented in CMOS and theoretically become a single IC.

If some quantity in the real physical world depends on a number of variables, and that variables can be changed, and a certain combination of those variables bring the quantity of interest to an optimum level, then there is the possibility of achieving that level without even knowing the physical mechanism. The key is to have an appropriate measurement of the quantity using a sensor and using that as a multi-variable objective function in a suitable optimization algorithm. The optimizer can then automatically try different combinations in real-time until it reaches the optimum point. The objective function here was the intensity of the vibrations and its variables were the two voltages applied to the vibrating system. DBS which is used for treating the Parkinson's disease is a very similar problem except that there are more variables like pulse width and frequency, but the same theory applies there.

The hardware system investigated in this thesis has demonstrated the feasibility of reducing vibrations through the use of wireless vibration sensors attached to the vibration source. The concept can be potentially applied in building a closed-loop system to reduce or eliminate vibrations.

Another major area for further research is the optimization algorithm. Simulated Annealing proved to be a very robust and an easy-to-apply algorithm. The algorithm was able to iterate the right way and find the correct combination of voltages that will reduce or eliminate vibrations. None of the other attempted optimizers could minimize the function generated from the acquired vibration signal. Though there are more optimization techniques can be examined more closely and that can result in faster and more reliable convergence. However, it will be necessary to test the system with data from an actual patient. Also, if the indication is more consistent and less random, the optimum point could be more easily found. Different signal processing techniques can be applied to achieve a better indication for passing to the optimizer.

Appendix A

Main MatLab Functions Written for Optimization

```matlab
function f = Vibration(x)

%Parameters
BufferSize = 12; %Samples
SampleRate = 120; %Hz
Settle = 0.5; %Seconds
Duration = 0.5; %Seconds
FilterPass = [1 10]; %Hz

t = BufferSize/SampleRate;
n = floor((Duration/t));
TimeSpan = 0:1/SampleRate:((BufferSize*n) - 1)/SampleRate;

x1 = 5*x(1);
x2 = 5*x(2);

a = zeros(1, n*BufferSize);

%Analog Output Lower and Settle
DAQAO(x1, x2);
pause(Settle);
for k = 1:n
    pause(t);
    %Read Sensor
    r = ReadBuffer('file.txt');
    a(1 + BufferSize*(k - 1):BufferSize*k) = r;
end

f = ProcessData(a, TimeSpan, FilterPass, [x1 x2]);

end
```

```
function f = ProcessData(x, TimeSpan, FilterPass)

%Filter then RMS
T = timeseries(detrend(x), TimeSpan);
F = get(idealfilter(T, FilterPass, 'pass'), 'Data');
f = norm(F(1, :))/sqrt(size(TimeSpan, 2));

end

function f = NoisyFunc(x)

f1 = 0.0019196*x(1)^4 + 0.22837*x(1)^3 - 3.0209*x(1)^2 + 7.2808*x(1) +
28.811;
f2 = 0.0019196*x(2)^4 + 0.22837*x(2)^3 - 3.0209*x(2)^2 + 7.2808*x(2) +
28.811;

r1 = rand*10 - 5;
r2 = rand*10 - 5;

f = (f1 + r1)*(f2 + r2);

end
```

Appendix B

Key Functions for Reading Wireless Sensor Data

```
void theZStar_OnBurstDataReceived(object sender, byte sensor)
{
   // Are these data from our sensor 0
   if (sensor == 0)
      while (theSensor.GetBurstData()) // Get all pending data in Burst Data FIFO Buffer
      {
         // Update labels with values of actual acceleration for all axes
         ShowData();
      }
}

private void btnStartFile_Click(object sender, EventArgs e)
{
   btnStartFile.Enabled = false;
   btnStopFile.Enabled = true;
   tmrSample.Enabled = true;
}

private void btnStopFile_Click(object sender, EventArgs e)
{
   btnStopFile.Enabled = false;
   btnStartFile.Enabled = true;
   tmrSample.Enabled = false;
}
```

```
        void OpenPort()
        {
            ComPortInfo[] ports = ZStar3.GetComPorts();
            // Check ZStarLib ComPort status and selected item in List box
            if (!theZStar.IsPortOpen)
            {
                // If ComPort exist and ZStarLib has closed port
                // Try to open selected port
                if (!theZStar.OpenPort(ports[1].PortNum))
                {
                    // If Failed, show Error message
                    MessageBox.Show("OpenPort failed!", "ZStarLib connection", MessageBoxButtons.OK,
MessageBoxIcon.Asterisk);
                    // And go out
                    return;
                }
                // Check USB_Stick type (has to be known!!)
                if (theZStar.ZStarUsbStickType == ZStar3.UsbStickType.Unknown)
                {
                    // Unknown USB Stick type
                    // Close opened port
                    theZStar.ClosePort();
                    // And show Error Message
                    MessageBox.Show("This is not ZStar3 Device!!!");
                    return;
                }
                // Disable for all sensor burst data
                theZStar.BurstDataReceiveEnableMask = 0x0000;
                // Keep new sensors(without power switch) sensor awake up when ZStarLib runs
                theZStar.SleepDisabled = true;
            }
```

```
}
private void btnStartSense_Click(object sender, EventArgs e)
{
    btnStartSense.Enabled = false;
    txtFile.Enabled = false;
    nudBuffer.Enabled = false;
    nudRate.Enabled = false;
    tmrSample.Interval = (int)Math.Floor((double)1000 / (double)nudRate.Value);
    length = Convert.ToInt32(nudBuffer.Value);
    OpenPort();
    prgStart.Visible = true;
    for (int i = 0; i < 100; i++)
    {
        prgStart.Value = prgStart.Value + 1;
        Thread.Sleep(70);
    }
    prgStart.Visible = false;
    theSensor.BurstDataReceiveEnable = true;
    theZStar.BurstModeEnabled = true;
    if (nudRate.Value > 30)
    { if (nudRate.Value > 60)
        { theSensor.DataRateSet(ZStar3.DataRate.DataRate_120Hz); }
        else
        { theSensor.DataRateSet(ZStar3.DataRate.DataRate_60Hz); }
    }
    sw = File.CreateText(txtFile.Text);
    for (int i = 0; i < length; i++)
    { sw.WriteLine(0); }
    sw.Flush();
    sw.Close();
    btnTurnOff.Visible = true;
}
```

```
private void tmrSample_Tick(object sender, EventArgs e)
{
   try
   {
      double[] temp = new double[length];
      while (FileIsLocked(txtFile.Text)){}
      StreamReader sr = new StreamReader(txtFile.Text);
      int each = 1;
      for (int i = 0; i < each; i++){sr.ReadLine();}
      for (int i = 0; i < length - each; i++)
      {
         string s = sr.ReadLine();
         temp[i] = Convert.ToDouble(s);
      }
      sr.Close();
      for (int i = 0; i < each; i++)
      {
         temp[length - i - 1] = theSensor.AbsoluteG;
      }
      while (FileIsLocked(txtFile.Text)) {}
      sw = File.CreateText(txtFile.Text);
      for (int i = 0; i < length; i++){sw.WriteLine(temp[i]); }
      sw.Flush();
      sw.Close();
   }
   catch {}
}
```

```
void ClosePort()
  {
      // Switch of sleepDisabled Sensor capabilities
      theZStar.SleepDisabled = false;
      // Switch of Burst mode of ZStar
      theZStar.BurstDataReceiveEnableMask = 0x0000;
      theZStar.BurstModeEnabled = false;

      // Close Port
      theZStar.ClosePort();
  }

  private void btnTurnOff_Click(object sender, EventArgs e)
  {
      btnStartSense.Enabled = false;
      btnStartFile.Enabled = false;
      btnStopFile.Enabled = false;
      theSensor.SwitchOff();
      btnTurnOff.Enabled = false;
  }
```

References

[1] **H. Norton,** "*Transducer Fundamentals",* Prentice Hall, 1989.

[2] **Th. Arampatzis, J. Lygeros, and S. Manesis, "***A Survey of Applications of Wireless Sensors and Wireless Sensor Networks",* Proceedings of the 13th Mediterranean Conference on Control and Automation, 2005.

[3] **W. J. Fleming, "***Overview of Automotive Sensors",* IEEE Sensors Journal, Vol. 1, 2001.

[4] **A. Sadat, H. Qu, C. Yu, J. S. Yuan and H. Xie,** *"Low-Power CMOS Wireless MEMS Motion Sensor for Physiological Activity Monitoring",* IEEE Transactions on Circuits and Systems, Vol. 52, 2005.

[5] **C. R. Fuller, A. H. Flotow,** *"Active Control of Sound and Vibration",* IEEE, 1995.

[6] **Y. B. Kim, W. G. Hwang, C. D. Kee and H. B. Yi,** *"Active vibration control of a suspension system using an electromagnetic damper",* Proc Instn Mech Engrs Part D, Vol. 215, 2001.

[7] **C. C. Boucher, S. J. Elliott, P. A. Nelson, "***Effect of errors in the plant model on the performance of algorithms for adaptive feedforward control",* IEEE Proceedings-F Vol. 138 No. 4, 1991.

[8] **S. D. Snyder, N. Tanaka,** *"On feedforward active control of sound and vibration using vibration error signals",* Acoustic society of America, 1993.

[9] **C. Bohn, A. Cortabarria, V. H.artel and K. Kowalczyk,** *"Active control of engine-induced vibrations in automotive vehicles using disturbance observer gain scheduling",* Control Engineering Practice, Elsevier, Vol. 12, 2004.

[10] **L. Schwiebert, S. K. S. Gupta, J. Weinmann,** *"Research challenges in Wireless Networks of Biomedical Sensors",* ACM SIGMOBILE, 2001.

[11] **B. P. L. Lo, G. Z. Yang,** *"Key Technical Challenges and Current Implementations of Body Sensor Networks",* Department of Computing, Imperial College London, 2005.

[12] **J. A. Obeso, C. W. Olanow and J. G. Nutt,** *"Levodopa motor complications in Parkinson's Disease",* Elsavier TINS, Vol. 23, 2000.

[13] *"Speak Campaigns" website,* www.speakcampaigns.org, March 2010.

[14] **S. Patel, D. Sherrill, R. Hughes, T. Hester, N. Huggins, T. Lie-Nemeth, D. Standaert, P. Bonato,** *"Analysis of the Severity of Dyskinesia in Patients with Parkinson's Disease via Wearable Sensors",* Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, 2006.

[15] **J. I. Hoff, A. A. Plas, E. A. H. Wagemans and J. J. van Hilten,** *"Accelerometric Assessment of Levodopa-Induced Dyskinesias in Parkinson's Disease",* Movement Disorder Society, Vol. 16, 2001.

[16] **S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. H. Growdon, M. Welsh, P. Bonato,** *"Analysis of Feature Space for Monitoring Persons with Parkinson's Disease With Application to a Wireless Wearable Sensor System",* 29th IEEE EMBS, 2007.

[17] **J. S. Perlmutter, J. W. Mink,** *"Deep Brain Stimulation"* Annual Review of Neuroscience, 2006.

[18] **C. W. Olanow, R. L. Watts, W. C. Koller,** *"An Algorithm (Decision Tree) for the Management o Parkinson's Disease",* Neurology, Vol. 56 Suppl 5, 2001.

[19] **P. R. Burkhard, H. Shale, J. W. Langston and J. W. Tetrud,** *"Quantification of Dyskinesia in Parkinson's Disease: Validation of a Novel Instrumentation Method",* Movement Disorders, Vol. 14 No. 5, 1999.

[20] **K. Lorincz, B. Kuris, S. M. Ayer, S. Patel, P. Bonato and M. Welsh,** *"Wearable Wireless Sensor Network to Assess Clinical Status in Patients with Neurological Disorders",* IPSN, 2007.

[21] **A. S. Tanenbaum,** *"Computer networks",* Prentice Hall PTR, 2002.

[22] *"Radio frequency identification for item management",* ISO/IEC 18000 Standard, 2004.

[23] *"ZigBee Specifications",* ZigBee Alliance Document 053474r17, 2008.

[24] *"Real Time Locating Systems",* ANSI/INCITS 371 Standard, 2003.

[25] **L. Bierl,** *"MSP430 Mixed-Signal Controller Application Reports",* Texas Instruments SLAA024, 2000.

[26] *"TRC104 Datasheet",* RFM, 2009.

[27] **H. Hinkelmann and others,** *"Design Concepts for a Dynamically Reconfigurable Wireless Sensor Node",* Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems, 2006.

[28] **J. L. Hill,** *"System Architecture for Wireless Sensor Networks",* University of California, Berkeley, 2003.

[29] **D. H. Shih and others,** *"Taxonomy and survey of RFID anti-collision protocols",* Elsevier, 2006.

[30] *"Freescale Accelerometers" website,* www.freescale.com/xyz, November 2009.

[31] **M. Dorigo, G. Di Caro,** *"Ant Algorithms for Discrete Optimization",* MIT Artificial Life 5, 1999.

[32] **W. L. Goffe, G. D. Ferrier, J. Rogers,** *"Global Optimization of Statistical Functions with Simulated Annealing",* University of Texas, 1993.

[33] **J. Nocedal, S. J. Wright,** *"Numerical Optimization",* Second Edition, Springer, 2006.

[34] *CIRFE Internal Documents.*