DESIGN, OPTIMIZATION, AND IMPLEMENTATION OF A VOLUME CONDUCTION ENERGY TRANSFER PLATFORM FOR IMPLANTABLE DEVICES

by

Steven A. Hackworth

B.S. in Electrical Engineering, University of Pittsburgh, 2004

M.S. in Electrical Engineering, University of Pittsburgh, 2005

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Steven A. Hackworth

It was defended on

January 14, 2010

and approved by

Mingui Sun, PhD, Professor, Neurological Surgery, Electrical and Computer Engineering

Robert J. Sclabassi, PhD, M.D., Faculty, Neurological Surgery

Marlin H. Mickle, PhD, Nickolas A. DeCecco Professor, Electrical and Computer Engineering

Richard Friedman, PhD

Zhi-Hong Mao, PhD, Assistant Professor, Electrical and Computer Engineering

Allen C. Cheng, PhD, Assistant Professor, Electrical and Computer Engineering

Dissertation Director: Mingui Sun, PhD, Professor, Neurological Surgery, Electrical and Computer Engineering

DESIGN, OPTIMIZATION, AND IMPLEMENTATION OF A VOLUME CONDUCTION ENERGY TRANSFER PLATFORM FOR IMPLANTABLE DEVICES

Steven A. Hackworth, PhD

University of Pittsburgh, 2010

Two significant problems are present which impede the widespread utilization of many implantable devices with great potential: 1) the lack of availability of an efficient energy source suitable for long-term operation, and 2) the lack of a robust, low-power communication path which does not rely on wired connectivity. The creation of a feasible solution to these two power and communication issues is critical to the success of many future implantable devices. This foundational work details the development of a general solution for the above issues, in a power and communications platform technology for implantable devices. The platform is developed based on the volume conduction technology explored in our laboratory. Ultimate devices are small in size, with the incorporation of a rechargeable battery and electrodes used for interfacing with external components through the skin. An external patch, or "energy pad," containing low-profile electrodes and circuitry, is used as the external interface for recharging and communicating with implanted devices. System design focuses on reliability and ease of integration into a variety of implantable systems, making them feasible for clinical application. Because this is the first system that uses volume conduction for both power and communication purposes, a novel "X-Δ model" of the system is created for use in analyzing the energy transfer of such systems to assist in engineering design. The model, which incorporates components to represent actual current pathways in the skin, is also used in finding theoretical maximum limits of volume conduction energy transfer efficiency for specific skin-electrode setups, proving the technology as a viable option for practical implanted devices.

Keywords: volume conduction, skin model, implantable device, energy transfer, optimization.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

Many persons have been instrumental in making this dissertation possible; not only the research, but the life and vitality behind it. My fianc*é*e Megan has been a constant light, sharing her love and always supporting me through difficult periods. She is so wonderful for respecting the time requirements of a PhD, if not being too fond of them. My parents as well have also encouraged me through their love and an unspoken pride. I am thankful for the freedom they have given me to think independently and forge my own paths. To them and the rest of my family I am most grateful, for without them the journey to this point would not be the same.

Of course I must thank my advisors, Dr. Sun and Dr. Sclabassi, for the opportunity to work in their laboratory. Through their guidance and support through the years, I have grown as a scientist and a professional, and I feel they have pushed me in such a way as to challenge me for the better. I also must thank Dr. Mickle for his support and commentary, which always provides a transparent view of the world and underscores what is most important. I can not thank him enough for helping to get me involved in research projects at the undergraduate level and being instrumental in my admission to graduate school and in my early graduate studies. I am grateful to the other members of my committee as well: to Dr. Friedman, for acting as a beacon of encouragement and bringing a sense of adventure and exploration to my research, and to Drs. Mao and Cheng, for their sharp comments and suggestions, along with their young and fresh view of the research environment.

My labmates, both past and present, have also played a significant role in my course of research and study. Without their discussions, help, and friendship, this journey would have been so much more difficult. Gus, Elli, Tolga, Leo, Qiang, Jun, Ning, Yuecheng, Professor Tang, Wenyan, Chengliu, Seda... I'm glad to have had the chance to work with all of

these extremely intelligent people. Also responsible for my sanity are my friends from high school and college. Zack, Doreen, Luke, Katie, Courtney... the strength of their friendship has been a constant through the years, and that foundation helps tremendously when times seem rough.

I would be remiss to exclude appreciation for all those who have had a less visible presence in the whole PhD process, from administration, to computer technical help, to pig surgery. Thanks to Sandy, Angela, Cam, Jim, Bill, Wendy, Dave, and Judy.

Finally, and not least of all, financial support for this research was provided by U.S. Army Medical Research and Material Command contract No. W81XWH-050C-0047 and Computational Diagnostics, Inc., Pittsburgh, PA.

1.0 INTRODUCTION

Throughout the literature on implantable devices, especially neural interfaces, numerous devices hold great potential. Despite novel designs and exciting progress, many of these have not yet been widely incorporated into clinical applications [105, 79, 103, 104, 50]. Two significant problems which continue to impede the widespread utilization of these designs are 1) the lack of availability of an efficient energy source which is suitable for long-term operation, and 2) the lack of a reliable, low-power communication path which does not depend on wired connectivity. Most researchers develop their own custom power and communication systems which require long development times and expensive fabrication processes. Other devices which hold great promise and scientific value, yet lack acceptable power and communication systems for clinical application, are not uncommon [93, 80, 38, 4, 66]. Although there do exist some commercial devices which rely on non-rechargeable battery power systems, such batteries require implantation of the device in the chest due to their large size. Consequently, subcutaneous cables must be used to connect the main device to distant locations in the body. Many medical experts agree that these designs are undesirable, as they present numerous problems such as hardware failure, repeated non-trivial surgeries, and high cost [40, 6, 67, 97, 28, 36]. The creation of feasible solutions to such power and communication issues is critical to the success of many future implantable devices. They are significant issues themselves, and need to be developed as enabling technologies to serve the implantable device community and move the field forward.

The Laboratory for Computational Neuroscience at the University of Pittsburgh has investigated such an enabling technology, namely volume conduction [91, 92, 89, 83, 87, 88, 102, 101, 76, 90, 47]. Unlike other commonly used technologies that rely on magnetic or propagating electromagnetic fields, volume conduction technology utilizes the conductive ions of the body as energy carriers. This technique affords certain advantages to the technology, such as being able to use simpler circuitry, lower frequency signals, and less power in general. Of course, certain disadvantages also present themselves, such as complications with the biological interface and limited data transfer rates. As with the use of all energy transfer technologies, the advantages and disadvantages of the volume conduction approach have to be balanced within the context of particular applications. As a relatively newly applied technology, however, volume conduction needs to be developed fully before it can be widely considered as a feasible option in the engineering design of implantable devices. The research described in this work attempts to address this issue, focusing on the design of a specific implantable volume conduction system as a proof-of-concept application of the technology, along with the development of a model for use in future engineering design of such systems.

1.1 DEMAND FOR IMPLANTABLE DEVICES

It has been estimated [2] that in the U.S. alone, there are approximately 1.5 million Parkinson's, 4.5 million Alzheimer's, 2.5 million epilepsy, 2 million spinal cord injury / amyotrophic lateral sclerosis / stroke, 10 million severe depressive, and 1 million legally blind patients. These numbers will only increase as the population ages. Many of these patients are left untreated, or are not treated effectively, due to the limitations of existing therapeutic options. Various implantable devices hold the potential for becoming viable treatment options for these and other patients, especially if device cost can be kept relatively low.

Although laboratories around the world have reported research on a variety of implantable devices, most of them remain only prototypes. At present, only three main types of neural implants are routinely utilized in clinical applications [31]. The most widely used is the cochlear implant, with approximately 70,000 implanted cases. The other two are the deep brain stimulator (DBS) for treating Parkinson's disease and dystonia, about 30,000 cases, and the vagus nerve stimulator (VNS) for treating epilepsy and depression, also about 30,000 cases. Numerous devices such as these don't get to see clinical use because of practical difficulties with power and communication aspects of their designs, as well as problems with patient compliance. The technology developed in this work will ideally be used to overcome some of these obstacles and allow many stalled devices to see their full potential in treating patients.

1.2 RESEARCH AIM

The research described herein has two primary objectives. The first objective is to demonstrate the feasibility of volume conduction energy transfer technology for use in implantable devices. The second objective is to develop a volume conduction system model which can be used in the engineering development of volume conduction systems. For the first objective, an implantable device that utilizes volume conduction for all powering and communication requirements is developed. This development involves 1) designing the system architecture and circuit components of the device, 2) creating a protocol to be used for the energy transfer via volume conduction, for both powering and communication, and 3) performing experiments with the device within a volume conduction environment. Not only do the experiments play an essential role in the design and verification processes during device development, but they are also essential in gathering important data for the second primary objective. Model creation is based on careful observance of electrical behavior within the system, both from impedance data collected during physical experiments and from simulation of the particular system using the finite element method. After the model is established, it can be analyzed mathematically to determine theoretical performance limits, e.g. maximum current and voltage transfer through the tissue, and guide system design to meet arbitrary device requirements. This kind of tool is essential for making volume conduction technology generally useful as a viable energy transfer option in future implantable devices.

1.3 THESIS OUTLINE

This thesis is presented in five main sections. The first section discusses the background of implantable device technology, commonly used energy transfer components, and the development of various skin models through the years. These topics are important foundational work from which to build and compare the volume conduction system and model. The second section of the thesis details the implantable device that has been built and the overall system for which it is used. Following is the third section, which details testing and verification of the device along with the gathering of data for construction of the model. Development and usage of the volume conduction system model is covered in the fourth section, and discussion of the entire project is given in the fifth, briefly followed by general conclusions about the work.

2.0 BACKGROUND

This chapter describes existing technology in the implantable device field, issues with electrode connections, various skin models in the literature, and previous work with volume conduction energy transfer. The most common applications for implantable devices are described, followed by methods for transferring energy to and from these devices. As noted in the introduction, many implantable devices exist within laboratory environments, without a feasible long-term energy transfer platform. Here we focus mainly on widely used or prominent devices as a basis for comparison with the volume conduction approach, for which it is intended to become a competetive and viable solution for various devices' energy transfer requirements. The skin models covered here also serve as a point of comparison for the system model developed later. Previously explored models all have their niche applications, but none is quite fitting or optimal for the volume conduction system designed in this work.

2.1 IMPLANTABLE SYSTEM DESIGNS

2.1.1 Nerve Stimulators

This group of devices is composed of Deep Brain Stimulators (or DBS, for treating movement disorders and depression [52, 53]), Vagus Nerve Stimulators (or VNS, for treating epilepsy and depression [43, 25, 42]), and pacemakers, among others. Most implantable stimulators consist of an implantable pulse generator (IPG) with wire leads connecting to the stimulation site. Figure 2.1 shows the placement of a DBS device. The IPG is normally implanted within the chest because of its relatively bulky size and shape. Such size is due to the internal battery which usually consumes about half of the space in the IPG [82].

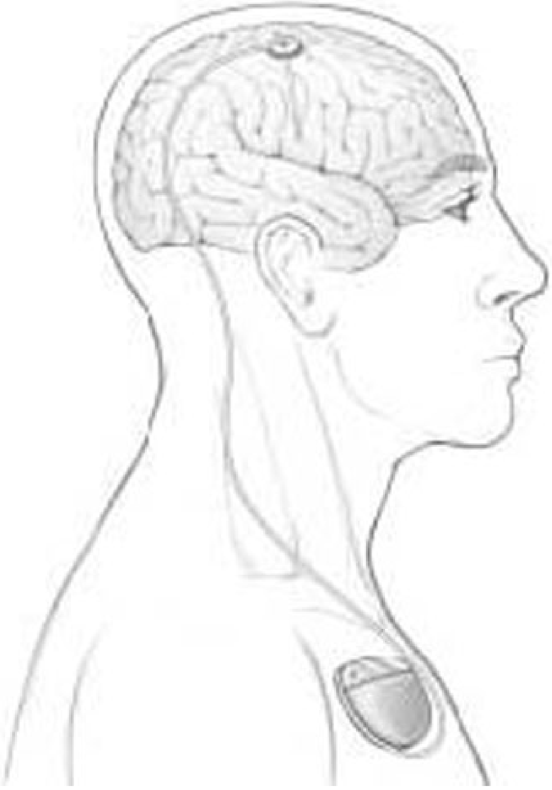


Figure 2.1: Placement of a DBS device. The IPG, containing the battery and control circuitry, is implanted within the chest and wires run through the neck to the top of the head to connect to the stimulating electrodes. Figure courtesy the Dana Foundation (www.dana.org).

2.1.2 Cochlear Implants

The cochlear implant [39, 55] is the most successful commercial neural implant available. Market competition has led to the sophistication of the hearing prosthesis' structural design, shown in figure 2.2. The system consists of two coils, internal and external, cochlear electrodes, and an external microphone and signal processor. The internal coil is surgically implanted in the skull behind the ear, and attached to the electrodes implanted in the cochlea. The external coil is held in place by a permanent magnet on the skin over the internal coil. The microphone is contained within a package worn behind the ear, which is attached to the separate speech processor which can be carried or worn on a belt. The internal and external coils are inductively coupled through the skin - the internal coil receives power and stimulation signals from the external coil, though the signal strength produced by the internal coil is limited. Design of this system has been optimized over the past decade for this particular application, so adopting a similar design for use in other implants would be difficult.

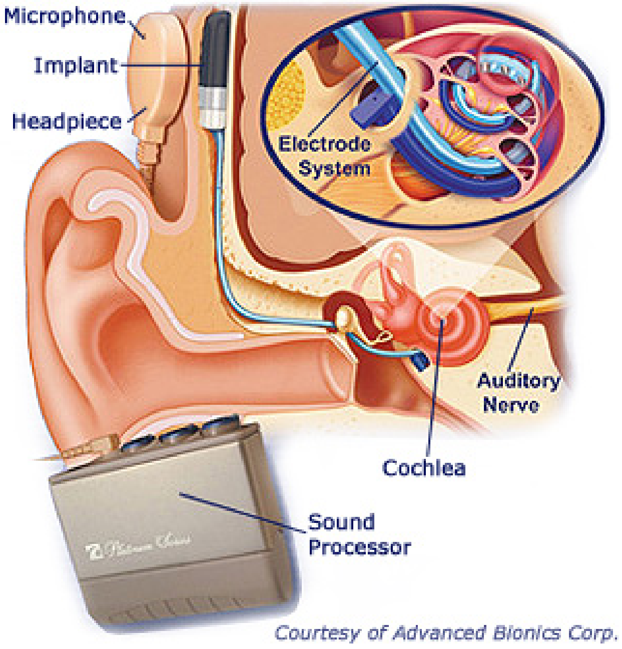


Figure 2.2: Cochlear implant system. The external coil inductively couples with the internal coil to send power and stimulation signals. The microphone is placed around the ear, with a connection to the signal processing unit worn elsewhere on the body.

2.1.3 Drug Delivery Systems

These implantable devices can take on various forms depending on the type of drug and targeted administration area. Intrathecal Baclofen (ITB) Therapy [12, 74], used to treat severe spasticity, involves the placement of a sealed drug reservoir, control circuitry, and battery into the lower abdomen, with an attached catheter inserted into the intrathecal space around the spinal cord. Baclofen is pumped from the drug reservoir through the catheter. The drug reservoir can be refilled via injection through a self-sealing septum, and the catheter can be accessed directly via its own access port. Figure 2.3 shows the ITB device body and catheter. Another form of drug delivery system currently undergoing research uses drug-loaded polymers as a virtual reservoir [45, 98, 48, 71, 94, 99]. Electrical stimulation is used to release the drug from the polymer. Such systems hold promise for diminishing immune response to neural implants or releasing anti-cancer drugs in a controlled fashion, among other applications.

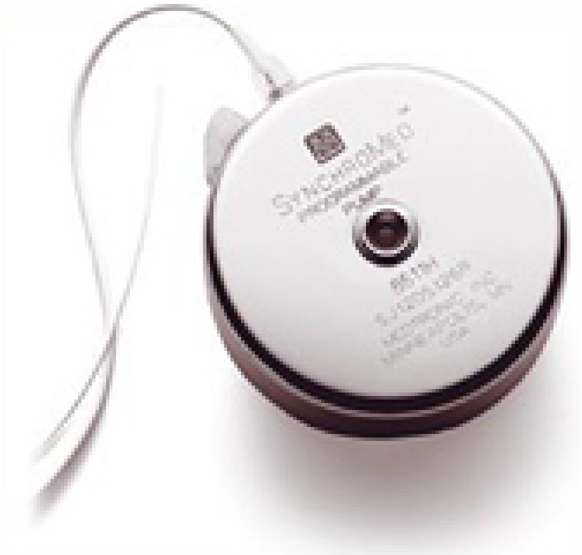


Figure 2.3: ITB Therapy pump. The device is implanted in the lower abdomen and the catheter is inserted into the intrathecal space around the spinal cord. Figure courtesy Medtronic, Inc. (www.medtronic.com).

2.1.4 Percutaneous Neural Prostheses

In percutaneous neural interfaces, most components are placed outside the body. A wire connection is made from a computer or dedicated hardware through the scalp and skull to an internal device [50, 77], which is usually an array of electrodes. Most neural recording studies on animals use wired connections because they facilitate system setup by avoiding the unsolved technical problems of wireless telemetry, and allow the use of standard computers and electronic components to obtain scientific data quickly and with minimal expense. Wire connections have also been utilized on human subjects for the evaluation of experimental neural implants. The left panel of figure 2.4 shows an experimental visual prosthesis in which a camera mounted on a pair of eyeglasses sends video signals to an external unit. This unit processes the signals and converts them to stimulation pulses which are sent to the brain using a thick bundle of wires. The right panel, published by Cyberkinetics Neurotechnology Systems, Inc. [1], shows the design of the BrainGate^{TM} system, which is being evaluated on human subjects. This device uses a percutaneous connector which links the external unit and the electrode array implanted on the motor cortex.

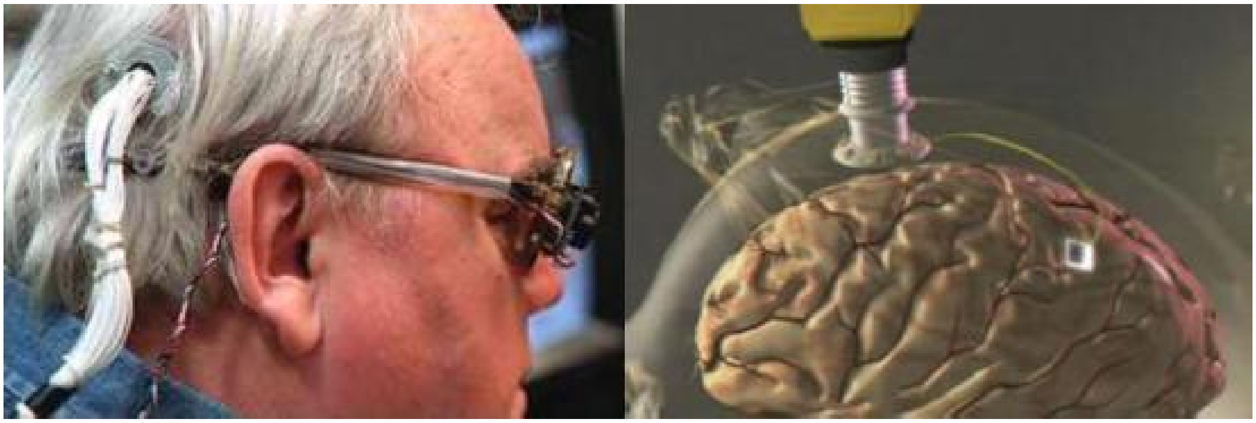


Figure 2.4: Left: Experimental visual implant utilizing percutaneous connections. Right: BrainGate^{TM} of Cyberkinetics Neurotechnology Systems, Inc. for neural control of assistive devices, such as a robot or computer display.

2.2 EXISTING ENERGY TRANSFER TECHNOLOGY FOR IMPLANTABLE DEVICES

2.2.1 Power Supply Technologies

Perhaps the most critical aspect for an active implantable device is its power supply. Any powering scheme dictates use of the implant and guides system design based on available energy. This section details the two main power supply methods used in commercial devices at present.

2.2.1.1 Battery Power A non-rechargeable battery is the most common power source utilized in commercial implantable devices. For example, the Medtronic Soletra$^{\mathrm{TM}}$ neurostimulator uses a 3.7-volt lithium ion battery which is sealed with the control circuitry inside an ovular titanium case (left panel, figure 2.5) [59, 60]. According to the manufacturer's specifications, battery life varies widely with energy use (right panel, figure 2.5), depending upon the stimulation settings. Replacement of these batteries involves replacement of the entire device, necessitating additional surgeries for the patient whenever the battery's energy expires.

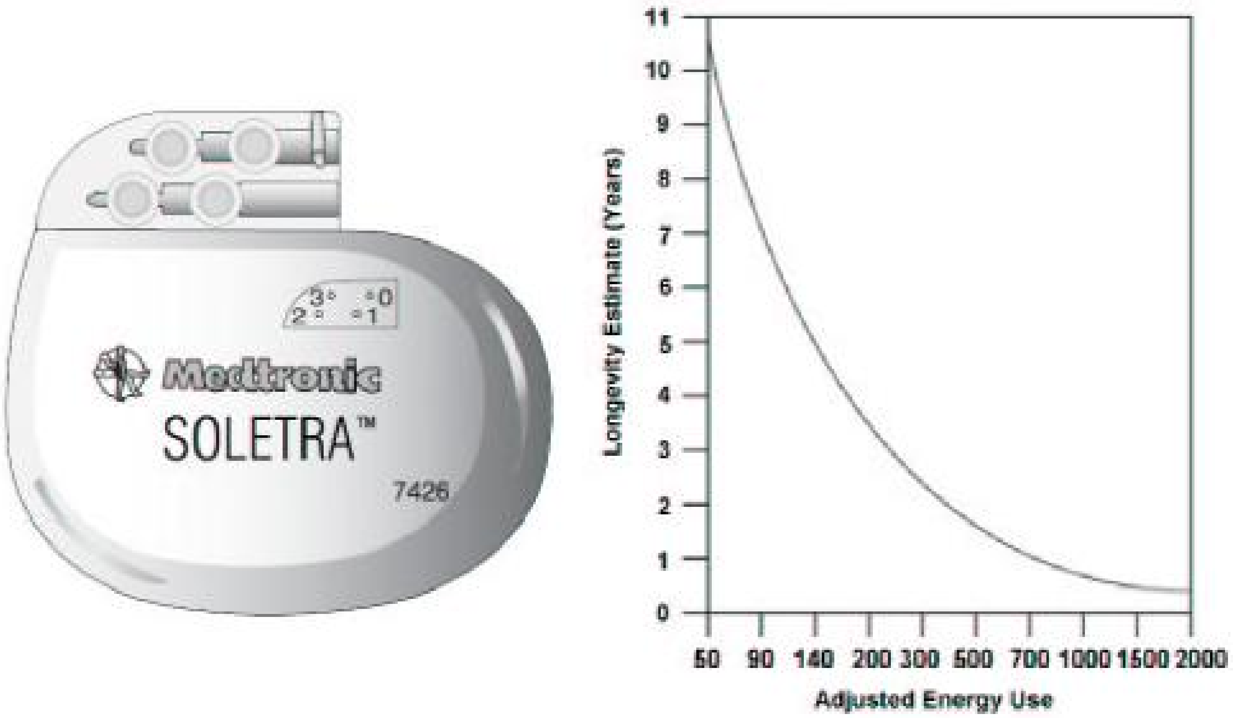


Figure 2.5: Left: IPG containing control circuitry and battery. The battery takes up more than half of the internal space. Right: Expected battery life of the Soletra$^{\mathrm{TM}}$, given average energy usage.

2.2.1.2 Inductive Coupling Magnetic inductive coupling is a prominent powering and communication method in the literature for implantable devices. A transformer-like device consisting of a primary and a secondary coil is used to couple the energy sending side with the receiving side, as shown in figure 2.6. System design relies greatly on the size of the coils used, as they determine the frequency and maximum practical coupling distance. Thorough treatments of the inductive coupling technique are available [50, 29] and it has been applied in several prototype systems [107, 19, 65, 8].

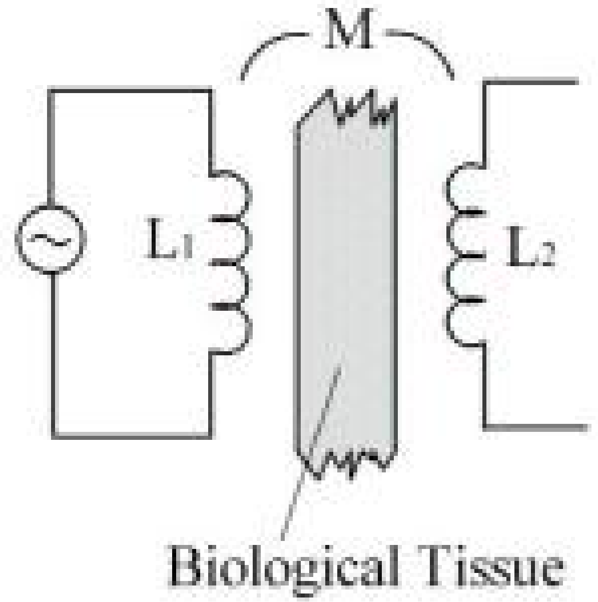


Figure 2.6: Coils L_1 and L_2 are coupled through the tissue via their mutual inductance M.

2.2.1.3 Lack of Alternatives At present, non-rechargeable batteries and inductive coupling are the only commercially used methods for powering implantable devices. These methods are not universally optimal, however. The volume conduction system developed in this work offers another alternative to meet the powering needs of future implantable devices.

2.2.2 Communication Methods

Communicating with active implantable devices is necessary for device diagnostics, changing device parameters, and turning the device on and off. The communication system is often governed by the available power and location of the device. Various methods have been proposed and used with varying degrees of success.

2.2.2.1 Radio Frequency Telemetry Radio frequency (RF) telemetry has been commonly utilized in implantable devices [103, 50, 95, 54]. With this method, an RF signal is transmitted using an antenna, which can either be a dipole antenna similar to the antenna used in a cell phone, or a coil of wire operating on the principle of inductive coupling described in section 2.2.1.2. The relatively high frequency usually employed in RF approaches allows for a high data transfer rate, which is important in certain applications such as multi-channel neural recording.

2.2.2.2 Optical Coupling Optical transcutaneous telemetry systems [32, 64, 21] involve an optical coupler with an infrared laser diode as the transmitter and a photodiode as the receiver. Similar to RF communication, this method suffers from high attenuation, and thus short range, through biological tissue.

2.2.2.3 Magnetic Control Perhaps the simplest of ”communication” methods, this method involves using the swipe of an external magnet to toggle a reed switch within the device [59], often used to turn the device on or off. This method is usually coupled with others which are more suitable for complex data transmission.

2.2.2.4 Volume Conduction Ionic fluids within biological tissues are capable of conducting electrical current which, when intentionally manipulated, can be used to transmit information. This experimental method has been used to send data from the inside of a dolphin to a pair of remotely located electrodes in sea water [54], transmit information from a sensor implanted within the leg of a cadaver to perform mechanical measurements [58, 49], and send information using a body bus described in a Microsoft patent [15]. The proposed volume conduction system relies on this same principle to transmit energy through the skin for both powering and communication. Preliminary studies from the Laboratory for Computational Neuroscience (LCN) showing the efficacy of volume conduction when used as such are detailed in section 2.5.

2.2.3 Need for Improvement

2.2.3.1 Problems with Existing System Designs

- **Percutaneous Systems** – These systems are impractical except for very special cases. Wired connections and risk of infection pose serious risks for any potential commercial use. It is evident that such designs are only temporary and will be replaced when appropriate wireless technologies become available.
- **Non-Rechargeable Battery Power** – Usually used in nerve stimulator systems, this power scheme is cause for a multitude of problems: due to the requirement of a large battery, the IPG must be implanted within the chest. This causes the need for a subcutaneous wired connection to the head or distant body part, which can be problematic. In the case of DBS devices, for example, breakage of the connections, microfractures, and migration of both the electrodes and leads due to forces exerted by movements of the patient, and faulty connectors between the electrodes and the extension wire have been documented in numerous cases [40, 6, 67, 97, 28]. It has been reported that the hardware failure rate can be as high as 20% in initial use [36]. These problems may require re-operation to replace parts of the device, or even removal of the entire system. Even if hardware failure does not become an issue, battery expiration after a period roughly between one to ten years requires surgery to replace the entire IPG.
- **Inductive Coupling and RF Systems** – Systems such as the cochlear implant that use inductive coupling experience similar problems with surgery and device malfunction as any other nerve stimulator [73]. However, these are rarely related to the powering or communication aspects of the system, which have been developed and improved over a span of decades. Inductive coupling does have its drawbacks though, as the restrictions

imposed by the coil geometry make the technology generally difficult to apply to other implantable devices. As the secondary coil's size decreases, its ability to capture magnetic flux declines, which can quickly reduce energy transfer efficiency. In order to maintain sufficient energy transfer, a large current must be applied to the primary coil, requiring large batteries to be carried by the user. Additionally, the relatively high frequencies employed in inductive coupling and RF telemetry put higher energy requirements on the device circuitry, creating difficulties in power supply design. Higher frequencies are also hindered by their low transmission efficiency and short range in biological tissue. One final drawback of RF telemetry is its susceptibility to outside interference. Certain applications require secure communication channels, which can be jammed or manipulated by not only chance interference, but also by malicious users.

2.2.3.2 Target Improvement Areas in System Design

- **Rechargeable Battery** – Recharging ability will allow the size of the battery to be greatly reduced, as it will only need to store enough energy for the period between recharging cycles, significantly less than the lifetime of the device. The reduction in battery size will reduce overall device size, allowing it to be placed closer to the application location. E.g. an IPG for DBS could be placed within the skull instead of the chest, eliminating the problematic connecting wires and leads. Recharging of the device will be kept simple with the application of a low profile energy pad or patch. Any pulsed power requirements unable to be met by the smaller battery can be fulfilled by the inclusion of a supercapacitor to store energy for short bursts when needed.
- **Volume Conduction Power/Communication Link** – Although volume conduction has been used mainly for communication in the past [54, 58, 49, 15], it has been shown that the method can be used to transfer sufficient energy across skin to recharge a battery, comparable to that of inductive coupling [91, 92]. Using volume conduction for both the power and communication channels in the system can overcome some of the drawbacks involved with inductive coupling. The relatively low frequencies used can reduce energy requirements of the device's circuitry. Security is also enhanced, as the only interface to the internal device is through a direct connection on the skin. This reduces

the possibility of outside interference. The use of volume conduction still has drawbacks, however. Like inductive coupling, energy transfer efficiency depends greatly on the size of the interface, namely the contact electrodes. Reduction of electrode size limits energy transfer. Volume conduction is also susceptible to environmental differences such as skin thickness and hydration. These variables affect system performance during application, requiring engineering design to cover a broader range of functionality. Finally, the electrode connections to the external side of the skin require some preparation to make good contact. Although minimal, this does add extra complexity to the system design.

2.3 ELECTRODE CONNECTIONS

The electrode-skin connection is a very important part of any volume conduction system and must be addressed. High impedance connections between the skin and either external or implanted devices can severely degrade signal integrity and dissipate power that could otherwise be delivered to other parts of the system. Another concern is the chemistry of the connection, dependent upon the electrode material and the surface to which it is connected. Especially with biological systems, a connection with poor chemistry will change over time and give unreliable performance. The main objective of any electrode attachment is thus to create a stable, low impedance connection. Various types of electrodes exist, as well as various methods of connecting them to the skin, both to the external and internal sides.

2.3.1 Current Transduction

One issue in the area of electrode connections is the conversion of ionic tissue current to electrical current in the implantable device, and vice versa. This conversion occurs at the electrode-electrolyte interface. Control of the potential and charge transfer at this interface has a profound effect on how well the implant coexists with the contacting tissue.

Current transduction can occur through two difference processes: non-faradaic and faradaic. In the non-faradaic process, charge builds up on either side of the interface, both in the electrode (e.g. electrons in metals) and as the aggregation of charged species in the electrolyte. This forms a capacitive double layer where no charge is actually transferred across the interface. The accumulation of species at the interface results in electrode polarization. In the faradaic process, electrons are transferred to or from the electrode to reactive species at the surface, causing oxidation or reduction reactions in the electrolyte.

There are two main types of electrodes: those for recording and those for stimulation (or passing relatively high levels of current). With recording electrodes, the main concern is their reaction with the environment, as relatively large amounts of current are not being passed across the electrode-electrolyte interface to stimulate electrochemical degradation. However, stimulating electrodes must pass current to and from the surrounding tissue. Ideally, strictly polarizable electrodes could be used in a non-faradaic process to prevent chemical reactions from occurring at the interface. Unfortunately, ideal electrodes do not practically exist, and commonly used metals can only support a maximum charge transfer of 20 μC/cm2, too little for adequate current [72]. Thus, mostly faradaic electrodes are commonly used for stimulation purposes. However, use of faradaic processes presents a problem in that not all processes are reversible. If the reactive species are not somehow kept in place at the electrode interface, diffusion and cellular activity will carry away the reactants, preventing the reverse reaction from taking place [30]. Irreversible processes can result in electrode degradation without any mechanism to restore its integrity. Even with reversible reactions, there exists a limit on the amount of current that can be supplied, called the reversible charge injection limit [30]. This is the charge transfer threshold beyond which irreversible changes are made to the electrode. Any applied current must remain within the threshold to avoid electrode degradation. Additionally, if the stimulating current waveform is not bidirectional, the reaction is not reversed. Thus, a stimulating waveform with zero DC component is necessary [72]. There is also a disparity between anodic and cathodic reactions which comes into play. The dynamics of either reaction can vary slightly when in the bodily environment, such that symmetric waveforms still have a tendency to favor one reaction and ultimately erode the electrode.

In addition to chemical reactions taking place at the electrode-tissue interface, the strict electrical parameters must also be taken into consideration. Significant current transfer relies on effective coupling at the electrode surface, i.e. low impedance between the electrode and the tissue. The impedance of this interface has a large effect on design of the implantable device, specifically when determining necessary voltage and power levels. Generally speaking, a low impedance will allow for easier charge transfer and thus require less power and driving voltage.

2.3.2 Electrode Materials

This section aims to examine various materials used as electrodes in implantable devices, with some insight as to their appropriateness. First and foremost amongst conductive materials are metals. There are two major categories, namely noble metals and non-noble metals. From the former, metals such as platinum, iridium, and gold are used. Noble metals are highly resistant to environmental corrosion, though corrosive effects such as loss of mass, unwanted film formation, and particulate debris are possible when used with higher current levels [30]. Platinum is most widely used, though it is usually alloyed with iridium to increase its strength. Iridium is often used by itself in implants, being stronger than platinum, but when used in such a manner, it is first treated in one of a number of ways to form a layer of iridium oxide on the surface [72, 51]. This oxide layer results in a much lower contact resistance at the electrode interface, and also boasts a larger charge injection limit due to the ease at which the oxide can switch valence levels. Non-noble metals used in implantable devices take advantage of the same type of reversible oxidation process. However, care must be taken not to exceed the charge injection limit because of the resulting irreversible faradaic reaction which dissolves the electrode [30]. Common non-noble metals are stainless steel 316 and 316LVM, and nickel-cobalt alloys. These conductors, though mechanically stronger, tend to have smaller charge injection limits than the noble metals. Additionally, nickel-cobalt alloys tend to also have higher electrical impedance at the electrode-tissue interface [30].

Along with metal conductors, much research is being done with silicon electrodes. Silicon electrodes have the advantage of being compatible with the CMOS fabrication process. As such, they can interface directly with CMOS electronics to simplify the implant system.

Titanium nitride electrodes also take advantage of this process. This material is very stable, capable of high charge injection while maintaining relatively low impedance [100].

2.4 ELECTRICAL MODELS OF THE SKIN

Here we examine in general the types of models used in the past to describe electrical behavior of the skin. These can be broken into two main categories: low voltage (small signal) models, commonly used in experimental monitoring situations, and moderate to high voltage models, used in various applications from iontophoresis or electroporation to surgical monitoring. Currently there is a lack of models relevant to the operating region of volume conduction technology, likely due to the relative dearth of research on volume conduction energy transfer in associated applications.

2.4.1 Small Signal Models

Most models describing small signal (microvolts to millivolts) behavior are based around the Debye or Cole models [23, 10, 41, 16, 69, 24], shown in figure 2.7. The original Debye model describes a finite resistance in series with a parallel conductance (variable) and capacitance. The mathematical equivalent impedance is given in (2.1). This rudimentary model is a close approximation of the impedance behavior measured across skin, though it was improved upon with the inclusion of a constant phase element (CPE) in place of the capacitor in the Cole model. The CPE is equivalent to a parallel connected resistor and capacitor, both frequency dependent in such a manner as to make the phase independent of frequency. This empirical model is used to describe behavior commonly seen in electrochemistry and tissue and cell suspensions. (2.2) is the mathematical expression describing the model's impedance.

$$Z = R_\infty + \frac{1}{G_{var} + j\omega C} \tag{2.1}$$

$$Z = R_\infty + \frac{1}{\Delta G + \Delta G\,(j\omega\tau_Z)^\alpha} \tag{2.2}$$

Although these well-established and physically explained [23, 69] models are effective at describing the skin's electrical behavior at low frequencies (10's of kilohertz and below), their appropriateness at higher frequencies has not been confirmed. Additionally, the experiments for which these models were designed involve low voltage and current density levels (below 2-4 volts and 1 mA/cm^2). At higher voltages and currents, at least at the low frequencies commonly used in these studies, the skin's behavior becomes increasingly non-linear [70, 18, 57].

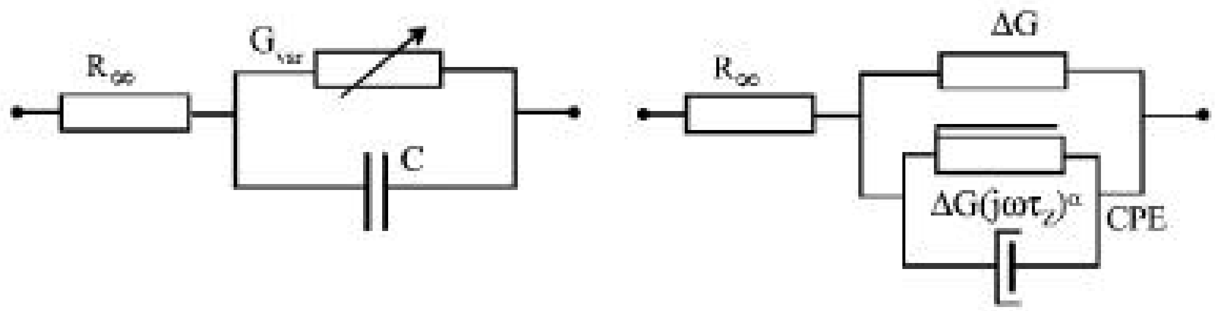


Figure 2.7: Left: Debye model of the impedance across the skin. Right: Cole model of the impedance across the skin. Recreated from [24].

2.4.2 High Voltage Models

For applied voltages in the tens to hundreds of volts range, circuit models are typically not employed because of the non-linear behavior at such high amplitudes. Instead, models focus on structural changes or polarization effects at the electrode-skin interface [34, 78, 56, 61, 9]. Most of these use various complex mathematical treatments involving chemistry, biology, and geometric principles. However, some do include circuit models with non-linear components based on such principles, each component corresponding to a physical structure of the model [14]. Again, unfortunately, none of the models treated in the literature can be applied to the volume conduction system developed in this work, as none are relevant to the combination of operating parameters (frequencies and voltages) used in said system.

2.5 PREVIOUS WORK WITH VOLUME CONDUCTION SYSTEMS

Through the past decade, the LCN has extended efforts to make volume conduction a viable energy transfer technique for implantable systems. Originally investigated as a transmission means for cortical EEG, it has evolved into a capable platform for transferring significant levels of energy to and from implantable devices. This section details the early stages of volume conduction investigation in the laboratory as well as the precursory efforts that led to the work in this thesis.

2.5.1 Multi-channel Biological Signal Transmission Device

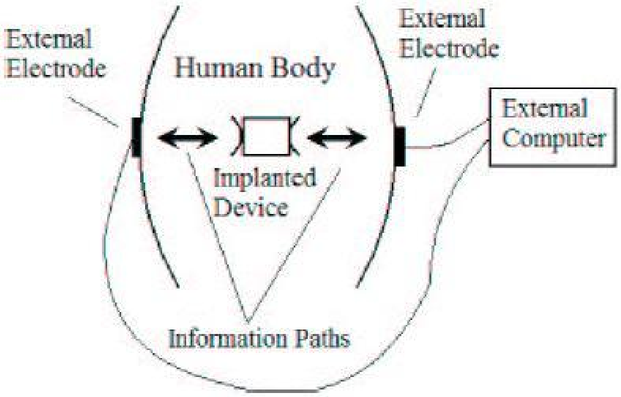


Figure 2.8: Volume conduction communication system diagram.

2.5.1.1 System Design This implantable device system uses volume conduction to send signals from inside the body to the surface of the skin. Figure 2.8 shows the general setup of the system. A device implanted within the body has a pair of antenna elements as shown. These elements are oriented towards a pair of external electrodes on the body surface. Electrically charged ions in the body fluid carry information from the inside to the outside of the human body. No RF signal is involved because the VC system uses a frequency in the kilohertz (KHz) range. This low transmission frequency reduces signal attenuation in the tissue, simplifies circuit design, consumes less energy than high frequency designs, and reduces the size of the implantable device [89, 83, 87, 88, 102, 101, 76, 90, 47, 85, 86, 84]. Such a low frequency design is not suitable for high-bandwidth data. Based on this system, various devices have been designed and constructed for use in transmitting cortical EEG signals from inside the skull to the surface of the scalp. The left panel of figure 2.9 shows a typical block diagram of the circuitry involved in one of these device. Small implanted batteries provide power for the internal Integrated Circuits (ICs), including an amplifier, filter, multiplexer, and timer. The measured signal(s) is(are) first amplified and filtered, then sampled at a rate of 4 KHz via the timer-driven multiplexer. The resulting output signal is applied to the specially curved electrodes forming two sides of the device body. These electrodes direct the signal to the surface of the body where it can easily be measured for the desired use. The right panel of figure 2.9 shows a constructed and encapsulated device, made using surface mount components and the miniature printed circuit board (PCB) construction process used in our laboratory.

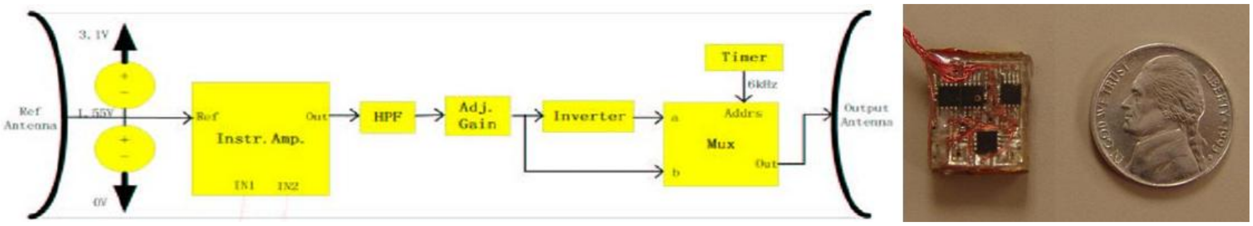


Figure 2.9: Left: Circuit block diagram for the cortical EEG transmission device designed according to the volume conduction communication system layout. Two small 1.5-V batteries power circuitry to amplify, filter, sample, and transmit EEG data. Right: Constructed device with nickel for size comparison. External wires are used for monitoring signals and power during device and system testing.

2.5.1.2 Experimental Validation

A 2-channel version of the EEG transmission device detailed in section 2.5.1.1 was tested and verified in the cranium of a freshly-sacrificed pig, generously donated by another research group. Because the pig was dead, separate 3- and 5-Hz sinusoidal signals simulating EEG were used as inputs to the device. Each input signal had an amplitude of approximately 500 V. After the device was placed within the cranium of the pig, the skull and skin were replaced and the scalp potential was recorded with needle electrodes. The experimental setup is shown in figure 2.10. Figure 2.11 shows the raw recorded potential from the scalp, in which can be seen the multiplexed 3- and 5-Hz signals with overlaid 60 Hz noise. The recovered 3- and 5-Hz signals are shown in figure 2.12. These experiments show that this volume conduction method for transmitting internal signals to the surface of the body with small, low power devices is feasible.

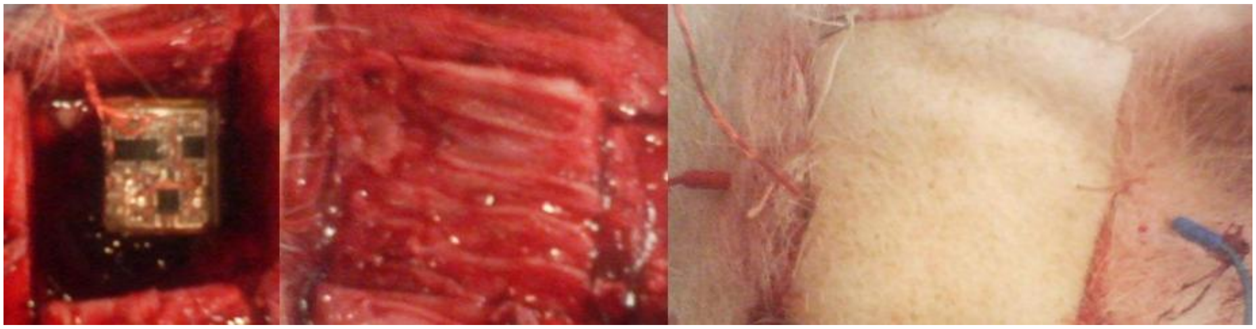


Figure 2.10: Left: Volume conduction device placed within the skull on top of the brain. Middle: Brain and implant re-covered with skull. Right: Scalp re-covered. Red and blue needle electrodes are used to record electric potential on the scalp. Wires to the implantable device provide the 3- and 5-Hz signals.

2.5.2 Preliminary Skin Energy Transfer Studies

Looking towards applying volume conduction as a complete energy solution for implantable devices, a system was designed to facilitate higher levels of energy transfer through the skin beyond those needed for communication signals. It involves the coupling of electrodes on either side of the skin, with the external electrodes connected to external powering circuitry and the internal electrodes attached to the implantable device. A preliminary model (called the "X-Model") of the system was developed [91, 92], as well as an attempt to optimize the electrode geometry.

2.5.2.1 X-Model of the Skin An electrostatic linear system model was used to represent the skin system, with electrode connections on the internal and external sides of the skin, an external recharging voltage, and an internal device battery voltage (as shown in

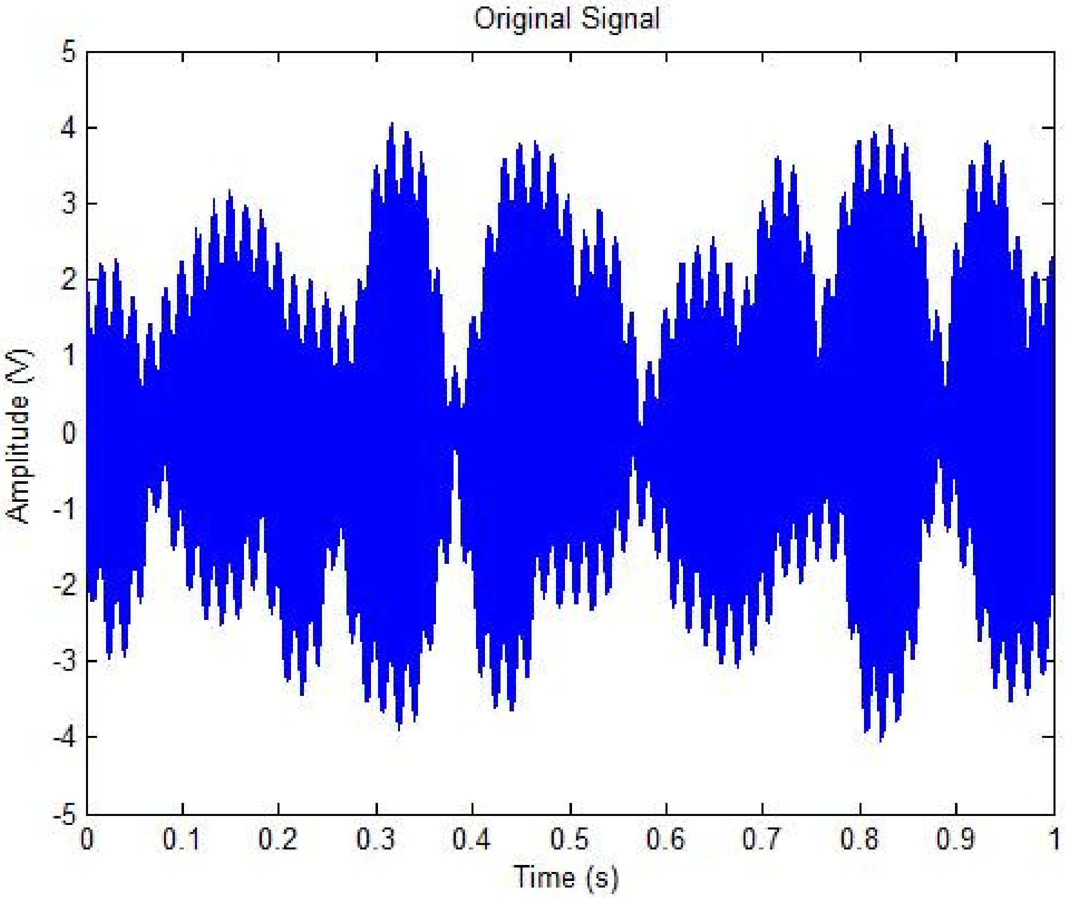


Figure 2.11: Raw electric potential recorded on the scalp from the EEG transmission device.

figure 2.13). Current from the external source enters the skin, with some recharging the internal battery and some shorting across the skin. The recharging efficiency is defined as the ratio of internal battery recharging current to injected current from the external source. Calculating this value from the circuit model gives the equation for efficiency in terms of the model parameters, displayed in (2.3).

$$\eta_I = \frac{I_2}{I_1} = \frac{1 - \left(1 + \frac{R_1+Z_1+Z_2}{Z_5}\right)\frac{V_2}{V_1}}{\left(1 + \frac{R_2+Z_3+Z_4}{Z_5}\right) - \frac{V_2}{V_1}} \tag{2.3}$$

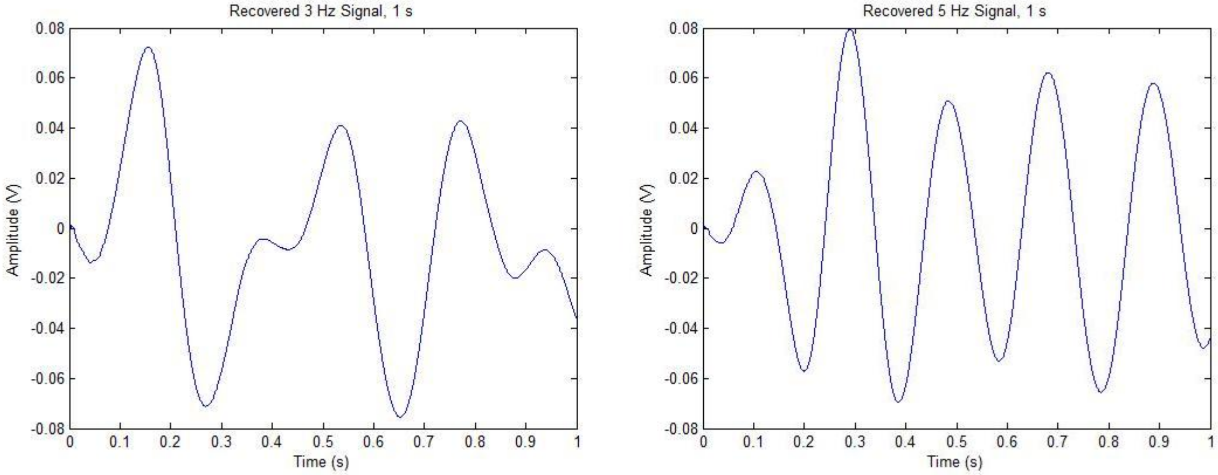


Figure 2.12: Left: Recovered 3-Hz signal from the recorded scalp potential. Distortion of the signal is caused by the low order filter not completely cutting out the 5-Hz signal. Right: Recovered 5-Hz signal from the recorded scalp potential.

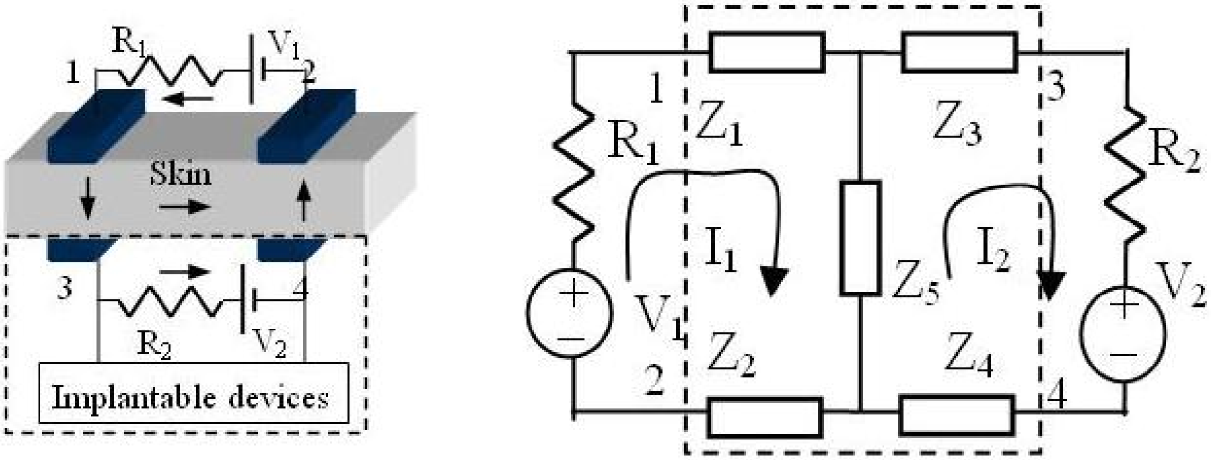


Figure 2.13: Left: Structural model of the volume conduction system with arrows showing current paths. Right: Circuit model (”X-Model”) of the system with lumped impedances representing contributions of the skin and electrode contacts.

$$IIR = \left(1 + \frac{R_1 + Z_1 + Z_2}{Z_5}\right) \quad , \quad OIR = \left(1 + \frac{R_2 + Z_3 + Z_4}{Z_5}\right) \tag{2.4}$$

Three main expressions in (2.3) govern the recharging current efficiency. The two impedance expressions, given in (2.4), are labeled as the input impedance ratio (IIR) and output impedance ratio (OIR). The other governing factor is the external-to-internal voltage ratio, V_1/V_2. Examples of these three expressions' specific effect on the efficiency are plotted in figure 2.14. It was found that the IIR and OIR govern maximum possible efficiency, while practical performance increase (higher efficiency) via increasing the voltage ratio is limited after a certain point.

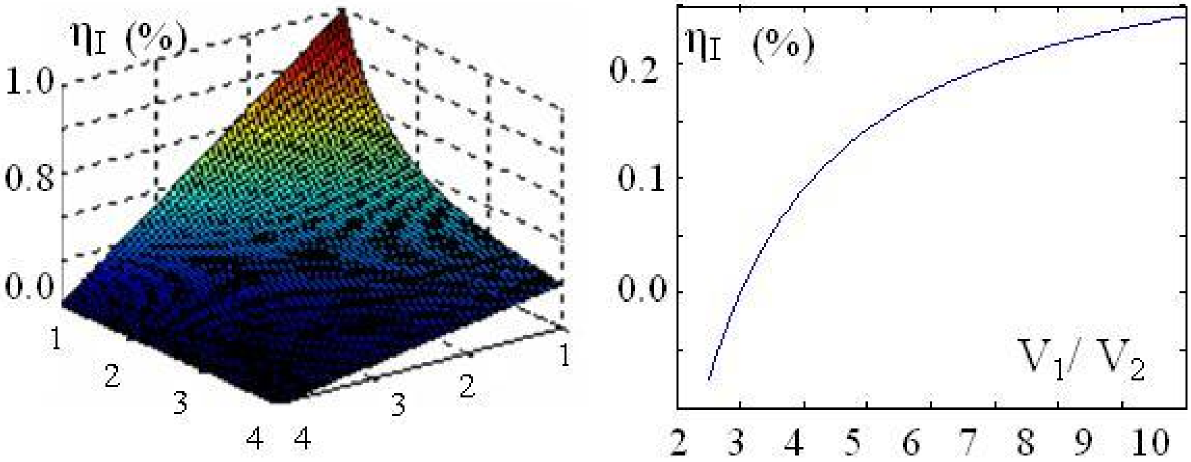


Figure 2.14: Left: Recharging current efficiency as a function of IIR and OIR with a constant voltage ratio, $\frac{V_1}{V_2}$, of 4. Right: Recharging current efficiency as a function of voltage ratio with constant IIR and OIR of 3.

It should be noted that the experiments to find the X-model parameters were performed at a single frequency of 5 KHz. Results using freshly harvested pig skin were unreliable at best, consistently producing inconsistent model parameters. The researchers are still unsure whether the inconsistencies are due to poor experimental procedure or a lacking model. Regardless, this preliminary work built the foundation for further refinement of the system and its model.

2.5.2.2 Electrode Geometry Analysis One of the important considerations for a volume conduction system is the electrode-skin connection. Not only does the type of material and skin preparation used determine the quality of the connection, but electrode geometry also plays a large role in determining current transfer efficiency through the volume conductor. Optimizing the geometry in general is futile, as any particular application will put various constraints on the electrode size. However, finite element simulations have been used to predict which geometries produce the highest efficiency given a limited area. Figure 2.15 shows the simulated geometries, including rectangular, triangular, concentric rings, and an electrode array. It can be seen from the results in figure 2.16 that with the specific simulation dimensions, the separation of the concentric ring electrodes is limited, thus limiting their current transfer efficiency. The electrode array also shows poor performance overall. The rectangular and triangular electrodes perform fairly well, however, showing that separation of electrodes is perhaps most important in preventing shorting current.

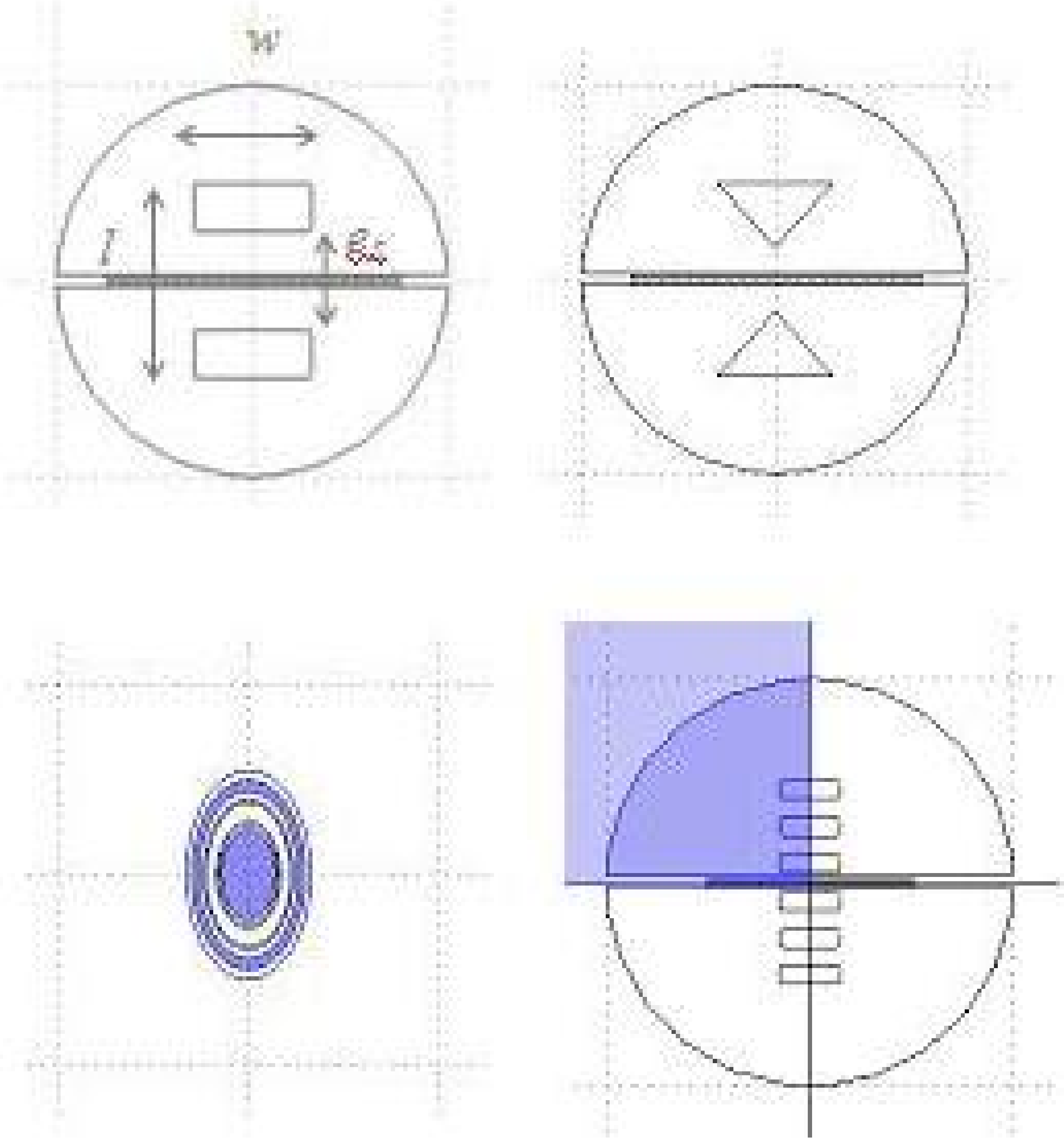


Figure 2.15: Simulated electrode geometries, all adhering to the same size constraints.

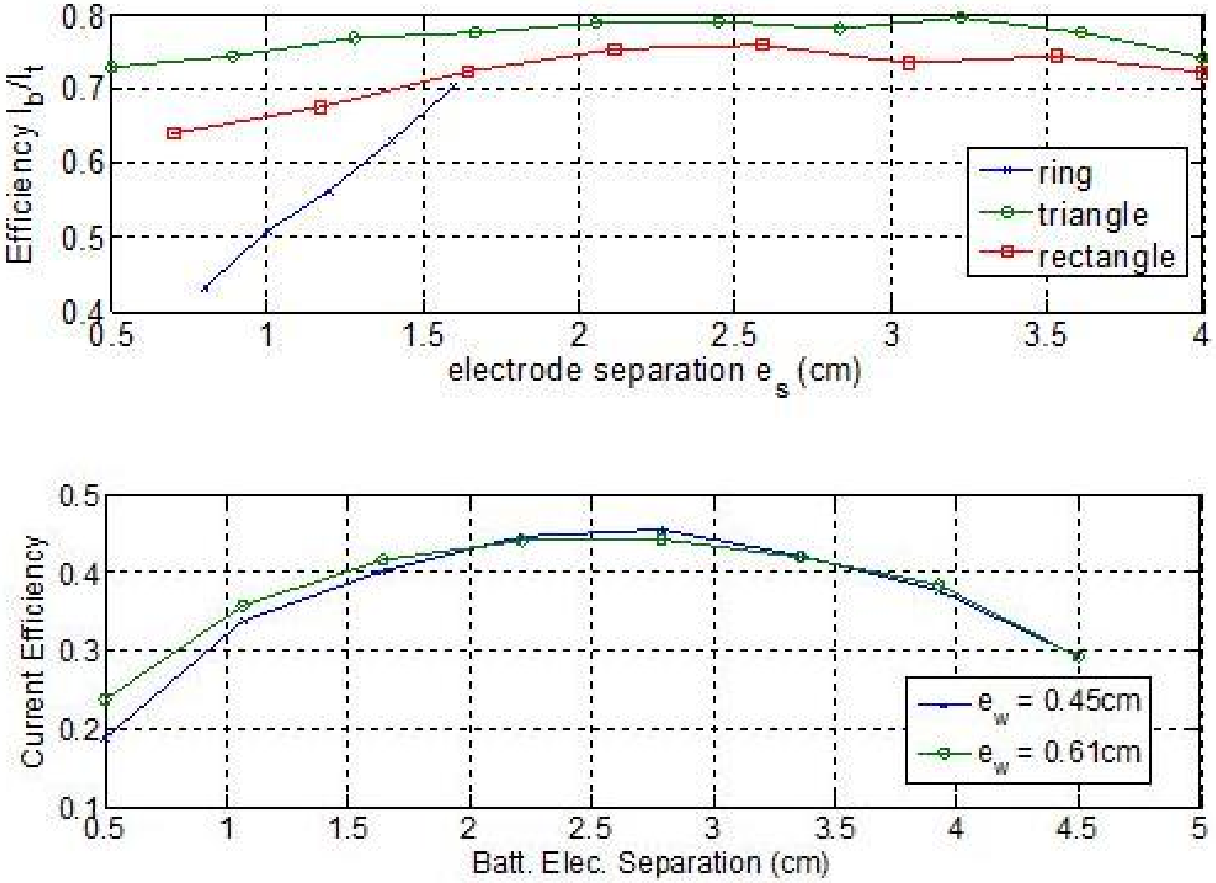


Figure 2.16: Electrode geometry simulation results. Top: Triangular and rectangular electrode configurations exhibit better performance than concentric ring electrodes under the same size restrictions. Bottom: The electrode array performs poorly compared to the other geometries.

3.0 SYSTEM AND DEVICE DESIGN

Towards the goal of testing and proving volume conduction energy transfer as a viable technology for implantable devices, a small implantable device has been designed and built which uses the technology as its energy platform. This chapter details the design of the device, starting with intended functionality requirements and including energy transfer protocols, system architecture development, associated circuitry realization, encapsulation of the device, and the electrode-skin interface. Original intentions for the device consisted of using it as a replacement for the IPG commonly used in DBS applications. Conceptually shown in figure 3.1, it was to be placed into the burr hole during DBS surgery and integrated into the skull, eliminating the more dangerous chest implantation surgery, the bulky IPG, and the associated cabling through the neck to the top of the head. Through the course of development, the aim has become more general, i.e. to make a device with volume conduction capabilities for use in researching the technology. The device detailed in this chapter, or part of it (e.g. the electrodes), was used in experiments to validate and characterize the volume conduction technology, described in chapter 4.

3.1 FUNCTIONAL OVERVIEW

Generally, the device is to be used as follows. It is placed under the skin or inside the body and is attached to electrodes in contact with the underside of the skin. These electrodes can either be part of the device body, as in figure 3.2, or separate, with connecting wires to the device. The latter approach allows for more flexibility in placement of the implant, but introduces additional points of failure. The electrode-skin connection forms a single

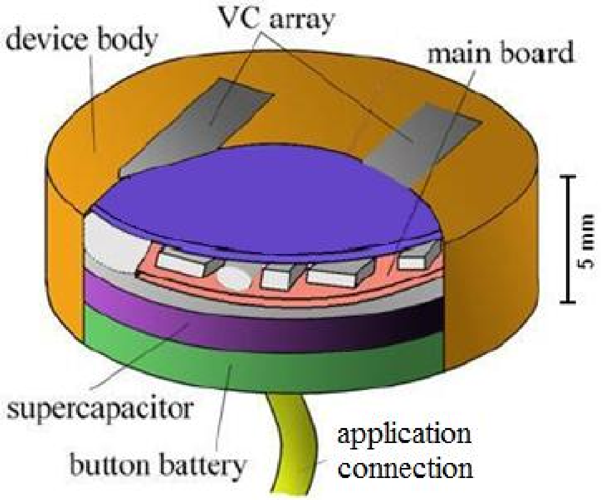


Figure 3.1: Theoretical layout of the implantable device body. Components are kept as thin as possible to keep a thin device profile to allow for freedom in placement of the implant.

port through which energy is transferred for both communication and recharging purposes. This energy transfer port is the foundation of volume conduction's implementation in this research; the ultimate functionality and form of the implantable device depends on the specific application and thus receives little attention in this work.

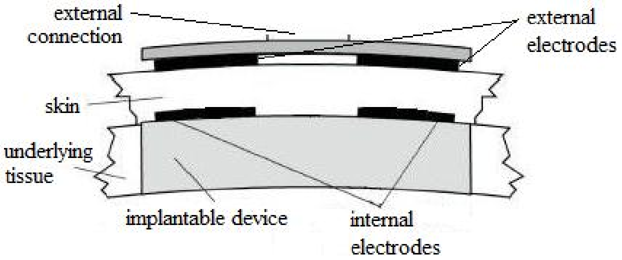


Figure 3.2: One possible implementation of the implantable device. Here the electrodes are part of the device body, which is implanted directly underneath the skin. External devices or circuitry are connected via the external electrode pad, which is connected to the skin when necessary.

To interface with the port from the outside world, an energy pad is attached to the external side of the skin via its own electrodes. Like the implant, the actual form and function of the external device will vary depending on the specific application, and thus is not crucial here. For purposes of testing device functionality, a hand-held controller has been made which plugs into an electrode pad that attaches to the skin. The controller has the capability to send recharging signals, for the purpose of recharging the implant's battery, or communication signals, which are received and acknowledged by the implantable device by sending a confirmation signal back to the controller. The controller gives feedback to the user, notifying if the correct acknowledgment signal was received, or if it was received at all.

Future implementations of the system will include low-profile standalone energy pads with battery and circuitry to create the recharging signal. A similar energy pad was created and used with the device in this research, but was not conducive to easy laboratory measurements. Because the main focus of the research is to characterize volume conduction technology, the handheld controller was ultimately used to produce recharging signals for the device.

3.2 ENERGY TRANSFER PROTOCOL

A significant aspect of the circuit design is realizing the energy transfer protocol for the system. Two layers exist to this protocol: 1) the physical layer, which dictates the actual form of the electric signals; 2) the software level, which dictates how the device and controller decode the physically represented data. While there is more flexibility in the software level of the protocol, the physical layer is dependent upon a number of factors associated with the entire application. These considerations are:

1. **Electrode Polarization** – Due to the nature of the electrode-skin interface, polarization effects could cause severe problems if not prevented (see section 2.3). To minimize the polarization due to current flow through the electrode-skin interface, all signals through the skin are of a bipolar nature, with the magnitude and time of current flow in one

direction cancelling that in the other. Conversion of DC battery voltage to AC (and back, for recharging the implant battery) is handled in the circuit design.

2. **Low Energy Usage** – The relatively small size of the battery makes it essential to conserve energy wherever possible. As such, the protocols are developed with the following two considerations: 1) using a relatively low frequency to reduce power consumption by the microcontroller of the device; 2) using a simple means to encode communication symbols to minimize computation and active device time.
3. **Availability of Circuit Components** – This point has a larger effect on the protocol design than desired. However, the reality is that using available off-the-shelf components drastically reduces design time and complexity. An integrated circuit (IC) that handles the receiving end of communication signals was chosen with the previous two factors taken into consideration. This directed subsequent protocol development.

With the above considerations, the physical and software protocol layers were developed. Each is detailed in the following sections.

3.2.1 Physical Layer

For communication, the protocol uses on-off-keying (OOK) of 125 KHz signals to represent high and low logic levels. That is, the presence of a 125 KHz signal represents a logic high level, and the absence of such a signal represents a logic low level. This satisfies the requirements mentioned above, namely using a relatively low frequency, bipolar signal with simple logic representation. This form of keying has other advantages for a volume conduction system. The electrode direction is irrelevant when applying the external electrode pad, eliminating the risk of applying signals in the wrong direction or otherwise. The 125 KHz signal is also easy to create, keeping circuit complexity low.

One thing to beware of when implementing a protocol like this is interference with other devices. The 125 KHz carrier signal may not be a problem itself, avoiding stimulation frequencies and utilizing an Industrial, Scientific, and Medical (ISM) radio frequency, but the OOK baseband signal frequency is much lower (value depending on the specific system). This lower frequency signal has the potential to cause unwanted stimulation in the body or interfere with other implantable recording or stimulating devices. To avoid this problem, future designs may have to use even higher frequencies, corresponding with higher power requirements. For this implementation, however, stimulation and interference are of little concern.

3.2.2 Software Layer

At the basic level, the software protocol uses Manchester encoding of data bits, each bit consisting of two logic levels with a transition in the middle of the bit time (low-to-high transition implies binary 0, high-to-low implies binary 1). This is done to guarantee synchronization between the transmitter and receiver. Beyond this level, each communication packet contains sets of bits as outlined in figure 3.3. Communication packets from the external controller to the implantable device contain two extra transmission blocks than those from the implantable device: an initial transmission section to wake up the communication IC in the implant, and a command byte to tell the device how to respond. Data showing these signals is illustrated in figure 4.6.

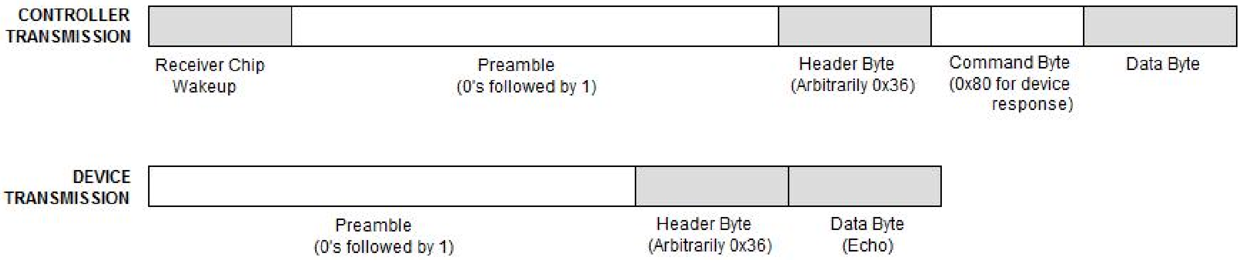

CONTROLLER TRANSMISSION
Receiver Chip Wakeup
Preamble (0's followed by 1)
Header Byte (Arbitrarily 0x36)
Command Byte (0x80 for device response)
Data Byte
DEVICE TRANSMISSION
Preamble (0's followed by 1)
Header Byte (Arbitrarily 0x36)
Data Byte (Echo)

Figure 3.3: Communication packets in the energy transfer protocol.

Because the recharging and communication functions share a single port, it is easiest to share the output stage circuitry between the two. As such, the recharging protocol is based on the communication protocol, but it is much simpler. It uses the same 125 KHz frequency, applied for as long as necessary to recharge the internal battery. As mentioned previously, the implantable device circuitry is responsible for converting the AC signal to

DC to recharge the implant battery. Since recharging is only performed in one direction, external-to-internal, the implantable device plays no role in the recharging protocol.

3.3 SYSTEM ARCHITECTURE

Circuit layouts for the implantable device and handheld controller are shown in figures 3.4 and 3.5, respectively. Tables 3.1 and 3.2 outline the major components of each. Further detailed descriptions of each component are given in the following section, 3.4.

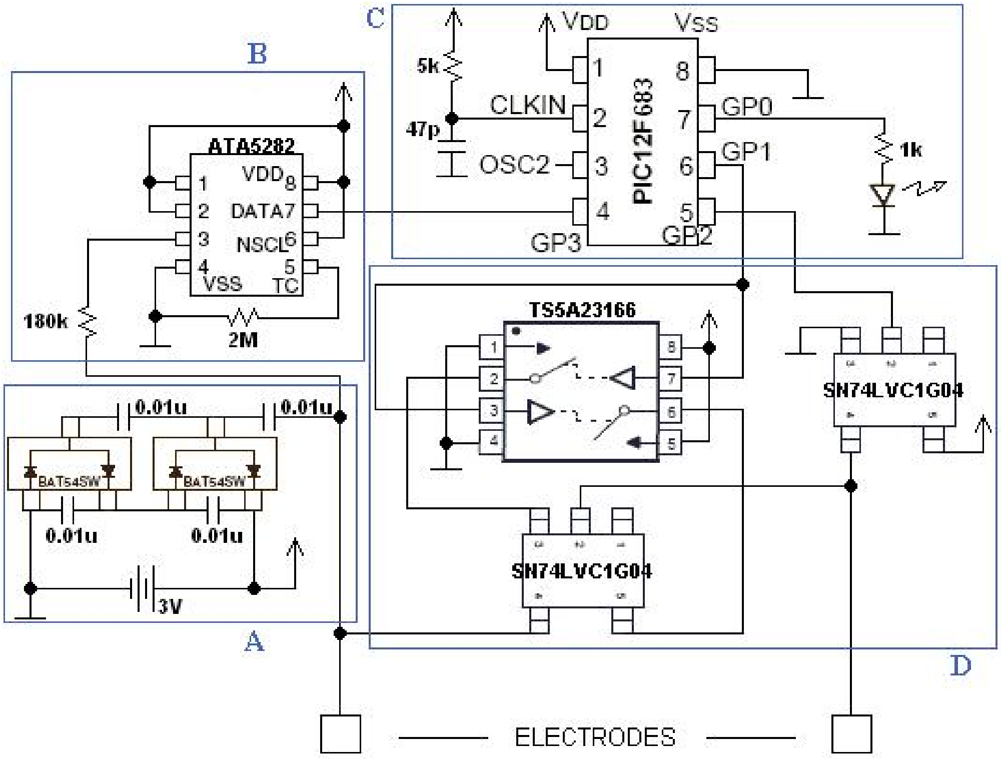


Figure 3.4: Chip-level circuit layout of the implantable device.

Table 3.1: Functional components of the implantable device circuit in figure 3.4.

Functional Component	Description
A	Battery and recharging circuitry (voltage multiplier)
B	Communication input stage (OOK IC)
C	Microcontroller and debugging LED
D	Communication output ICs (switches and inverters)

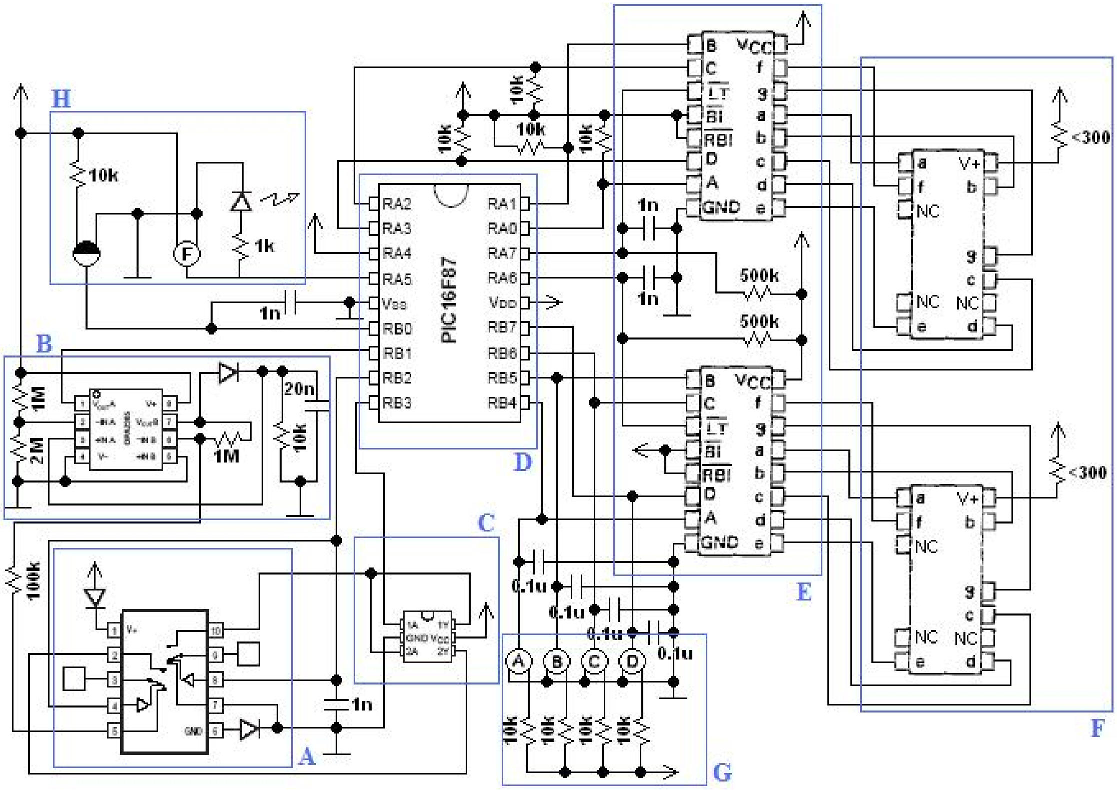
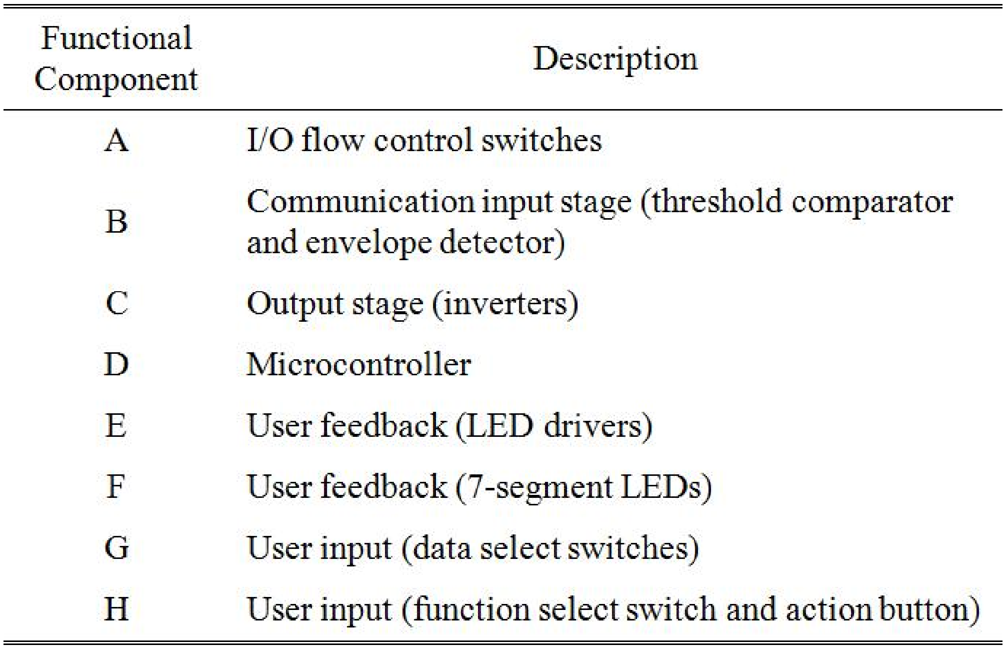


Figure 3.5: Chip-level circuit layout of the handheld controller.

Table 3.2: Functional components of the handheld controller circuit in figure 3.5.



Functional Component	Description
A	I/O flow control switches
B	Communication input stage (threshold comparator and envelope detector)
C	Output stage (inverters)
D	Microcontroller
E	User feedback (LED drivers)
F	User feedback (7-segment LEDs)
G	User input (data select switches)
H	User input (function select switch and action button)

3.4 FUNCTIONAL COMPONENT DESCRIPTIONS

3.4.1 Implantable Device

3.4.1.1 Battery and Recharging Circuitry All electronic components of the implantable device are powered by a rechargeable 3-volt lithium ion coin battery. The battery has a total capacity of about 40 mAh and can survive through almost 500 charge/discharge cycles. Future battery technology will only improve this part of the device. The circuit used to condition the recharging signal is a standard two-stage voltage multiplier, also known as a charge pump. Voltage multipliers are commonly used to amplify and rectify an AC input signal to produce a higher DC output. With the present setup, the output voltage is clamped at the battery voltage level. Applied recharging signals only need to reach a level of approximately 1 V at the internal device electrodes to begin charging the battery. A higher input voltage will not increase the output voltage, but rather increase the charging current and rate of charging.

Studies have shown that one-stage voltage multipliers are more efficient than two-stage multipliers [27], but they lack the ability to amplify voltage as much. From experience with this system, it has been found that the two-stage multiplier provides more reliable charging performance, despite being less efficient than a similar one-stage design. Experiments with this particular voltage multiplier also show that the signal frequency must be above about 70 KHz to achieve maximum recharging current from a given input voltage amplitude [27]. This frequency threshold is dependent upon the capacitance values used in the circuit. If the signal does not switch the diodes quickly enough, leaked charge from the capacitors reduces performance.

3.4.1.2 Communication Input Stage The communication input stage consists of a single ATA5282 amplitude shift keying (ASK) IC. It requires a specific 125 KHz input signal to wake up from its low power standby mode. As such, any non-communication signals are filtered out and prevented from reaching the microcontroller. After the specific header, the ATA5282 operates as a simple envelope detector with automatic gain control (AGC). This converts the OOK input of the communication signal into digital signals for the microcontroller.

3.4.1.3 Microcontroller The microcontroller, a low power PIC12F683, contains all the code for controlling the activity states and communication of the device. It uses an RC oscillator as a low-power clock source (4 MHz) and relies on the bit-syncing properties of Manchester coding to compensate for variations in the oscillator frequency. It remains in a low power state until receiving an interrupt from the communication IC, at which point it receives incoming data and responds appropriately. An LED is attached for debugging purposes during testing, to notify researchers of various errors or confirmations. Code for the microcontroller is listed in appendix A.

3.4.1.4 Output Stage The output stage consists of two cascaded inverters, each output connected to a different electrode. A digitally oscillating control signal can thus be used to create a bipolar output, as required by the communication protocol. The first inverter maintains a low output while the device is idle, essentially connecting one of the input electrodes to the circuit's ground reference so that input signals can be interpreted. The other inverter's power supply is gated by digital switches. The microcontroller manipulates the switches to connect power to the inverter only when necessary. This prevents a short-circuit from occurring during reception of a large amplitude signal on the electrodes.

3.4.2 External Controller

3.4.2.1 I/O Flow Control Unlike the implantable device, the controller actively connects the electrodes to either the input or output stages at all times. Two DPDT switches are used to control which stage is connected. Electrodes are symbolized by the empty squares connected to the switch IC. Using this scheme, the software on the controller knows and dictates information flow at all times.

3.4.2.2 Communication Input Stage The controller's input stage varies from that of the implant's by neglecting the use of the ASK IC. Original designs included the ATA5282, but it proved impossible to separate the outgoing 125 KHz wake-up signal from the onboard IC, thus interfering with the communication timing of return signals from the implantable device. A custom thresholder and envelope detector were thus constructed to perform a similar task. Elimination of the IC also relaxes timing constraints for the input communication signals, allowing the slower implant microcontroller margin for more timing error and making the system slightly more robust.

3.4.2.3 Output Stage Because the controller does not have to worry about input signals larger than its supply voltage, it does not need need to disconnect power from its output inverters (see section 3.4.1.4), and thus uses a single dual-inverter chip. Like the implant, these inverters are cascaded and connected to separate electrodes to create a bipolar output signal from the microcontroller's digital output.

3.4.2.4 Microcontroller The microcontroller, a low power PIC16LF87, contains all the code for interpreting user inputs, giving user feedback, and controlling and interpreting the recharging and communication signals of the device. It uses an internal 8 MHz oscillator while active, and enters in a standby mode after a certain amount of inactivity by the user. Code for the microcontroller is listed in appendix B.

3.4.2.5 User Interface The user interface of the handheld controller consists of a bank of four switches to control data value inputs (sixteen possible data values), one switch to set the transmission mode (either recharging signals or communication signals, status indicated by a single LED), a push button to activative transmission, and 7-segment LEDs to provide feedback. One LED shows the data selected by the user, while the other is reserved for displaying confirmation or error messages (as well as data received back in successful communication cases) after the controller communicates with the implantable device. After a period of about 30 seconds, the microcontroller turns off the LEDs and enters a low power state. Any input change will wake the microcontroller and activate the feedback LEDs again.

3.5 ELECTRODE CONNECTIONS

As discussed in section 2.3, quality electrode connections are essential for good performance and long term reliability of the volume conduction system. Given that the system requires charge to actually pass between the external controller, the skin, and the implantable device, a faradaic electrode-skin connection is desired. Although silver-chloride (AgCl) electrodes are popular for their faradaic ion transfer, their performance changes over short time periods when exposed to larger amounts of current, and they are not reusable, necessitating greater cost. Metal electrodes were necessary for ease in testing, and so gold-coated copper was chosen as the electrode material. These electrodes, while they do experience some degradation over time (see section 2.3.1), can be used multiple times before needing to be replated with gold, and can be made very easily.

Various physical electrodes were originally created with small profiles (see figure 3.6) for testing of the circuit. The size restrictions on these electrodes were simply too drastic, however, greatly inhibiting system performance. According to the preliminary work in 2.5.2.2, electrode separation plays the largest role in keeping current transmission efficiency high, and a basic geometry of two separate electrodes gives better results than concentric circles or an electrode array. The locations of solder connection points on the electrodes may also affect current distribution at the electrode surface, but such investigation remains for future work. A larger set of electrodes, shown in 3.7, was created to compensate. These electrodes determine the size of the energy pad, which is not an important factor for initial system testing and characterization. System performance with these electrodes greatly improved, and so they are used as the final electrodes for the test system.

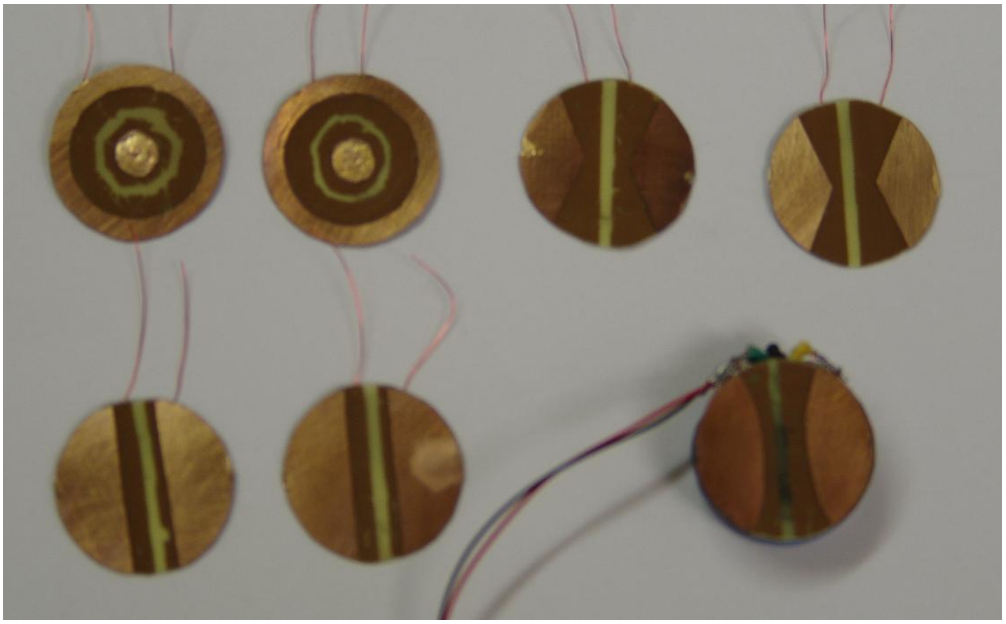


Figure 3.6: Small electrode designs, each the size of a U.S. nickel. These ultimately proved too small for effective current transfer across skin.

Another factor besides the electrode material is the contact preparation involved before attaching the electrodes. The skin consists of multiple layers, with a single external stratum corneum layer contributing to most of its electrical barrier properties, as described in 5.1. Slight abrasion of this external stratum corneum layer, either through application of standard skin prep gels (e.g. NuPrep^{TM} skin prep gel) or fine sandpaper abrasion, decreases contact

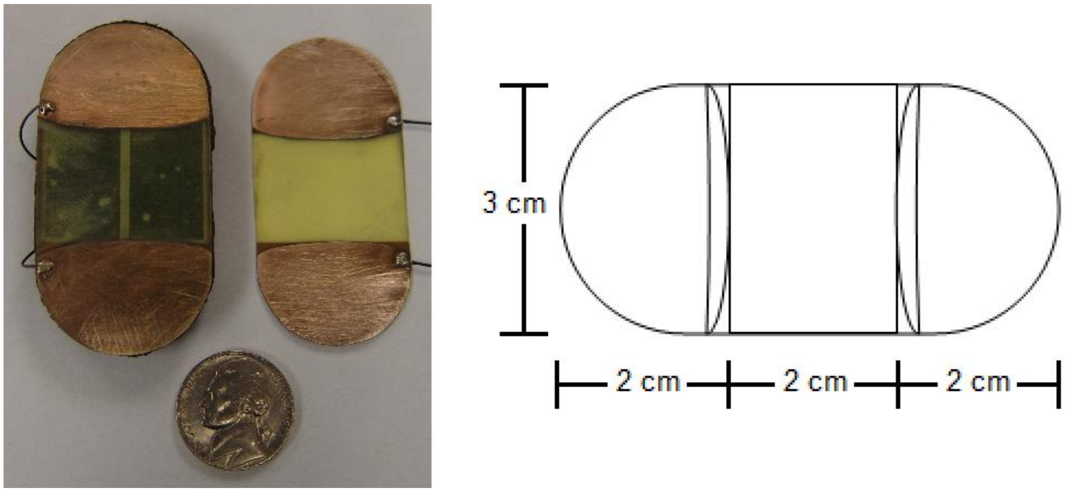


Figure 3.7: Large gold-plated copper electrodes used in the final system. In the left panel, the left electrodes are on an energy pad used for external connections. The right electrodes connect the underside of the skin to the implantable device.

impedance drastically. Applying a conductive paste (e.g. GRASS EC2® electrode cream) to the external electrodes also helps to decrease the contact impedance and increase system performance and reliability. From experimental measurements, the electrode connections to the internal side of the skin seem to need no special preparation to form a stable, low impedance interface (see chapters 4 and 5). This is likely due to the full conductive contact provided by saline, blood, and other ion-containing liquids on the underside of the skin.

3.6 DEVICE CONSTRUCTION

Both the handheld controller and implantable device were constructed so as to be practically usable equipment. The implantable device has been through many iterations, before finally arriving at the final realization in the right panel of figure 3.9. The device is built on a 0.005" printed circuit board (PCB) (layout shown in figure 3.8) and encapsulated in epoxy to seal and protect the circuitry from its environment. The wire leads give the implant flexibility to attach to distant batteries or electrodes, providing freedom of placement during testing and ample test points to take measurements. The controller, pictured in figure 3.10, is powered by four AA batteries, the holder for which provides weight and dimension for easy usage.

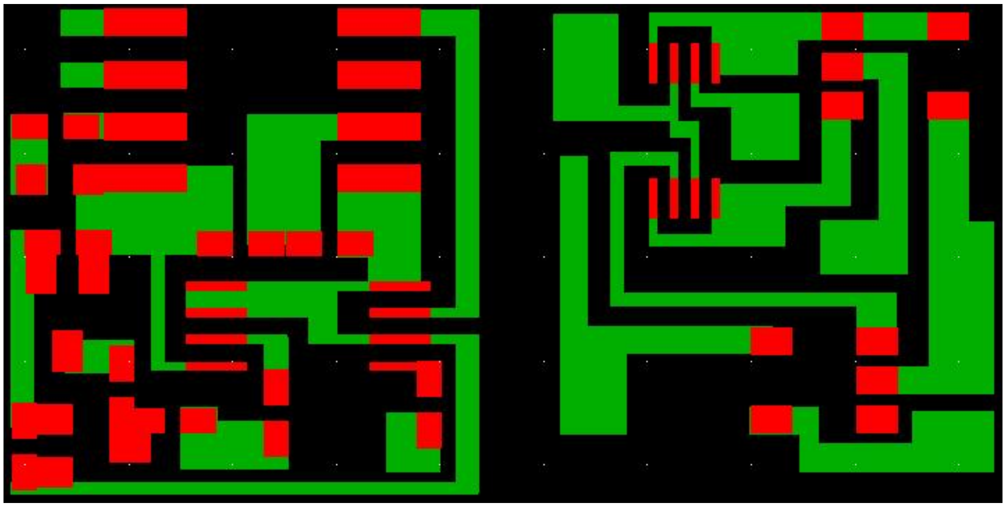


Figure 3.8: PCB layout for the implantable device.

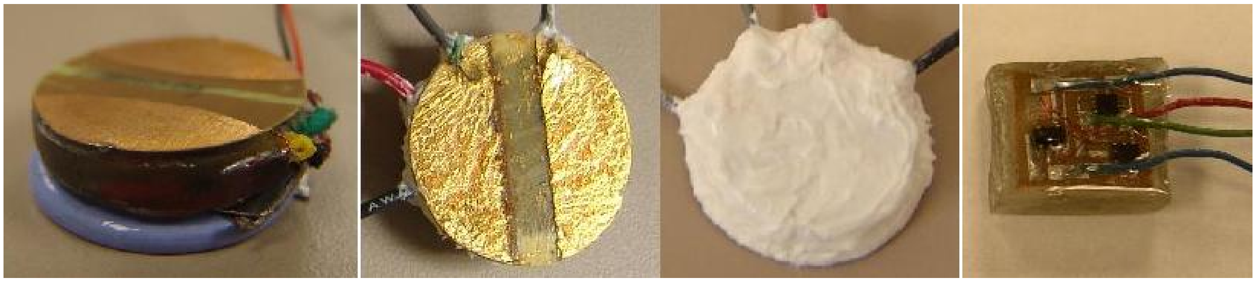


Figure 3.9: Left: First generation of the implantable device with battery included. The small electrodes hindered performance. Middle: The second generation device used slightly larger electrodes and used an improved communication protocol. Right: The third and final generation of the implantable device fixed a debilitating circuitry flaw in the previous two generations and uses distant electrodes to contact the underside of the skin.

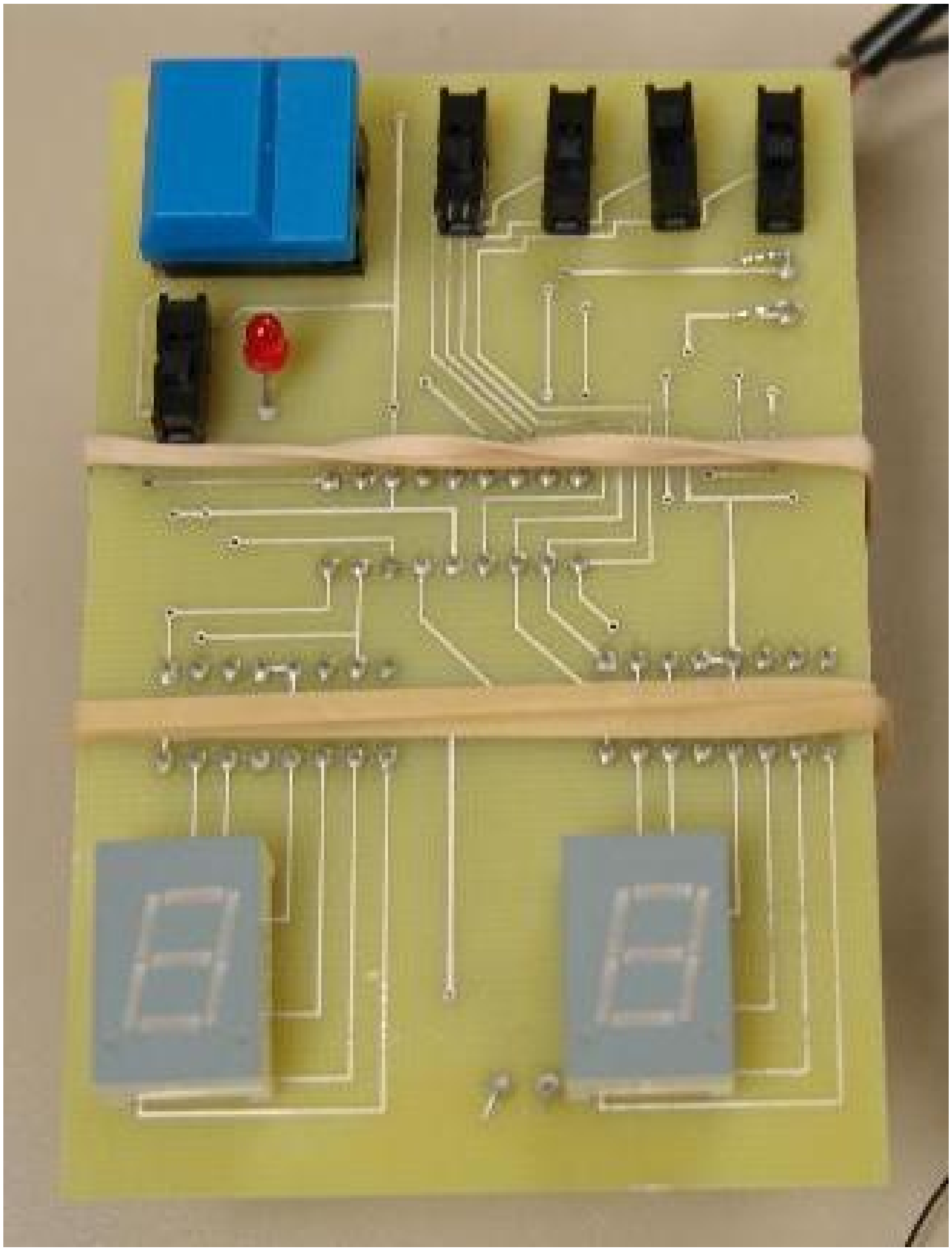


Figure 3.10: Handheld controller with rubber bands holding 4 AA batteries on the back. The single black switch toggles between communication and recharging mode, indicated by the red LED. The four black switches set data bits as either 0's or 1's. The blue button signals the controller to send a data packet or is held to provide a recharging signal, and the 7-segment LEDs on the bottom give visual feedback.

4.0 DEVICE TESTING AND VERIFICATION

This chapter discusses the various experiments performed during the design and development of the volume conduction implantable device and its system model. Experiments began using slabs of agar to represent the skin as a volume conductor. A head model incorporating such agar and a separate conductor for the skull was made to represent the initial intended application environment (DBS). Upon reaching a reliable state of functionality, the device was tested using pieces of pig skin from sacrificed swine as a model for human skin. A final experiment involved testing the device in a live pig under anesthesia. During the experiments involving pig skin, both live and dead, impedance measurements were taken to be used in creation of the model in chapter 5.

4.1 AGAR AND HEAD MODEL

Commonly used as a substrate for culturing in microbiology, agar is a type of gelatin made from structural sugars in certain types of seaweed. It works well for modeling biological tissues because it is easily molded into desired shapes and has mechanical properties similar to soft tissues. By adding salt (NaCl or other ions) to the powdered mixture during preparation, the electrical conductivity of agar can be adjusted to match that of various tissues. Although agar makes a uniform volume conductor with no layers like the skin, it is sufficient for representing the skin during preliminary testing of the implantable device. Various research groups have had success in using it as an electrically accurate physical model for their studies [75, 81, 37].

To determine the correct amount of salt to add to the agar mixture for conductivity tuning, a number of samples were created with different amounts of NaCl. Their conductivities (or perhaps more properly, their "admittivities") were measured and fit as a function of NaCl concentration, shown in figure 4.1. The roughly linear fit gives a good idea of the range of concentrations to use in making agar for experiments. Although the measurements were performed at a frequency of 5 KHz, the conductivity of wet skin and salt solutions does not change much over the kilohertz range up to 1 MHz [17]. Precise conductivity is also not a necessity, since the agar is used as an inexact test bed for debugging the implantable system's timing and functionality. In all experimental procedures involving agar, the agar was made with 0.63 grams NaCl per liter of water, aiming for a conductivity of 0.21 S/m, similar to that of the skin in the kilohertz region [17].

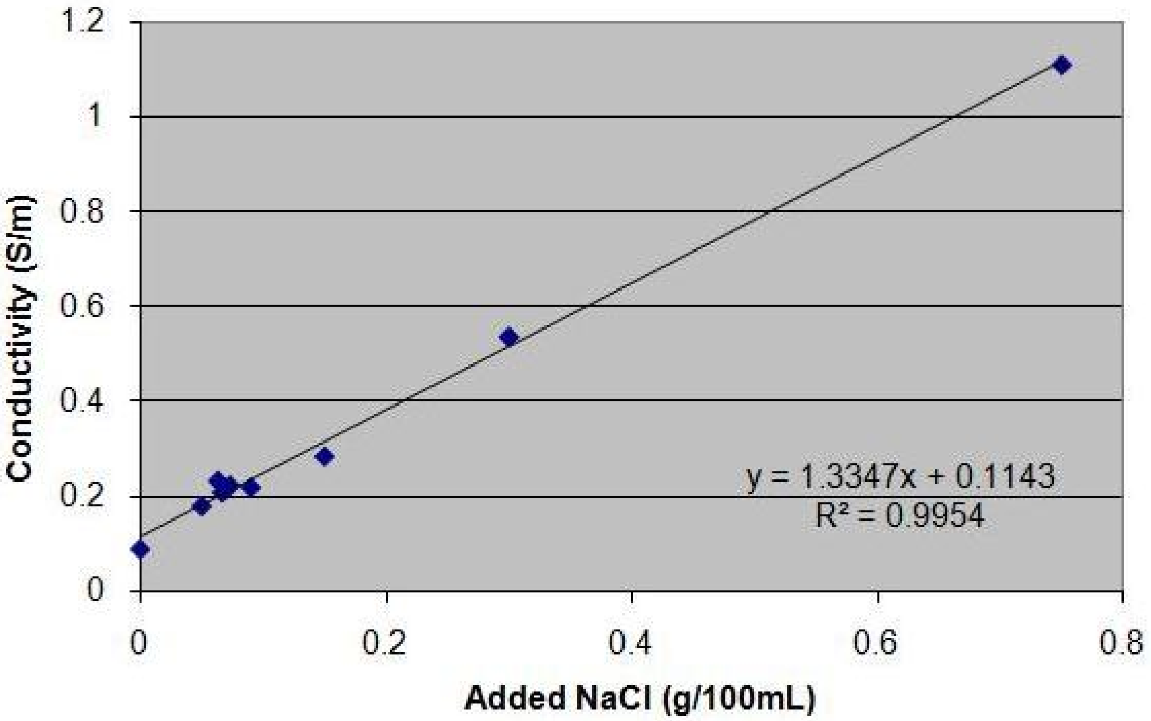


Figure 4.1: Calibration measurements for tuning agar conductivity with added NaCl.

To test the system in an environment similar to the initial intended DBS environment, a head model (or phantom) was created with electrical properties close to that of a real human head. Agar, as described above, was used for the scalp and brain. Inspired by [62], a mixture of SPI Conductobed Carbon Black-filled conductive epoxy and barium titanate ($BaTiO_3$) dielectric was used to match the physical hardness and conductivity of the skull. The conductive carbon black epoxy gives the material its conductivity, while the barium titanate adds a dielectric effect and mediates the admittivity of the mixture. Similar to the agar conductivity calibration, various samples of the epoxy/$BaTiO_3$ mixture with different amounts of added $BaTiO_3$ (by percent-volume) were created to calibrate the material's conductivity. Figure 4.2 shows the measured sample conductivities, the results matching the same general form as those in [35]. Based on these results, the final epoxy mixture for the skull was made with 25-percent-volume $BaTiO_3$, giving an appropriate conductivity of about 0.01 S/m according to a 1:1/17:1 ratio of scalp:skull:brain conductivity [68].

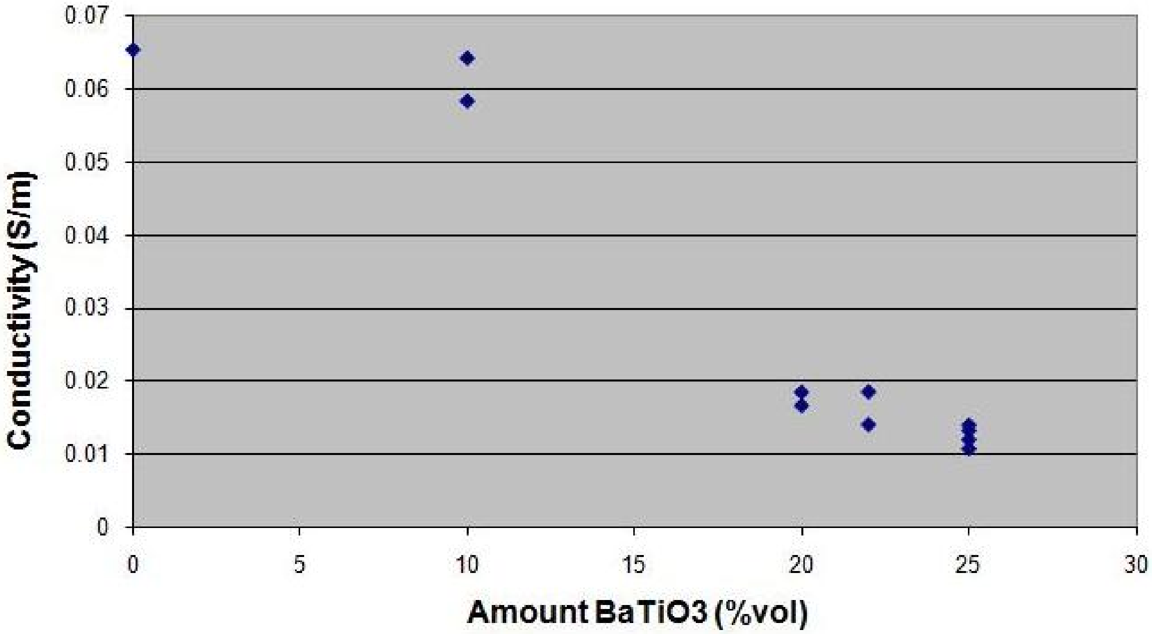


Figure 4.2: Calibration measurements for tuning carbon black epoxy/$BaTiO_3$ conductivity.

Actual construction of the head model was performed using various molding and casting techniques. The base was constructed from a liquid plastic polyurethane, cast from a Styrofoam head model. The epoxy skull required a silicone cast with model brain insert, both created from a life-size plastic skull, shown in figure 4.3. After creation of the base and skull components, an implantable device for testing was placed into a hole in the skull and the agar for the scalp and brain were cast around it. Figure 4.4 shows the completed head model. Using the head phantom for testing the implantable device was instrumental in developing the system's structure and functionality. It greatly helped in diagnosing system behavior, developing the communication protocol, correctly programming the internal and external microcontrollers, and improving the system's reliability.

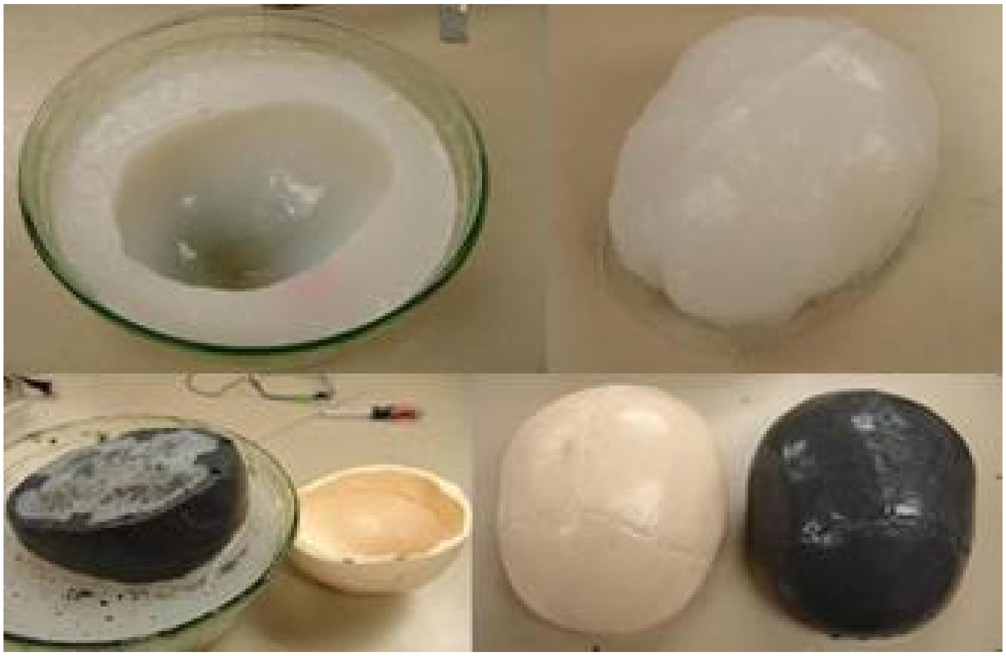


Figure 4.3: Carbon black epoxy/$BaTiO_3$ skull phantom, cast from a life-size plastic skull model.

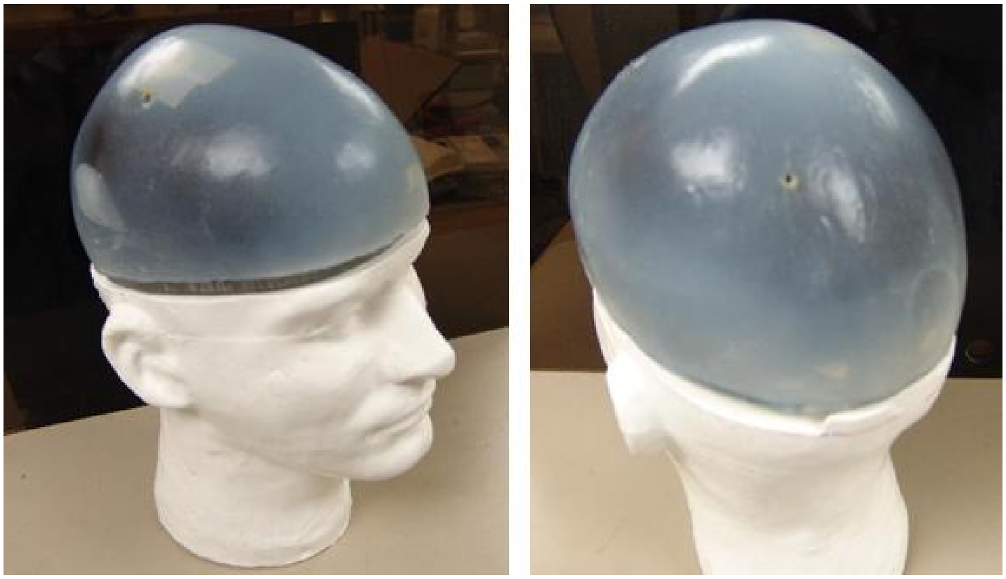


Figure 4.4: Completed head model with agar for scalp and brain.

4.2 VERIFICATION IN LIVE PIG EXPERIMENT

The volume conduction system developed and verified through head phantom testing, after also being tested through numerous pig skin experiments, was verified to work in an animal model (swine) of the intended application setting. Figure 4.5 shows placement of the final device in the skull of the pig, with electrodes simply placed above the device under the scalp for simplicity. Wires extending from the device allow monitoring of the communication and recharging signals, as well as the battery current of the device.

All sixteen possible communication data packets were transmitted to and acknowledged by the internal device with no errors. Figure 4.6 shows the recorded electrical signals at the device electrodes during transmission of a '5' using the communication protocol. Individual sections of the packets as outlined in figure 3.3 are marked.

Recharging signals were applied to the external electrodes so that the device's recharging current could be measured. Because the handheld controller only has one setting for the

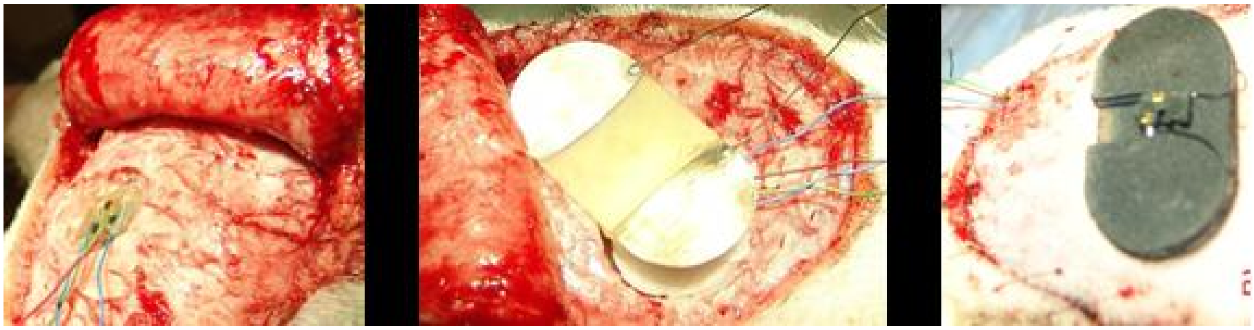


Figure 4.5: Implantation of the volume conduction device *in vivo.* The scalp was 8 mm thick. Left: Device placed within the skull. Middle: Internal electrodes placed under the scalp. Right: Scalp sutured shut and external electrode pad pasted in position over the internal electrodes.

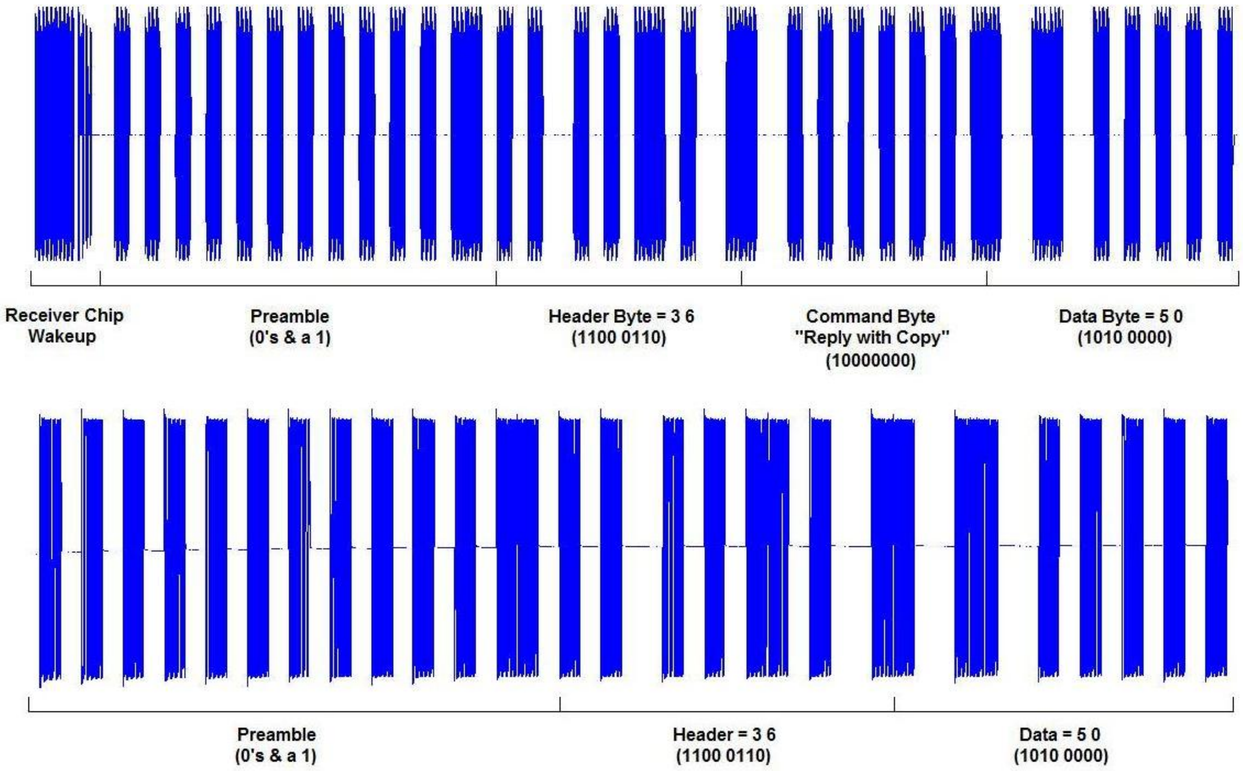


Figure 4.6: Top: Communication signal from the handheld controller as recorded at the internal electrodes. Bottom: Communication signal from the implanted device in response to the controller's signal. The separate sections of each packet are marked.

recharging signal, the signals were instead applied via a function generator. All possessed the designed 125 KHz frequency (with the square waveform including higher frequencies as well), but waveform amplitudes were changed to see the effect on the recharging current. The difference between sinusoidal and square wave inputs was also examined. For comparison, the same measurements were taken with the recharging signal connected directly to the implantable device electrodes. Results are plotted in figure 4.7. Note that the current is that of the device battery: positive current indicates the battery is being recharged and negative current indicates the battery is losing charge. The recharging profiles are similar for each set of measurements, though the higher frequency components of the square waveform likely lose less energy through the skin as compared to the pure sinusoidal waveform (right panel, 4.7), seeing as the sinusoidal and square waveforms have almost equal effect when applied directly to the internal electrodes (left panel, 4.7). The roughly piecewise linear nature of both recharging profiles is an indication that voltage attenuation across the skin is a linear function, supporting the model assumptions made in the next chapter (5).

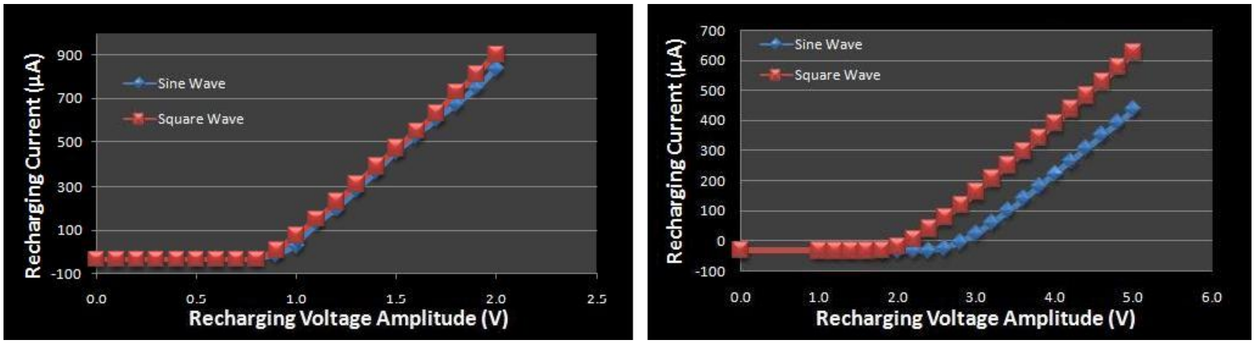


Figure 4.7: Left: Device recharging current as a function of voltage amplitude of recharging signals applied directly to the implantable device electrodes. Right: Similar plot with recharging signals applied to the electrodes on the external side of the skin. Positive recharging current indicates current going into the battery to actively increase its charge.

During the experiment, impedance measurements were also taken on the scalp using a Gamry machine. These measurements are described in the next section.

4.3 SYSTEM IMPEDANCE MEASUREMENTS

These experiments play a very important role in this thesis. Their results are essential to understanding the behavior of the electrode-skin system on which the entire implantable device system depends. The measurements described here are also used in development and verification of the system model in chapter 5.

Two major types of impedance experiments are detailed, one involving use of a Gamry machine for electrochemical measurements, and another involving use of a vector network analyzer (VNA) for impedance measurements across a large frequency range.

Most of these experiments involve the use of pig skin as a model for human skin, in which the actual implantable devices will eventually be used. Numerous studies have shown that swine skin is indeed a close approximation to human skin, and almost certainly better than other commonly used non-primate animal models [20, 33, 22]. Importantly, pig skin contains the same major structures and form as human skin [63], as well as being roughly the same thickness, depending on location.

For all experiments involving skin except for the *in vivo* measurements, the electrode-skin setup is identical. A petri dish is filled with a thin layer of saline solution, less than 2 mm, and ideally less than 1 mm. One pair of thin electrodes (see figure 3.7) is placed in the dish with its contact surfaces facing upward. The piece of skin, first shaved of hair, is then placed in the dish with its internal side facing downward. The thin saline film keeps the bottom side of the skin moist during the experiment, while the external side of the skin is exposed to the air similar to a real application situation. The external side of the skin is then lightly abraded with fine skin prep sandpaper and the external electrodes are attached using conductive electrode paste, with care taken to align them over the electrodes underneath. The resulting system can then be seen as a four-terminal network, as in figure 4.8. For each piece of skin tested, measurements are taken across each unique pair of terminals in the system (leaving the other two terminals disconnected) for the desired frequency range, resulting in six series of measurements in the form of impedance vs. frequency. During measurement across each pair of terminals, the exact orientation of the connections theoretically has an effect on the measured impedance due to the creation of relatively small half-cell potentials across the electrode-electrolyte or electrolyte-skin interfaces. However, the combination of highly polarizable gold electrodes (and highly non-polarizable conductive paste at the skin interface) and small sensing currents and voltages drastically reduces the creation of such half-cell potentials, also reducing any effect on the measured results.

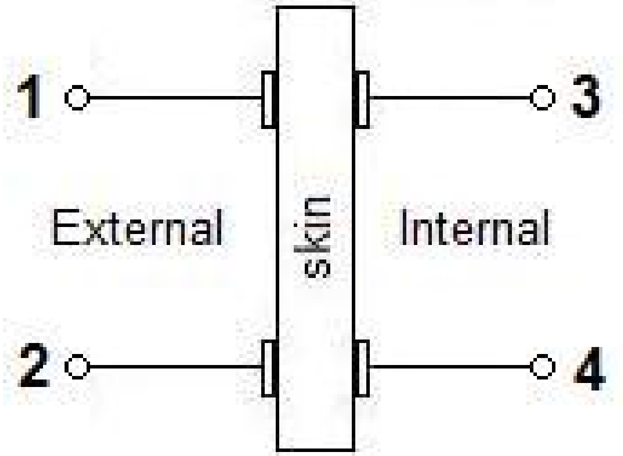


Figure 4.8: Diagram of the general skin measurement setup. The two pairs of electrodes on the external and internal sides of the skin form a four-terminal network.

Not controlled during the experiments is ambient temperature. During the experiments with the Gamry machine, ambient temperature was measured to be about $69° - 70°$F, but no such measurements were taken during the VNA measurements. Also of note is the time between pig sacrifice and skin testing. According to [11], tissue conductivity remains relatively constant within an hour after sacrifice, and longer for some tissues. All experimental measurements here were taken within two hours after animal sacrifice to ensure the skin samples experienced as little change in electrical properties as possible. Samples were transported in containers of room-temperature saline and never frozen, which could cause irreversible damage.

4.3.1 Gamry Measurements

4.3.1.1 Gamry Machine Usage The Gamry Potentiostat, FAS2/Femtostat, under control of the Gamry Framework software from Gamry Instruments, is a piece of equipment that facilitates automated electrochemical measurements. The software runs on a Dell Dimension 2400 desktop computer. When set up for Electrical Impedance Spectroscopy measurements, it uses a two-electrode system to measure two-terminal impedances across a range of frequencies. Possible frequencies range from millihertz to 100 kilohertz. For measurements taken here, the frequency range was set to 1 - 100 KHz. Obviously this frequency range is below that used by the implantable device, but these measurements are used in a more general manner to gain information about system behavior and to help in establishing the system model in the next chapter.

4.3.1.2 Impedance Measurements The Gamry machine was used to take impedance measurements in three separate experiments, detailed in 4.3. The first experiment was performed on a piece of excised skin from the scalp of a pig (skin #7 for future reference) about 7 mm thick. The second experiment was performed on a live pig scalp (skin #8 for future reference), about 7 mm thick, during testing of the implantable device in a live pig (see section 4.2). The third experiment was performed on a piece of excised skin from the ear of a pig (skin #9 for future reference), about 3 mm thick. Ear skin was chosen per the suggestion of [33] as the closest match to human skin. Figure 4.9 shows the three pieces of skin used. An example of the measured impedance data for skin #8 is shown in figure 4.10.

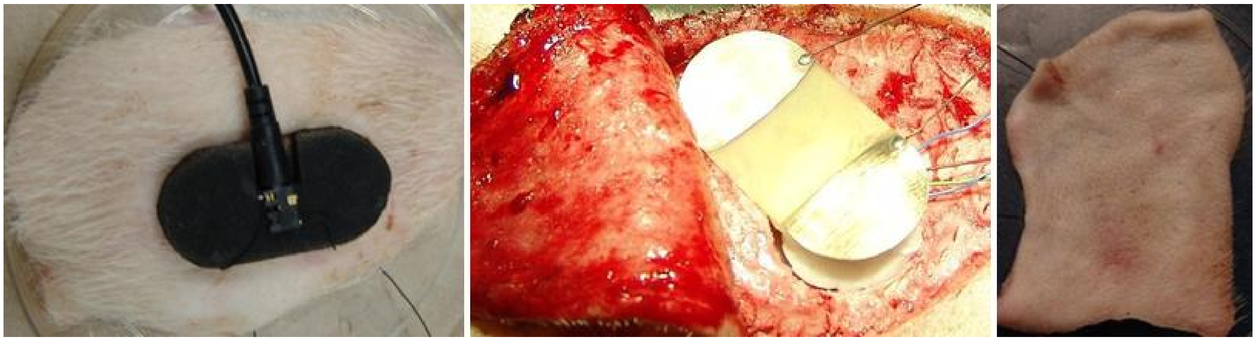


Figure 4.9: Pig skin used for low frequency measurements with the Gamry machine. Left: Excised scalp skin (#7), 7 mm, in petri dish with external electrodes (energy pad) attached. Middle: Live scalp skin (#8), 7 mm, and internal electrodes. The top of the scalp with external electrodes (energy pad) can be seen in the right panel of 4.5. Right: Excised ear skin (#9), 3 mm, in petri dish on top of internal electrodes.

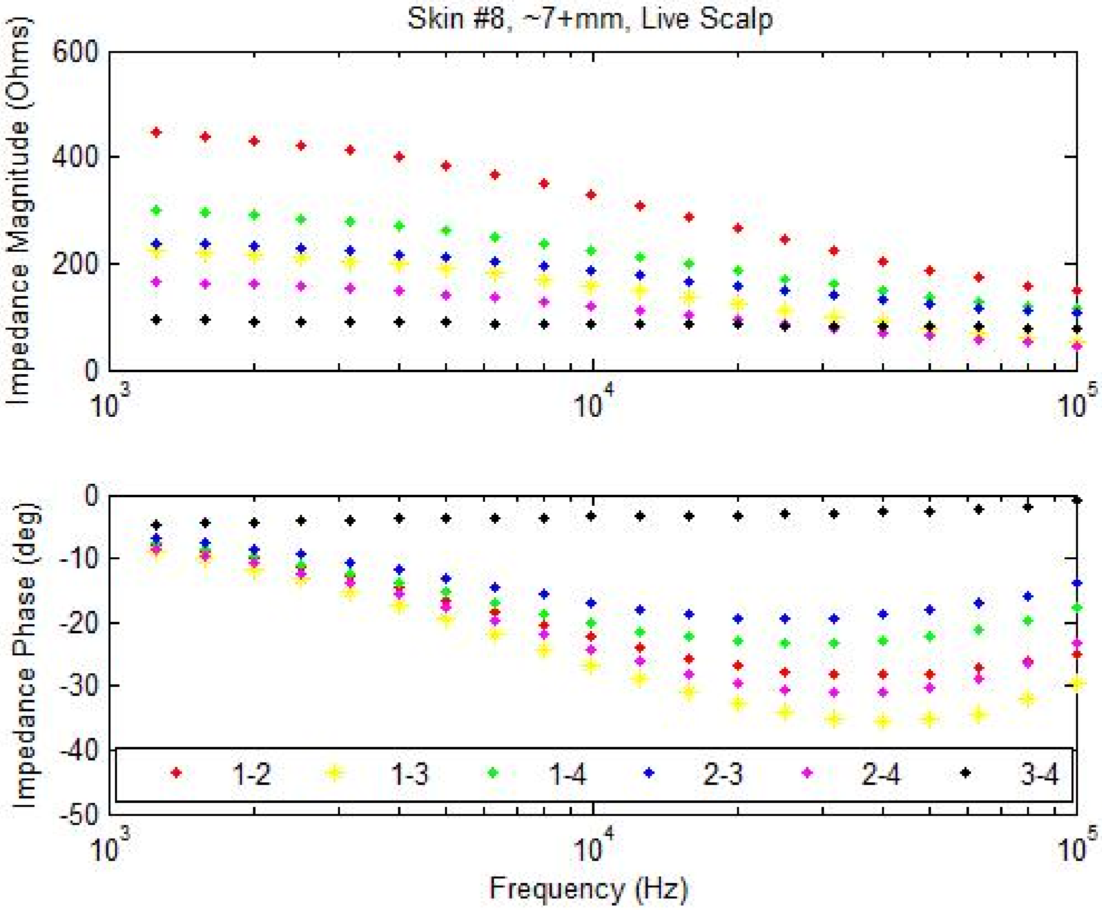


Figure 4.10: Impedance measurements on skin #8, live scalp, using the Gamry machine. Plotted are the magnitude and phase of the impedances between each unique pair of terminals, as defined in figure 4.8. Legend specifies pairs of numbered terminals for each measurement series.

4.3.2 Network Analyzer Measurements

4.3.2.1 Network Analyzer Usage The Agilent HP 8753ES Vector Network Analyzer (VNA) is able to reliably make impedance measurements from 30 KHz to the GHz range (among other functions). It is commonly used in high frequency signal and circuit characterization, but here it is used only for measuring the impedance between two terminals (one "port"). It does this by sending a signal (or "stimulus") to the connected device and measuring the reflected signal it receives back, an S_{11} measurement. From the difference in the transmitted and reflected signals, the VNA calculates the impedance seen at the connected device port. In this case, the device port is any one of the pairs of terminals of the electrode-skin system (see figure 4.8).

An important step in using the VNA is setting the proper stimulus, both frequency range and power level, for the intended measurements. Because one of the advantages of using volume conduction technology is the ability to use low frequencies to conserve power and avoid high frequency signal attenuation when transmitting signals, focus stays in the frequency range below 1 MHz. Thus the frequency range of the stimulus is limited to 30 KHz - 1 MHz. The stimulus power level is set to 10 dBm, corresponding to 10 mW. In a recharging scenario for the implantable system, this would correspond to a signal through the electrodes of 2.5 V and 4 mA. Considering a recharging current of about 1 mA (see figure 4.7) and a two-stage voltage multiplier efficiency of 25% [27], it is very realistic to expect 4 mA at the electrode-skin connection with a voltage differential of 2 - 2.5 V. Thus, the 10 dBm stimulus power level is a reasonable choice to use for impedance measurements.

4.3.2.2 Impedance Measurements The VNA was used to take measurements on six different pieces of pig skin. These pieces came from two different animals, the first 3 from the flank (skin #'s 1, 2, and 3) of one pig, and the last 3 from the flank (skin #'s 4 and 5) and underbelly (skin #6) of a second. The sections of skin are shown in figure 4.11. Each experiment was performed as described in section 4.3. An example of the measured impedance data for skin #2 is shown in figure 4.12.

In addition to measuring impedance values for the electrode-skin setups, two other important impedance values of the system were measured. These are the impedance of the electrode connections themselves with no skin in between, and the impedance of the implantable device as seen from the electrode connection terminals. The reason for the former is to verify that the electrodes themselves, i.e. the soldered wires, clips, and electrode/paste interface, contribute little to the electrode-skin impedance measurements, or to discover their contribution if otherwise. The device impedance, beyond knowing for thoroughness, is used

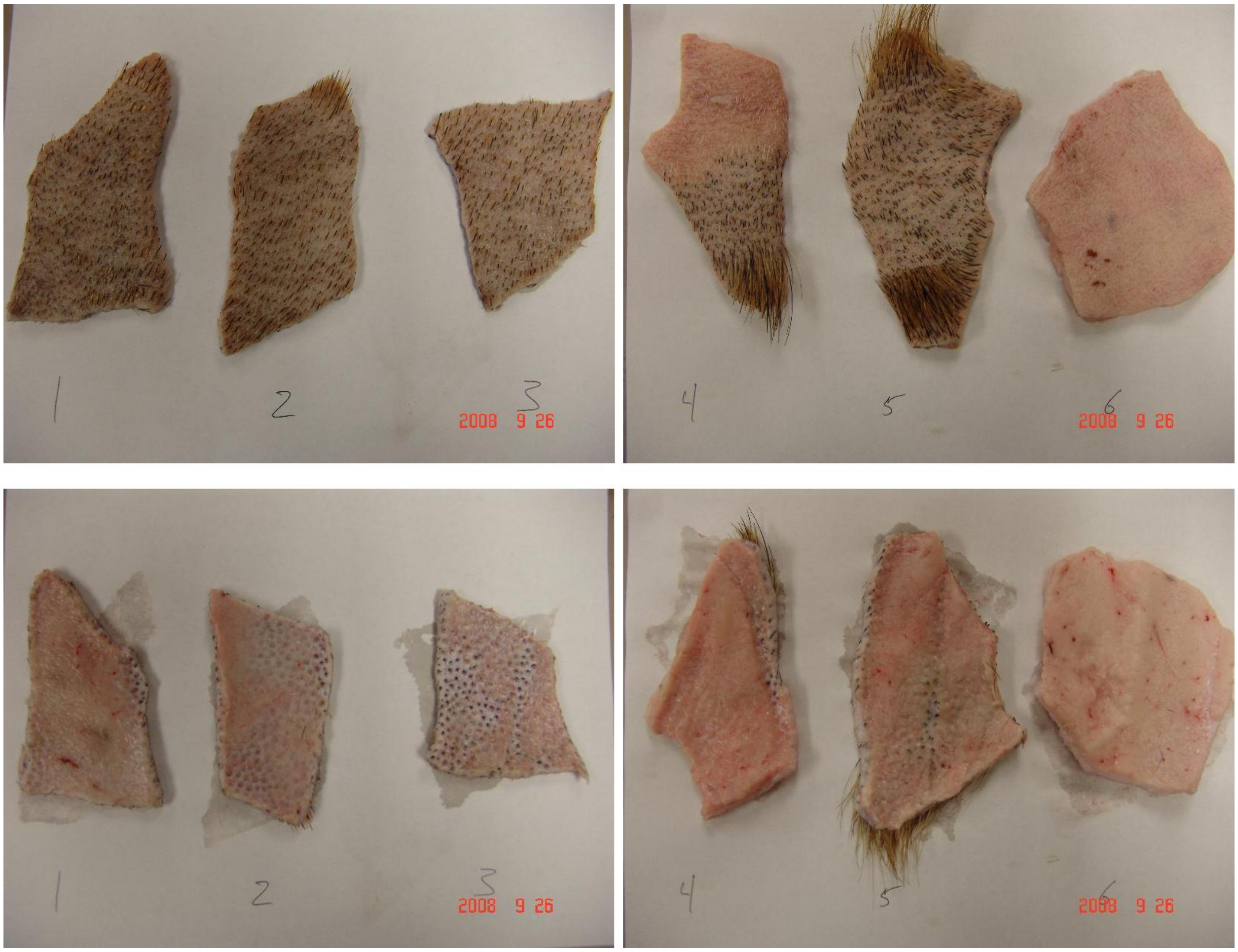


Figure 4.11: Pig skin used for high frequency measurements with the VNA. Skin thicknesses as numbered: 1) 6 mm ; 2) 5 mm ; 3) 4.5 mm ; 4) 7 mm ; 5) 4.5 mm ; 6) 7 mm. Skin from one pig is on the left. Skin from the second pig is on the right. The top row shows the external sides of each piece after being shaved, while the bottom row shows the internal side.

to compare with ideal load impedances found from optimizing the system with the model described in chapter 5.

For the electrode connection measurements, the two pairs of electrodes (figure 3.7) were simply pasted together with the same conductive electrode paste used on the external electrodes for the previous impedance measurement experiments. The VNA was then used to measure the impedance between connected terminals, namely pairs 1-3 and 2-4. The implantable device impedance is similarly measured using the VNA. Results from the electrode

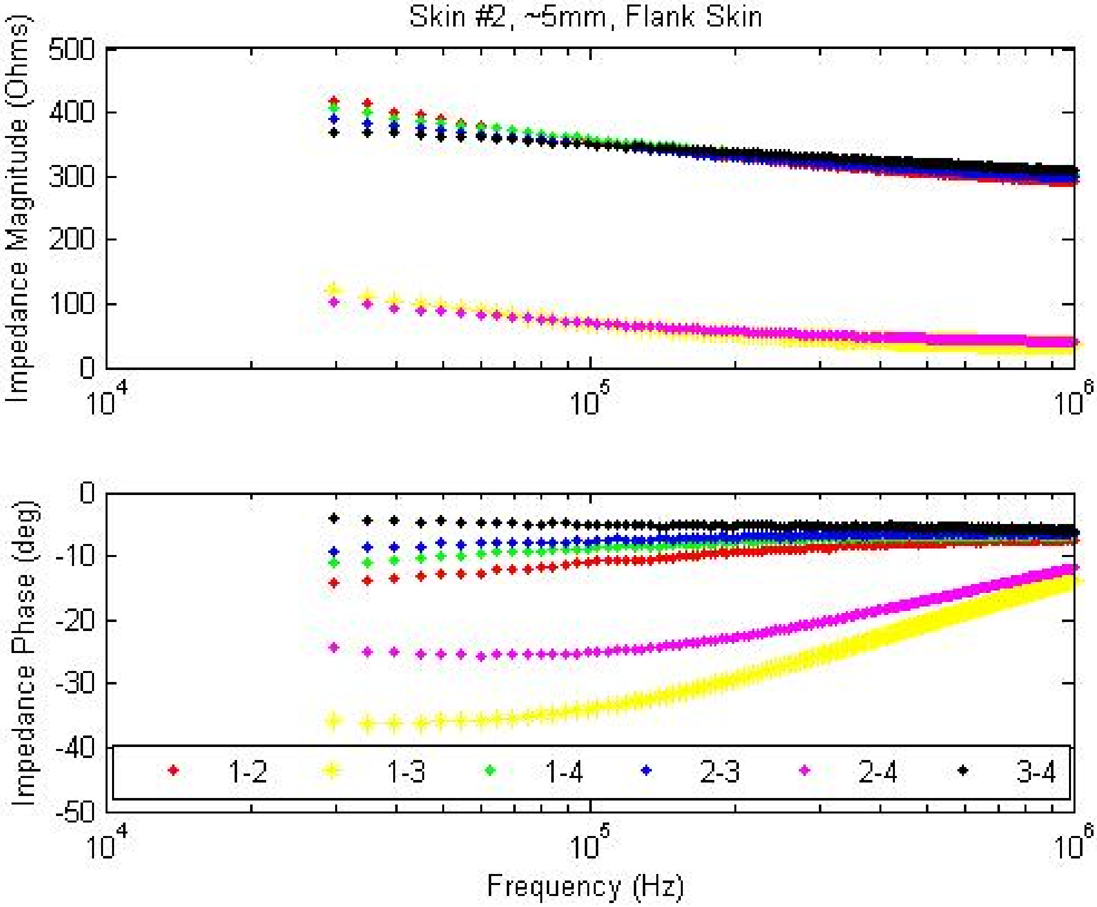


Figure 4.12: Impedance measurements on skin #2, flank skin, using the VNA. Plotted are the magnitude and phase of the impedances between each unique pair of terminals, as defined in figure 4.8. Legend specifies pairs of numbered terminals for each measurement series.

measurements, displayed in figure 4.13, show the electrodes have very low impedance on the order of ten ohms or less, confirming that they contribute little to the measured electrode-skin impedances. The implantable device impedance is plotted in figure 4.14.

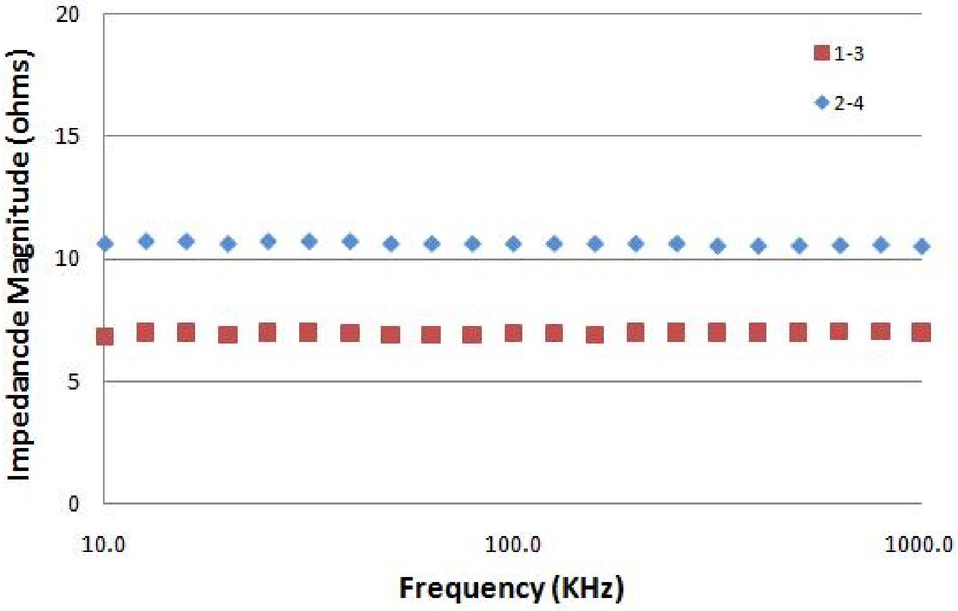


Figure 4.13: Impedance measurements of electrode connections without skin.

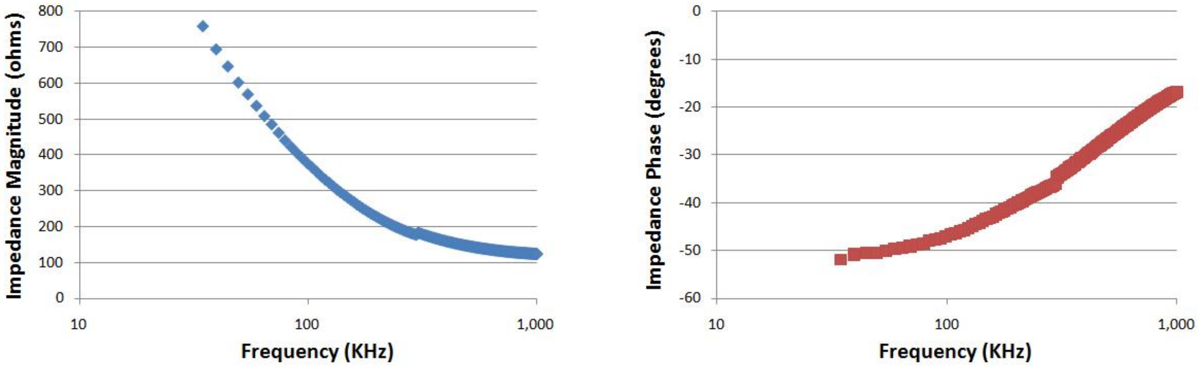


Figure 4.14: Impedance measurements of the implantable device as seen from the electrode connection terminals.

5.0 ELECTRODE-SKIN MODEL DEVELOPMENT AND USE

Volume conduction energy transfer technology has been shown to be a feasible solution for the energy requirements of implantable devices [26, 91, 89], but systems that incorporate the technology must still be custom-designed for each particular application. Currently, there is no established framework in which to develop a volume conduction system, which necessitates long development schedules or use of other more established, yet perhaps less appropriate, technologies to fulfill a device's powering and communication needs. This work aims to establish a standard volume conduction system model which researchers, engineers, and developers can utilize to determine valuable system parameters and constraints necessary during the design process. Such a tool will make volume conduction technology more available as an energy transfer platform for future implantable devices.

This chapter details the design and use of the volume conduction system model, specifically an impedance model of the electrode-skin system, starting with its physiological and electrical bases. The model is then introduced and its similarities to and advantages over other models are described. Verification of the model is performed by examining its ability to represent actual electrode-skin impedance measurements. Details about use of the model in an engineering capacity are then explained.

5.1 MODEL BASIS

Development of the model begins with an examination of the system it represents. The diagram in the left panel of figure 2.13 is a good representation of the physical structure: two pairs of electrodes on either side of a portion of skin. The model in the right panel of the figure is also a fair approximation of the system, but it lacks adequate representation of the different skin layers and current pathways through electrical components of the system. Instead, it assumes a more uniform volume conductor and takes a general "black box" approach in modeling the skin. Given that device safety is one of the most important factors in an implantable system, representation of conductive pathways in a model so that the designer can know and adjust the current density distributions within the skin is essential. For this reason, a simulation of the system to determine the current pathways through the skin in this volume conduction setting is performed to guide the development of the model.

FEMLAB 3.1, a relatively simple finite element modeling program, was used to construct and simulate a FEM model of the electrode-skin system, shown in figure 5.1. The FEM model shares the same practical dimensions as the actual system, i.e. 2 cm wide electrode surfaces with 2 cm of separation between electrodes on a single side of the skin, which is 5 mm thick. The electrodes are thicker than the 0.005-inch thick electrodes of the constructed system, but their high conductivity makes this difference irrelevant. The implantable device is given a conductivity of 0.02 S/m so that its "input resistance" is roughly 300 ohms, matching that in figure 4.14 at ˜125 KHz. The simulation was performed as a Conductive Medium in DC, with the upper external electrode boundary set to +5 V potential and the lower external electrode boundary set to -5 V potential. The DC condition is a reasonable assumption if one assumes the simulation examines such a small window of time that any oscillatory signals in the actual system appear as DC.

Inherently assumed by the FEM model structure is uniformity of the entire skin. This appears contrary to the real world situation, where the skin consists of many layers of cells with varying degrees of density and moisture. Most notable is the stratum corneum, however, a very thin multi-lamellar structure which contributes most of the skin's physical, chemical, and electrical barrier properties [7, 46, 106]. With the stratum corneum present, even appendageal shunts such as sweat glands and hair tubes have relatively little effect on the skin's impedance layer [96]. However, removal of this barrier via abrasive means (which is done for all external electrode connections in this work) allows the electrodes to interface directly with the underlying stratum granulosum. This layer of the skin and the layers beneath have a relatively uniform conductivity, justifying the use of the uniform volume conductor in the FEM model.

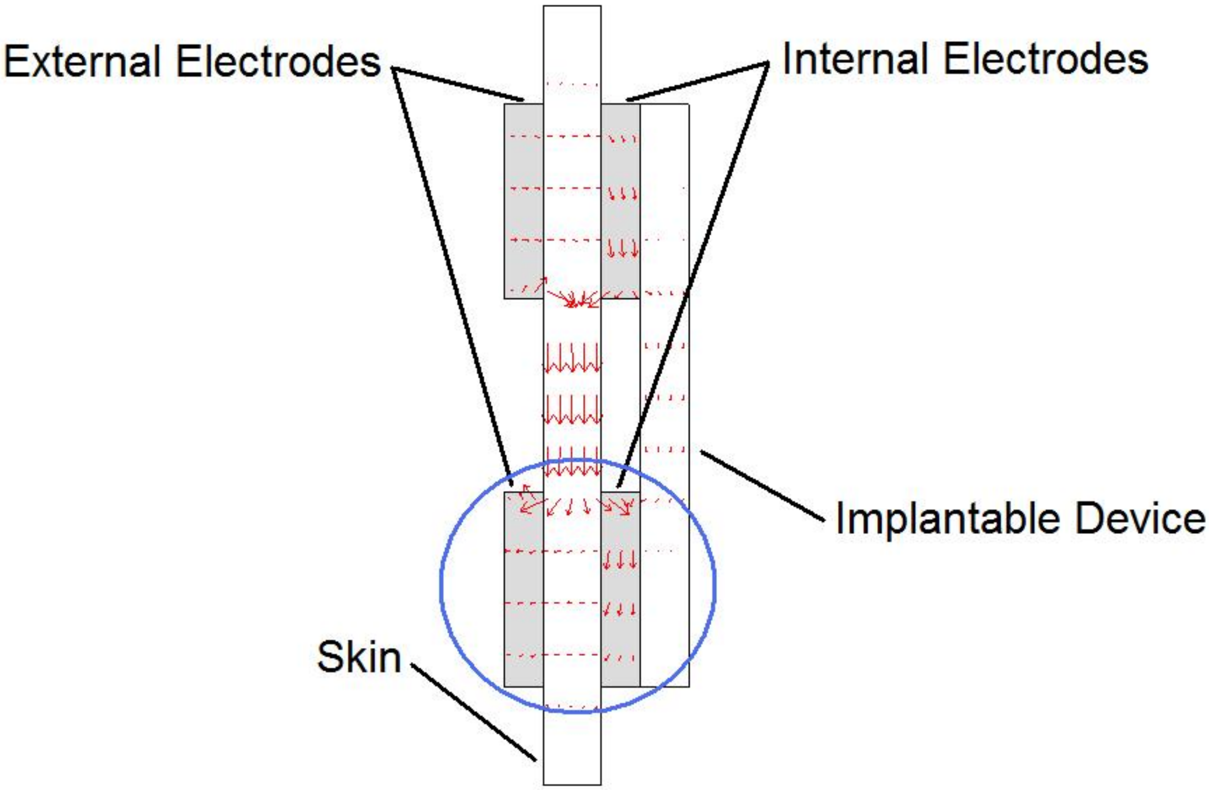


Figure 5.1: FEM model used to simulate the electrode-skin system. See text for the description of the physical setup. Arrows in the figure represent current density in the system induced by the applied voltages on the external electrodes. Circled in blue is the section corresponding to zoomed-in figure 5.2.

Examining figure 5.1 more closely reveals important electrical behavior not captured by the X-model described in section 2.5.2.1. Zooming in on the electrode connections (figure 5.2) reveals that not all current going from one external electrode to the other enters the skin and passes directly to the other electrode or through the implantable device. Some current passes through the skin to the opposing internal electrode in a distributed fashion, only to re-enter the skin before passing to the other internal electrode (as seen in the figure). The same behavior is mirrored at the other two opposing electrodes. The specific current distribution obviously depends upon the skin thickness and electrode width and separation, but the associated impedance pathways are not accurately captured by the previous X-model. Examining figure 5.2, the X-model's impedances are ambiguous as to what they represent. For this reason, the original X-model must be modified to give a complete view of the electrical behavior in the skin for the designer of an implantable device. These observations lead to the model layout detailed in the following section. This more accurate model allows analysis of current distributions at specific areas within the skin, giving designers the capability to modify system parameters and establish desired current densities within physical and safety constraints.

5.2 X-Δ IMPEDANCE MODEL

Given the need for an accurate impedance model for the electrode-skin system and considering the points discussed in the previous section, the model in the right panel of figure 5.3, called the X-Δ model, was created. Its derivation from the FEM simulation is illustrated in figure 5.4. The parameters Zt, Zd, Zs, and Zc correspond to "through," "deep," "shallow," and "cross" impedances, respectively. This new model is electrically equivalent to the X-Model if looked at as a "black box." However, as noted in the previous section, it more accurately represents the electrical behavior within the electrode-skin system, with the aim of giving a designer more complete information about current or voltage activity in the system as a whole.

Of note are certain assumptions or characteristics inherent or implied in the model:

- **Vertical Symmetry** – The choice to use idential impedances at each opposing set of external and internal electrodes was made mainly for simplicity in using the model. However, it is a reasonable assumption considering such connections are made within a small area of the skin that should not vary much in physical nature. As stated earlier, removal of the stratum corneum eliminates most of the skin's barrier properties, leaving a uniform volume conductor. The difference in Zs and Zd accounts for the different electrode interfaces on either side of the skin.

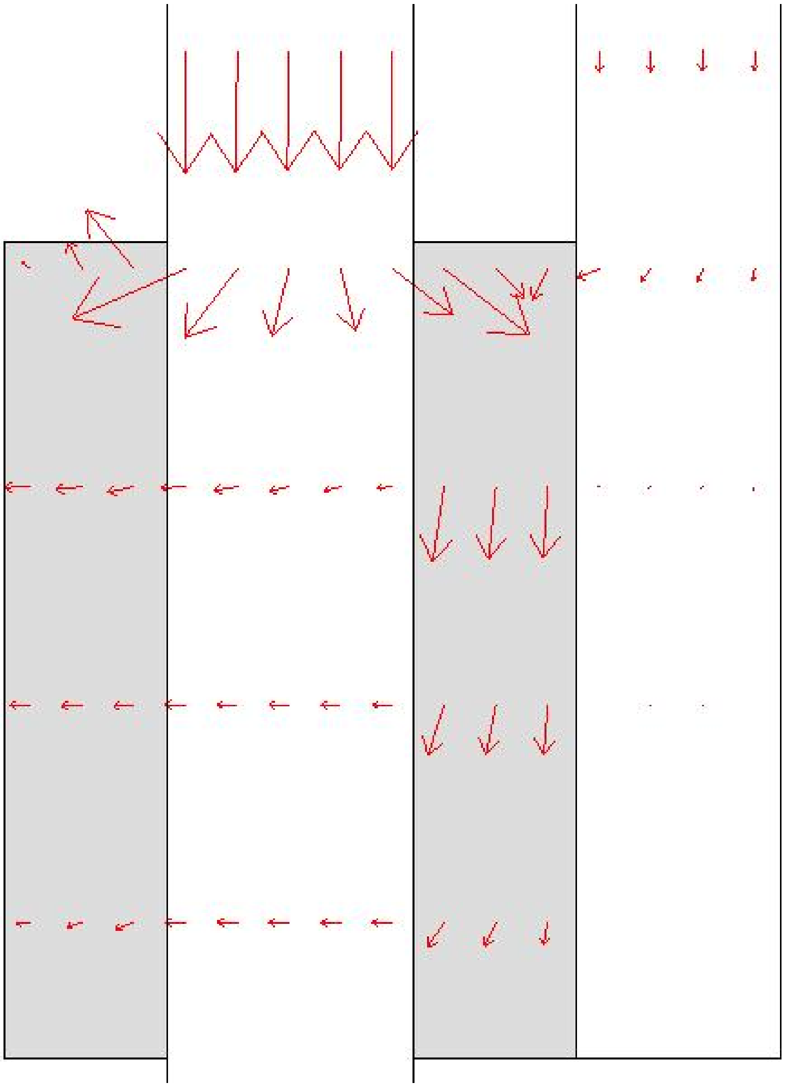


Figure 5.2: Zoomed-in portion of the FEM model used to simulate the electrode-skin system. Arrows in the figure represent current density in the system induced by the applied voltages on the external electrodes. Here it is shown that current passing between external and internal electrodes is actually larger than the current entering or exiting the implantable device, which is unpredicted by the X-model discussed in 2.5.2.1.

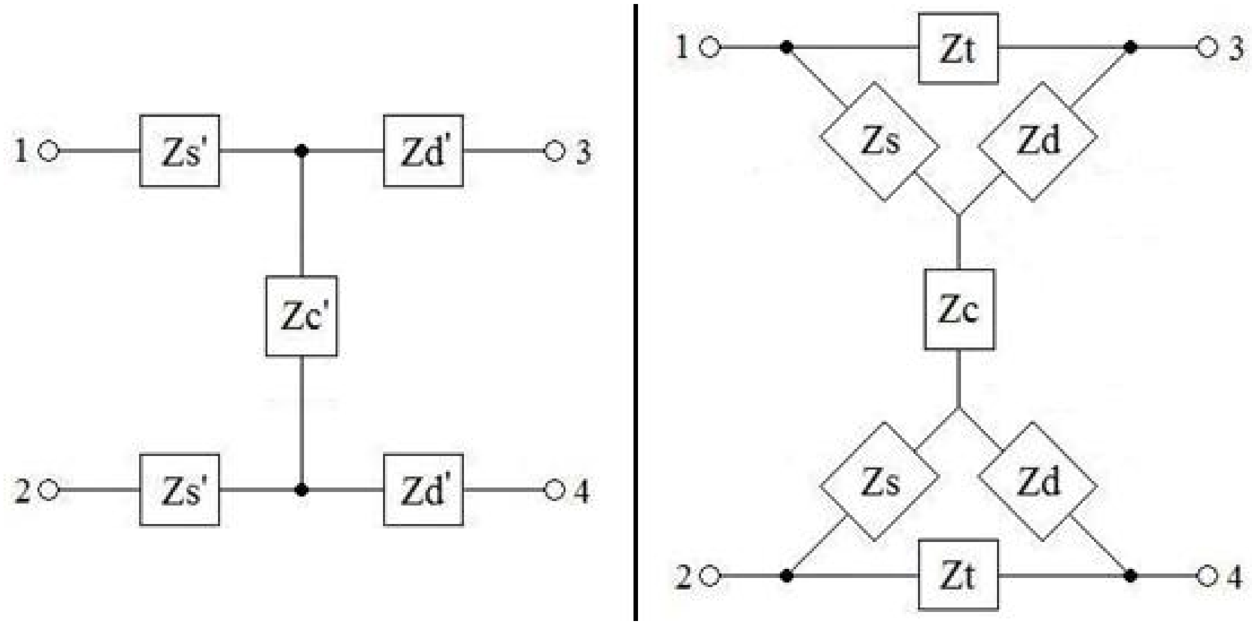


Figure 5.3: Left: Previous X-Model for comparison and future reference in parameter verification. Right: Newly created X-Δ model of the volume conduction electrode-skin system.

- **Linearity** – According to [10], a linear model involving resistors and dielectric-based capacitors is generally accepted as an appropriate model for the skin. There is still the issue of non-linear characteristics arising as applied current density increases. However, studies on electrode-electrolyte interfaces have shown that the current density limit of linearity increases as frequency increases [78, 57]. For example, model components retained linearity up to 10 mA/cm^2 at 100 Hz and up to 1000 mA/cm^2 at 10 KHz. Since the region of operation for this volume conduction system is generally above 100 KHz and the maximum current density measured during use is usually less than 5 mA/cm^2, it is likely that the system can be considered linear for this type of application. It should be explicitly stated that an assumption of linearity is only made over a small region around any specific frequency, and not with respect to frequency. This assumption should not hinder practical applications for the model, considering many communication protocols such as frequency or phase modulation require only small shifts in frequency. Addition-

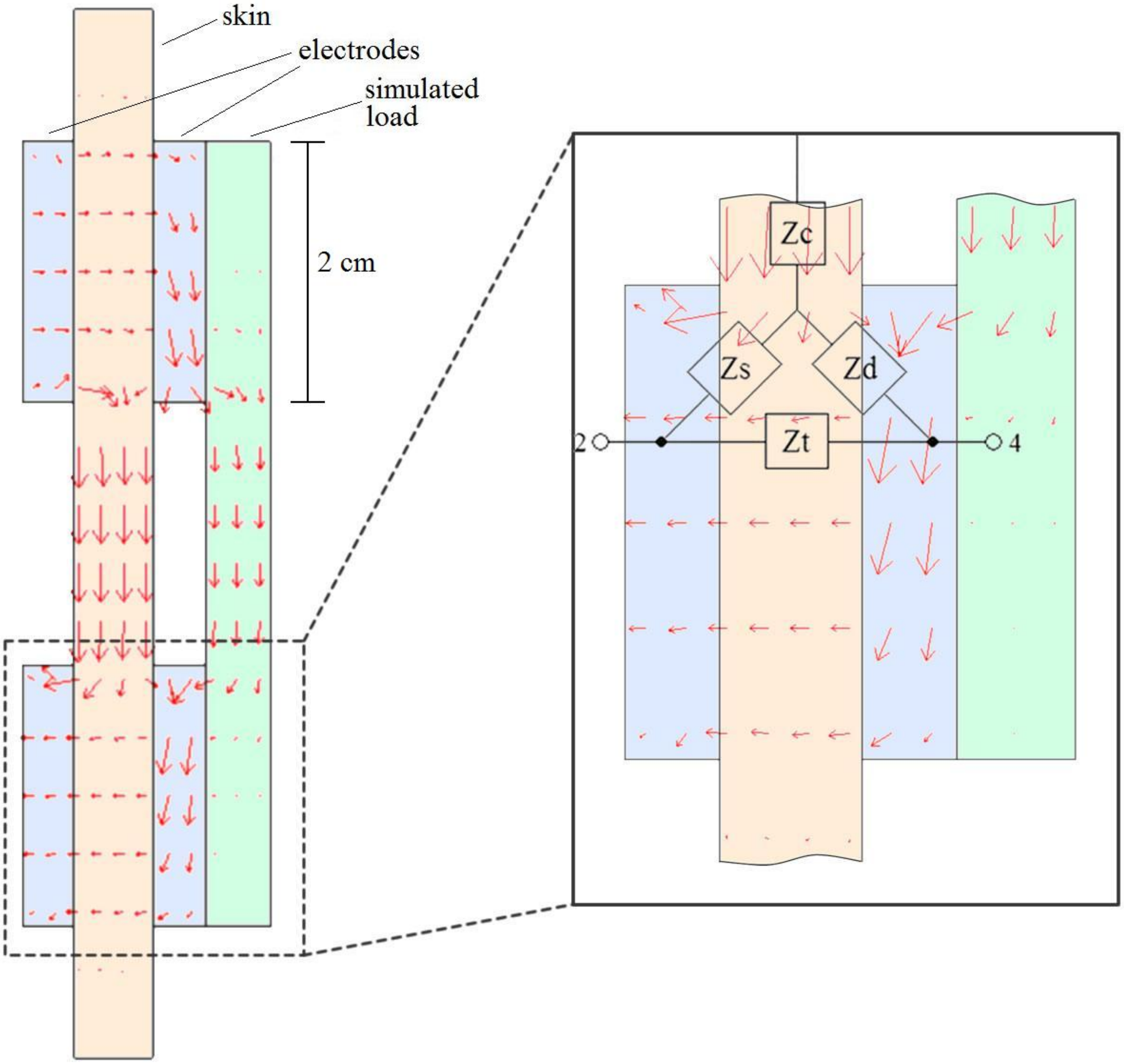


Figure 5.4: X-Δ model as derived from the FEM system simulation.

ally, no specific non-linear characteristics, such as those in [24], are attributed to the impedances.

- **Frequency Dependence** – Currently the model is not comprehensive so as to include frequency-dependent behavior in each parameter. That is, the model's component values will change between one frequency and another. Currently, the parameters' functional dependence on frequency is not established; instead the parameters are recalculated from impedance measurements at each frequency. It will be shown that the model's integrity can be verified across the range of frequencies measured, but with different model parameters at each individual frequency. Future work may address this issue by exploring the frequency dependence of each parameter.

5.3 MODEL VERIFICATION

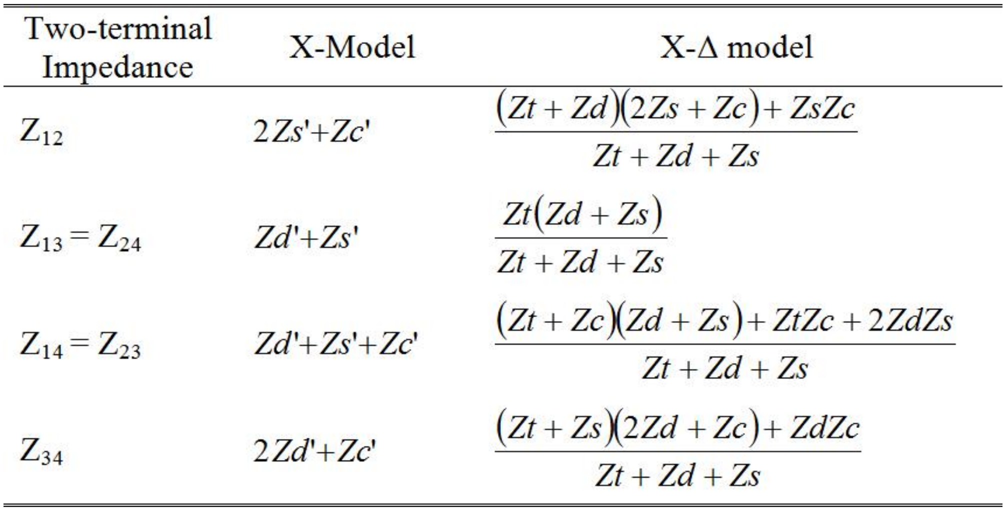
Verification of the X-Δ model involves two processes: 1) finding parameter (impedance) values for the model, and 2) confirming that the parameters correctly represent the actual system. These two processes are detailed in the sections below.

5.3.1 Calculation of Model Parameters

Given the model structures in section 5.2 and the measurements from section 4.3, the parameters of the X- and X-Δ models can be calculated. Examining figure 5.3, expressions can be found for equivalent two-terminal impedances of the models (table 5.1) corresponding to the two-terminal impedance measurements taken in section 4.3. Z_{ij} corresponds to the impedance between terminals i and j as indicated in figure 4.8.

Examining table 5.1, it can be seen that for the X-Model, a set of linear equations is easily solvable using three independent measurements. However, solving the more complicated two-terminal impedance equations associated with the X-Δ model proves more difficult than solving a set of linear equations. Because a concise analytical solution is mathematically intractable, a method of fitting the model parameters to the measured impedances via an

Table 5.1: Equivalent two-terminal impedances for the X-Model and X-Δ Model.



Two-terminal Impedance	X-Model	X-Δ model
Z_{12}	$2Zs'+Zc'$	$\frac{(Zt+Zd)(2Zs+Zc)+ZsZc}{Zt+Zd+Zs}$
$Z_{13}=Z_{24}$	$Zd'+Zs'$	$\frac{Zt(Zd+Zs)}{Zt+Zd+Zs}$
$Z_{14}=Z_{23}$	$Zd'+Zs'+Zc'$	$\frac{(Zt+Zc)(Zd+Zs)+ZtZc+2ZdZs}{Zt+Zd+Zs}$
Z_{34}	$2Zd'+Zc'$	$\frac{(Zt+Zs)(2Zd+Zc)+ZdZc}{Zt+Zd+Zs}$

optimization routine is used. It involves searching the solution space of an error function to find parameters which minimize the error. Specifically, a conjugate gradient method was utilized, with a section search method used to determine step-size at each iteration. MATLAB code used to realize the conjugate gradient algorithm and perform various calculations is given in appendices G-N.

An error function is defined as the squared magnitude of the difference between the measured terminal impedance and the equivalent terminal impedance of the model, as in 5.1. The partial derivatives of this function with respect to each variable, the real and imaginary parts of the impedance parameters, are used in the optimization algorithm to traverse the function space to its minimum. With four model parameters to optimize (Zt, Zd, Zs, Zc), each one having real and imaginary parts, the optimization problem contains eight free variables. The error function for each impedance value is expressed as

$$E_{ij} = \left|Z_{ij} - M_{ij}\right|^2 = \mathrm{Re}\{Z_{ij} - M_{ij}\}^2 + \mathrm{Im}\{Z_{ij} - M_{ij}\}^2 \quad ; \quad ij = \{12,13,14,23,24,34\}, \tag{5.1}$$

where $ij = \{12, 13, 14, 23, 24, 34\}$ and M_{ij} corresponds to the equivalent two-terminal impedance of the X-Δ model given in table 5.1. The derivative of the above function is

$$\frac{\partial E_{ij}}{\partial v_k} = 2\left(\mathrm{Re}\{Z_{ij} - M_{ij}\} \frac{\partial \mathrm{Re}\{M_{ij}\}}{\partial v_k} + \mathrm{Im}\{Z_{ij} - M_{ij}\} \frac{\partial \mathrm{Im}\{M_{ij}\}}{\partial v_k} \right), \tag{5.2}$$

where v_k is one of the eight free variables, the real and imaginary parts of each impedance parameter. Because the parameter fitting must optimize all impedance expressions at once, the problem becomes a multi-objective optimization problem. For simplicity, the objective function is expressed as a weighted sum of the individual error functions, 5.1. Because each of these error functions is convex, the objective function is thus also convex, implying one optimal solution exists. Additionally, to reduce computation time and eliminate redundant data, only four of the six possible error functions are used. Again, this assumes that the impedance measuring setup in 4.3 was configured properly and measurement noise is low, so that measurements $Z_{13} \approx Z_{24}$ and $Z_{14} \approx Z_{23}$. The resulting objective function and its derivatives are

$$F = w_1 E_{12} + w_2 E_{13} + w_3 E_{14} + w_4 E_{34}, \tag{5.3}$$

$$\frac{\partial F}{\partial v_k} = w_1 \frac{\partial E_{12}}{\partial v_k} + w_2 \frac{\partial E_{13}}{\partial v_k} + w_3 \frac{\partial E_{14}}{\partial v_k} + w_4 \frac{\partial E_{34}}{\partial v_k}, \tag{5.4}$$

where w_n is the weight associated with its respective error function. Each of the derivatives in 5.4 are rather long and complex, and so a program called wxMaxima was used to compute them. The wxMaxima computations can be found in appendices C-F.

Given the objective function 5.3, parameters for the X-Δ model were computed using the conjugate gradient optimization method. Figure 5.5 shows the magnitude and phase of these impedances across the range of measured frequencies. The plots for each parameter contain results for all nine pieces of skin detailed in section 4.3.

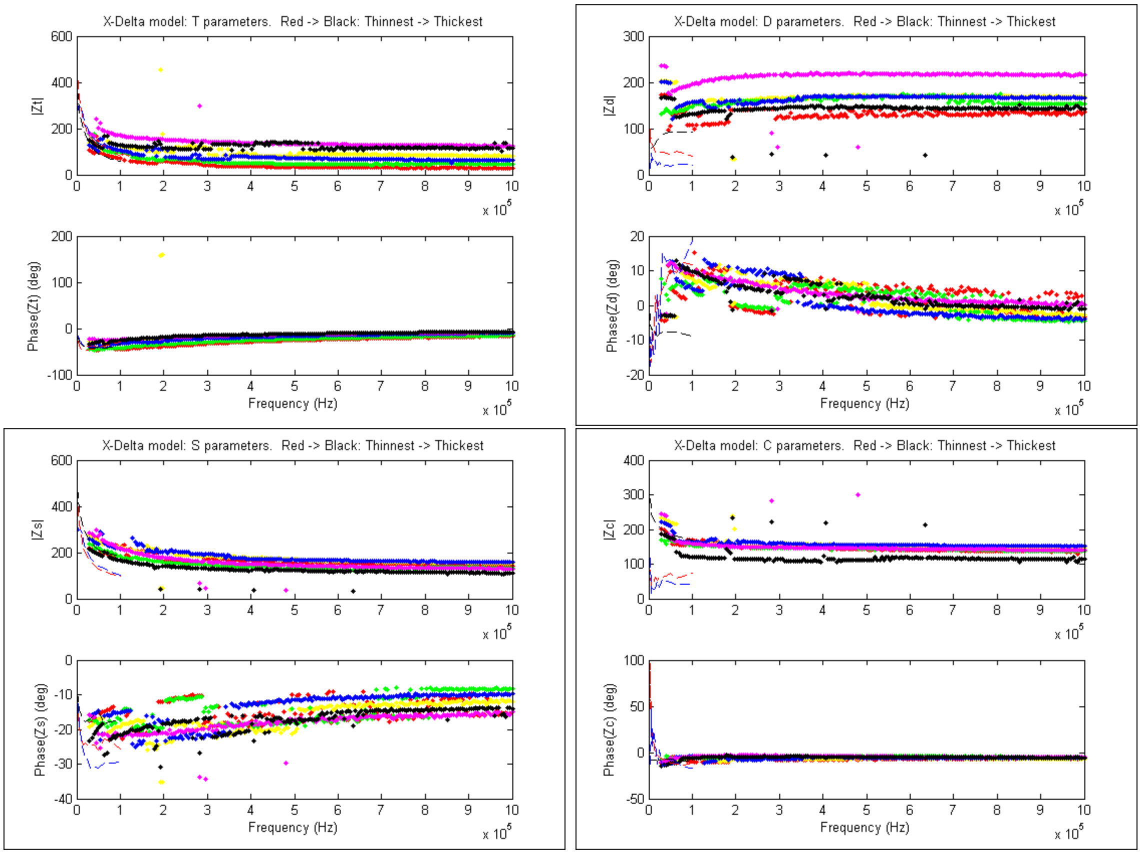
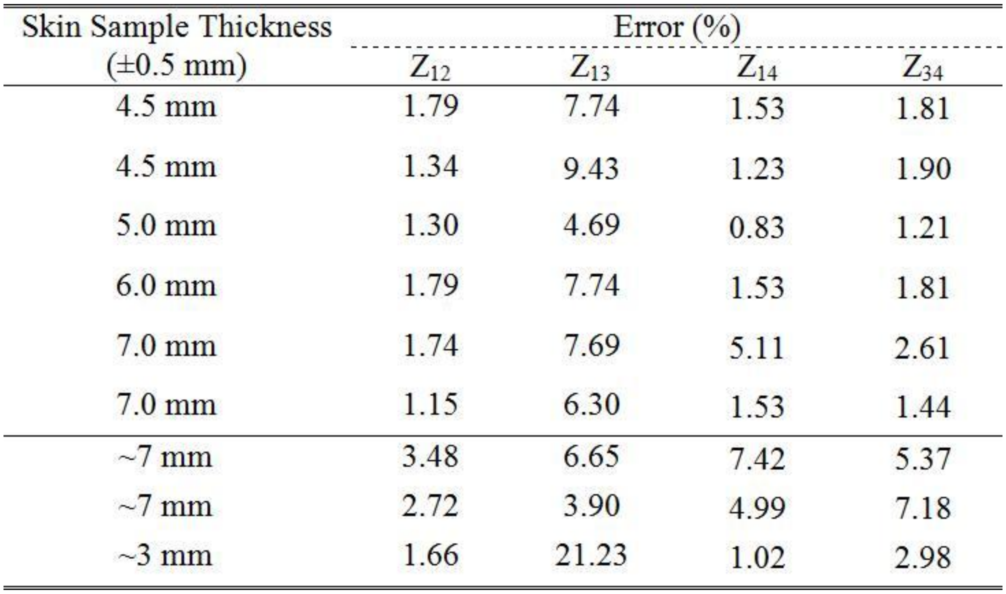


Figure 5.5: X-Δ model parameters calculated via optimization method using impedance measurements. The magnitude and phase of each of Zt, Zd, Zs, and Zc are shown as functions of frequency for all nine pieces of skin tested. Skin #1: 6 mm - blue. Skin #2: 5 mm - green. Skin #3: 4-5 mm - red. Skin #4: 7 mm - magenta. Skin #5: 4-5 mm - yellow. Skin #6: 7 mm - black. Skin #7: 7+ mm - dashed red. Skin #8: 7+ mm - dashed blue. Skin #9: 3+ mm - dashed black.

5.3.2 Equivalent Impedance Error Analysis

The impedance parameters for the X-Δ model being calculated, its equivalent two-terminal impedances are compared to the measured two-terminal impedances. Table 5.2 shows the average error across frequencies for each unique pair of terminals and each piece of skin. Error is consistently below 10% except for Z_{13} calculated for the thin ear skin. Median errors across frequencies are even lower, hinting that some outliers may skew the averaged data. Error is generally higher, though still low, from the pieces of skin measured with the Gamry machine. It is unknown whether the increased error is due to the different experimental setup using the Gamry machine or if the model may not represent the electrode-skin system as well at the lower frequencies. For skin #3 specifically, the large 21.23% error for component Z_{13} is likely a result of the greater relative contribution of the saline and conductive paste, which are not explicitly included in the model, to the skin system's impedance, due to the small thickness of the skin. Additionally, averaging the errors may mask error trends across frequencies, which could show if the model represents the system better or worse at certain frequencies. Future studies should examine this phenomenon. Regardless, such generally small errors demonstrate that the X-Δ model accurately represents the impedance behavior of the system across the measured frequency range.

Table 5.2: Average error across frequencies between calculated X-Δ model two-terminal impedances and measured impedances. The first six entries (rows) use measurements results from the VNA (30-1000 KHz). The last three entries use results from the Gamry machine (1-100 KHz).



Skin Sample Thickness	Error (%)			
(±0.5 mm)	Z_{12}	Z_{13}	Z_{14}	Z_{34}
4.5 mm	1.79	7.74	1.53	1.81
4.5 mm	1.34	9.43	1.23	1.90
5.0 mm	1.30	4.69	0.83	1.21
6.0 mm	1.79	7.74	1.53	1.81
7.0 mm	1.74	7.69	5.11	2.61
7.0 mm	1.15	6.30	1.53	1.44
~7 mm	3.48	6.65	7.42	5.37
~7 mm	2.72	3.90	4.99	7.18
~3 mm	1.66	21.23	1.02	2.98

5.4 PRACTICAL USE OF THE MODEL IN ENGINEERING APPLICATIONS

Here, the practical use of the X-Δ model is explored for engineering implantable system characteristics. It is developed to be used as a tool for analyzing volume conduction implantable device systems and helping to determine circuit design parameters as well as power requirements and system efficiencies. Shown in figure 5.6 is a circuit model of the entire implantable system with a connected external energy pad or similar. The X-Δ model represents the impedances of the skin-electrode component, V and Zi represent an external circuit with energy source for putting energy into the system, and Zl represents the equivalent impedance of the implantable device.

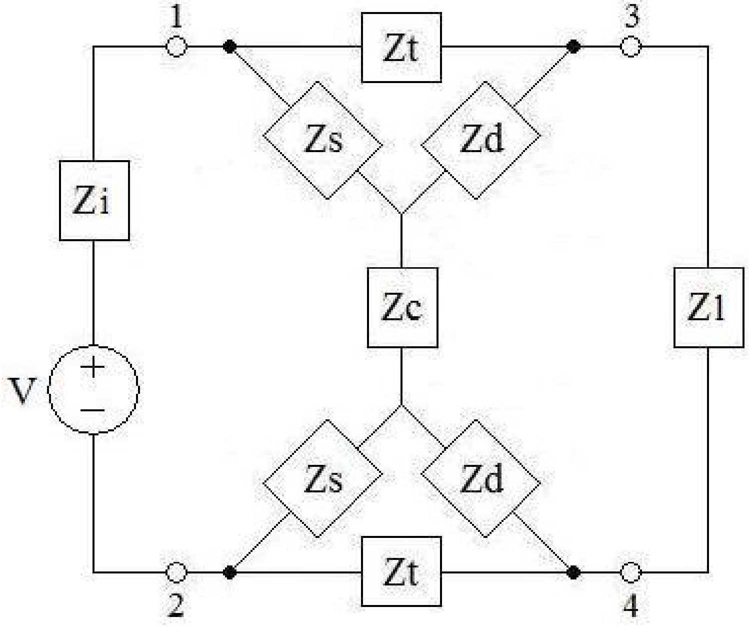


Figure 5.6: Circuit model of an implantable system incorporating the X-Δ model with source (external) and load (internal) impedances. A system designer can use this model to optimize desired energy transfer parameters.

Given any particular type of skin and electrode connections, the X-Δ model parameters (Zt, Zd, Zs, Zc) will be fixed. The designer of an implantable system only has the flexibility to modify Zi, Zl, and V to meet his design constraints. A common approach to such a problem is to aim for maximum power transfer by matching the input and output impedances (complex conjugates) at each set of terminals. However, this approach has certain drawbacks: 1) maximum power transfer does not imply maximum efficiency; 2) other electrical characteristics such as voltage and current are more important than power in this particular case; 3) analytically solving the matching problem is very difficult. These points considered, focus is shifted to determining the current and voltage transfer functions (or transfer efficiencies). These two functions complement one another in the design process. The current transfer function is important in determining just how much of an external battery's energy goes into the implantable device, which affects the choice of both batteries, i.e one for the device and another for the recharging circuitry. Also, most implantable devices need a certain threshold voltage to even begin charging, and the voltage transfer efficiency can be used to determine the minimum external voltage needed. While the X-Δ model parameters will establish theoretical limits on these efficiencies, the designer will be able to see the effect of the source and load impedances on the transfer functions and choose components accordingly. Both the theoretical limits and impedance effects on current and voltage transfer functions will be examined. The X-Δ model parameters established from previous experiments and optimization will be used.

Voltage and current transfer functions are calculated using mesh current analysis on the system in figure 5.6. The currents are shown in figure 5.7, with associated mesh current equations in 5.5. The current transfer function is analyzed first because it is slightly simpler than the voltage transfer function.

$$\begin{aligned}
I1*(2Zs+Zc) &= V+I2*(2Zs)+I3*Zc\\
I2*(Zs+Zt+Zd) &= I1*Zs+I3*Zd\\
I3*(Zl+2Zd+Zc) &= I1*Zc+I2*(2Zd)
\end{aligned} \tag{5.5}$$

5.4.1 Current Transfer Function

Expressions for the individual currents are easily found from equation 5.5, but they are relatively long. These expressions are useful elsewhere in the implantable system design process when examining current flow through specific parts of the skin and modifying system

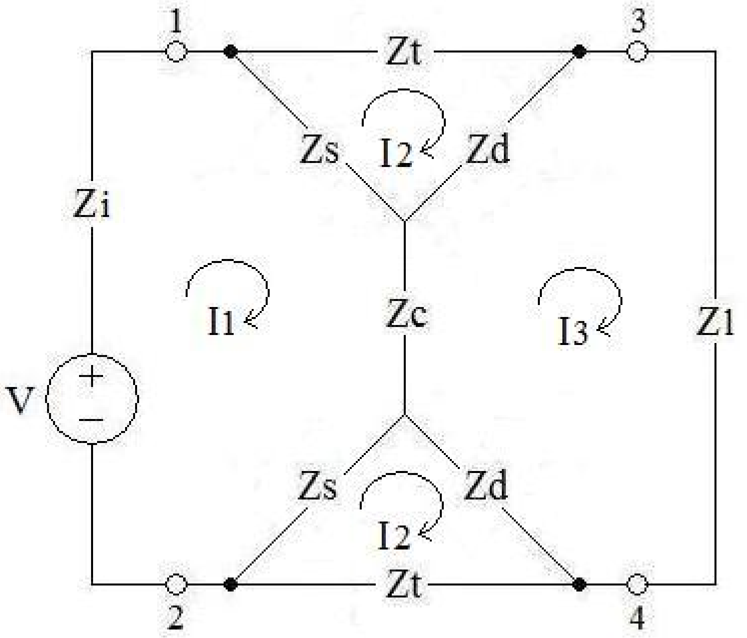


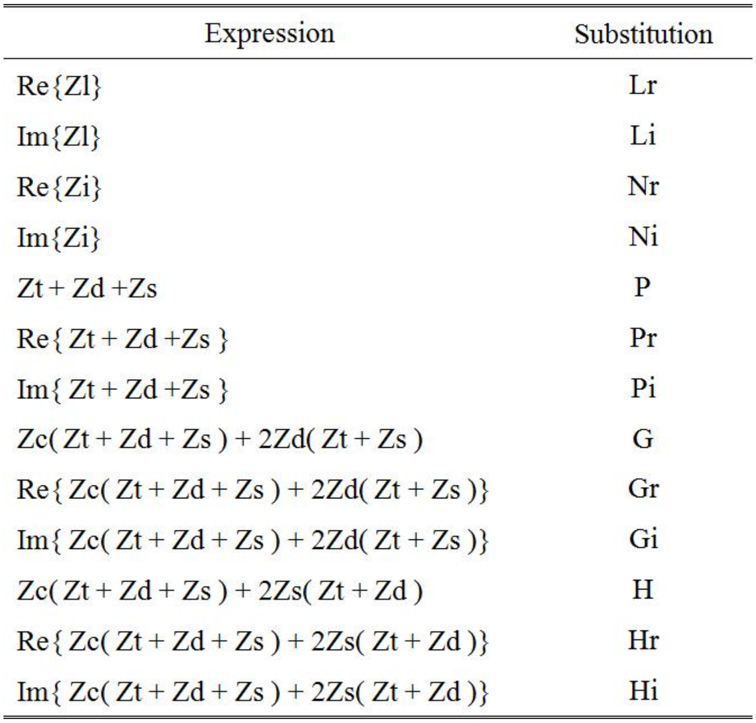
Figure 5.7: Mesh current definitions for use in deriving voltage and current transfer functions. Due to the symmetric nature of the model, I_2 is the same in both the top and bottom Δ components.

parameters to keep these currents within established current exposure limits. However, here the focus is solely on the current transfer efficiency, namely $\frac{I_3}{I_1}$, or the ratio of load current to input current. The expression for current transfer efficiency, from the individual current expressions for I_3 and I_1, is

$$\frac{I3}{I1}=\frac{Zc(Zt+Zd+Zs)+2ZdZs}{(Zc+Zl)(Zt+Zd+Zs)+2Zd(Zt+Zs)}. \tag{5.6}$$

Note that Zi and V are not factors in equation 5.6. This means that the load impedance Zl is the only variable which affects the current transfer efficiency once the X-Δ model parameters are set. First, the theoretical maximum efficiency is examined. To maximize $\frac{I_3}{I_1}$ as a function of Zl, the denominator of 5.6 needs to be minimized, or equivalently, the squared magnitude of the denominator. Some substitutions are made to simplify the mathematical expressions, as detailed in table 5.3.

Table 5.3: Expression substitutions for optimization of current and voltage transfer functions.



Expression	Substitution
Re{Zl}	Lr
Im{Zl}	Li
Re{Zi}	Nr
Im{Zi}	Ni
Zt + Zd +Zs	P
Re{ Zt + Zd +Zs }	Pr
Im{ Zt + Zd +Zs }	Pi
Zc(Zt + Zd + Zs) + 2Zd(Zt + Zs)	G
Re{ Zc(Zt + Zd + Zs) + 2Zd(Zt + Zs)}	Gr
Im{ Zc(Zt + Zd + Zs) + 2Zd(Zt + Zs)}	Gi
Zc(Zt + Zd + Zs) + 2Zs(Zt + Zd)	H
Re{ Zc(Zt + Zd + Zs) + 2Zs(Zt + Zd)}	Hr
Im{ Zc(Zt + Zd + Zs) + 2Zs(Zt + Zd)}	Hi

Thus, the denominator equals ($Zl{*}P{+}G$). Minimizing the magnitude of this denominator is equivalent to minimizing

$$Wc = |Zl * P + G|^2, \tag{5.7}$$

which is easier for computation. The complex nature of the impedances (real and imaginary parts) makes the problem slightly more complicated, and so the complex number is separated into its real and imaginary parts:

$$\begin{aligned} \mathrm{Re}\{Zl * P + G\} &= LrPr - LiPi + Gr \\ \mathrm{Im}\{Zl * P + G\} &= LrPi + LiPr + Gi \end{aligned} . \tag{5.8}$$

Because there are two variables to optimize, a standard multi-variable optimization is performed, requiring first derivatives and the Hessian matrix. Noting that for a complex number C, $\frac{d}{dr}|C|^2 = 2\mathtt{Re}\{C\}\frac{d}{dr}\mathtt{Re}\{C\} + 2\mathtt{Im}\{C\}\frac{d}{dr}\mathtt{Im}\{C\}$, these first and second derivatives are:

$$\begin{aligned}\frac{\partial Wc}{\partial Lr} &= 2\,\mathrm{Re}\{Lr*P+G\}\frac{\partial}{\partial Lr}\mathrm{Re}\{Lr*P+G\}+2\,\mathrm{Im}\{Lr*P+G\}\frac{\partial}{\partial Lr}\mathrm{Im}\{Lr*P+G\}\\ &= 2\left[Lr\left(Pr^2+Pi^2\right)+PrGr+PiGi\right]\end{aligned} \tag{5.9}$$

$$\begin{aligned}\frac{\partial Wc}{\partial Li} &= 2\,\mathrm{Re}\{Lr*P+G\}\frac{\partial}{\partial Li}\mathrm{Re}\{Lr*P+G\}+2\,\mathrm{Im}\{Lr*P+G\}\frac{\partial}{\partial Li}\mathrm{Im}\{Lr*P+G\}\\ &= 2\left[Li\left(Pr^2+Pi^2\right)+PrGi+PiGr\right]\end{aligned} \tag{5.10}$$

$$\begin{bmatrix}\dfrac{\partial^2}{\partial Lr^2} & \dfrac{\partial^2}{\partial LrLi}\\ \dfrac{\partial^2}{\partial LiLr} & \dfrac{\partial^2}{\partial Li^2}\end{bmatrix} Wc = \begin{bmatrix}2\left(Pr^2+Pi^2\right) & 0\\ 0 & 2\left(Pr^2+Pi^2\right)\end{bmatrix} \tag{5.11}$$

Seeing that $\frac{\partial^2 Wc}{\partial Lr^2} \times \frac{\partial^2 Wc}{\partial Li^2} = 4(Pr^2+Pi^2)^2 > 0$, an extremum exists, and because $\frac{\partial^2 Wc}{\partial Lr^2} > 0$, that extremum is a local minimum. Solving for the minimum by setting the first derivatives equal to zero gives

$$Lr = -\frac{PrGr + PiGi}{(Pr^2 + Pi^2)} \tag{5.12}$$

$$Li = \frac{PiGr - PrGi}{(Pr^2 + Pi^2)} \tag{5.13}$$

Now, given the form of the data in figure 5.5 and looking more closely at the expression in equation 5.12, it can be seen that the optimal value for the real part of the load impedance Zl is negative. This is the case because no constraints were placed on the optimization. Knowing that the real part of Zl can not realistically be negative, and that the objective function for optimization is convex (a squared magnitude function), the closest value of Lr to the objective minimum must be zero. As such, the value of Zl which minimizes the denominator of 5.6 and thus maximizes the current transfer efficiency, $\frac{I_3}{I_1}$, is

$$Zl = jLi = j\frac{PiGr - PrGi}{(Pr^2 + Pi^2)}. \tag{5.14}$$

Using this value in equation 5.6 with the values for the X-Δ model impedance parameters already calculated for each piece of skin, maximum current transfer efficiences are plotted in figure 5.8 for the range of frequencies measured. The values range from 65% to just above 90% for different pieces of skin, and the efficiency for any one piece of skin tends to remain the same, with only little upward or downward tendency as frequency increases. Ideally, the trend would be similar for all pieces. The reason for the difference could be any number of factors, from slight differences in experimental setups to inherent differences in the pieces of skin tested. Future studies may examine causes of any particular upward or downward trend in efficiency.

Beyond simply the theoretical maximum current transfer efficiency, designers may want to explore the effects of varying impedances on the efficiency. Figure 5.9 shows the effects of varying the load impedance over a range of values for skin #2, with a fixed frequency of 124 KHz. The maximum efficiency, just over 80%, occurs at approximately $j30\Omega$, corresponding to an inductance of about $38.5\mu H$. Obviously an input impedance of such value would be difficult (or impossible!) to obtain in an implantable device. With the impedance of the implantable device measured in section 4.3, the maximum current transfer efficiency is approximately 45%. An engineer designing an implantable device circuit could make efforts to modify the input impedance of the device to achieve a desirable current transfer ratio within the limit, or know just how much current would be necessary to inject from the source side for a given received current. He could also examine the individual currents within the skin, corresponding to the different model components, to determine if such levels fall within established safety standards [3].

5.4.2 Voltage Transfer Function

Deriving the voltage transfer function follows the same procedure as with the current transfer function, but the expression is more complicated, with more variable parameters. Expressions for the individual voltages are easily found from equation 5.5, but, again, they are relatively long. These expressions are useful elsewhere in the implantable system design process when examining electric field exposure limits of the skin and modifying system pa-

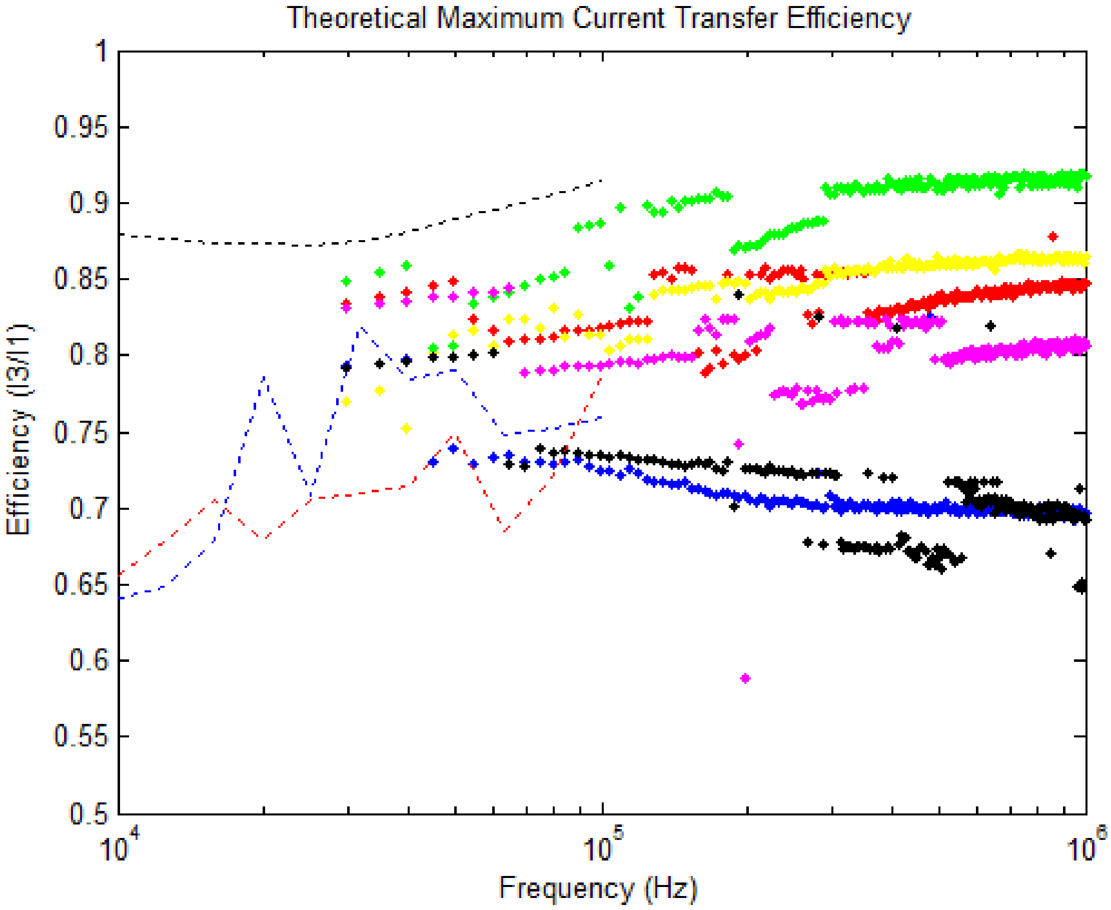


Figure 5.8: Theoretical maximum current transfer efficiencies are shown as functions of frequency for all nine pieces of skin tested. Skin #1: 6 mm - blue. Skin #2: 5 mm - green. Skin #3: 4-5 mm - red. Skin #4: 7 mm - magenta. Skin #5: 4-5 mm - yellow. Skin #6: 7 mm - black. Skin #7: 7+ mm - dashed red. Skin #8: 7+ mm - dashed blue. Skin #9: 3+ mm - dashed black.

rameters to keep these fields within safe ranges. Here the focus is solely on the voltage transfer efficiency, namely $\frac{Vl}{V}$, or the ratio of load voltage (Vl) to input voltage. The expression for voltage transfer efficiency, from the individual voltage expressions for Vl and V, is

$$\frac{Vl}{V}=\frac{Zl[Zc(Zt+Zd+Zs)+2ZdZs]}{Zl[(Zc+Zi)(Zt+Zd+Zs)+2Zs(Zt+Zd)]+Zi[Zc(Zt+Zd+Zs)+2Zd(Zt+Zs)]+2Zt[Zc(Zd+Zs)+2ZdZs]}. \quad (5.15)$$

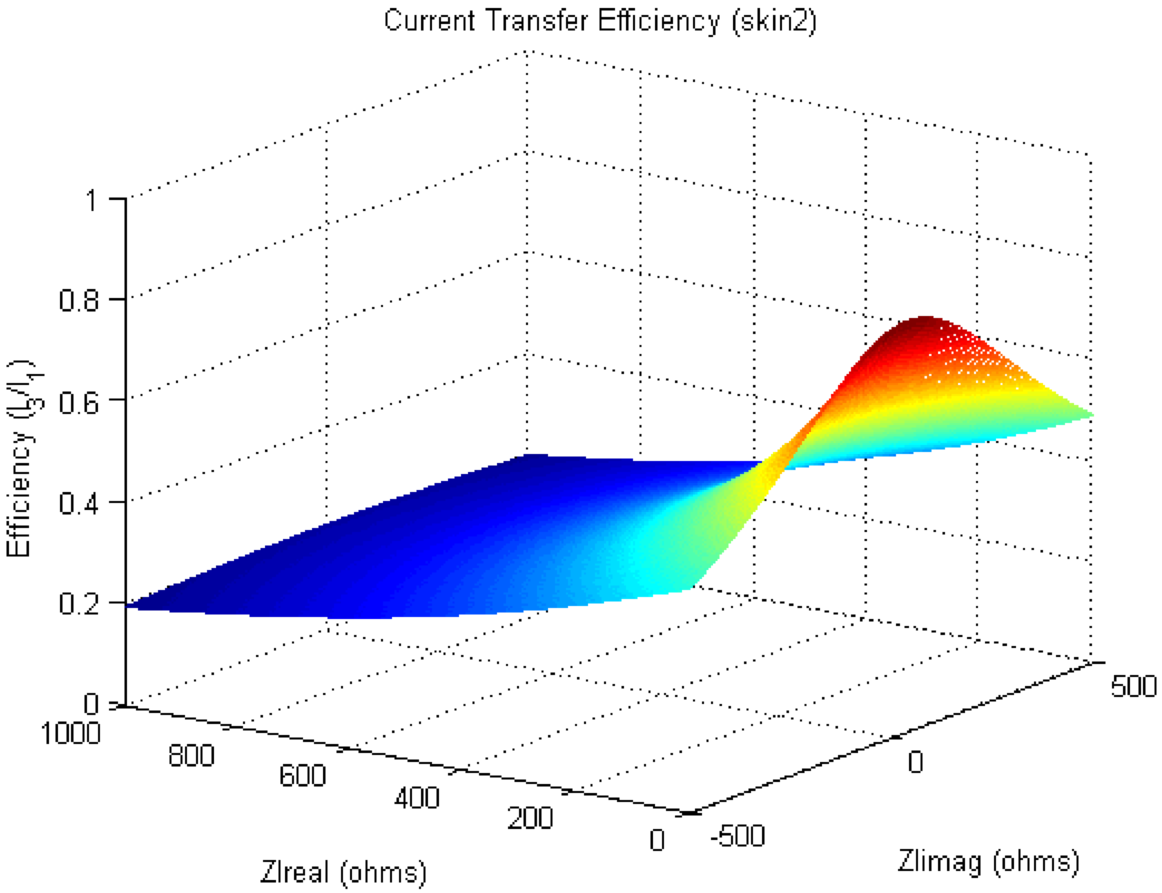


Figure 5.9: Current transfer efficiency for skin #2 as a function of load impedance.

For this expression, both Zi and Zl are variable parameters. To simplify the expression, it is first assumed that Zi is a constant and the optimization to find the maximum voltage transfer ratio is performed with respect to Zl. The derivative of 5.15 is calculated as shown in 5.16. Setting this equal to zero to find the optimum point implies that $Zl \to \infty$. The second and higher order derivatives at such point are all infinity, but given the form of the equation and knowing that the minimum must be at $Zl = 0$, reason that this point must be the maximum. Whether the real or imaginary part of Zl approaches infinity is irrelevant, as the result would be the same.

$$\frac{\partial}{\partial Zl}\left(\frac{Vl}{V}\right)=\frac{ZcP+2ZdZs}{\left(Zl\left(ZiP+H\right)+ZiG+2Zt\left[Zc\left(Zd+Zs\right)+2ZdZs\right]\right)^{2}}\,. \tag{5.16}$$

Putting $Zl \to \infty$ back into equation 5.15 gives a new objective function to optimize, where Zi is the only variable parameter. This equation, 5.17, takes the same form as that in the current transfer efficiency optimization (5.6), except Zi is now the variable to optimize instead of Zl.

$$\frac{Vl}{V}=\frac{Zc\left(Zt+Zd+Zs\right)+2ZdZs}{\left(Zc+Zi\right)\left(Zt+Zd+Zs\right)+2Zs\left(Zt+Zd\right)}=\frac{ZcP+2ZdZs}{ZiP+H}\,. \tag{5.17}$$

Following the same optimization procedure as for the current transfer efficiency, to maximize $\frac{Vl}{V}$ as a function of Zi, the denominator of 5.17 needs to be minimized. Equivalently, the squared magnitude of the denominator can be minimized, easing computation:

$$Wv = |Zi * P + H|^2. \tag{5.18}$$

The complex part of the denominator is separated into its real and imaginary parts:

$$\begin{aligned}\operatorname{Re}\{Zi * P + H\} &= NrPr - NiPi + Hr\\ \operatorname{Im}\{Zi * P + H\} &= NrPi + NiPr + Hi\,.\end{aligned} \tag{5.19}$$

Again, the standard multi-variable optimization is performed, requiring first derivatives and the Hessian matrix. These first and second derivatives are:

$$\begin{aligned}\frac{\partial Wv}{\partial Nr} &= 2\operatorname{Re}\{Nr * P + H\}\frac{\partial}{\partial Nr}\operatorname{Re}\{Nr * P + H\} + 2\operatorname{Im}\{Nr * P + H\}\frac{\partial}{\partial Nr}\operatorname{Im}\{Nr * P + H\}\\ &= 2\left[Nr\left(Pr^2 + Pi^2\right) + PrHr + PiHi\right]\end{aligned} \tag{5.20}$$

$$\begin{aligned}\frac{\partial Wv}{\partial Ni} &= 2\operatorname{Re}\{Nr * P + H\}\frac{\partial}{\partial Ni}\operatorname{Re}\{Nr * P + H\} + 2\operatorname{Im}\{Nr * P + H\}\frac{\partial}{\partial Ni}\operatorname{Im}\{Nr * P + H\}\\ &= 2\left[Ni\left(Pr^2 + Pi^2\right) + PrHi + PiHr\right]\end{aligned} \tag{5.21}$$

$$\begin{bmatrix} \dfrac{\partial^2}{\partial Nr^2} & \dfrac{\partial^2}{\partial NrNi} \\ \dfrac{\partial^2}{\partial NiNr} & \dfrac{\partial^2}{\partial Ni^2} \end{bmatrix} Wv = \begin{bmatrix} 2\left(Pr^2 + Pi^2\right) & 0 \\ 0 & 2\left(Pr^2 + Pi^2\right) \end{bmatrix} \tag{5.22}$$

Seeing that $\frac{\partial^2 Wv}{\partial Nr^2} \times \frac{\partial^2 Wv}{\partial Ni^2} = 4(Pr^2 + Pi^2)^2 > 0$, an extremum exists, and because $\frac{\partial^2 Wv}{\partial Nr^2} > 0$, that extremum is a local minimum. Solving for the minimum by setting the first derivatives equal to zero gives

$$Nr = -\frac{PrHr + PiHi}{(Pr^2 + Pi^2)} \tag{5.23}$$

$$Ni = \frac{PiHr - PrHi}{(Pr^2 + Pi^2)} \tag{5.24}$$

Following the same reasoning as with 5.12, the real part of Zi, Nr, must be zero. As such, the value of Zi which minimizes the denominator of 5.15 and thus maximizes the voltage transfer efficiency, $\frac{Vl}{V}$, is

$$Zi = jNi = j\frac{PiHr - PrHi}{(Pr^2 + Pi^2)}. \tag{5.25}$$

Using this value in equation 5.15 with the values for the X-Δ model impedance parameters already calculated for each piece of skin, maximum voltage transfer efficiencies are plotted in figure 5.10 for the range of frequencies measured. The values range from 20% to just over 90% for different pieces of skin, and the efficiency for any one piece of skin tends to remain relatively constant with a slight upward trend as frequency increases, except for those measured at lower frequencies, whose efficiencies increase more drastically. Unlike the current transfer efficiency trends, here they are similar for all pieces of skin. Future studies may examine causes of the upward trend in efficiency as frequency increases, although this behavior is typical of many capacitive networks.

Beyond simply the theoretical maximum voltage transfer efficiency, designers may want to explore the effects of varying impedances on the efficiency. Figure 5.11 shows the effects of varying the load impedance over a range of values for skin #3, with a fixed frequency of 124 KHz. The maximum efficiency, about 87%, occurs at approximately $j70\Omega$, corresponding to

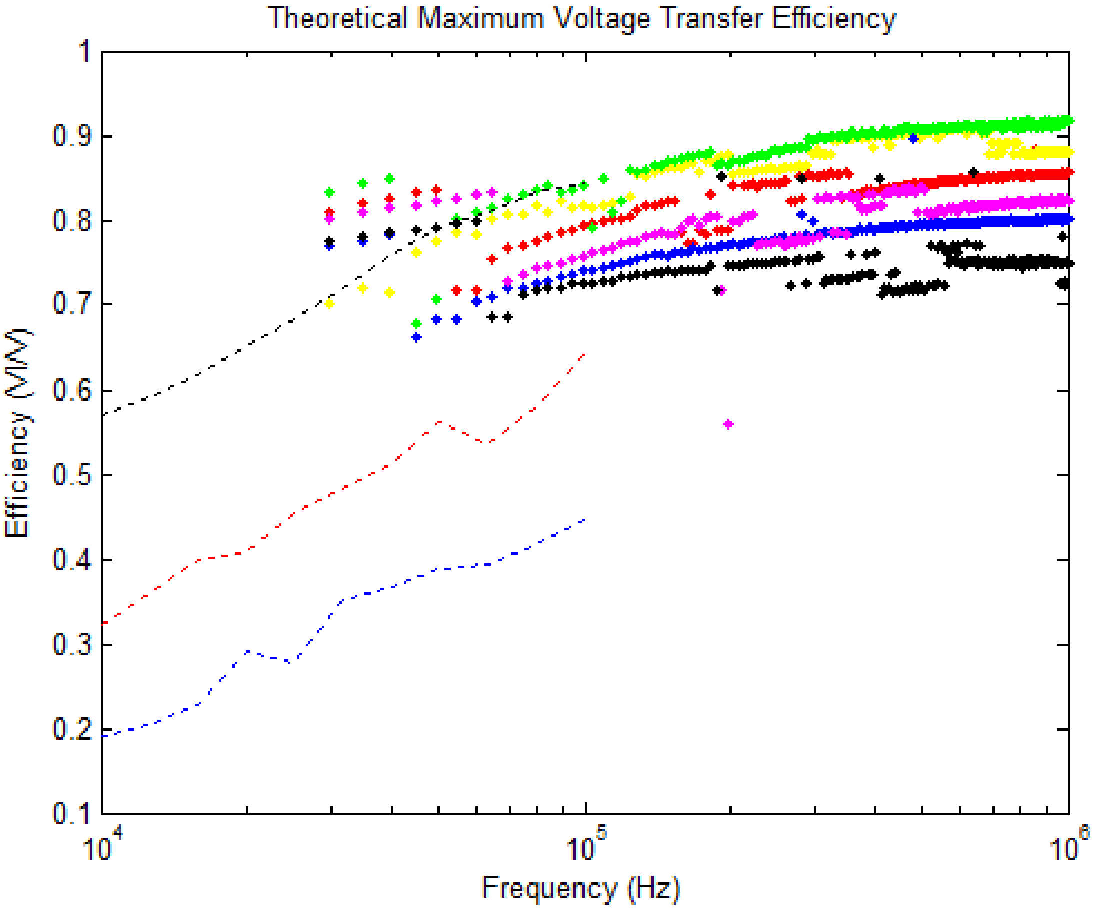


Figure 5.10: Theoretical maximum voltage transfer efficiencies are shown as functions of frequency for all nine pieces of skin tested. Skin #1: 6 mm - blue. Skin #2: 5 mm - green. Skin #3: 4-5 mm - red. Skin #4: 7 mm - magenta. Skin #5: 4-5 mm - yellow. Skin #6: 7 mm - black. Skin #7: 7+ mm - dashed red. Skin #8: 7+ mm - dashed blue. Skin #9: 3+ mm - dashed black.

an inductance of about $89.8\mu H$. Obviously an input impedance of such value is impractical for an implantable device. Calculating the efficiency using the impedance of the implantable device measured in section 4.3 as the load, and assuming $Zi = 0$, the theoretical maximum voltage transfer efficiency is approximately 65%. An engineer designing an implantable device circuit could make efforts to modify the input impedance of the device to achieve a desirable voltage transfer ratio within the limit, or determine how much voltage to apply from the source side for a given received voltage at the receiving electrodes.

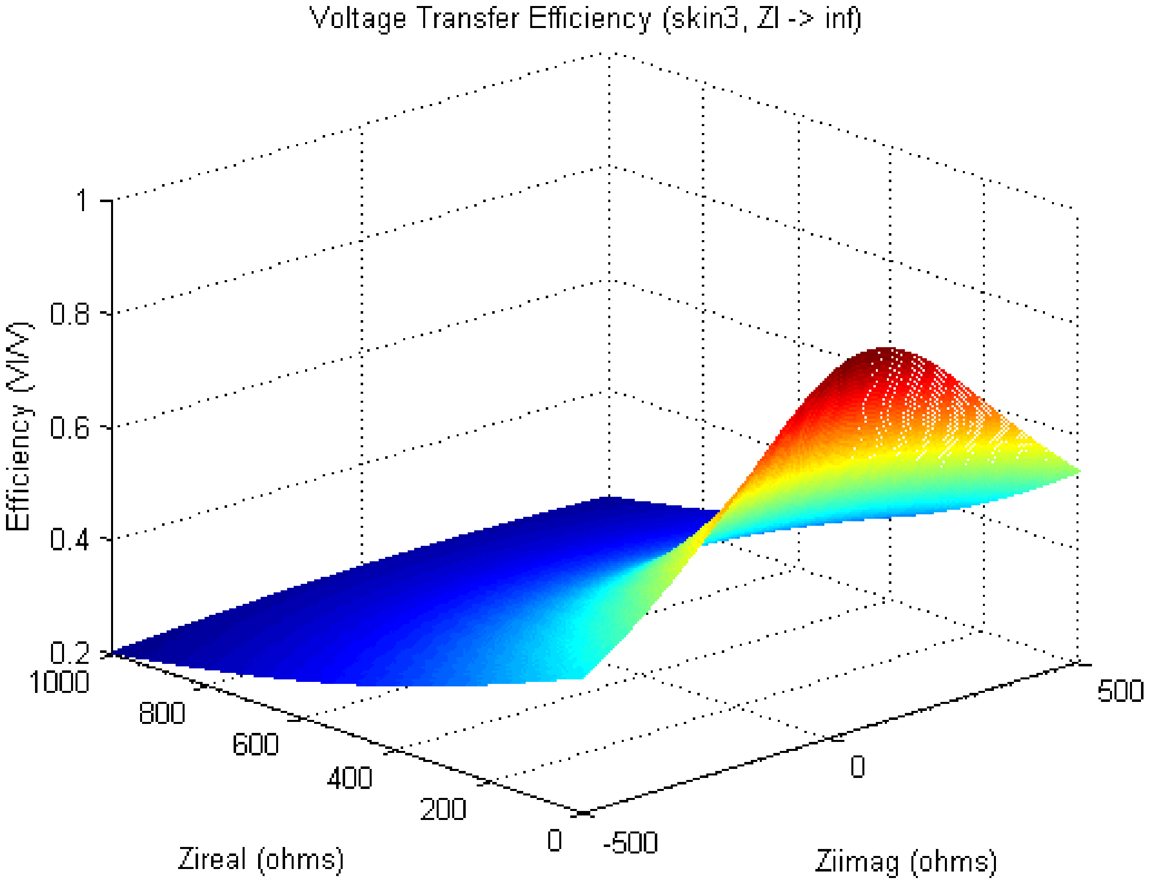


Figure 5.11: Voltage transfer efficiency for skin #3 as a function of load impedance.

6.0 DISCUSSION

Numerous issues which were either not addressed or addressed only briefly in the body of the text require further attention. From remarks about the device and experimental procedures to comments about the significance of the work, these are all important issues and deserve consideration, either in the context of this work or with regard to future studies.

Truly, the development of the device and model for this system is only a small part of the larger picture. The research here serves as a proof of concept study for a technology, volume conduction energy transfer, which holds much potential for not only the implantable device field, but the medical field in general. As discussed in 2.2, options for powering many important implantable devices are very limited. Because such devices are usually chronically implanted, batteries can rarely provide power for the lifetime of the device on a single charge. Until power requirements are scaled down to a level where implants can get their energy from renewable sources inside the body, some sort of replacement or recharging scheme must be used. The risks associated with repeated battery replacement surgeries make such a method seem outdated, but it has become commonplace for various neural implants because there is simply no better method at present. The other option, powering and/or recharging via inductive coupling energy transfer, requires relatively high power and larger components when compared to the volume conduction method, boasting efficiencies no higher than the theoretical maximums discussed in 5.4. Because it has been developed for decades though, inductive coupling has reached a point where, at first review, it seems superior to other infant technologies that attempt to break into the area, thus denying them the chance to really be fully developed. Here, a device has been designed, built, and successfully tested which relies on a relatively unexplored technology. A model of the system of which it is a part has been developed, with demonstrations of how that model can be used as a tool to establish important design parameters for similar devices. By proving this volume conduction technology and providing a tool for future work, it is hoped that the advantages and usefulness of the technology will be highlighted and that it will be one of a number of emerging technologies responsible for bringing useful devices from research to commercialization.

Regarding the device itself, it is currently purely a test platform for proving the technology, although additional functions could be easily added for various implantable applications. Even the relatively outdated microcontroller used in this design has plenty of spare resources which could be used to handle signal sensing, data collection, or stimulus driving. Newer technology is much more capable of performing more advanced tasks with lower power requirements. To be used as a chronically implantable device, the epoxy coating would also have to be modified due to its toxicity to biological tissue. Numerous materials such as silicone, polyurethane, and similar have been established for biocompatible usage, and many other studies have focused on modifying other materials to make them biocompatible.

Some implantable device systems require the exchange of energy with many distributed devices throughout the body. In its present state, the developed system is incapable of interfacing with multiple devices from the external side of the body. The modeled current pathways only provide for one channel in and out of the body through the skin, precluding the use of a “one-to-many” system. Such systems can indeed be realized using volume conduction technology, but their fundamental design and protocols differ greatly from that studied here, requiring different directional volume conduction antennas instead of electrode coupling across the skin, and necessitating multiple device communication schemes. The distributed nature of such a system also implies greater distance between devices, which has negative effects on power transfer efficiency. The development of a volume conduction model for a distributed implantable system is relegated to other separate studies.

An important point about recharging the battery is the maximum number of cycles the battery can withstand. The goal of this implantable volume conduction system is to use a smaller device with a smaller battery and recharge the battery when necessary, with the idea that this scheme will keep the device functional much longer than alternate systems which use one large non-rechargeable battery. However, given that even rechargeable batteries are limited to a certain number of life cycles, greatly dependent on usage and storage, the rechargeable advantage disappears with respect to device replacement cycles. There is also risk of potential functional complications associated with rechargeable batteries due to their changing behavior as the number of recharging cycles increases [13]. Electrical changes in the battery could cause many problems if not totally prepared for in the circuit design, thus increasing system development and usage complication. The addition of supercapacitors to complement rechargeable batteries is a common suggestion, and would serve to lessen negative battery effects by reducing the amount and frequency of battery recharging. The long term use of supercapacitors with rechargeable batteries has yet to be investigated in the laboratory, but such technology is advancing rapidly and will likely be included as a standard design component in future implantable devices. Though supercapacitors can't completely eliminate eventual battery degradation, other advantages of the rechargeable battery scheme still exist, mainly those associated with small size, such as lower risk surgeries, elimination of cabling, and reduced bodily rejection, giving credit to the choice to use a rechargeable battery.

The experiments detailed in this work are the culminations of many previous experiments, involving tests through resistive networks, slabs of agar, and swine skin from the scalp, ear, and flank regions. Many of the experiments during the first half of the research project served only as debugging trials for the implantable device hardware and software during development of the device. Once the device was working reliably, focus shifted to examining the properties of the skin itself. These early experiments highlighted the plethora of variables to consider during device design. Power, current, voltage, waveform, frequency, skin thickness, electrode material and geometry, and skin preparation all have significant effects on the behavior of the system. Multiple iterations of experiments were required to refine procedures with respect to each of these parameters. Even after electrical and electrode parameters were established, some experimental results had to be discarded due to poor experimental setup. Poor calibration of the testing equipment led to unreliable data, and allowing too much time between sacrifice and measurement led to situations where the skin properties changed rapidly, becoming very susceptible to changes in moisture content. The shallow saline bath is perhaps the best method of keeping the skin "intact," since it simulates having a supply of internal fluids while leaving the external skin exposed to the environment. After refining all of these variables, the availability of samples in the time frame up to the conclusion of the research was limited, restricting the data set to the nine pieces of skin in this thesis. However, even though the actual number of skin samples is small, the model is proven through hundreds of frequency points and results are consistent between similar pieces of skin.

The model developed in this research serves to represent the electrode-skin system in an electrical operating region relatively unexplored in the literature. Many other studies have attempted to model the skin for different applications. These include surgical monitoring and electroporation, which use high voltage, pulsed stimuli; iontophoresis, which uses moderate voltage, low frequency stimuli; RF energy transfer, which uses moderate voltage, high frequency stimuli; and experimental monitoring or signal recording, which focuses on low voltage, low frequency signals. The signals used in this study would be classified as low to moderate voltages at moderate frequencies. As such, common models used in other studies (see section 2.4) are not applicable to the present area. Some early generations of the device, influenced by models from the literature and previous laboratory work in volume conduction (see section 2.5), did operate at lower frequencies below 10 KHz, but they ultimately proved inadequate for overall system design. Upon first attempting to model the system in the moderate voltage/frequency range, a "black box" characterization method using network synthesis techniques [5, 44] was pursued. This technique was abandoned after it was determined that it would not give much additional insight into understanding electrical phenomena within the skin, from where a bulk of the ultimate design constraints come. An accurate physical-based model was needed to overcome the shortcomings of the previous X-Model, which bordered on following the "black box" approach. Future volume conduction studies and experiments will need to focus more on this modeling problem to illuminate the skin system's behavior over the present operating region, as the energy transfer medium is perhaps the most important part of the system.

Actual use of the developed X-Δ model does not guarantee a fully optimized system. In its current state, it is simply a tool to help with analyzing the system and its responses, meant to be used in conjunction with other device and external circuitry design tools to provide engineering limits. Future research should improve the model and transform it into part of an official design procedure. In the present context, however, pre-implantation design efforts may be limited due to the lack of available system measurements before implantation. The model parameters for each piece of skin in this study were fit using measured impedance values from each respective skin, an approach which may not be feasible in the commercial use of implantable systems, again highlighting the need for additional research on the skin and its properties to make a robust model that is adaptable to different application scenarios without a wealth of information on the specific biological environment. This may not be as large of a problem as it sounds, though; *in vivo* calibration and measurements are performed during the implantation of some implantable nerve stimulators (e.g. DBS). A similar process could be adopted for volume conduction systems to obtain system parameters of the targeted implant site and eliminate the problem of calibrating a system "blindly," based only on general impedance trends from previous studies.

One of the most important issues that deserves future research is the examination and adaptation of the X-Δ model to incorporate frequency dependent characteristics. Especially at the lower frequencies tested in this research, behaviors such as the current and voltage transfer functions change with frequency. Obviously this is not abnormal for linear systems with passive electrical components, but it is likely that non-linear frequency dependence exists. Figure 6.1 shows the error between the equivalent impedance of the X-Δ model calculated at certain frequencies for skin #1 and measured impedances across the whole range of frequencies tested. It can be seen that the model parameters calculated at one frequency can not accurately represent the actual skin impedances at other frequencies, using the current model. Including frequency dependency in future versions of the model will make it more universal and increase utility as a design tool. Of course, again, this necessitates further studies of the electrode-skin system in the electrical operating region useful for volume conduction energy transfer applications.

In addition to or in concert with exploring frequency dependency characteristics of the X-Δ model, further additions could be made. The most important addition would be the inclusion of explicit components representing the electrode-electrolyte-skin interfaces on either side of the skin. Currently these are essentially lumped into the X-Δ model compo-

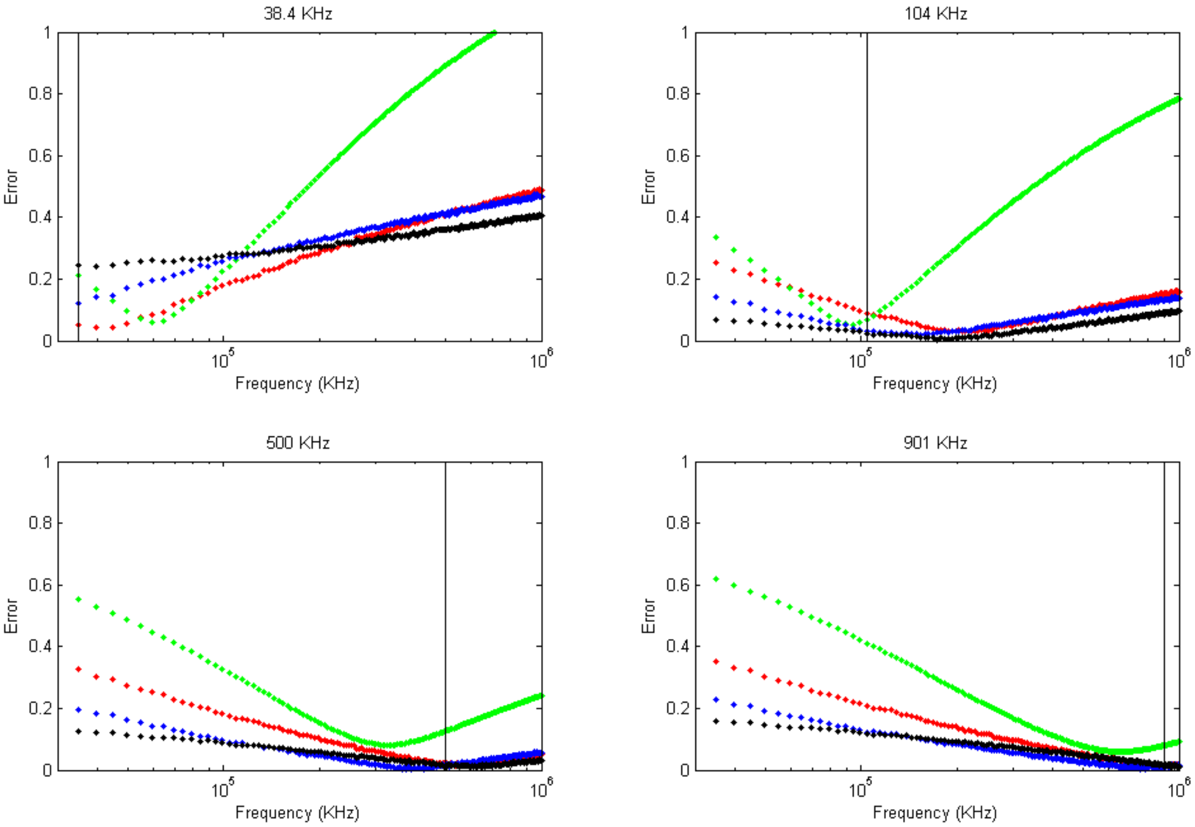


Figure 6.1: Error between the equivalent impedance of the X-Δ model calculated at certain frequencies and measured impedances across the whole range of frequencies tested for skin #1.

nents, which serve the main purpose of identifying current pathways through the skin. The interfaces, however, are critical in capturing the complete behavior of the system, as interpreted from the errors in table 5.2 (most noticeably those associated with skin #3). Beyond understanding and completeness, incorporating these interface components into the model would likely reduce error in system representation and allow for greater flexibility in design adjustments based on modified system setup or environmental conditions (moisture, skin responses, electrode size, shape, material, etc.). In future studies, physiological details could be included in the model, adding representation of non-linear behavior due to chemical or biological processes inherent in the system. For example, changes in conductivity resulting from sweat gland activity or application of varying currents are known to occur in the skin.

This would also allow for additional design options, e.g. applying chemicals to the skin to change its conductivity in specific areas for more direct control of current flow within the system.

Many of the above modeling concepts would require bodies of work before becoming applicable, but it has been shown how even the present model can accurately represent the system and help in the design of implantable systems using volume conduction. Further refinement will only make the model more relevant for a wider array of applications, reduce system representation error, and give engineers additional options for implantable system implementation.

7.0 CONCLUSIONS

In this dissertation, a platform for exploring the efficacy of volume conduction energy transfer for implantable devices was developed and tested. The implant system, which incorporates an implantable device, an external handheld controller, and skin connection electrodes, was tested and verified through agar skin models, a complete head model, a pig skin animal model, and within a live pig. The system displays the capability for communication and recharging using volume conduction technology, setting the stage for further development and integration of the technology into future devices. The technology provides a low power, secure energy solution that has potential for decreasing device size, eliminating problematic implant cabling, reducing implant surgery risk, and reducing the chance of bodily complications. Additionally, a novel X-Δ model of the implantable system has been created, which addresses limits in system representation inherent in certain previous models. The model has been shown to accurately represent the system's electrode-skin components over the targeted voltage and frequency regions for volume conduction applications, displaying small error between its equivalent impedances and actual measured impedances. Use of the model as a tool in implantable system design and development has been explained and discussed, helping to establish volume conduction technology as a viable contender for providing the energy needs of various implantable devices.

APPENDIX A

MICROCONTROLLER CODE (PIC12F683) FOR IMPLANTABLE DEVICE

```
// Steven A. Hackworth <sah24@pitt.edu> in conjunction with Xueling Lu, Energy Pad Project
// Laboratory for Computational Neuroscience
// University of Pittsburgh/UPMC
// Started: March 6, 2006
// Last Updated: April 21, 2008

// Code for internally implanted device


#include <12F683.h>
#use delay(clock=4000000) //clock frequency is 4 MHz
#use fast_io(A)
#fuseS NOWDT, NOCPD, NOPROTECT, NOMCLR, PUT, RC, BROWNOUT, NOIESO, FCMEN

//void ATAWakeup(); // external circuit will use an envelope detector circuit
void ManchesterLow();
void ManchesterHigh();
void Pulse2ms();
void turnon_WDT();
void turnon_IOC();
void turnoff_IOC();

void main(){

/*
  In MPLAB, set up the following configuration
external RC oscillator at 4 MHz
WDT (disabled, controlled by SWDTEN bit)
Code protector off
Master Clear Internal
*/

int i,count;
int header=0,command=0,data=0;
int1 lastb;

set_tris_a(0x08);  // GP3,4,5 is input; GP0,1,2 is output

//set WDT period to ~8s
#asm
BCF 0x18,0x04 ;bit 4 clear
BSF 0x18,0x03 ;bit 3 set
BSF 0x18,0x02 ;bit 2 set
BCF 0x18,0x01 ;bit 1 clear
#endasm

output_low(PIN_A1); // controls switch enable
output_low(PIN_A0); // controls LED indicator
output_low(PIN_A2); // controls output


    while(TRUE){
mystart: delay_us(400);
turnon_IOC();
```

```
sleep();
turnoff_IOC();
delay_ms(1);

// output_low(PIN_A1); // handled in sub-functions

//looking for bit 1 to start collecting data
lastb=0;

// follow preamble of 0's until it ends with a 1
while(TRUE)
{
count=0;
// must be a transition in middle of bit time
while (lastb == input(PIN_A3))
{
count++;
if (count > 250) // more than 1 ms
{
goto mystart;
}
}

if (lastb)
{
goto head;
}

delay_ms(1); // wait >1/2 bit time
delay_us(200);
lastb = input(PIN_A3); // update bit value in case transition in boundary
 }


head: delay_ms(1); // wait >1/2 bit time until start of first header bit
delay_us(200);
lastb = input(PIN_A3);

// input header byte
for (i=0;i<8;i++)
{
count=0;
//must be a transition in middle of bit time
while (lastb == input(PIN_A3))
{
count++;
if (count > 250)
{
goto mystart;
}
}

lastb = input(PIN_A3);
if (!lastb)
bit_set(header,i); // store data in header variable, LSB first

delay_ms(1); // wait >1/2 bit time
delay_us(200);
lastb = input(PIN_A3);
}

// input command byte (not needed right now, but it's still here)
for (i=0;i<8;i++)
{
count=0;
//must be a transition in middle of bit time
while (lastb == input(PIN_A3))
{
count++;
if (count > 250)
{
goto mystart;
}
}

lastb = input(PIN_A3);
if (!lastb)
bit_set(command,i); // store data in command variable, LSB first

delay_ms(1); // wait >1/2 bit time
```

```
delay_us(200);
lastb = input(PIN_A3);
       }

// input data byte
for (i=0;i<8;i++)
{
count=0;
//must be a transition in middle of bit time
while (lastb == input(PIN_A3))
{
count++;
if (count > 250){
goto mystart;
}
}

lastb = input(PIN_A3);
if (!lastb)
bit_set(data,i); // store data in data variable, LSB first

delay_ms(1); // wait >1/2 bit time
delay_us(200);
lastb = input(PIN_A3);
}


if (header == 0x63)
{
  //interpret the command
  if (command == 0x01) // copycat command, return received data
{
// ATAWakeup(); // Wakeup the controller's ATA5282 chip
// not needed since external circuit has envelope detector

delay_us(275); // add delay for total of ~500 us after last received bit

// transmit pre-header, 0's followed by a 1
for(i=0;i<12;i++)
{
ManchesterLow();
}
ManchesterHigh();

for(i=0;i<8;i++)
{
if(bit_test(header,i))
ManchesterHigh();
else
ManchesterLow();
}

for(i=0;i<8;i++)
{
if(bit_test(data,i))
ManchesterHigh();
else
ManchesterLow();
}

// output_low(PIN_A2); // handled in sub-functions
// output_low(PIN_A1); // handled in sub-functions

output_high(PIN_A0); // LED confirmation
delay_ms(250);
output_low(PIN_A0);
delay_ms(250);
output_high(PIN_A0);
delay_ms(250);
output_low(PIN_A0);
delay_ms(250);
output_high(PIN_A0);
delay_ms(250);
output_low(PIN_A0);
delay_ms(1000);
  }

  if (command == 0x02) //LED function
{
output_high(PIN_A0);
delay_ms(1000);
output_low(PIN_A0);
delay_ms(500);
output_high(PIN_A0);
delay_ms(1000);
output_low(PIN_A0);
delay_ms(1000);
  }

delay_us(400);
turnon_WDT();
     sleep();
}

header = 0;
command = 0;
data = 0;

} //end of while

} //end of main

/*
void ATAWakeup()
{
int i; // loop control variables

//// transmit preamble and header for ATA5282

output_high(PIN_A1); // flip switches for I/O direction -> output

// transmit pre-header, 320 periods of 125 KHz (16x20 for fewer loop checks)
for(i=0;i<16;i++)
{
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3); // 10
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
```

```
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2); // 20
}

output_low(PIN_A1); // disconnect output stage so it's not driving 5 volts
delay_us(256); // delay before header

// transmit header, 4 cycles of 16 periods on, 16 periods off
for(i=0;i<4;i++)
{
output_high(PIN_A1); // connect the output stage
delay_us(3); // extend connected voltage to be first half cycle

output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
```

```
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3); // 10
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);

output_low(PIN_A1); // disconnect the output stage so it's not driving 5 volts
delay_us(118); // subtract 8 us for loop, 2 us for PIN_A1 setting
}
delay_us(355); // delay after header for ATA5282 synch
}
*/

void ManchesterLow()
{
delay_ms(2);
Pulse2ms();
}


void ManchesterHigh()
{
Pulse2ms();
delay_ms(2);
}


void Pulse2ms()
// this function outputs a ~125 KHz square wave for 2 milliseconds
{
int i;

output_high(PIN_A1); // connect the output stage
// delay_us(1); // extend connected voltage to be first half cycle
for(i=0;i<10;i++) // add cycles for 2 ms on time (may need to work with this timing)
// oscillator is too slow -> loop check is too long for proper timing
// only 10 extra small delays amidst 250 cycles should be okay
{
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
```

```
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3); // 10
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3); // 20
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
delay_us(3);
output_high(PIN_A2);
delay_us(3);
output_low(PIN_A2);
```

```
}
output_low(PIN_A1); // disconnect the output stage so it’s not driving 3 volts
}

void turnon_WDT()
{

int dummy=0; // dummy variable to eliminate GPIO input mismatch

#asm
BCF 0x0B,0x03 ;GPIE in intcon register cleared
BSF 0x03,0x05  ;Bank 1
BCF 0x96,0x03 ;disable interrupt on change on A3
BCF 0x03,0x05  ;Bank 0
#endasm

dummy = input(PIN_A3); // read port to clear any mismatch condition
dummy = input(PIN_A3);

#asm
BCF 0x0B,0x00 ;GPIF in intcon cleared
BSF 0x18,0x00 ;bit 0 set   //SWDTEN turned on
#endasm

// notification that device is going to sleep
output_high(PIN_A0);
delay_ms(250);
output_low(PIN_A0);
delay_ms(250);
output_high(PIN_A0);
delay_ms(250);
output_low(PIN_A0);

restart_wdt();
}


void turnon_IOC()
{

int dummy=0; // dummy variable to eliminate GPIO input mismatch

// notification for readiness to receive command
output_high(PIN_A0);
delay_ms(250);
output_low(PIN_A0);

#asm
BCF 0x18,0x00 ;bit 0 clear   //SWDTEN turned off
#endasm

dummy = input(PIN_A3); // read port to clear any mismatch condition
dummy = input(PIN_A3);

#asm
BCF 0x0B,0x00 ;GPIF in intcon cleared
BSF 0x03,0x05  ;Bank 1
BSF 0x96,0x03 ;Enable interrupt on change on A3
BCF 0x03,0x05  ;Bank 0
BSF 0x0B,0x03 ;GPIE in intcon register set
#endasm

}


void turnoff_IOC()
{

int dummy=0; // dummy variable to eliminate GPIO input mismatch

#asm
BCF 0x0B,0x03 ;GPIE in intcon register cleared
BSF 0x03,0x05  ;Bank 1
BCF 0x96,0x03 ;disable interrupt on change on A3
BCF 0x03,0x05  ;Bank 0
#endasm

dummy = input(PIN_A3); // read port to clear any mismatch condition
dummy = input(PIN_A3);
```

```
#asm
BCF 0x0B,0x00 ;GPIF in intcon cleared
#endasm

}
```

APPENDIX B

MICROCONTROLLER CODE (PIC16F87) FOR HANDHELD CONTROLLER

```
// Steven A. Hackworth <sah24@pitt.edu>, Energy Pad Project
// Laboratory for Computational Neuroscience
// University of Pittsburgh/UPMC
// Started: May 4, 2006
// Last Updated: March 28, 2008

// Code for controlling the external programmer


#include <16F87.h>
#fuses INTRC_IO, NOWDT, PUT, NOMCLR, BROWNOUT, NOLVP, NOCPD, NOWRT, NODEBUG, NOPROTECT, FCMEN, NOIESO
#use delay(clock=8000000)
#use FAST_IO(A)
#use FAST_IO(B)

// interrupt flag variables to monitor interrupts in software
//#BIT INT0IF = 0x0B.1
//#BIT RBIF = 0x0B.0
//#BIT TMR0IF = 0x0B.2

#BYTE PORTA = 0x05

// these variables are used to keep track of the number of TMR0 timeouts
// this is ultimately used to turn off the LEDs after a certain amount of inactivity
//int counter = 0;
//int subcounter = 0;

// flags to check which interrupt was called, if any
// basically I don't want to go and see what this C Compiler
// does to take care of the inherent interrupt flags
int1 dataChangeFlag = 0;
int1 transmitFlag = 0;

// function list
#int_ext
void Transmit();
#int_rb
void DataChange();
//#int_rtcc
//void TimeOut();
void Charge();
void Copycat();
void ATAWakeup();
void ManchesterHigh();
void ManchesterLow();
void Pulse1ms();
void ErrorFunction();
void ErrorPreHeader();
void ErrorHeader1();
void ErrorData1();
void ErrorHeader2(int & header);
void ErrorData2(int & receiveddata);
void ShowData(int & receiveddata);

void main()
{

// select bank 1 for OSCCON register (set bit 5 of the STATUS reg)
// internal oscillator set to 8 MHz from default of 32 kHz (set bits 4-6 of OSCCON reg)
#asm
BSF 0x3,0x5
BSF 0xF,0x4
BSF 0xF,0x5
BSF 0xF,0x6
BCF 0x3,0x5
#endasm

set_tris_a(0x30);
// port A
// 7 -> output: LED on/off for returned data LED
// 6 -> output: LED on/off for user data LED
// 5 -> input:  function select
// 4 -> input:  TMR0 clock input tied high to halt TMR0 when necessary
// <3:0> -> outputs: binary code for BCD-to-7-segment decoder for received data
set_tris_b(0xF3);
// port B
// <7:4> -> inputs: binary code for BCD-to-7-segment decoder for user input data
// 3 -> output: send data out line
// 2 -> output: switch control for input/output direction
// 1 -> input:  received data in line
// 0 -> input:  transmit button to start communication


// initialize output states
output_low(PIN_A7);
output_low(PIN_A6);
output_low(PIN_A3);
output_low(PIN_A2);
output_low(PIN_A1);
output_low(PIN_A0);
output_low(PIN_B3);
output_low(PIN_B2);

// SETUP_TIMER_0(RTCC_DIV_256|RTCC_EXT_L_TO_H);  // function may not have en-dash between TIMER and 0?

ENABLE_INTERRUPTS(GLOBAL);
ENABLE_INTERRUPTS(INT_EXT);
EXT_INT_EDGE(L_TO_H);
ENABLE_INTERRUPTS(INT_RB);
// ENABLE_INTERRUPTS(INT_RTCC);

setup_wdt(WDT_1152MS); // set WDT timeout to 1152 ms
setup_wdt(WDT_TIMES_32); // set total WDT timeout to ~37 seconds

// enable wake up on interrupt from EXT and RB
#asm
BSF 0x0b, 0x4
#endasm

while(TRUE) // always running
{
// turn off LED displays if the controller woke up from a WDT timeout
if(dataChangeFlag == 0)
{
if(transmitFlag == 0)
{
output_low(PIN_A6);
output_low(PIN_A7);
}
}
dataChangeFlag = 0;
transmitFlag = 0;

restart_wdt();
#asm
BSF 0x18,0x00 ;bit 0 set   //SWDTEN turned on
#endasm


// sleep while waiting for interrupt flag
sleep();
```

```
/* if(TMR0IF == 1) // TMR0 timeout
{
TimeOut(counter, subcounter);
}

if(INT0IF == 1) // transmit button pressed
{
Transmit(counter, subcounter);
}

if(RBIF == 1) // user input data changed
{
DataChange(counter, subcounter);
}
*/
}
}


#int_ext
void Transmit()
// this function is called when the user presses the button to communicate
{
#asm
BCF 0x18,0x00 ;bit 0 set   //SWDTEN turned off
#endasm

output_low(PIN_A7); // turn off returned data 7-segment LED if it's on

if(input(PIN_A5)) // check function select input
{
Charge();
}
else
{
output_high(PIN_A6); // user input data LED on
Copycat();
// counter = 0; // reset timeout counters
// subcounter = 0;
// set_timer0(0); // reset TMR0
// TMR0IF = 0; // clear TMR0 interrupt flag, in case TMR0 is already on
}
delay_ms(1000); // software delay for switch bounce

transmitFlag = 1;
}


#int_rb
void DataChange()
// this function is called when the user changes the data input
// it serves to reset the timeout timing control
{
#asm
BCF 0x18,0x00 ;bit 0 set   //SWDTEN turned off
#endasm

output_low(PIN_A7); // returned data LED off
output_high(PIN_A6); // user input data LED on
// set_timer0(0); // reset the timer
// counter = 0; // reset timeout count variables
// subcounter = 0;
// SETUP_TIMER_0(RTCC_DIV_256|RTCC_INTERNAL); // enable timer counting
delay_ms(100); // software switch debounce
// RBIF = 0; // clear interrupt flag
dataChangeFlag = 1;
}

/*
#int_rtcc
void TimeOut()
// this function is called if TMR0 overflows
// TMR0 is set up to have a timeout period of ~.0327s with the prescaler
// with the counter variables, the 7-segment LEDs will be turned off after ~30 sec. of inactivity
{
if(counter < 4) // not completely timed out
{
if(subcounter == 255)
```

```
{
counter += 1; // increment main counter if subcounter rolls over
}
subcounter += 1;
}
else // timed out from inactivity
{
output_low(PIN_A6); // turn 7-segment LEDs off
output_low(PIN_A7);
SETUP_TIMER_0(RTCC_DIV_256|RTCC_EXT_L_TO_H); // stop timer counting
counter = 0; // reset timeout count variables
subcounter = 0;
}

TMR0IF = 0; // clear interrupt flag
}
*/

void Charge()
{
output_high(PIN_B2); // flip switches for I/O direction -> output
// delay_us(5); // make sure switch state has changed


// delay for switch bounce
delay_ms(100);

// transmit charging signal by holding transmit button
while(input(PIN_B0))
{
output_high(PIN_B3);
delay_us(3);
#asm
NOP
#endasm
output_low(PIN_B3);
delay_us(1);
#asm
NOP
#endasm
}

output_low(PIN_B3); // make sure output line is low

output_low(PIN_B2); // flip switches for I/O diretion -> input
// delay_us(5); // make sure switch state has changed
}


void Copycat()
{
int i; // loop control variables
int receiveddata = 0; // data received from internal device
int userdataout = 0; // data to be sent, input by user
int1 lastb = 0; // variable used in controlling data input flow
int count = 0; // variable to guard against spurious incoming data
int header = 0; // variable to store header bits

userdataout = input_b(); // read user data input
userdataout = (userdataout & 0xF0); // mask the lower bits (4:7 are data)
rotate_left(&userdataout, 1); // put data in the 4 LSBs
rotate_left(&userdataout, 1);
rotate_left(&userdataout, 1);
rotate_left(&userdataout, 1);
output_high(PIN_A6); // show user input data on 7-segment display


ATAWakeup(); // Wakeup the internal ATA5282 chip


//// communication packet for implant circuitry contains four parts: preamble, header, command, data

// transmit preamble, 0's followed by a 1
for(i=0;i<12;i++)
{
ManchesterLow();
}
ManchesterHigh();

// transmit header, 0x63, LSB first
```

```
ManchesterHigh();
ManchesterHigh();
ManchesterLow();
ManchesterLow();
ManchesterLow();
ManchesterHigh();
ManchesterHigh();
ManchesterLow();

// transmit Copycat command, 0x01, LSB first
ManchesterHigh();
ManchesterLow();
ManchesterLow();
ManchesterLow();
ManchesterLow();
ManchesterLow();
ManchesterLow();
ManchesterLow();

// transmit data
for(i=0;i<8;i++) // 4 actual data bits first, last 4 bits are filler
{
if(bit_test(userdataout,i))
{
ManchesterHigh();
}
else
{
ManchesterLow();
}
}
// delay_us(3);

// output_low(PIN_B2); // flip switches for I/O direction -> input
// done in Pulse1ms sub-function already
delay_ms(1); // delay until internal circuitry starts transmitting
delay_us(500);


//// receive and store data

// input pre-header
while(TRUE)
{
count = 0;
while(lastb == input(PIN_B1)) // wait for middle-of-bit transition; lastb was initialized to 0
{
delay_us(10);
count++;
if (count > 250) // error out if there is no transition after a ~long period of time (> 1 ms at least)
{
ErrorPreHeader();
goto end;
}
}

if(lastb == 1) // if end of pre-header
{
goto head; // jump to code to input header
}

delay_ms(3); // delay until start of next bit
lastb = input(PIN_B1); // check input for bit value
}

// input header
head: delay_ms(3); // delay until start of first header bit
lastb = input(PIN_B1); // get first header bit value

// input header byte
for(i=0;i<8;i++)
{
count = 0; // reset the timeout variable
while(lastb == input(PIN_B1)) // wait for middle-of-bit transition
{
delay_us(10);
count++;
if(count > 250) // error out if there is no transition after a ~long period of time (> 1 ms at least)
{
ErrorHeader1();
```

```
goto end;
}
}

if(lastb)
{
bit_set(header, i); // store data in header variable, LSB first
}
delay_ms(3); // delay until start of next bit
lastb = input(PIN_B1); // get next bit value
}

// if header is correct, input data, otherwise give error
if(header == 0x63)
{
// input data value
for(i=0;i<8;i++)
{
count = 0; // reset the timeout variable
while(lastb == input(PIN_B1)) // wait for middle-of-bit transition
{
delay_us(10);
count++;
if(count > 250) // error out if there is no transition after a ~long period of time (> 1 ms at least)
{
ErrorData1();
goto end;
}
}

if(lastb)
{
bit_set(receiveddata, i); // store data in data variable, LSB first
}

delay_ms(3); // delay until start of next bit
lastb = input(PIN_B1); // get next bit value
}
receiveddata = (receiveddata & 0x0F); // mask the higher bits of received data (only using 4 bits)
}
else
{
ErrorHeader2(header);
goto end;
}

// check if data is correct
if(receiveddata != userdataout) // error if data is incorrect
{
ErrorData2(receiveddata);
}
else
{
ShowData(receiveddata); // show correct data otherwise
}

end: delay_ms(300); //SETUP_TIMER_0(RTCC_DIV_256|RTCC_INTERNAL); // enable timer counting
}


void ATAWakeup()
{
int i,j; // loop control variables

//// transmit preamble and header for ATA5282

output_high(PIN_B2); // flip switches for I/O direction -> output

// transmit pre-header, 320 periods of 125 KHz (2x160 b/c int limit is 255)
for(i=0;i<2;i++)
{
for(j=0;j<160;j++)
{
output_high(PIN_B3);

if(159 == j)
{
#asm
NOP
#endasm
```

```
break;
}

delay_us(1);
output_low(PIN_B3);
// delay_us(1); // loop time is enough delay
}
output_low(PIN_B3); // since loop is broken before PIN_B3 is reset
}
output_low(PIN_B2); // disconnect output stage so it's not driving 5 volts
delay_us(256); // delay before header

// transmit header, 4 cycles of 16 periods on, 16 periods off
for(i=0;i<4;i++)
{
output_high(PIN_B2); // connect the output stage
delay_us(1); // extend connected voltage to be first half cycle
for(j=0;j<16;j++)
{
output_high(PIN_B3);

if(15 == j)
{
#asm
NOP
#endasm
break;
}

delay_us(1);
output_low(PIN_B3);
// delay_us(1); // loop time is enough delay
}
output_low(PIN_B3); // since loop is broken before PIN_B3 is reset
output_low(PIN_B2); // disconnect the output stage so it's not driving 5 volts
delay_us(123); // subtract 4 us for loop, 1 us for PIN_B2 setting
}
delay_us(355); // delay after header for ATA5282 synch
}


void ManchesterHigh()
// high-to-low transition in middle of bit time represents 1
{
Pulse1ms();
delay_ms(1);
}


void ManchesterLow()
// low-to-high transition in middle of bit time represents 0
{
delay_ms(1);
Pulse1ms();
}


void Pulse1ms()
// this forms one-half (high level) of a Manchester-coded bit through the ATA5282
{
int i;

output_high(PIN_B2); // connect the output stage
delay_us(1); // extend connected voltage to be first half cycle
for(i=0;i<125;i++) // add cycles for 1 ms on time
{
output_high(PIN_B3);
delay_us(3);
#asm
NOP
#endasm
output_low(PIN_B3);
}
output_low(PIN_B2); // disconnect the output stage so it's not driving 5 volts
}


void ErrorPreHeader()
// this function is called if there is a problem in actually receiving the pre-header in the Copycat function
// it makes the received data feedback LED blink quickly with an error code of '-'
```

```
{
int i;

output_high(PIN_A3);
output_low(PIN_A2);
output_high(PIN_A1);
output_low(PIN_A0);

for(i=0;i<4;i++)
{
output_high(PIN_A7);
delay_ms(750);
output_low(PIN_A7);
delay_ms(700);
}
}


void ErrorHeader1()
// this function is called if there is a problem in actually receiving the header in the Copycat function
// it makes the received data feedback LED blink quickly with an error code of 'E'
{
int i;

output_high(PIN_A3);
output_low(PIN_A2);
output_high(PIN_A1);
output_high(PIN_A0);

for(i=0;i<4;i++)
{
output_high(PIN_A7);
delay_ms(750);
output_low(PIN_A7);
delay_ms(700);
}
}


void ErrorData1()
// this function is called if there is a problem in actually receiving data in the Copycat function
// it makes the received data feedback LED blink quickly with an error code of 'd'
{
int i;

output_high(PIN_A3);
output_high(PIN_A2);
output_low(PIN_A1);
output_high(PIN_A0);

for(i=0;i<4;i++)
{
output_high(PIN_A7);
delay_ms(750);
output_low(PIN_A7);
delay_ms(700);
}
}

void ErrorHeader2(int & header)
// this function is called if the received header is incorrect in the Copycat function
// it makes the received data feedback LED blink slowly with an error code of 'E', then shows the received header
{
int i;

output_high(PIN_A3);
output_low(PIN_A2);
output_high(PIN_A1);
output_high(PIN_A0);

for(i=0;i<4;i++)
{
output_high(PIN_A7);
delay_ms(1250);
output_low(PIN_A7);
delay_ms(1200);
}

for(i=0;i<4;i++)
{
```

```
if(bit_test(header, i))
{
bit_set(PORTA, i);
}
else
{
bit_clear(PORTA, i);
}
}

output_high(PIN_A7);
}


void ErrorData2(int & receiveddata)
// this function is called if the received data is incorrect in the Copycat function
// it makes the received data feedback LED blink slowly with an error code of 'd', then shows the received data
{
int i;

output_high(PIN_A3);
output_high(PIN_A2);
output_low(PIN_A1);
output_high(PIN_A0);

for(i=0;i<4;i++)
{
output_high(PIN_A7);
delay_ms(1250);
output_low(PIN_A7);
delay_ms(1200);
}

for(i=0;i<4;i++)
{
if(bit_test(receiveddata, i))
{
bit_set(PORTA, i);
}
else
{
bit_clear(PORTA, i);
}
}

output_high(PIN_A7);
}


void ShowData(int & receiveddata)
// this function displays the correctly received data from the Copycat function
{
int i;

for(i=0;i<4;i++)
{
if(bit_test(receiveddata, i))
{
bit_set(PORTA, i);
}
else
{
bit_clear(PORTA, i);
}
}

output_high(PIN_A7); // quick flash for confirmation then hold
delay_ms(250);
output_low(PIN_A7);
delay_ms(250);
output_high(PIN_A7);
delay_ms(250);
output_low(PIN_A7);
delay_ms(250);
output_high(PIN_A7);

delay_ms(3000); // hold for three seconds
}
```

APPENDIX C

MAXIMA CALCULATIONS FOR FINDING DERIVATIVES OF Z_{12} EQUIVALENT IMPEDANCES OF THE X-Δ MODEL

```
(%i1) ((tr+dr+%i*(ti+di))*(2*sr+cr+%i*(2*si+ci))+(sr+%i*si)*(cr+%i*ci))/(tr+dr+sr+%i*(ti
 +di+si));
```

(%o1)

$$\frac{(2\,sr + i\ (2\,si + ci) + cr)\ (tr + i\ (ti + di) + dr) + (cr + i\,ci)\ (sr + i\,si)}{tr + i\ (ti + si + di) + sr + dr}$$

```
(%i2) realpart(%o1);
```

(%o2)

$$\Big(\,(tr + sr + dr)\ ((2\,sr + cr)\ (tr + dr) - (2\,si + ci)\ (ti + di) + cr\,sr - ci\,si) -$$

$$(-ti - si - di)\ ((2\,si + ci)\ (tr + dr) + (2\,sr + cr)\ (ti + di) + ci\,sr + cr\,si)\Big)/$$

$$\Big((tr + sr + dr)^2 + (ti + si + di)^2\Big)$$

```
(%i3) imagpart(%o1);
```

(%o3)

$$\Big(\,(-ti - si - di)\ ((2\,sr + cr)\ (tr + dr) - (2\,si + ci)\ (ti + di) + cr\,sr - ci\,si) +$$

$$(tr + sr + dr)\ ((2\,si + ci)\ (tr + dr) + (2\,sr + cr)\ (ti + di) + ci\,sr + cr\,si)\Big)/$$

$$\Big((tr + sr + dr)^2 + (ti + si + di)^2\Big)$$

```
(%i4) diff(%o2, tr);
```

(%o4)

$$\frac{(2\,sr + cr)\ (tr + sr + dr) + (2\,sr + cr)\ (tr + dr) - (2\,si + ci)\ (ti + di) - (2\,si + ci)\ (-ti - si - di) + cr\,sr - ci\,si}{(tr + sr + dr)^2 + (ti + si + di)^2} -$$

$$\Big(2\ (tr + sr + dr)\ ((tr + sr + dr)\ ((2\,sr + cr)\ (tr + dr) - (2\,si + ci)\ (ti + di) + cr\,sr - ci\,si) -$$

$$(-ti - si - di)\ ((2\,si + ci)\ (tr + dr) + (2\,sr + cr)\ (ti + di) + ci\,sr + cr\,si))\Big)/$$

$$\Big((tr + sr + dr)^2 + (ti + si + di)^2\Big)^2$$

```
(%i5) diff(%o2, ti);
```

(%o5)

$$\frac{(-2\,si-ci)\;(tr+sr+dr)+(2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)-(2\,sr+cr)\;(-ti-si-di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si)-$$

$$(-ti-si-di)\;((2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i6) diff(%o2, dr);
```

(%o6)

$$\frac{(2\,sr+cr)\;(tr+sr+dr)+(2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)-(2\,si+ci)\;(-ti-si-di)+cr\,sr-ci\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((tr+sr+dr)\;((2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si)-$$

$$(-ti-si-di)\;((2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i7) diff(%o2, di);
```

(%o7)

$$\frac{(-2\,si-ci)\;(tr+sr+dr)+(2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)-(2\,sr+cr)\;(-ti-si-di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si)-$$

$$(-ti-si-di)\;((2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i8) diff(%o2, sr);
```

(%o8)

$$\frac{(tr+sr+dr)\;(2\;(tr+dr)+cr)+(2\,sr+cr)\;(tr+dr)-(-ti-si-di)\;(2\;(ti+di)+ci)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((tr+sr+dr)\;((2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si)-$$

$$(-ti-si-di)\;((2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i9) diff(%o2, si);
```

(%o9)

$$\frac{-(-ti-si-di)\;(2\;(tr+dr)+cr)+(-2\;(ti+di)-ci)\;(tr+sr+dr)+(2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((2\,sr+cr)\;(tr+dr)-(2\,si+ci)\;(ti+di)+cr\,sr-ci\,si)-$$

$$(-ti-si-di)\;((2\,si+ci)\;(tr+dr)+(2\,sr+cr)\;(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i10) diff(%o2, cr);
```

(%o10)

$$\frac{(tr+sr+dr)^2-(-ti-si-di)\;(ti+si+di)}{(tr+sr+dr)^2+(ti+si+di)^2}$$

```
(%i11) diff(%o2, ci);
```

(%o11) $$0$$

```
(%i12) diff(%o3, tr);
```

(%o12)
$$\frac{(2\,si+ci)\,(tr+sr+dr)+(2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+(2\,sr+cr)\,(-ti-si-di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$
$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$
$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i13) diff(%o3, ti);
```

(%o13)
$$\frac{(2\,sr+cr)\,(tr+sr+dr)-(2\,sr+cr)\,(tr+dr)+(2\,si+ci)\,(ti+di)+(-2\,si-ci)\,(-ti-si-di)-cr\,sr+ci\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$
$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$
$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i14) diff(%o3, dr);
```

(%o14)
$$\frac{(2\,si+ci)\,(tr+sr+dr)+(2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+(2\,sr+cr)\,(-ti-si-di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$
$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$
$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i15) diff(%o3, di);
```

(%o15)
$$\frac{(2\,sr+cr)\,(tr+sr+dr)-(2\,sr+cr)\,(tr+dr)+(2\,si+ci)\,(ti+di)+(-2\,si-ci)\,(-ti-si-di)-cr\,sr+ci\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$
$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$
$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i16) diff(%o3, sr);
```

(%o16)
$$\frac{(-ti-si-di)\,(2\,(tr+dr)+cr)+(2\,(ti+di)+ci)\,(tr+sr+dr)+(2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$
$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$
$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i17) diff(%o3, si);
```

(%o17)

$$\frac{(tr+sr+dr)\,(2\,(tr+dr)+cr)-(2\,sr+cr)\,(tr+dr)+(-ti-si-di)\,(-2\,(ti+di)-ci)+(2\,si+ci)\,(ti+di)-cr\,sr+ci\,si}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((2\,sr+cr)\,(tr+dr)-(2\,si+ci)\,(ti+di)+cr\,sr-ci\,si)+$$

$$(tr+sr+dr)\,((2\,si+ci)\,(tr+dr)+(2\,sr+cr)\,(ti+di)+ci\,sr+cr\,si))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i18) diff(%o3, cr);
```

(%o18)

$$\frac{(ti+si+di)\,(tr+sr+dr)+(-ti-si-di)\,(tr+sr+dr)}{(tr+sr+dr)^2+(ti+si+di)^2}$$

```
(%i19) diff(%o3, ci);
```

(%o19)

$$\frac{(tr+sr+dr)^2+(-ti-si-di)^2}{(tr+sr+dr)^2+(ti+si+di)^2}$$

APPENDIX D

MAXIMA CALCULATIONS FOR FINDING DERIVATIVES OF Z_{13} EQUIVALENT IMPEDANCES OF THE X-Δ MODEL

```
(%i1) ((tr+%i*ti)*(dr+sr+%i*(di+si)))/(tr+dr+sr+%i*(ti+di+si));
```

(%o1)

$$\frac{(sr + i\,(si + di) + dr)\,(tr + i\,ti)}{tr + i\,(ti + si + di) + sr + dr}$$

```
(%i2) realpart(%o1);
```

(%o2)

$$\frac{(sr + dr)\,(tr\,(tr + sr + dr) - (-ti - si - di)\,ti) - (si + di)\,(ti\,(tr + sr + dr) + (-ti - si - di)\,tr)}{(tr + sr + dr)^2 + (ti + si + di)^2}$$

```
(%i3) imagpart(%o1);
```

(%o3)

$$\frac{(si + di)\,(tr\,(tr + sr + dr) - (-ti - si - di)\,ti) + (sr + dr)\,(ti\,(tr + sr + dr) + (-ti - si - di)\,tr)}{(tr + sr + dr)^2 + (ti + si + di)^2}$$

```
(%i4) diff(%o2, tr);
```

(%o4)

$$\frac{(sr + dr)\,(2\,tr + sr + dr) - (-si - di)\,(si + di)}{(tr + sr + dr)^2 + (ti + si + di)^2} -$$

$$\frac{2\,(tr + sr + dr)\,((sr + dr)\,(tr\,(tr + sr + dr) - (-ti - si - di)\,ti) - (si + di)\,(ti\,(tr + sr + dr) + (-ti - si - di)\,tr))}{\left((tr + sr + dr)^2 + (ti + si + di)^2\right)^2}$$

```
(%i5) diff(%o2, ti);
```

(%o5)

$$\frac{(sr + dr)\,(2\,ti + si + di) - (si + di)\,(sr + dr)}{(tr + sr + dr)^2 + (ti + si + di)^2} -$$

$$\frac{2\,(ti + si + di)\,((sr + dr)\,(tr\,(tr + sr + dr) - (-ti - si - di)\,ti) - (si + di)\,(ti\,(tr + sr + dr) + (-ti - si - di)\,tr))}{\left((tr + sr + dr)^2 + (ti + si + di)^2\right)^2}$$

```
(%i6) diff(%o2, dr);
```

(%o6)
$$\frac{tr\ (tr+sr+dr)+(sr+dr)\ tr-(-ti-si-di)\ ti-(si+di)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (tr+sr+dr)\ ((sr+dr)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)-(si+di)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i7) diff(%o2, di);
```

(%o7)
$$\frac{-ti\ (tr+sr+dr)-(-ti-si-di)\ tr+(si+di)\ tr+(sr+dr)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (ti+si+di)\ ((sr+dr)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)-(si+di)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i8) diff(%o2, sr);
```

(%o8)
$$\frac{tr\ (tr+sr+dr)+(sr+dr)\ tr-(-ti-si-di)\ ti-(si+di)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (tr+sr+dr)\ ((sr+dr)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)-(si+di)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i9) diff(%o2, si);
```

(%o9)
$$\frac{-ti\ (tr+sr+dr)-(-ti-si-di)\ tr+(si+di)\ tr+(sr+dr)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (ti+si+di)\ ((sr+dr)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)-(si+di)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i10) diff(%o2, cr);
```

(%o10)
$$0$$

```
(%i11) diff(%o2, ci);
```

(%o11)
$$0$$

```
(%i12) diff(%o3, tr);
```

(%o12)
$$\frac{(si+di)\ (2\,tr+sr+dr)+(-si-di)\ (sr+dr)}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (tr+sr+dr)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i13) diff(%o3, ti);
```

(%o13)
$$\frac{(si+di)\ (2\,ti+si+di)+(sr+dr)^2}{(tr+sr+dr)^2+(ti+si+di)^2}-\frac{2\ (ti+si+di)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i14) diff(%o3, dr);
```

(%o14)

$$\frac{ti\ (tr+sr+dr)+(-ti-si-di)\ tr+(si+di)\ tr+(sr+dr)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\frac{2\ (tr+sr+dr)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i15) diff(%o3, di);
```

(%o15)

$$\frac{tr\ (tr+sr+dr)-(sr+dr)\ tr-(-ti-si-di)\ ti+(si+di)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\frac{2\ (ti+si+di)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i16) diff(%o3, sr);
```

(%o16)

$$\frac{ti\ (tr+sr+dr)+(-ti-si-di)\ tr+(si+di)\ tr+(sr+dr)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\frac{2\ (tr+sr+dr)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i17) diff(%o3, si);
```

(%o17)

$$\frac{tr\ (tr+sr+dr)-(sr+dr)\ tr-(-ti-si-di)\ ti+(si+di)\ ti}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\frac{2\ (ti+si+di)\ ((si+di)\ (tr\ (tr+sr+dr)-(-ti-si-di)\ ti)+(sr+dr)\ (ti\ (tr+sr+dr)+(-ti-si-di)\ tr))}{\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2}$$

```
(%i18) diff(%o3, cr);
```

(%o18) $$0$$

```
(%i19) diff(%o3, ci);
```

(%o19) $$0$$

APPENDIX E

MAXIMA CALCULATIONS FOR FINDING DERIVATIVES OF Z_{14} EQUIVALENT IMPEDANCES OF THE X-Δ MODEL

```
(%i1) ((tr+cr+%i*(ti+ci))*(dr+sr+%i*(di+si))+(tr+%i*ti)*(cr+%i*ci)+2*(dr+%i*di)*(sr+%i*si))/(tr
 +dr+sr+%i*(ti+di+si));
```

(%o1)
$$\frac{(sr + i\ (si + di) + dr)\ (tr + i\ (ti + ci) + cr) + (cr + i\,ci)\ (tr + i\,ti) + 2\ (dr + i\,di)\ (sr + i\,si)}{tr + i\ (ti + si + di) + sr + dr}$$

```
(%i2) realpart(%o1);
```

(%o2)
$$\Big((tr + sr + dr)\ ((sr + dr)\ (tr + cr) + cr\,tr - (si + di)\ (ti + ci) - ci\,ti + 2\ (dr\,sr - di\,si)) -$$
$$(-ti - si - di)\ ((si + di)\ (tr + cr) + ci\,tr + (sr + dr)\ (ti + ci) + cr\,ti + 2\ (di\,sr + dr\,si)) \Big) /$$
$$\Big((tr + sr + dr)^2 + (ti + si + di)^2 \Big)$$

```
(%i3) imagpart(%o1);
```

(%o3)
$$\Big((-ti - si - di)\ ((sr + dr)\ (tr + cr) + cr\,tr - (si + di)\ (ti + ci) - ci\,ti + 2\ (dr\,sr - di\,si)) +$$
$$(tr + sr + dr)\ ((si + di)\ (tr + cr) + ci\,tr + (sr + dr)\ (ti + ci) + cr\,ti + 2\ (di\,sr + dr\,si)) \Big) /$$
$$\Big((tr + sr + dr)^2 + (ti + si + di)^2 \Big)$$

```
(%i4) diff(%o2, tr);
```

(%o4)
$$\frac{(sr + dr + cr)\ (tr + sr + dr) + (sr + dr)\ (tr + cr) + cr\,tr - (si + di)\ (ti + ci) - ci\,ti - (si + di + ci)\ (-ti - si - di) + 2\ (dr\,sr - di\,si)}{(tr + sr + dr)^2 + (ti + si + di)^2} -$$
$$\Big(2\ (tr + sr + dr)\ ((tr + sr + dr)\ ((sr + dr)\ (tr + cr) + cr\,tr - (si + di)\ (ti + ci) - ci\,ti + 2\ (dr\,sr - di\,si)) -$$
$$(-ti - si - di)\ ((si + di)\ (tr + cr) + ci\,tr + (sr + dr)\ (ti + ci) + cr\,ti + 2\ (di\,sr + dr\,si))) \Big) /$$
$$\Big((tr + sr + dr)^2 + (ti + si + di)^2 \Big)^2$$

```
(%i5) diff(%o2, ti);
```

(%o5)

$$\frac{(-si-di-ci)\;(tr+sr+dr)+(si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti-(sr+dr+cr)\;(-ti-si-di)+2\;(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((sr+dr)\;(tr+cr)+cr\,tr-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si))-$$

$$(-ti-si-di)\;((si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i6) diff(%o2, dr);
```

(%o6)

$$\frac{(tr+sr+dr)\;(tr+2\,sr+cr)+(sr+dr)\;(tr+cr)+cr\,tr-(-ti-si-di)\;(ti+2\,si+ci)-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((tr+sr+dr)\;((sr+dr)\;(tr+cr)+cr\,tr-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si))-$$

$$(-ti-si-di)\;((si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i7) diff(%o2, di);
```

(%o7)

$$\frac{-(-ti-si-di)\;(tr+2\,sr+cr)+(-ti-2\,si-ci)\;(tr+sr+dr)+(si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((sr+dr)\;(tr+cr)+cr\,tr-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si))-$$

$$(-ti-si-di)\;((si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i8) diff(%o2, sr);
```

(%o8)

$$\frac{(tr+2\,dr+cr)\;(tr+sr+dr)+(sr+dr)\;(tr+cr)+cr\,tr-(-ti-si-di)\;(ti+2\,di+ci)-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((tr+sr+dr)\;((sr+dr)\;(tr+cr)+cr\,tr-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si))-$$

$$(-ti-si-di)\;((si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i9) diff(%o2, si);
```

(%o9)

$$\frac{(-ti-2\,di-ci)\;(tr+sr+dr)-(-ti-si-di)\;(tr+2\,dr+cr)+(si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((tr+sr+dr)\;((sr+dr)\;(tr+cr)+cr\,tr-(si+di)\;(ti+ci)-ci\,ti+2\;(dr\,sr-di\,si))-$$

$$(-ti-si-di)\;((si+di)\;(tr+cr)+ci\,tr+(sr+dr)\;(ti+ci)+cr\,ti+2\;(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i10) diff(%o2, cr);
```

(%o10)

$$\frac{(tr+sr+dr)^2-(-ti-si-di)\;(ti+si+di)}{(tr+sr+dr)^2+(ti+si+di)^2}$$

```
(%i11) diff(%o2, ci);
```

(%o11) 0

```
(%i12) diff(%o3, tr);
```

(%o12)
$$\frac{(si+di+ci)\,(tr+sr+dr)+(si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+(sr+dr+cr)\,(-ti-si-di)+2\,(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i13) diff(%o3, ti);
```

(%o13)
$$\frac{(sr+dr+cr)\,(tr+sr+dr)-(sr+dr)\,(tr+cr)-cr\,tr+(si+di)\,(ti+ci)+ci\,ti+(-si-di-ci)\,(-ti-si-di)-2\,(dr\,sr-di\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i14) diff(%o3, dr);
```

(%o14)
$$\frac{(-ti-si-di)\,(tr+2\,sr+cr)+(ti+2\,si+ci)\,(tr+sr+dr)+(si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i15) diff(%o3, di);
```

(%o15)
$$\frac{(tr+sr+dr)\,(tr+2\,sr+cr)-(sr+dr)\,(tr+cr)-cr\,tr+(si+di)\,(ti+ci)+ci\,ti+(-ti-2\,si-ci)\,(-ti-si-di)-2\,(dr\,sr-di\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i16) diff(%o3, sr);
```

(%o16)
$$\frac{(ti+2\,di+ci)\,(tr+sr+dr)+(-ti-si-di)\,(tr+2\,dr+cr)+(si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(tr+sr+dr)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i17) diff(%o3, si);
```

(%o17)

$$\frac{(tr+2\,dr+cr)\,(tr+sr+dr)-(sr+dr)\,(tr+cr)-cr\,tr+(si+di)\,(ti+ci)+ci\,ti+(-ti-2\,di-ci)\,(-ti-si-di)-2\,(dr\,sr-di\,si)}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((-ti-si-di)\,((sr+dr)\,(tr+cr)+cr\,tr-(si+di)\,(ti+ci)-ci\,ti+2\,(dr\,sr-di\,si))+$$

$$(tr+sr+dr)\,((si+di)\,(tr+cr)+ci\,tr+(sr+dr)\,(ti+ci)+cr\,ti+2\,(di\,sr+dr\,si)))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i18) diff(%o3, cr);
```

(%o18)
$$\frac{(ti+si+di)\,(tr+sr+dr)+(-ti-si-di)\,(tr+sr+dr)}{(tr+sr+dr)^2+(ti+si+di)^2}$$

```
(%i19) diff(%o3, ci);
```

(%o19)
$$\frac{(tr+sr+dr)^2+(-ti-si-di)^2}{(tr+sr+dr)^2+(ti+si+di)^2}$$

APPENDIX F

MAXIMA CALCULATIONS FOR FINDING DERIVATIVES OF Z_{34} EQUIVALENT IMPEDANCES OF THE X-Δ MODEL

```
(%i1) ((tr+sr+%i*(ti+si))*(2*dr+cr+%i*(2*di+ci))+(dr+%i*di)*(cr+%i*ci))/(tr+dr+sr+%i*(ti
 +di+si));
```

(%o1)

$$\frac{(2\,dr+i\,(2\,di+ci)+cr)\,(tr+i\,(ti+si)+sr)+(cr+i\,ci)\,(dr+i\,di)}{tr+i\,(ti+si+di)+sr+dr}$$

```
(%i2) realpart(%o1);
```

(%o2)

$$\Big((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$
$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di)\Big)/$$
$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)$$

```
(%i3) imagpart(%o1);
```

(%o3)

$$\Big((-ti-si-di)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)+$$
$$(tr+sr+dr)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di)\Big)/$$
$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)$$

```
(%i4) diff(%o2, tr);
```

(%o4)

$$\frac{(2\,dr+cr)\,(tr+sr+dr)+(2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)-(2\,di+ci)\,(-ti-si-di)+cr\,dr-ci\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$
$$\Big(2\,(tr+sr+dr)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$
$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$
$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i5) diff(%o2, ti);
```

(%o5) $$\frac{(-2\,di-ci)\,(tr+sr+dr)+(2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)-(2\,dr+cr)\,(-ti-si-di)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$

$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i6) diff(%o2, dr);
```

(%o6) $$\frac{(tr+sr+dr)\,(2\,(tr+sr)+cr)+(2\,dr+cr)\,(tr+sr)-(-ti-si-di)\,(2\,(ti+si)+ci)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(tr+sr+dr)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$

$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i7) diff(%o2, di);
```

(%o7) $$\frac{-(-ti-si-di)\,(2\,(tr+sr)+cr)+(-2\,(ti+si)-ci)\,(tr+sr+dr)+(2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$

$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i8) diff(%o2, sr);
```

(%o8) $$\frac{(2\,dr+cr)\,(tr+sr+dr)+(2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)-(2\,di+ci)\,(-ti-si-di)+cr\,dr-ci\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(tr+sr+dr)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$

$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i9) diff(%o2, si);
```

(%o9) $$\frac{(-2\,di-ci)\,(tr+sr+dr)+(2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)-(2\,dr+cr)\,(-ti-si-di)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\,(ti+si+di)\,((tr+sr+dr)\,((2\,dr+cr)\,(tr+sr)-(2\,di+ci)\,(ti+si)+cr\,dr-ci\,di)-$$

$$(-ti-si-di)\,((2\,di+ci)\,(tr+sr)+(2\,dr+cr)\,(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\left((tr+sr+dr)^2+(ti+si+di)^2\right)^2$$

```
(%i10) diff(%o2, cr);
```

(%o10) $$\frac{(tr+sr+dr)^2-(-ti-si-di)\,(ti+si+di)}{(tr+sr+dr)^2+(ti+si+di)^2}$$

```
(%i11) diff(%o2, ci);
```

(%o11) $$0$$

```
(%i12) diff(%o3, tr);
```

(%o12)

$$\frac{(2\,di+ci)\;(tr+sr+dr)+(2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+(2\,dr+cr)\;(-ti-si-di)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((-ti-si-di)\;((2\,dr+cr)\;(tr+sr)-(2\,di+ci)\;(ti+si)+cr\,dr-ci\,di)+$$

$$(tr+sr+dr)\;((2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i13) diff(%o3, ti);
```

(%o13)

$$\frac{(2\,dr+cr)\;(tr+sr+dr)-(2\,dr+cr)\;(tr+sr)+(2\,di+ci)\;(ti+si)+(-2\,di-ci)\;(-ti-si-di)-cr\,dr+ci\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((-ti-si-di)\;((2\,dr+cr)\;(tr+sr)-(2\,di+ci)\;(ti+si)+cr\,dr-ci\,di)+$$

$$(tr+sr+dr)\;((2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i14) diff(%o3, dr);
```

(%o14)

$$\frac{(-ti-si-di)\;(2\;(tr+sr)+cr)+(2\;(ti+si)+ci)\;(tr+sr+dr)+(2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((-ti-si-di)\;((2\,dr+cr)\;(tr+sr)-(2\,di+ci)\;(ti+si)+cr\,dr-ci\,di)+$$

$$(tr+sr+dr)\;((2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i15) diff(%o3, di);
```

(%o15)

$$\frac{(tr+sr+dr)\;(2\;(tr+sr)+cr)-(2\,dr+cr)\;(tr+sr)+(-ti-si-di)\;(-2\;(ti+si)-ci)+(2\,di+ci)\;(ti+si)-cr\,dr+ci\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(ti+si+di)\;((-ti-si-di)\;((2\,dr+cr)\;(tr+sr)-(2\,di+ci)\;(ti+si)+cr\,dr-ci\,di)+$$

$$(tr+sr+dr)\;((2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i16) diff(%o3, sr);
```

(%o16)

$$\frac{(2\,di+ci)\;(tr+sr+dr)+(2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+(2\,dr+cr)\;(-ti-si-di)+ci\,dr+cr\,di}{(tr+sr+dr)^2+(ti+si+di)^2}-$$

$$\Big(2\;(tr+sr+dr)\;((-ti-si-di)\;((2\,dr+cr)\;(tr+sr)-(2\,di+ci)\;(ti+si)+cr\,dr-ci\,di)+$$

$$(tr+sr+dr)\;((2\,di+ci)\;(tr+sr)+(2\,dr+cr)\;(ti+si)+ci\,dr+cr\,di))\Big)/$$

$$\Big((tr+sr+dr)^2+(ti+si+di)^2\Big)^2$$

```
(%i17) diff(%o3, si);
```

(%o17)

$$\frac{\left(2\,dr+cr\right)\,\left(tr+sr+dr\right)-\left(2\,dr+cr\right)\,\left(tr+sr\right)+\left(2\,di+ci\right)\,\left(ti+si\right)+\left(-2\,di-ci\right)\,\left(-ti-si-di\right)-cr\,dr+ci\,di}{\left(tr+sr+dr\right)^2+\left(ti+si+di\right)^2}-$$

$$\Big(2\,\left(ti+si+di\right)\,\left(\left(-ti-si-di\right)\,\left(\left(2\,dr+cr\right)\,\left(tr+sr\right)-\left(2\,di+ci\right)\,\left(ti+si\right)+cr\,dr-ci\,di\right)+$$

$$\left(tr+sr+dr\right)\,\left(\left(2\,di+ci\right)\,\left(tr+sr\right)+\left(2\,dr+cr\right)\,\left(ti+si\right)+ci\,dr+cr\,di\right)\right)\Big)/$$

$$\Big(\left(tr+sr+dr\right)^2+\left(ti+si+di\right)^2\Big)^2$$

```
(%i18) diff(%o3, cr);
```

(%o18)

$$\frac{\left(ti+si+di\right)\,\left(tr+sr+dr\right)+\left(-ti-si-di\right)\,\left(tr+sr+dr\right)}{\left(tr+sr+dr\right)^2+\left(ti+si+di\right)^2}$$

```
(%i19) diff(%o3, ci);
```

(%o19)

$$\frac{\left(tr+sr+dr\right)^2+\left(-ti-si-di\right)^2}{\left(tr+sr+dr\right)^2+\left(ti+si+di\right)^2}$$

APPENDIX G

MATLAB CODE FOR CALCULATING X-Δ MODEL PARAMETERS FROM IMPEDANCE MEASUREMENTS

```
% CalcXYModelParams.m

% Find parameters of an X-Y impedance model using
% an optimization algorithm and parameters from an X impedance model as
% starting points

% only use skin1 - skin6 for the VNA calculations
% all measurements are from abdominal skin
%
% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm
% skin7 = dead scalp, no thickness measurement
% skin8 = live scalp, no thickness measurement, looks like 7+ mm from pictures
% skin9 = dead ear, ~3+ mm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all


% weights for weighted multi-objective optimization
% if they're too high (above 1 or so?), the optimization can diverge from
% the starting point, not properly converging on a solution
w = [ 0.25 ; 0.25 ; 0.25 ; 0.25 ];
%w = [ 0.5 ; 0.5 ; 0.5 ; 0.5 ];
%w = [ 0.75 ; 0.75 ; 0.75 ; 0.75 ];
%w = [ 1 ; 1 ; 1 ; 1 ];


% interval size variable for CG method
ab = [ 1E-6 ; 1E3 ];

% interval search stop width for CG method
intvl = 1E-2;

% stopping criterion
e = 1E-3;      % used as derivative limit for CG method
%e = 1E-3;       % used as variance limit for GL method

% maximum number of algorithm cycles
stopCycles = 100;       % used for CG method
%stopCycles = 100;        % used for GL method

% maximum number of points to search at each iteration for GL method
% numSrchPnts = 3000;
% load impedance measurements
loadZmeas

% calculate X-model parameters
[Zdx1,Zsx1,Zcx1,normError12x1,normError13x1,normError14x1,normError34x1] = CalcXModelParams(freq10,Z12skin1,Z13skin1,Z14skin1,Z34skin1);
[Zdx2,Zsx2,Zcx2,normError12x2,normError13x2,normError14x2,normError34x2] = CalcXModelParams(freq10,Z12skin2,Z13skin2,Z14skin2,Z34skin2);
[Zdx3,Zsx3,Zcx3,normError12x3,normError13x3,normError14x3,normError34x3] = CalcXModelParams(freq10,Z12skin3,Z13skin3,Z14skin3,Z34skin3);
[Zdx4,Zsx4,Zcx4,normError12x4,normError13x4,normError14x4,normError34x4] = CalcXModelParams(freq10,Z12skin4,Z13skin4,Z14skin4,Z34skin4);
[Zdx5,Zsx5,Zcx5,normError12x5,normError13x5,normError14x5,normError34x5] = CalcXModelParams(freq10,Z12skin5,Z13skin5,Z14skin5,Z34skin5);
[Zdx6,Zsx6,Zcx6,normError12x6,normError13x6,normError14x6,normError34x6] = CalcXModelParams(freq10,Z12skin6,Z13skin6,Z14skin6,Z34skin6);
[Zdx7,Zsx7,Zcx7,normError12x7,normError13x7,normError14x7,normError34x7] = CalcXModelParams(freq1,Z12skin7,Z13skin7,Z14skin7,Z34skin7);
[Zdx8,Zsx8,Zcx8,normError12x8,normError13x8,normError14x8,normError34x8] = CalcXModelParams(freq1,Z12skin8,Z13skin8,Z14skin8,Z34skin8);
[Zdx9,Zsx9,Zcx9,normError12x9,normError13x9,normError14x9,normError34x9] = CalcXModelParams(freq1,Z12skin9,Z13skin9,Z14skin9,Z34skin9);


%%% skin 1
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin1(k)) ; imag(Z13skin1(k)) ; real(0.5*Z34skin1(k)) ; imag(0.5*Z34skin1(k)) ;
        real(0.5*Z12skin1(k)) ; imag(0.5*Z12skin1(k)) ; real(0.5*Z14skin1(k)) ; imag(0.5*Z14skin1(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin1(k) ; Z13skin1(k) ; Z14skin1(k) ; Z34skin1(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin1(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin1(k));
        normError13xy = abs(Z13skin1(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin1(k));
        normError14xy = abs(Z14skin1(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin1(k));
        normError34xy = abs(Z34skin1(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin1(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin1(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin1(k)) ];
        normError13xy = [ normError13xy abs(Z13skin1(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin1(k)) ];
        normError14xy = [ normError14xy abs(Z14skin1(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin1(k)) ];
        normError34xy = [ normError34xy abs(Z34skin1(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin1(k)) ];
    end
end

Ztxy1 = Ztxy;
Zdxy1 = Zdxy;
Zsxy1 = Zsxy;
Zcxy1 = Zcxy;
normError12xy1 = normError12xy;
normError13xy1 = normError13xy;
normError14xy1 = normError14xy;
normError34xy1 = normError34xy;
fTracelength1 = fTracelength;
cTrace1 = cTrace;
mTrace1 = mTrace;
```

```
%%% skin 2
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin2(k)) ; imag(Z13skin2(k)) ; real(0.5*Z34skin2(k)) ; imag(0.5*Z34skin2(k)) ;
        real(0.5*Z12skin2(k)) ; imag(0.5*Z12skin2(k)) ; real(0.5*Z14skin2(k)) ; imag(0.5*Z14skin2(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin2(k) ; Z13skin2(k) ; Z14skin2(k) ; Z34skin2(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin2(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin2(k));
        normError13xy = abs(Z13skin2(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin2(k));
        normError14xy = abs(Z14skin2(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin2(k));
        normError34xy = abs(Z34skin2(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin2(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin2(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin2(k)) ];
        normError13xy = [ normError13xy abs(Z13skin2(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin2(k)) ];
        normError14xy = [ normError14xy abs(Z14skin2(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin2(k)) ];
        normError34xy = [ normError34xy abs(Z34skin2(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin2(k)) ];
    end
end

Ztxy2 = Ztxy;
Zdxy2 = Zdxy;
Zsxy2 = Zsxy;
Zcxy2 = Zcxy;
normError12xy2 = normError12xy;
normError13xy2 = normError13xy;
normError14xy2 = normError14xy;
normError34xy2 = normError34xy;
fTracelength2 = fTracelength;
cTrace2 = cTrace;
mTrace2 = mTrace;


%%% skin 3
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin3(k)) ; imag(Z13skin3(k)) ; real(0.5*Z34skin3(k)) ; imag(0.5*Z34skin3(k)) ;
        real(0.5*Z12skin3(k)) ; imag(0.5*Z12skin3(k)) ; real(0.5*Z14skin3(k)) ; imag(0.5*Z14skin3(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin3(k) ; Z13skin3(k) ; Z14skin3(k) ; Z34skin3(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);
    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin3(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin3(k));
        normError13xy = abs(Z13skin3(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin3(k));
        normError14xy = abs(Z14skin3(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin3(k));
        normError34xy = abs(Z34skin3(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin3(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin3(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin3(k)) ];
        normError13xy = [ normError13xy abs(Z13skin3(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin3(k)) ];
        normError14xy = [ normError14xy abs(Z14skin3(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin3(k)) ];
        normError34xy = [ normError34xy abs(Z34skin3(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin3(k)) ];
    end
end

Ztxy3 = Ztxy;
Zdxy3 = Zdxy;
Zsxy3 = Zsxy;
Zcxy3 = Zcxy;
normError12xy3 = normError12xy;
normError13xy3 = normError13xy;
normError14xy3 = normError14xy;
normError34xy3 = normError34xy;
fTracelength3 = fTracelength;
cTrace3 = cTrace;
mTrace3 = mTrace;


%%% skin 4
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin4(k)) ; imag(Z13skin4(k)) ; real(0.5*Z34skin4(k)) ; imag(0.5*Z34skin4(k)) ;
        real(0.5*Z12skin4(k)) ; imag(0.5*Z12skin4(k)) ; real(0.5*Z14skin4(k)) ; imag(0.5*Z14skin4(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin4(k) ; Z13skin4(k) ; Z14skin4(k) ; Z34skin4(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin4(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin4(k));
        normError13xy = abs(Z13skin4(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin4(k));
        normError14xy = abs(Z14skin4(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin4(k));
        normError34xy = abs(Z34skin4(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin4(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin4(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin4(k)) ];
        normError13xy = [ normError13xy abs(Z13skin4(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin4(k)) ];
        normError14xy = [ normError14xy abs(Z14skin4(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin4(k)) ];
        normError34xy = [ normError34xy abs(Z34skin4(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin4(k)) ];
    end
end

Ztxy4 = Ztxy;
Zdxy4 = Zdxy;
Zsxy4 = Zsxy;
Zcxy4 = Zcxy;
normError12xy4 = normError12xy;
normError13xy4 = normError13xy;
normError14xy4 = normError14xy;
normError34xy4 = normError34xy;
fTracelength4 = fTracelength;
cTrace4 = cTrace;
mTrace4 = mTrace;


%%% skin 5
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin5(k)) ; imag(Z13skin5(k)) ; real(0.5*Z34skin5(k)) ; imag(0.5*Z34skin5(k)) ;
        real(0.5*Z12skin5(k)) ; imag(0.5*Z12skin5(k)) ; real(0.5*Z14skin5(k)) ; imag(0.5*Z14skin5(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin5(k) ; Z13skin5(k) ; Z14skin5(k) ; Z34skin5(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin5(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin5(k));
        normError13xy = abs(Z13skin5(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin5(k));
        normError14xy = abs(Z14skin5(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin5(k));
        normError34xy = abs(Z34skin5(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin5(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin5(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin5(k)) ];
        normError13xy = [ normError13xy abs(Z13skin5(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin5(k)) ];
        normError14xy = [ normError14xy abs(Z14skin5(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin5(k)) ];
        normError34xy = [ normError34xy abs(Z34skin5(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin5(k)) ];
    end
end

Ztxy5 = Ztxy;
Zdxy5 = Zdxy;
Zsxy5 = Zsxy;
Zcxy5 = Zcxy;
normError12xy5 = normError12xy;
normError13xy5 = normError13xy;
normError14xy5 = normError14xy;
normError34xy5 = normError34xy;
fTracelength5 = fTracelength;
cTrace5 = cTrace;
mTrace5 = mTrace;


%%% skin 6
for k=1:length(freq10)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin6(k)) ; imag(Z13skin6(k)) ; real(0.5*Z34skin6(k)) ; imag(0.5*Z34skin6(k)) ;
        real(0.5*Z12skin6(k)) ; imag(0.5*Z12skin6(k)) ; real(0.5*Z14skin6(k)) ; imag(0.5*Z14skin6(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin6(k) ; Z13skin6(k) ; Z14skin6(k) ; Z34skin6(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin6(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin6(k));
        normError13xy = abs(Z13skin6(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin6(k));
        normError14xy = abs(Z14skin6(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin6(k));
        normError34xy = abs(Z34skin6(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin6(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin6(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin6(k)) ];
        normError13xy = [ normError13xy abs(Z13skin6(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin6(k)) ];
        normError14xy = [ normError14xy abs(Z14skin6(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin6(k)) ];
        normError34xy = [ normError34xy abs(Z34skin6(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin6(k)) ];
    end
end

Ztxy6 = Ztxy;
Zdxy6 = Zdxy;
```

```
Zsxy6 = Zsxy;
Zcxy6 = Zcxy;
normError12xy6 = normError12xy;
normError13xy6 = normError13xy;
normError14xy6 = normError14xy;
normError34xy6 = normError34xy;
fTracelength6 = fTracelength;
cTrace6 = cTrace;
mTrace6 = mTrace;


%%% skin 7
for k=1:length(freq1)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin7(k)) ; imag(Z13skin7(k)) ; real(0.25*Z34skin7(k)) ; imag(0.25*Z34skin7(k)) ;
        real(0.5*Z12skin7(k)) ; imag(0.5*Z12skin7(k)) ; real(0.5*Z34skin7(k)) ; imag(0.5*Z34skin7(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin7(k) ; Z13skin7(k) ; Z14skin7(k) ; Z34skin7(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin7(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin7(k));
        normError13xy = abs(Z13skin7(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin7(k));
        normError14xy = abs(Z14skin7(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin7(k));
        normError34xy = abs(Z34skin7(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin7(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin7(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin7(k)) ];
        normError13xy = [ normError13xy abs(Z13skin7(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin7(k)) ];
        normError14xy = [ normError14xy abs(Z14skin7(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin7(k)) ];
        normError34xy = [ normError34xy abs(Z34skin7(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin7(k)) ];
    end
end

Ztxy7 = Ztxy;
Zdxy7 = Zdxy;
Zsxy7 = Zsxy;
Zcxy7 = Zcxy;
normError12xy7 = normError12xy;
normError13xy7 = normError13xy;
normError14xy7 = normError14xy;
normError34xy7 = normError34xy;
fTracelength7 = fTracelength;
cTrace7 = cTrace;
mTrace7 = mTrace;


%%% skin 8
for k=1:length(freq1)
```

```
    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin8(k)) ; imag(Z13skin8(k)) ; real(0.25*Z34skin8(k)) ; imag(0.25*Z34skin8(k)) ;
        real(0.5*Z12skin8(k)) ; imag(0.5*Z12skin8(k)) ; real(0.5*Z34skin8(k)) ; imag(0.5*Z34skin8(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin8(k) ; Z13skin8(k) ; Z14skin8(k) ; Z34skin8(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin8(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin8(k));
        normError13xy = abs(Z13skin8(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin8(k));
        normError14xy = abs(Z14skin8(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin8(k));
        normError34xy = abs(Z34skin8(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin8(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin8(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin8(k)) ];
        normError13xy = [ normError13xy abs(Z13skin8(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin8(k)) ];
        normError14xy = [ normError14xy abs(Z14skin8(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin8(k)) ];
        normError34xy = [ normError34xy abs(Z34skin8(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin8(k)) ];
    end
end

Ztxy8 = Ztxy;
Zdxy8 = Zdxy;
Zsxy8 = Zsxy;
Zcxy8 = Zcxy;
normError12xy8 = normError12xy;
normError13xy8 = normError13xy;
normError14xy8 = normError14xy;
normError34xy8 = normError34xy;
fTracelength8 = fTracelength;
cTrace8 = cTrace;
mTrace8 = mTrace;


%%% skin 9
for k=1:length(freq1)

    % initial guesses for the model parameters
    tdsc0 = [ real(Z13skin9(k)) ; imag(Z13skin9(k)) ; real(0.25*Z34skin9(k)) ; imag(0.25*Z34skin9(k)) ;
        real(0.5*Z12skin9(k)) ; imag(0.5*Z12skin9(k)) ; real(0.5*Z34skin9(k)) ; imag(0.5*Z34skin9(k)) ];

    % measured impedances at the indexed frequency
    Z = [ Z12skin9(k) ; Z13skin9(k) ; Z14skin9(k) ; Z34skin9(k) ];

    % use algorithm to find optimal parameters to fit the measurements
    [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
    [c,m] = min(fTrace);

    % record parameters for this frequency
    if (k==1)
        cTrace = c;
        mTrace = m;
        fTracelength = length(fTrace);

        Ztxy = tdsc(1,m)+1i*tdsc(2,m);
        Zdxy = tdsc(3,m)+1i*tdsc(4,m);
        Zsxy = tdsc(5,m)+1i*tdsc(6,m);
        Zcxy = tdsc(7,m)+1i*tdsc(8,m);
        normError12xy = abs(Z12skin9(k) - ( ( (Ztxy+Zdxy)*(2*Zsxy+Zcxy)+Zsxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z12skin9(k));
        normError13xy = abs(Z13skin9(k) - ( ( Ztxy*(Zdxy+Zsxy) ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z13skin9(k));
        normError14xy = abs(Z14skin9(k) - ( ( (Ztxy+Zcxy)*(Zdxy+Zsxy)+Ztxy*Zcxy+2*Zdxy*Zsxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z14skin9(k));
        normError34xy = abs(Z34skin9(k) - ( ( (Ztxy+Zsxy)*(2*Zdxy+Zcxy)+Zdxy*Zcxy ) / ( Ztxy+Zdxy+Zsxy ) ))/abs(Z34skin9(k));
    else
        cTrace = [ cTrace c ];
        mTrace = [ mTrace m ];
        fTracelength = [ fTracelength ; length(fTrace) ];

        Ztxy = [ Ztxy tdsc(1,m)+1i*tdsc(2,m) ];
        Zdxy = [ Zdxy tdsc(3,m)+1i*tdsc(4,m) ];
        Zsxy = [ Zsxy tdsc(5,m)+1i*tdsc(6,m) ];
        Zcxy = [ Zcxy tdsc(7,m)+1i*tdsc(8,m) ];
        normError12xy = [ normError12xy abs(Z12skin9(k) - ( ( (Ztxy(length(Ztxy))+Zdxy(length(Zdxy)))*(2*Zsxy(length(Zsxy))+
Zcxy(length(Zcxy)))+Zsxy(length(Zsxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z12skin9(k)) ];
        normError13xy = [ normError13xy abs(Z13skin9(k) - ( ( Ztxy(length(Ztxy))*(Zdxy(length(Zdxy))+Zsxy(length(Zsxy))) ) /
 ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z13skin9(k)) ];
        normError14xy = [ normError14xy abs(Z14skin9(k) - ( ( (Ztxy(length(Ztxy))+Zcxy(length(Zcxy)))*(Zdxy(length(Zdxy))+
Zsxy(length(Zsxy)))+Ztxy(length(Ztxy))*Zcxy(length(Zcxy))+2*Zdxy(length(Zdxy))*Zsxy(length(Zsxy)) ) / ( Ztxy(length(Ztxy))+
Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/abs(Z14skin9(k)) ];
        normError34xy = [ normError34xy abs(Z34skin9(k) - ( ( (Ztxy(length(Ztxy))+Zsxy(length(Zsxy)))*(2*Zdxy(length(Zdxy))+
Zcxy(length(Zcxy)))+Zdxy(length(Zdxy))*Zcxy(length(Zcxy)) ) / ( Ztxy(length(Ztxy))+Zdxy(length(Zdxy))+Zsxy(length(Zsxy)) ) ))/
abs(Z34skin9(k)) ];
    end
end

Ztxy9 = Ztxy;
Zdxy9 = Zdxy;
Zsxy9 = Zsxy;
Zcxy9 = Zcxy;
normError12xy9 = normError12xy;
normError13xy9 = normError13xy;
normError14xy9 = normError14xy;
normError34xy9 = normError34xy;
fTracelength9 = fTracelength;
cTrace9 = cTrace;
mTrace9 = mTrace;


%%% Additional post-processing for cleaning data outliers and comparing models
for j=1:5
    CleanXYModelParams;
end
CalcXParamsFromXY
```

APPENDIX H

MATLAB CODE FOR CALCULATING X MODEL PARAMETERS FROM IMPEDANCE MEASUREMENTS

```
% CalcXModelParams.m

% Calculate parameters of an X impedance model and the associated errors
% between actual measurements and parameters

% only use skin1 - skin6 for the high frequency calculations
%
% only use skin1 - skin6 for the VNA calculations
% all measurements are from abdominal skin
%
% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Zdx,Zsx,Zcx,normError12x,normError13x,normError14x,normError34x] = CalcXModelParams(freq,Z12,Z13,Z14,Z34)

%%% Z12 through Z34 are vectors containing impedance measurements for the
%%% desired piece of skin
%%% freq is a vector containing frequency values for each measurement

%%% used to be called in standalone function
% close all


% load impedance measurements
%%% used to be used in standalone function
% loadZmeas

% inverted matrix to multiply by the measurements
% only 3 impedance vectors are used of the 4, these can be manually changed
% if desired.  I've just been using Z12, Z13, and Z34
A = [ 2 0 1 ; 1 1 0 ; 0 2 1 ];     % [ S ; D ; C ] , [ Z12 ; Z13 ; Z34 ]
Ainv = [ 0.25 0.5 -0.25 ; -0.25 0.5 0.25 ; 0.5 -1 0.5 ];
%A = [ 2 0 1 ; 1 1 0 ; 1 1 1 ];     % [ S ; D ; C ] , [ Z12 ; Z13 ; Z14 ]
%Ainv = [ 0.5 0.5 -0.5 ; -0.5 0.5 0.5 ; 0 -1 1 ];
%A = [ 1 1 0 ; 1 1 1 ; 0 2 1 ];     % [ S ; D ; C ] , [ Z13 ; Z14 ; Z34 ]
%Ainv = [ 0.5 0.5 -0.5 ; 0.5 -0.5 0.5 ; -1 1 0 ];

for k=1:length(freq)

    % measured impedances at the indexed frequency
    Z = [ Z12(k) ; Z13(k) ; Z34(k) ];
%    Z = [ Z12(k) ; Z13(k) ; Z14(k) ];
%    Z = [ Z13(k) ; Z14(k) ; Z34(k) ];

    % calculate S,D,C
    sdc = Ainv * Z;

    % record parameters for this frequency
    if (k==1)

        Zsx = sdc(1);
        Zdx = sdc(2);
        Zcx = sdc(3);

        normError12x = abs(Z12(k) - ( 2*Zsx + Zcx ))/abs(Z12(k));
        normError13x = abs(Z13(k) - ( Zsx + Zdx ))/abs(Z13(k));
        normError14x = abs(Z14(k) - ( Zsx + Zdx + Zcx ))/abs(Z14(k));
        normError34x = abs(Z34(k) - ( 2*Zdx + Zcx ))/abs(Z34(k));

    else

        Zsx = [ Zsx sdc(1) ];
        Zdx = [ Zdx sdc(2) ];
        Zcx = [ Zcx sdc(3) ];

        normError12x = [ normError12x abs(Z12(k) - ( 2*Zsx(length(Zsx)) + Zcx(length(Zcx)) ))/abs(Z12(k)) ];
        normError13x = [ normError13x abs(Z13(k) - ( Zsx(length(Zsx)) + Zdx(length(Zdx)) ))/abs(Z13(k)) ];
        normError14x = [ normError14x abs(Z14(k) - ( Zsx(length(Zsx)) + Zdx(length(Zdx)) + Zcx(length(Zcx)) ))/abs(Z14(k)) ];
        normError34x = [ normError34x abs(Z34(k) - ( 2*Zdx(length(Zdx)) + Zcx(length(Zcx)) ))/abs(Z34(k)) ];

    end

end
```

APPENDIX I

MATLAB CODE FOR LOADING EXPERIMENTALLY MEASURED IMPEDANCE VALUES FOR CALCULATION

```
% loadZmeas.m

%%% Steven A. Hackworth
%%% load Z measurements from data files

% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm
% skin7 = dead scalp, no thickness measurement
% skin8 = live scalp, no thickness measurement, looks like 7+ mm from pictures
% skin9 = dead ear, ~3+ mm


%%% The .dat files contain 13 columns.  The first column contains the
%%% frequencies at which measurements were recorded.  The following columns
%%% contain, in order, the real and imaginary parts, respectively, of
%%% measurements between terminals 1-2, 1-3, 1-4, 3-4, 2-3, and 2-4.
%%% i.e. the second and third columns contain real and imaginary parts,
%%% respectively, of impedance measurements between terminals 1 and 2
Zrealimagskin1 = load(’Zrealimagskin1.dat’);
Zrealimagskin2 = load(’Zrealimagskin2.dat’);
Zrealimagskin3 = load(’Zrealimagskin3.dat’);
Zrealimagskin4 = load(’Zrealimagskin4.dat’);
Zrealimagskin5 = load(’Zrealimagskin5.dat’);
Zrealimagskin6 = load(’Zrealimagskin6.dat’);
Zrealimagskin7 = load(’Zrealimagskin7.dat’);
Zrealimagskin8 = load(’Zrealimagskin8.dat’);
Zrealimagskin9 = load(’Zrealimagskin9.dat’);

%%% making frequency vectors of varying sizes to ease plotting
freq1 = Zrealimagskin7(:,1);
freq10 = Zrealimagskin1(:,1);
freq100 = Zrealimagskin1(16:length(Zrealimagskin1(:,1)),1);

Z12skin1 = Zrealimagskin1(:,2) + 1i*Zrealimagskin1(:,3);
Z13skin1 = Zrealimagskin1(:,4) + 1i*Zrealimagskin1(:,5);
Z14skin1 = Zrealimagskin1(:,6) + 1i*Zrealimagskin1(:,7);
Z34skin1 = Zrealimagskin1(:,8) + 1i*Zrealimagskin1(:,9);
Z23skin1 = Zrealimagskin1(:,10) + 1i*Zrealimagskin1(:,11);
Z24skin1 = Zrealimagskin1(:,12) + 1i*Zrealimagskin1(:,13);

Z12skin2 = Zrealimagskin2(:,2) + 1i*Zrealimagskin2(:,3);
Z13skin2 = Zrealimagskin2(:,4) + 1i*Zrealimagskin2(:,5);
Z14skin2 = Zrealimagskin2(:,6) + 1i*Zrealimagskin2(:,7);
Z34skin2 = Zrealimagskin2(:,8) + 1i*Zrealimagskin2(:,9);
Z23skin2 = Zrealimagskin2(:,10) + 1i*Zrealimagskin2(:,11);
Z24skin2 = Zrealimagskin2(:,12) + 1i*Zrealimagskin2(:,13);
```

```
Z12skin3 = Zrealimagskin3(:,2) + 1i*Zrealimagskin3(:,3);
Z13skin3 = Zrealimagskin3(:,4) + 1i*Zrealimagskin3(:,5);
Z14skin3 = Zrealimagskin3(:,6) + 1i*Zrealimagskin3(:,7);
Z34skin3 = Zrealimagskin3(:,8) + 1i*Zrealimagskin3(:,9);
Z23skin3 = Zrealimagskin3(:,10) + 1i*Zrealimagskin3(:,11);
Z24skin3 = Zrealimagskin3(:,12) + 1i*Zrealimagskin3(:,13);

Z12skin4 = Zrealimagskin4(:,2) + 1i*Zrealimagskin4(:,3);
Z13skin4 = Zrealimagskin4(:,4) + 1i*Zrealimagskin4(:,5);
Z14skin4 = Zrealimagskin4(:,6) + 1i*Zrealimagskin4(:,7);
Z34skin4 = Zrealimagskin4(:,8) + 1i*Zrealimagskin4(:,9);
Z23skin4 = Zrealimagskin4(:,10) + 1i*Zrealimagskin4(:,11);
Z24skin4 = Zrealimagskin4(:,12) + 1i*Zrealimagskin4(:,13);

Z12skin5 = Zrealimagskin5(:,2) + 1i*Zrealimagskin5(:,3);
Z13skin5 = Zrealimagskin5(:,4) + 1i*Zrealimagskin5(:,5);
Z14skin5 = Zrealimagskin5(:,6) + 1i*Zrealimagskin5(:,7);
Z34skin5 = Zrealimagskin5(:,8) + 1i*Zrealimagskin5(:,9);
Z23skin5 = Zrealimagskin5(:,10) + 1i*Zrealimagskin5(:,11);
Z24skin5 = Zrealimagskin5(:,12) + 1i*Zrealimagskin5(:,13);

Z12skin6 = Zrealimagskin6(:,2) + 1i*Zrealimagskin6(:,3);
Z13skin6 = Zrealimagskin6(:,4) + 1i*Zrealimagskin6(:,5);
Z14skin6 = Zrealimagskin6(:,6) + 1i*Zrealimagskin6(:,7);
Z34skin6 = Zrealimagskin6(:,8) + 1i*Zrealimagskin6(:,9);
Z23skin6 = Zrealimagskin6(:,10) + 1i*Zrealimagskin6(:,11);
Z24skin6 = Zrealimagskin6(:,12) + 1i*Zrealimagskin6(:,13);

%%% cut out measurements below 100 KHz for various plotting
Z12skin1high = Z12skin1(16:length(Z12skin1(:,1)));
Z13skin1high = Z13skin1(16:length(Z13skin1(:,1)));
Z14skin1high = Z14skin1(16:length(Z14skin1(:,1)));
Z34skin1high = Z34skin1(16:length(Z34skin1(:,1)));
Z23skin1high = Z23skin1(16:length(Z23skin1(:,1)));
Z24skin1high = Z24skin1(16:length(Z24skin1(:,1)));

Z12skin2high = Z12skin2(16:length(Z12skin2(:,1)));
Z13skin2high = Z13skin2(16:length(Z13skin2(:,1)));
Z14skin2high = Z14skin2(16:length(Z14skin2(:,1)));
Z34skin2high = Z34skin2(16:length(Z34skin2(:,1)));
Z23skin2high = Z23skin2(16:length(Z23skin2(:,1)));
Z24skin2high = Z24skin2(16:length(Z24skin2(:,1)));

Z12skin3high = Z12skin3(16:length(Z12skin3(:,1)));
Z13skin3high = Z13skin3(16:length(Z13skin3(:,1)));
Z14skin3high = Z14skin3(16:length(Z14skin3(:,1)));
Z34skin3high = Z34skin3(16:length(Z34skin3(:,1)));
Z23skin3high = Z23skin3(16:length(Z23skin3(:,1)));
Z24skin3high = Z24skin3(16:length(Z24skin3(:,1)));

Z12skin4high = Z12skin4(16:length(Z12skin4(:,1)));
Z13skin4high = Z13skin4(16:length(Z13skin4(:,1)));
Z14skin4high = Z14skin4(16:length(Z14skin4(:,1)));
Z34skin4high = Z34skin4(16:length(Z34skin4(:,1)));
Z23skin4high = Z23skin4(16:length(Z23skin4(:,1)));
Z24skin4high = Z24skin4(16:length(Z24skin4(:,1)));

Z12skin5high = Z12skin5(16:length(Z12skin5(:,1)));
Z13skin5high = Z13skin5(16:length(Z13skin5(:,1)));
Z14skin5high = Z14skin5(16:length(Z14skin5(:,1)));
Z34skin5high = Z34skin5(16:length(Z34skin5(:,1)));
Z23skin5high = Z23skin5(16:length(Z23skin5(:,1)));
Z24skin5high = Z24skin5(16:length(Z24skin5(:,1)));

Z12skin6high = Z12skin6(16:length(Z12skin6(:,1)));
Z13skin6high = Z13skin6(16:length(Z13skin6(:,1)));
Z14skin6high = Z14skin6(16:length(Z14skin6(:,1)));
Z34skin6high = Z34skin6(16:length(Z34skin6(:,1)));
Z23skin6high = Z23skin6(16:length(Z23skin6(:,1)));
Z24skin6high = Z24skin6(16:length(Z24skin6(:,1)));


Z12skin7 = Zrealimagskin7(:,2) + 1i*Zrealimagskin7(:,3);
Z13skin7 = Zrealimagskin7(:,4) + 1i*Zrealimagskin7(:,5);
Z14skin7 = Zrealimagskin7(:,6) + 1i*Zrealimagskin7(:,7);
Z34skin7 = Zrealimagskin7(:,8) + 1i*Zrealimagskin7(:,9);
Z23skin7 = Zrealimagskin7(:,10) + 1i*Zrealimagskin7(:,11);
Z24skin7 = Zrealimagskin7(:,12) + 1i*Zrealimagskin7(:,13);

Z12skin8 = Zrealimagskin8(:,2) + 1i*Zrealimagskin8(:,3);
```

```
Z13skin8 = Zrealimagskin8(:,4) + 1i*Zrealimagskin8(:,5);
Z14skin8 = Zrealimagskin8(:,6) + 1i*Zrealimagskin8(:,7);
Z34skin8 = Zrealimagskin8(:,8) + 1i*Zrealimagskin8(:,9);
Z23skin8 = Zrealimagskin8(:,10) + 1i*Zrealimagskin8(:,11);
Z24skin8 = Zrealimagskin8(:,12) + 1i*Zrealimagskin8(:,13);

Z12skin9 = Zrealimagskin9(:,2) + 1i*Zrealimagskin9(:,3);
Z13skin9 = Zrealimagskin9(:,4) + 1i*Zrealimagskin9(:,5);
Z14skin9 = Zrealimagskin9(:,6) + 1i*Zrealimagskin9(:,7);
Z34skin9 = Zrealimagskin9(:,8) + 1i*Zrealimagskin9(:,9);
Z23skin9 = Zrealimagskin9(:,10) + 1i*Zrealimagskin9(:,11);
Z24skin9 = Zrealimagskin9(:,12) + 1i*Zrealimagskin9(:,13);


clear Zrealimagskin1
clear Zrealimagskin2
clear Zrealimagskin3
clear Zrealimagskin4
clear Zrealimagskin5
clear Zrealimagskin6
clear Zrealimagskin7
clear Zrealimagskin8
clear Zrealimagskin9
```

APPENDIX J

MATLAB CODE FOR PERFORMING THE CONJUGATE GRADIENT SEARCH METHOD

```
% CGSearchXYModelWeightedMulti.m

% Performs a conjugate gradient optimization algorithm with  step size
% determined by the golden section search method to find impedance
% parameters of an x-y impedance model given the input
% impedance measured between terminals 1-2, 1-3, 1-4, and 3-4

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [tdsc,fTrace,gradient] = CGSearchXYModelWeightedMulti(tdsc0,Z,w,ab,intvl,e,stopCycles)

% tdsc = record of progressive values traversed by the algorithm
% fTrace = record of function values at successive points
% tdsc0 = starting values for t, d, s, c (8 element vector [tr;ti;dr;di;sr;si;cr;ci])
% Z = vector of measured impedance values [Z12;Z13;Z14;Z34]
% w = vector of multi-objective weights [w12;w13;w14;w34]
% ab = 2-element positive vector with starting search interval, b>a
% intvl = interval search stop width, must be positive
% e = stopping criterion, must be positive
% stopCycles = max. # of iterations, must be positive

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%% check input variables to make sure they meet function requirements
%----------------------------
if ( length(tdsc0)~=8 )
    disp('Error in CGSearchXYModelWeightedMulti: Starting point "tdsc0" must be an 8 element column vector.');
    return;
end

if ( length(Z) ~= 4 )
    disp('Error in CGSearchXYModelWeightedMulti: Impedance measurements "Z" must be a 4 element vector.');
    return;
end

if ( length(w) ~= 4 )
    disp('Error in CGSearchXYModelWeightedMulti: Weights "w" must be a 4 element vector.');
    return;
end

if ( length(ab) ~= 2 )
    disp('Error in CGSearchXYModelWeightedMulti: Search interval "ab" must be a 2 element vector.');
    return;
end

a0 = ab(1);  b0 = ab(2);

if ( 0==a0 || 0>a0 || 0==b0 || 0>b0 )
    disp('Error in CGSearchXYModelWeightedMulti: Interval parameters "ab" must be positive.');
    return;
```

```
end

if ( a0==b0 || a0>b0 )
    disp('Error in CGSearchXYModelWeightedMulti: Interval end "b0" must be greater than interval start "a0".');
    return;
end

if ( 0==intvl || 0>intvl )
    disp('Error in CGSearchXYModelWeightedMulti: Interval search stop width "intvl" must be positive.');
    return;
end

if ( 0==e || 0>e )
    disp('Error in CGSearchXYModelWeightedMulti: Stopping criterion "e" must be positive.');
    return;
end

if ( 0==stopCycles || 0>stopCycles )
    disp('Error in CGSearchXYModelWeightedMulti: Max. # of iterations "stopCycles" must be positive.');
    return;
end
%----------------------------


% record starting point and objective function starting value
tdsc = tdsc0;
fTrace = XYModelObjFunct(tdsc0,Z,w);

% variable for counting # of iterations
n = 1;

%%% first iteration of CG (F-R) algorithm is different than those following

    % calculate gradient at current point (negative of the direction)

    [gradf,gradNorm] = XYModelGrad(tdsc,Z,w);
    d = -gradf;

    % record gradient evolution
    gradient = gradf;

    % perform interval search to find step size
    a = a0;  b = b0;
    while ( (b-a) > intvl )

        alpha1 = 0.382*(b-a) + a;
        alpha2 = 0.618*(b-a) + a;

        % calculate t,d,s,c values and corresponding function values at alpha's
        tdsc1 = tdsc(:,length(tdsc(1,:)))-alpha1*gradf;
        tdsc2 = tdsc(:,length(tdsc(1,:)))-alpha2*gradf;
        g1 = XYModelObjFunct(tdsc1,Z,w);
        g2 = XYModelObjFunct(tdsc2,Z,w);

        if ( g1 < g2 )
            b = alpha2;
        elseif ( g1 > g2 )
            a = alpha1;
        else
            a = alpha1;  b = alpha2;
        end
    end

    % append new t,d,s,c values
    tdsc=[tdsc tdsc+a*d];

    % store this iteration's values of gradient and direction
    gradfOld = gradf;
    dOld = d;

    % record objective function after first iteration
    fTrace = [ fTrace XYModelObjFunct(tdsc(:,length(tdsc(1,:))),Z,w)];


% iterations after the first
while ( n < stopCycles )

    % add 1 to the loop count
    n = n+1;
    % calculate gradient at current point

    [gradf,gradNorm] = XYModelGrad(tdsc(:,length(tdsc(1,:))),Z,w);

    % compare norm of the gradient with stopping criterion
    if ( gradNorm < e )
        break;
    end

    % record gradient evolution
    gradient = [gradient gradf];

    % calculate beta and new direction d
        % Fletcher-Reeves method for calculating beta
        beta = gradf'*gradf/(gradfOld'*gradfOld);

        % Polak-Ribiere method for calculating beta
        %beta = max(0, gradf'*(gradf-gradfOld)/(gradfOld'*gradfOld));

        if ( mod(n,length(tdsc0)) == 0 )    % "restart" the algorithm every # cycles (dimension #)
            d = -gradf;
        else
            d = -gradf + beta*dOld;
        end


    % perform interval search to find step size
    a = a0;  b = b0;
    while ( (b-a) > intvl )

        alpha1 = 0.382*(b-a) + a;
        alpha2 = 0.618*(b-a) + a;

        % calculate t,d,s values and corresponding function values at alpha's
        tdsc1 = tdsc(:,length(tdsc(1,:)))-alpha1*gradf;
        tdsc2 = tdsc(:,length(tdsc(1,:)))-alpha2*gradf;
        g1 = XYModelObjFunct(tdsc1,Z,w);
        g2 = XYModelObjFunct(tdsc2,Z,w);

        if ( g1 < g2 )
            b = alpha2;
        elseif ( g1 > g2 )
            a = alpha1;
        else
            a = alpha1;  b = alpha2;
        end
    end

    % append new t,d,s,c values
    tdsc=[tdsc tdsc(:,length(tdsc(1,:)))+a*d];

    % store this iteration's values of gradient and direction
    gradfOld = gradf;
    dOld = d;

    % record objective function after each iteration
    fTrace = [ fTrace XYModelObjFunct(tdsc(:,length(tdsc(1,:))),Z,w)];

end
```

APPENDIX K

MATLAB CODE FOR CALCULATING THE OBJECTIVE FUNCTION FOR OPTIMIZATION

```
% XYModelObjFunct.m

% Calculates the objective (error) function for the x-y model

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function objFunct = XYModelObjFunct(tdsc,Z,w)

% tdsc = current values of t,d,s,c [tr;ti;dr;di;sr;si;cr;ci]
% Z = vector of measured impedance values [Z12;Z13;Z14;Z34]
% w = vector of weights for the objective function [w12;w13;w14;w34]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%% check input variables to make sure they meet function requirements
%----------------------------
if ( length(tdsc(:,1))~=8 )
    disp('Error in XYModelObjFunct: "tdsc" must be an 8-element vector.');
    return;
end

if ( length(Z) ~= 4 )
    disp('Error in XYModelObjFunct: Impedance measurements "Z" must be a 4 element vector.');
    return;
end

if ( length(w) ~= 4 )
    disp('Error in XYModelObjFunct: Weights "w" must be a 4 element vector.');
    return;
end

%----------------------------

Z12 = Z(1);  Z13 = Z(2);  Z14 = Z(3);  Z34 = Z(4);
w12 = w(1);  w13 = w(2);  w14 = w(3);  w34 = w(4);

t = tdsc(1)+1i*tdsc(2);
d = tdsc(3)+1i*tdsc(4);
s = tdsc(5)+1i*tdsc(6);
c = tdsc(7)+1i*tdsc(8);

objFunct = w12*abs( Z12 - ( (t+d)*(2*s+c)+s*c ) / ( t+d+s ) )^2 +
 w13*abs(Z13 - ( t*(d+s) ) / ( t+d+s ) )^2 +
 w14*abs( Z14 - ( (t+c)*(d+s)+t*c+2*d*s ) / ( t+d+s ) )^2 +
 w34*abs( Z34 - ( (t+s)*(2*d+c)+d*c ) / ( t+d+s ) )^2;
```

APPENDIX L

MATLAB CODE FOR CALCULATING THE GRADIENT OF THE OBJECTIVE FUNCTION FOR OPTIMIZATION

```
% XYModelGrad.m

% Calculates the gradients for the x-y model

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [gradf,gradNorm] = XYModelGrad(tdsc,Z,w)

% gradf = column vector containing calculated gradients (eight partials)
% gradfNorm = norm of the gradient
% tdsc = current values of t,d,s,c [tr;ti;dr;di;sr;si;cr;ci]
% Z = vector of measured impedance values [Z12;Z13;Z14;Z34]
% w = vector of multi-objective weights [w12;w13;w14;w34]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%% check input variables to make sure they meet function requirements
%++++++++++++++
if ( length(tdsc)~=8 )
    disp('Error in XYModelGrad: Current position "tdsc" must be an 8 element vector.');
    return;
end

if ( length(Z) ~= 4 )
    disp('Error in XYModelGrad: Impedance measurements "Z" must be a 4 element vector.');
    return;
end

Z12 = Z(1);  Z13 = Z(2);  Z14 = Z(3);  Z34 = Z(4);

if ( length(w) ~= 4 )
    disp('Error in XYModelGrad: Weights "w" must be a 4 element vector.');
    return;
end

w12 = w(1);  w13 = w(2);  w14 = w(3);  w34 = w(4);

%++++++++++++++


    tr = tdsc(1);
    ti = tdsc(2);
    t = tr+1i*ti;
    dr = tdsc(3);
    di = tdsc(4);
    d = dr+1i*di;
    sr = tdsc(5);
    si = tdsc(6);
    s = sr+1i*si;
    cr = tdsc(7);
    ci = tdsc(8);
    c = cr+1i*ci;

        % common variables to reduce computation time

        Z12eqn = ( (t+d)*(2*s+c)+s*c ) / ( t+d+s );
        Z13eqn = ( t*(d+s) ) / ( t+d+s );
        Z14eqn = ( (t+c)*(d+s)+t*c+2*d*s ) / ( t+d+s );
        Z34eqn = ( (t+s)*(2*d+c)+d*c ) / ( t+d+s );
        diffreal12 = real(Z12) - real(Z12eqn);
        diffimag12 = imag(Z12) - imag(Z12eqn);
        diffreal13 = real(Z13) - real(Z13eqn);
        diffimag13 = imag(Z13) - imag(Z13eqn);
        diffreal14 = real(Z14) - real(Z14eqn);
        diffimag14 = imag(Z14) - imag(Z14eqn);
        diffreal34 = real(Z34) - real(Z34eqn);
        diffimag34 = imag(Z34) - imag(Z34eqn);
        % no squaring
%         derivfactor12 = -w12 / ( 2*sqrt(diffreal12^2+diffimag12^2) );
%         derivfactor13 = -w13 / ( 2*sqrt(diffreal13^2+diffimag13^2) );
%         derivfactor14 = -w14 / ( 2*sqrt(diffreal14^2+diffimag14^2) );
%         derivfactor34 = -w34 / ( 2*sqrt(diffreal34^2+diffimag34^2) );

        % squared
        derivfactor12 = -w12;
        derivfactor13 = -w13;
        derivfactor14 = -w14;
        derivfactor34 = -w34;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    rtds = (tr+sr+dr);
    itds = (ti+si+di);
    r2sc = (2*sr+cr);
    i2sc = (2*si+ci);
    rtd = (tr+dr);
    itd = (ti+di);
    itds = (ti+si+di);
    rds = (sr+dr);
    ids = (si+di);
    rtc = (tr+cr);
    itc = (ti+ci);
    srcr = cr*sr;
    sici = ci*si;
    srci = ci*sr;
    sicr = cr*si;
    drsr = dr*sr;
    disi = di*si;
    drsi = dr*si;
    disr = di*sr;
    trcr = cr*tr;
    tici = ci*ti;
    trci = ci*tr;
    ticr = cr*ti;
    drcr = cr*dr;
    dici = ci*di;
    drci = ci*dr;
    dicr = cr*di;
    r2dc = (2*dr+cr);
    i2dc = (2*di+ci);
    rts = (tr+sr);
    its = (ti+si);
    r2tds = (2*tr+sr+dr);
    i2tds = (2*ti+si+di);
    rdsc = (sr+dr+cr);
    idsc = (si+di+ci);
    rt2sc = (tr+2*sr+cr);
    it2sc = (ti+2*si+ci);
    rt2dc = (tr+2*dr+cr);
    it2dc = (ti+2*di+ci);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        denommag12 = rtds^2+itds^2;
        denommag13 = rtds^2+itds^2;
        denommag14 = rtds^2+itds^2;
        denommag34 = rtds^2+itds^2;
        numrealpart12 = rtds*(r2sc*rtd-i2sc*itd+srcr-sici)+itds*(i2sc*rtd+r2sc*itd+srci+sicr);
        numimagpart12 = -itds*(r2sc*rtd-i2sc*itd+srcr-sici)+rtds*(i2sc*rtd+r2sc*itd+srci+sicr);
        numrealpart13 = rds*(tr*rtds+itds*ti)-ids*(ti*rtds-itds*tr);
        numimagpart13 = ids*(tr*rtds+itds*ti)+rds*(ti*rtds-itds*tr);
        numrealpart14 = rtds*(rds*rtc+trcr-ids*itc-tici+2*(drsr-disi))+itds*(ids*rtc+trci+rds*itc+ticr+2*(disr+drsi));
        numimagpart14 = -itds*(rds*rtc+trcr-ids*itc-tici+2*(drsr-disi))+rtds*(ids*rtc+trci+rds*itc+ticr+2*(disr+drsi));
        numrealpart34 = rtds*(r2dc*rts-i2dc*its+drcr-dici)+itds*(i2dc*rts+r2dc*its+drci+dicr);
```

```
     numimagpart34 = -itds*(r2dc*rts-i2dc*its+drcr-dici)+rtds*(i2dc*rts+r2dc*its+drci+dicr);


   gradftr12r1 = r2sc*rtds+r2sc*rtd-i2sc*itd-i2sc*-itds+srcr-sici;
   gradftr12r2 = 2*rtds*numrealpart12;
   gradftr12i1 = i2sc*rtds+i2sc*rtd+r2sc*itd+r2sc*-itds+srci+sicr;
   gradftr12i2 = 2*rtds*numimagpart12;
   gradfti12r1 = -i2sc*rtds+i2sc*rtd+r2sc*itd-r2sc*-itds+srci+sicr;
   gradfti12r2 = 2*itds*numrealpart12;
   gradfti12i1 = r2sc*rtds-r2sc*rtd+i2sc*itd-i2sc*-itds-srcr+sici;
   gradfti12i2 = 2*itds*numimagpart12;
   gradfdr12r1 = r2sc*rtds+r2sc*rtd-i2sc*itd-i2sc*-itds+srcr-sici;
   gradfdr12r2 = 2*rtds*numrealpart12;
   gradfdr12i1 = i2sc*rtds+i2sc*rtd+r2sc*itd+r2sc*-itds+srci+sicr;
   gradfdr12i2 = 2*rtds*numimagpart12;
   gradfdi12r1 = -i2sc*rtds+i2sc*rtd+r2sc*itd-r2sc*-itds+srci+sicr;
   gradfdi12r2 = 2*itds*numrealpart12;
   gradfdi12i1 = r2sc*rtds-r2sc*rtd+i2sc*itd-i2sc*-itds-srcr+sici;
   gradfdi12i2 = 2*itds*numimagpart12;
   gradfsr12r1 = rtds*(2*rtd+cr)+r2sc*rtd+itds*(2*itd+ci)-i2sc*itd+srcr-sici;
   gradfsr12r2 = 2*rtds*numrealpart12;
   gradfsr12i1 = -itds*(2*rtd+cr)+(2*itd+ci)*rtds+i2sc*rtd+r2sc*itd+srci+sicr;
   gradfsr12i2 = 2*rtds*numimagpart12;
   gradfsi12r1 = +itds*(2*rtd+cr)-(2*itd+ci)*rtds+i2sc*rtd+r2sc*itd+srci+sicr;
   gradfsi12r2 = 2*itds*numrealpart12;
   gradfsi12i1 = rtds*(2*rtd+cr)-r2sc*rtd-itds*-(2*itd+ci)+i2sc*itd-srcr+sici;
   gradfsi12i2 = 2*itds*numimagpart12;
   gradfcr12r1 = rtds^2+itds*itds;
   gradfcr12r2 = 0;
   gradfcr12i1 = itds*rtds-itds*rtds;
   gradfcr12i2 = 0;
   gradfci12r1 = 0;
   gradfci12r2 = 0;
   gradfci12i1 = rtds^2-itds^2;
   gradfci12i2 = 0;

   gradftr12 = derivfactor12 * ( 2*diffreal12*( (gradftr12r1)/denommag12 - (gradftr12r2)/denommag12^2 ) +
2*diffimag12*( (gradftr12i1)/denommag12 - (gradftr12i2)/denommag12^2 ) );
   gradfti12 = derivfactor12 * ( 2*diffreal12*( (gradfti12r1)/denommag12 - (gradfti12r2)/denommag12^2 ) +
2*diffimag12*( (gradfti12i1)/denommag12 - (gradfti12i2)/denommag12^2 ) );
   gradfdr12 = derivfactor12 * ( 2*diffreal12*( (gradfdr12r1)/denommag12 - (gradfdr12r2)/denommag12^2 ) +
2*diffimag12*( (gradfdr12i1)/denommag12 - (gradfdr12i2)/denommag12^2 ) );
   gradfdi12 = derivfactor12 * ( 2*diffreal12*( (gradfdi12r1)/denommag12 - (gradfdi12r2)/denommag12^2 ) +
2*diffimag12*( (gradfdi12i1)/denommag12 - (gradfdi12i2)/denommag12^2 ) );
   gradfsr12 = derivfactor12 * ( 2*diffreal12*( (gradfsr12r1)/denommag12 - (gradfsr12r2)/denommag12^2 ) +
2*diffimag12*( (gradfsr12i1)/denommag12 - (gradfsr12i2)/denommag12^2 ) );
   gradfsi12 = derivfactor12 * ( 2*diffreal12*( (gradfsi12r1)/denommag12 - (gradfsi12r2)/denommag12^2 ) +
2*diffimag12*( (gradfsi12i1)/denommag12 - (gradfsi12i2)/denommag12^2 ) );
   gradfcr12 = derivfactor12 * ( 2*diffreal12*( (gradfcr12r1)/denommag12 - (gradfcr12r2)/denommag12^2 ) +
2*diffimag12*( (gradfcr12i1)/denommag12 - (gradfcr12i2)/denommag12^2 ) );
   gradfci12 = derivfactor12 * ( 2*diffreal12*( (gradfci12r1)/denommag12 - (gradfci12r2)/denommag12^2 ) +
2*diffimag12*( (gradfci12i1)/denommag12 - (gradfci12i2)/denommag12^2 ) );


   gradftr13r1 = rds*r2tds+ids*ids;
   gradftr13r2 = 2*rtds*numrealpart13;
   gradftr13i1 = ids*r2tds-ids*rds;
   gradftr13i2 = 2*rtds*numimagpart13;
   gradfti13r1 = rds*i2tds-ids*rds;
   gradfti13r2 = 2*itds*numrealpart13;
   gradfti13i1 = ids*i2tds+rds^2;
   gradfti13i2 = 2*itds*numimagpart13;
   gradfdr13r1 = tr*rtds+rds*tr+itds*ti-ids*ti;
   gradfdr13r2 = 2*rtds*numrealpart13;
   gradfdr13i1 = ti*rtds-itds*tr+ids*tr+rds*ti;
   gradfdr13i2 = 2*rtds*numimagpart13;
   gradfdi13r1 = -ti*rtds+itds*tr+ids*tr+rds*ti;
   gradfdi13r2 = 2*itds*numrealpart13;
   gradfdi13i1 = tr*rtds-rds*tr+itds*ti+ids*ti;
   gradfdi13i2 = 2*itds*numimagpart13;
   gradfsr13r1 = tr*rtds+rds*tr+itds*ti-ids*ti;
   gradfsr13r2 = 2*rtds*numrealpart13;
   gradfsr13i1 = ti*rtds-itds*tr+ids*tr+rds*ti;
   gradfsr13i2 = 2*rtds*numimagpart13;
   gradfsi13r1 = -ti*rtds+itds*tr+ids*tr+rds*ti;
   gradfsi13r2 = 2*itds*numrealpart13;
   gradfsi13i1 = tr*rtds-rds*tr+itds*ti+ids*ti;
   gradfsi13i2 = 2*itds*numimagpart13;
   gradfcr13r1 = 0;
   gradfcr13r2 = 0;
   gradfcr13i1 = 0;
   gradfcr13i2 = 0;
   gradfci13r1 = 0;
   gradfci13r2 = 0;
   gradfci13i1 = 0;
   gradfci13i2 = 0;

   gradftr13 = derivfactor13 * ( 2*diffreal13*( (gradftr13r1)/denommag13 - (gradftr13r2)/denommag13^2 ) +
2*diffimag13*( (gradftr13i1)/denommag13 - (gradftr13i2)/denommag13^2 ) );
   gradfti13 = derivfactor13 * ( 2*diffreal13*( (gradfti13r1)/denommag13 - (gradfti13r2)/denommag13^2 ) +
2*diffimag13*( (gradfti13i1)/denommag13 - (gradfti13i2)/denommag13^2 ) );
   gradfdr13 = derivfactor13 * ( 2*diffreal13*( (gradfdr13r1)/denommag13 - (gradfdr13r2)/denommag13^2 ) +
2*diffimag13*( (gradfdr13i1)/denommag13 - (gradfdr13i2)/denommag13^2 ) );
   gradfdi13 = derivfactor13 * ( 2*diffreal13*( (gradfdi13r1)/denommag13 - (gradfdi13r2)/denommag13^2 ) +
2*diffimag13*( (gradfdi13i1)/denommag13 - (gradfdi13i2)/denommag13^2 ) );
   gradfsr13 = derivfactor13 * ( 2*diffreal13*( (gradfsr13r1)/denommag13 - (gradfsr13r2)/denommag13^2 ) +
2*diffimag13*( (gradfsr13i1)/denommag13 - (gradfsr13i2)/denommag13^2 ) );
   gradfsi13 = derivfactor13 * ( 2*diffreal13*( (gradfsi13r1)/denommag13 - (gradfsi13r2)/denommag13^2 ) +
2*diffimag13*( (gradfsi13i1)/denommag13 - (gradfsi13i2)/denommag13^2 ) );
   gradfcr13 = derivfactor13 * ( 2*diffreal13*( (gradfcr13r1)/denommag13 - (gradfcr13r2)/denommag13^2 ) +
2*diffimag13*( (gradfcr13i1)/denommag13 - (gradfcr13i2)/denommag13^2 ) );
   gradfci13 = derivfactor13 * ( 2*diffreal13*( (gradfci13r1)/denommag13 - (gradfci13r2)/denommag13^2 ) +
2*diffimag13*( (gradfci13i1)/denommag13 - (gradfci13i2)/denommag13^2 ) );


   gradftr14r1 = rdsc*rtds+rds*rtc+trcr-ids*itc-tici-idsc*-itds+2*(drsr-disi);
   gradftr14r2 = 2*rtds*numrealpart14;
   gradftr14i1 = idsc*rtds+ids*rtc+trci+rds*itc+ticr+rdsc*-itds+2*(disr+drsi);
   gradftr14i2 = 2*rtds*numimagpart14;
   gradfti14r1 = -idsc*rtds+ids*rtc+trci+rds*itc+ticr-rdsc*-itds+2*(disr+drsi);
   gradfti14r2 = 2*itds*numrealpart14;
   gradfti14i1 = rdsc*rtds-rds*rtc-trcr+ids*itc+tici-idsc*-itds-2*(drsr-disi);
   gradfti14i2 = 2*itds*numimagpart14;
   gradfdr14r1 = rtds*rt2sc+rds*rtc+trcr+itds*it2sc-ids*itc-tici+2*(drsr-disi);
   gradfdr14r2 = 2*rtds*numrealpart14;
   gradfdr14i1 = -itds*rt2sc+it2sc*rtds+ids*rtc+trci+rds*itc+ticr+2*(disr+drsi);
   gradfdr14i2 = 2*rtds*numimagpart14;
   gradfdi14r1 = +itds*rt2sc-it2sc*rtds+ids*rtc+trci+rds*itc+ticr+2*(disr+drsi);
   gradfdi14r2 = 2*itds*numrealpart14;
   gradfdi14i1 = rtds*rt2sc-rds*rtc-trcr+ids*itc+tici-it2sc*-itds-2*(drsr-disi);
   gradfdi14i2 = 2*itds*numimagpart14;
   gradfsr14r1 = rt2dc*rtds+rds*rtc+trcr+itds*it2dc-ids*itc-tici+2*(drsr-disi);
   gradfsr14r2 = 2*rtds*numrealpart14;
   gradfsr14i1 = it2dc*rtds-itds*rt2dc+ids*rtc+trci+rds*itc+ticr+2*(disr+drsi);
   gradfsr14i2 = 2*rtds*numimagpart14;
   gradfsi14r1 = -it2dc*rtds+itds*rt2dc+ids*rtc+trci+rds*itc+ticr+2*(disr+drsi);
   gradfsi14r2 = 2*itds*numrealpart14;
   gradfsi14i1 = rt2dc*rtds-rds*rtc-trcr+ids*itc+tici-it2dc*-itds-2*(drsr-disi);
   gradfsi14i2 = 2*itds*numimagpart14;
   gradfcr14r1 = rtds^2+itds*itds;
   gradfcr14r2 = 0;
   gradfcr14i1 = itds*rtds-itds*rtds;
   gradfcr14i2 = 0;
   gradfci14r1 = 0;
   gradfci14r2 = 0;
   gradfci14i1 = rtds^2-itds^2;
   gradfci14i2 = 0;

   gradftr14 = derivfactor14 * ( 2*diffreal14*( (gradftr14r1)/denommag14 - (gradftr14r2)/denommag14^2 ) +
2*diffimag14*( (gradftr14i1)/denommag14 - (gradftr14i2)/denommag14^2 ) );
   gradfti14 = derivfactor14 * ( 2*diffreal14*( (gradfti14r1)/denommag14 - (gradfti14r2)/denommag14^2 ) +
2*diffimag14*( (gradfti14i1)/denommag14 - (gradfti14i2)/denommag14^2 ) );
   gradfdr14 = derivfactor14 * ( 2*diffreal14*( (gradfdr14r1)/denommag14 - (gradfdr14r2)/denommag14^2 ) +
2*diffimag14*( (gradfdr14i1)/denommag14 - (gradfdr14i2)/denommag14^2 ) );
   gradfdi14 = derivfactor14 * ( 2*diffreal14*( (gradfdi14r1)/denommag14 - (gradfdi14r2)/denommag14^2 ) +
2*diffimag14*( (gradfdi14i1)/denommag14 - (gradfdi14i2)/denommag14^2 ) );
   gradfsr14 = derivfactor14 * ( 2*diffreal14*( (gradfsr14r1)/denommag14 - (gradfsr14r2)/denommag14^2 ) +
2*diffimag14*( (gradfsr14i1)/denommag14 - (gradfsr14i2)/denommag14^2 ) );
   gradfsi14 = derivfactor14 * ( 2*diffreal14*( (gradfsi14r1)/denommag14 - (gradfsi14r2)/denommag14^2 ) +
2*diffimag14*( (gradfsi14i1)/denommag14 - (gradfsi14i2)/denommag14^2 ) );
   gradfcr14 = derivfactor14 * ( 2*diffreal14*( (gradfcr14r1)/denommag14 - (gradfcr14r2)/denommag14^2 ) +
2*diffimag14*( (gradfcr14i1)/denommag14 - (gradfcr14i2)/denommag14^2 ) );
   gradfci14 = derivfactor14 * ( 2*diffreal14*( (gradfci14r1)/denommag14 - (gradfci14r2)/denommag14^2 ) +
2*diffimag14*( (gradfci14i1)/denommag14 - (gradfci14i2)/denommag14^2 ) );


   gradftr34r1 = r2dc*rtds+r2dc*rts-i2dc*its-i2dc*-itds+drcr-dici;
   gradftr34r2 = 2*rtds*numrealpart34;
   gradftr34i1 = i2dc*rtds+i2dc*rts+r2dc*its+r2dc*-itds+drci+dicr;
   gradftr34i2 = 2*rtds*numimagpart34;
   gradfti34r1 = -i2dc*rtds+i2dc*rts+r2dc*its-r2dc*-itds+drci+dicr;
   gradfti34r2 = 2*itds*numrealpart34;
   gradfti34i1 = r2dc*rtds-r2dc*rts+i2dc*its-i2dc*-itds-drcr+dici;
   gradfti34i2 = 2*itds*numimagpart34;
   gradfdr34r1 = rtds*(2*rts+cr)+r2dc*rts+itds*(2*its+ci)-i2dc*its+drcr-dici;
   gradfdr34r2 = 2*rtds*numrealpart34;
   gradfdr34i1 = -itds*(2*rts+cr)+(2*its+ci)*rtds+i2dc*rts+r2dc*its+drci+dicr;
   gradfdr34i2 = 2*rtds*numimagpart34;
   gradfdi34r1 = +itds*(2*rts+cr)+(-2*its-ci)*rtds+i2dc*rts+r2dc*its+drci+dicr;
   gradfdi34r2 = 2*itds*numrealpart34;
   gradfdi34i1 = rtds*(2*rts+cr)-r2dc*rts-itds*(-2*its-ci)+i2dc*its-drcr+dici;
   gradfdi34i2 = 2*itds*numimagpart34;
   gradfsr34r1 = r2dc*rtds+r2dc*rts-i2dc*its-i2dc*-itds+drcr-dici;
   gradfsr34r2 = 2*rtds*numrealpart34;
   gradfsr34i1 = i2dc*rtds+i2dc*rts+r2dc*its+r2dc*-itds+drci+dicr;
   gradfsr34i2 = 2*rtds*numimagpart34;
   gradfsi34r1 = -i2dc*rtds+i2dc*rts+r2dc*its-r2dc*-itds+drci+dicr;
   gradfsi34r2 = 2*itds*numrealpart34;
   gradfsi34i1 = r2dc*rtds-r2dc*rts+i2dc*its-i2dc*-itds-drcr+dici;
   gradfsi34i2 = 2*itds*numimagpart34;
   gradfcr34r1 = rtds^2+itds*itds;
   gradfcr34r2 = 0;
   gradfcr34i1 = itds*rtds-itds*rtds;
   gradfcr34i2 = 0;
   gradfci34r1 = 0;
   gradfci34r2 = 0;
   gradfci34i1 = rtds^2-itds^2;
   gradfci34i2 = 0;

   gradftr34 = derivfactor34 * ( 2*diffreal34*( (gradftr34r1)/denommag34 - (gradftr34r2)/denommag34^2 ) +
2*diffimag34*( (gradftr34i1)/denommag34 - (gradftr34i2)/denommag34^2 ) );
   gradfti34 = derivfactor34 * ( 2*diffreal34*( (gradfti34r1)/denommag34 - (gradfti34r2)/denommag34^2 ) +
2*diffimag34*( (gradfti34i1)/denommag34 - (gradfti34i2)/denommag34^2 ) );
   gradfdr34 = derivfactor34 * ( 2*diffreal34*( (gradfdr34r1)/denommag34 - (gradfdr34r2)/denommag34^2 ) +
2*diffimag34*( (gradfdr34i1)/denommag34 - (gradfdr34i2)/denommag34^2 ) );
   gradfdi34 = derivfactor34 * ( 2*diffreal34*( (gradfdi34r1)/denommag34 - (gradfdi34r2)/denommag34^2 ) +
2*diffimag34*( (gradfdi34i1)/denommag34 - (gradfdi34i2)/denommag34^2 ) );
   gradfsr34 = derivfactor34 * ( 2*diffreal34*( (gradfsr34r1)/denommag34 - (gradfsr34r2)/denommag34^2 ) +
2*diffimag34*( (gradfsr34i1)/denommag34 - (gradfsr34i2)/denommag34^2 ) );
   gradfsi34 = derivfactor34 * ( 2*diffreal34*( (gradfsi34r1)/denommag34 - (gradfsi34r2)/denommag34^2 ) +
2*diffimag34*( (gradfsi34i1)/denommag34 - (gradfsi34i2)/denommag34^2 ) );
   gradfcr34 = derivfactor34 * ( 2*diffreal34*( (gradfcr34r1)/denommag34 - (gradfcr34r2)/denommag34^2 ) +
2*diffimag34*( (gradfcr34i1)/denommag34 - (gradfcr34i2)/denommag34^2 ) );
   gradfci34 = derivfactor34 * ( 2*diffreal34*( (gradfci34r1)/denommag34 - (gradfci34r2)/denommag34^2 ) +
2*diffimag34*( (gradfci34i1)/denommag34 - (gradfci34i2)/denommag34^2 ) );


   % combine partial derivatives for each objective
   gradftr = gradftr12 + gradftr13 + gradftr14 + gradftr34;
   gradfti = gradfti12 + gradfti13 + gradfti14 + gradfti34;
   gradfdr = gradfdr12 + gradfdr13 + gradfdr14 + gradfdr34;
   gradfdi = gradfdi12 + gradfdi13 + gradfdi14 + gradfdi34;
   gradfsr = gradfsr12 + gradfsr13 + gradfsr14 + gradfsr34;
   gradfsi = gradfsi12 + gradfsi13 + gradfsi14 + gradfsi34;
   gradfcr = gradfcr12 + gradfcr13 + gradfcr14 + gradfcr34;
   gradfci = gradfci12 + gradfci13 + gradfci14 + gradfci34;

   gradf = [ gradftr ; gradfti ; gradfdr ; gradfdi ; gradfsr ; gradfsi ; gradfcr ; gradfci ];
   gradNorm = sqrt(gradftr*conj(gradftr) + gradfti*conj(gradfti) +
gradfdr*conj(gradfdr) + gradfdi*conj(gradfdi) +
gradfsr*conj(gradfsr) + gradfsi*conj(gradfsi) +
gradfcr*conj(gradfcr) + gradfci*conj(gradfci));
```

APPENDIX M

MATLAB CODE FOR ELIMINATING OUTLIERS IN THE CALCULATED MODEL PARAMETERS

```
% CleanXYModelParams.m

% Sort through calculated parameters of an X-Y impedance model and get rid
% of obvious extreme outliers to facilitate subsequent calculations

% only use skin1 - skin6 for the VNA calculations
% all measurements are from abdominal skin
%
% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm
% skin7 = dead scalp, no thickness measurement
% skin8 = live scalp, no thickness measurement, looks like 7+ mm from pictures
% skin9 = dead ear, ~3+ mm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%% skin 1
for k=2:(length(freq10)-1)

    if ( abs(Ztxy1(k)) > 300 )
        Ztxy1(k) = (Ztxy1(k-1) + Ztxy1(k+1))/2;
    end

    if ( abs(Zdxy1(k)) > 300 )
        Zdxy1(k) = (Zdxy1(k-1) + Zdxy1(k+1))/2;
    end

    if ( abs(Zsxy1(k)) > 300 )
        Zsxy1(k) = (Zsxy1(k-1) + Zsxy1(k+1))/2;
    end

    if ( abs(Zcxy1(k)) > 300 )
        Zcxy1(k) = (Zcxy1(k-1) + Zcxy1(k+1))/2;
    end
end


%%% skin 2
for k=2:(length(freq10)-1)

    if ( abs(Ztxy2(k)) > 300 )
        Ztxy2(k) = (Ztxy2(k-1) + Ztxy2(k+1))/2;
    end

    if ( abs(Zdxy2(k)) > 300 )
        Zdxy2(k) = (Zdxy2(k-1) + Zdxy2(k+1))/2;
    end

    if ( abs(Zsxy2(k)) > 300 )
        Zsxy2(k) = (Zsxy2(k-1) + Zsxy2(k+1))/2;
    end

    if ( abs(Zcxy2(k)) > 300 )
        Zcxy2(k) = (Zcxy2(k-1) + Zcxy2(k+1))/2;
    end
end


%%% skin 3
for k=2:(length(freq10)-1)

    if ( abs(Ztxy3(k)) > 300 )
        Ztxy3(k) = (Ztxy3(k-1) + Ztxy3(k+1))/2;
    end

    if ( abs(Zdxy3(k)) > 300 )
        Zdxy3(k) = (Zdxy3(k-1) + Zdxy3(k+1))/2;
    end

    if ( abs(Zsxy3(k)) > 300 )
        Zsxy3(k) = (Zsxy3(k-1) + Zsxy3(k+1))/2;
    end

    if ( abs(Zcxy3(k)) > 300 )
        Zcxy3(k) = (Zcxy3(k-1) + Zcxy3(k+1))/2;
    end
end


%%% skin 4
for k=2:(length(freq10)-1)

    if ( abs(Ztxy4(k)) > 300 )
        Ztxy4(k) = (Ztxy4(k-1) + Ztxy4(k+1))/2;
    end

    if ( abs(Zdxy4(k)) > 300 )
        Zdxy4(k) = (Zdxy4(k-1) + Zdxy4(k+1))/2;
    end

    if ( abs(Zsxy4(k)) > 300 )
        Zsxy4(k) = (Zsxy4(k-1) + Zsxy4(k+1))/2;
    end

    if ( abs(Zcxy4(k)) > 300 )
        Zcxy4(k) = (Zcxy4(k-1) + Zcxy4(k+1))/2;
    end
end


%%% skin 5
for k=2:(length(freq10)-1)

    if ( abs(Ztxy5(k)) > 300 )
        Ztxy5(k) = (Ztxy5(k-1) + Ztxy5(k+1))/2;
    end

    if ( abs(Zdxy5(k)) > 300 )
        Zdxy5(k) = (Zdxy5(k-1) + Zdxy5(k+1))/2;
    end

    if ( abs(Zsxy5(k)) > 300 )
        Zsxy5(k) = (Zsxy5(k-1) + Zsxy5(k+1))/2;
    end

    if ( abs(Zcxy5(k)) > 300 )
        Zcxy5(k) = (Zcxy5(k-1) + Zcxy5(k+1))/2;
    end
end


%%% skin 6
for k=2:(length(freq10)-1)

    if ( abs(Ztxy6(k)) > 300 )
        Ztxy6(k) = (Ztxy6(k-1) + Ztxy6(k+1))/2;
    end

    if ( abs(Zdxy6(k)) > 300 )
        Zdxy6(k) = (Zdxy6(k-1) + Zdxy6(k+1))/2;
    end

    if ( abs(Zsxy6(k)) > 300 )
        Zsxy6(k) = (Zsxy6(k-1) + Zsxy6(k+1))/2;
    end

    if ( abs(Zcxy6(k)) > 300 )
        Zcxy6(k) = (Zcxy6(k-1) + Zcxy6(k+1))/2;
    end
end


%%% skin 7
for k=2:(length(freq1)-1)

    if ( abs(Ztxy7(k)) > 300 )
        Ztxy7(k) = (Ztxy7(k-1) + Ztxy7(k+1))/2;
    end

    if ( abs(Zdxy7(k)) > 300 )
        Zdxy7(k) = (Zdxy7(k-1) + Zdxy7(k+1))/2;
    end

    if ( abs(Zsxy7(k)) > 300 )
        Zsxy7(k) = (Zsxy7(k-1) + Zsxy7(k+1))/2;
    end

    if ( abs(Zcxy7(k)) > 300 )
        Zcxy7(k) = (Zcxy7(k-1) + Zcxy7(k+1))/2;
    end
end


%%% skin 8
for k=2:(length(freq1)-1)

    if ( abs(Ztxy8(k)) > 300 )
        Ztxy8(k) = (Ztxy8(k-1) + Ztxy8(k+1))/2;
    end

    if ( abs(Zdxy8(k)) > 300 )
        Zdxy8(k) = (Zdxy8(k-1) + Zdxy8(k+1))/2;
    end

    if ( abs(Zsxy8(k)) > 300 )
        Zsxy8(k) = (Zsxy8(k-1) + Zsxy8(k+1))/2;
    end

    if ( abs(Zcxy8(k)) > 300 )
        Zcxy8(k) = (Zcxy8(k-1) + Zcxy8(k+1))/2;
    end
end


%%% skin 9
for k=2:(length(freq1)-1)

    if ( abs(Ztxy9(k)) > 300 )
        Ztxy9(k) = (Ztxy9(k-1) + Ztxy9(k+1))/2;
    end

    if ( abs(Zdxy9(k)) > 300 )
        Zdxy9(k) = (Zdxy9(k-1) + Zdxy9(k+1))/2;
    end

    if ( abs(Zsxy9(k)) > 300 )
        Zsxy9(k) = (Zsxy9(k-1) + Zsxy9(k+1))/2;
    end

    if ( abs(Zcxy9(k)) > 300 )
        Zcxy9(k) = (Zcxy9(k-1) + Zcxy9(k+1))/2;
    end
end
```

APPENDIX N

MATLAB CODE FOR CALCULATING EQUIVALENT X-MODEL PARAMETERS FROM X-Δ MODEL PARAMETERS

```
% CalcXParamsFromXY.m

% Find equivalent parameters to an x impedance model
% from x-y parameters.  Also calculates error between the two


close all

%%% skin 1
denom = Ztxy1 + Zdxy1 + Zsxy1;

Zdxyx1 = (Ztxy1.*Zdxy1)./denom;
Zsxyx1 = (Ztxy1.*Zsxy1)./denom;
Zcxyx1 = Zcxy1 + 2*(Zdxy1.*Zsxy1)./denom;
Zdxyxerrnorm1 = abs(Zdx1 - Zdxyx1)./abs(Zdx1);
Zsxyxerrnorm1 = abs(Zsx1 - Zsxyx1)./abs(Zsx1);
Zcxyxerrnorm1 = abs(Zcx1 - Zcxyx1)./abs(Zcx1);


%%% skin 2
denom = Ztxy2 + Zdxy2 + Zsxy2;

Zdxyx2 = (Ztxy2.*Zdxy2)./denom;
Zsxyx2 = (Ztxy2.*Zsxy2)./denom;
Zcxyx2 = Zcxy2 + 2*(Zdxy2.*Zsxy2)./denom;
Zdxyxerrnorm2 = abs(Zdx2 - Zdxyx2)./abs(Zdx2);
Zsxyxerrnorm2 = abs(Zsx2 - Zsxyx2)./abs(Zsx2);
Zcxyxerrnorm2 = abs(Zcx2 - Zcxyx2)./abs(Zcx2);


%%% skin 3
denom = Ztxy3 + Zdxy3 + Zsxy3;

Zdxyx3 = (Ztxy3.*Zdxy3)./denom;
Zsxyx3 = (Ztxy3.*Zsxy3)./denom;
Zcxyx3 = Zcxy3 + 2*(Zdxy3.*Zsxy3)./denom;
Zdxyxerrnorm3 = abs(Zdx3 - Zdxyx3)./abs(Zdx3);
Zsxyxerrnorm3 = abs(Zsx3 - Zsxyx3)./abs(Zsx3);
Zcxyxerrnorm3 = abs(Zcx3 - Zcxyx3)./abs(Zcx3);


%%% skin 4
denom = Ztxy4 + Zdxy4 + Zsxy4;

Zdxyx4 = (Ztxy4.*Zdxy4)./denom;
Zsxyx4 = (Ztxy4.*Zsxy4)./denom;
Zcxyx4 = Zcxy4 + 2*(Zdxy4.*Zsxy4)./denom;
Zdxyxerrnorm4 = abs(Zdx4 - Zdxyx4)./abs(Zdx4);
Zsxyxerrnorm4 = abs(Zsx4 - Zsxyx4)./abs(Zsx4);
Zcxyxerrnorm4 = abs(Zcx4 - Zcxyx4)./abs(Zcx4);
```

```
%%% skin 5
denom = Ztxy5 + Zdxy5 + Zsxy5;

Zdxyx5 = (Ztxy5.*Zdxy5)./denom;
Zsxyx5 = (Ztxy5.*Zsxy5)./denom;
Zcxyx5 = Zcxy5 + 2*(Zdxy5.*Zsxy5)./denom;
Zdxyxerrnorm5 = abs(Zdx5 - Zdxyx5)./abs(Zdx5);
Zsxyxerrnorm5 = abs(Zsx5 - Zsxyx5)./abs(Zsx5);
Zcxyxerrnorm5 = abs(Zcx5 - Zcxyx5)./abs(Zcx5);


%%% skin 6
denom = Ztxy6 + Zdxy6 + Zsxy6;

Zdxyx6 = (Ztxy6.*Zdxy6)./denom;
Zsxyx6 = (Ztxy6.*Zsxy6)./denom;
Zcxyx6 = Zcxy6 + 2*(Zdxy6.*Zsxy6)./denom;
Zdxyxerrnorm6 = abs(Zdx6 - Zdxyx6)./abs(Zdx6);
Zsxyxerrnorm6 = abs(Zsx6 - Zsxyx6)./abs(Zsx6);
Zcxyxerrnorm6 = abs(Zcx6 - Zcxyx6)./abs(Zcx6);


%%% skin 7
denom = Ztxy7 + Zdxy7 + Zsxy7;

Zdxyx7 = (Ztxy7.*Zdxy7)./denom;
Zsxyx7 = (Ztxy7.*Zsxy7)./denom;
Zcxyx7 = Zcxy7 + 2*(Zdxy7.*Zsxy7)./denom;
Zdxyxerrnorm7 = abs(Zdx7 - Zdxyx7)./abs(Zdx7);
Zsxyxerrnorm7 = abs(Zsx7 - Zsxyx7)./abs(Zsx7);
Zcxyxerrnorm7 = abs(Zcx7 - Zcxyx7)./abs(Zcx7);


%%% skin 8
denom = Ztxy8 + Zdxy8 + Zsxy8;

Zdxyx8 = (Ztxy8.*Zdxy8)./denom;
Zsxyx8 = (Ztxy8.*Zsxy8)./denom;
Zcxyx8 = Zcxy8 + 2*(Zdxy8.*Zsxy8)./denom;
Zdxyxerrnorm8 = abs(Zdx8 - Zdxyx8)./abs(Zdx8);
Zsxyxerrnorm8 = abs(Zsx8 - Zsxyx8)./abs(Zsx8);
Zcxyxerrnorm8 = abs(Zcx8 - Zcxyx8)./abs(Zcx8);


%%% skin 9
denom = Ztxy9 + Zdxy9 + Zsxy9;

Zdxyx9 = (Ztxy9.*Zdxy9)./denom;
Zsxyx9 = (Ztxy9.*Zsxy9)./denom;
Zcxyx9 = Zcxy9 + 2*(Zdxy9.*Zsxy9)./denom;
Zdxyxerrnorm9 = abs(Zdx9 - Zdxyx9)./abs(Zdx9);
Zsxyxerrnorm9 = abs(Zsx9 - Zsxyx9)./abs(Zsx9);
Zcxyxerrnorm9 = abs(Zcx9 - Zcxyx9)./abs(Zcx9);
```

APPENDIX O

MATLAB CODE FOR CALCULATING THEORETICAL MAXIMUM CURRENT TRANSFER EFFICIENCY

```
% CurrentTransferOptimization.m

% Find load impedance that optimizes current transfer given previously
% calculated model parameters

% only use skin1 - skin6 for the VNA calculations
% all measurements are from abdominal skin
%
% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm
% skin7 = dead scalp, no thickness measurement
% skin8 = live scalp, no thickness measurement, looks like 7+ mm from pictures
% skin9 = dead ear, ~3+ mm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% % interval size variable for CG method
% ab = [ 1E-6 ; 1E3 ];
%
% % interval search stop width for CG method
% intvl = 1E-2;
%
% % stopping criterion
% e = 1E-3;      % used as derivative limit for CG method
%
% % maximum number of algorithm cycles
% stopCycles = 100;       % used for CG method



%%% skin 1
for k=1:length(freq10)

    % initial guess for the load impedance
%     Zl0 = [ real(Zcxy1(k)) ; imag(Zcxy1(k)) ];

    % calculated parameters at the indexed frequency
    Z = [ Ztxy1(k) ; Zdxy1(k) ; Zsxy1(k) ; Zcxy1(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy1(k) + Zdxy1(k) + Zsxy1(k));
    tdsi = imag(Ztxy1(k) + Zdxy1(k) + Zsxy1(k));
    denomr = real(Zcxy1(k)*(Ztxy1(k) + Zdxy1(k) + Zsxy1(k)) + 2*Zdxy1(k)*(Ztxy1(k) + Zsxy1(k)));
    denomi = imag(Zcxy1(k)*(Ztxy1(k) + Zdxy1(k) + Zsxy1(k)) + 2*Zdxy1(k)*(Ztxy1(k) + Zsxy1(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );
    % use algorithm to find optimal parameters to fit the measurements
%     [Zl,fTrace,gradient] = CGSearchCurrentEff(Zl0,Z,ab,intvl,e,stopCycles);

    % find index of smallest error in case algorithm diverges
%     [c,m] = min(fTrace);

    % record value for this frequency
    if (k==1)
%         cTrace = c;
%         mTrace = m;
%         fTracelength = length(fTrace);

%         Zlcurr = Zl(1,m)+1i*Zl(2,m);
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy1(k)*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*Zsxy1(k) ) /
 ( (Zlcurr(k)+Zcxy1(k))*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*(Ztxy1(k)+Zsxy1(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
%         cTrace = [ cTrace c ];
%         mTrace = [ mTrace m ];
%         fTracelength = [ fTracelength ; length(fTrace) ];

%         Zlcurr = [ Zlcurr Zl(1,m)+1i*Zl(2,m) ];
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy1(k)*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*Zsxy1(k) ) /
 ( (Zlcurr(k)+Zcxy1(k))*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*(Ztxy1(k)+Zsxy1(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr1 = Zlcurr;
effcurrmag1 = effcurrmag;
effcurrphase1 = effcurrphase;
% fTracelength1 = fTracelength;
% cTrace1 = cTrace;
% mTrace1 = mTrace;


%%% skin 2
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy2(k) ; Zdxy2(k) ; Zsxy2(k) ; Zcxy2(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy2(k) + Zdxy2(k) + Zsxy2(k));
    tdsi = imag(Ztxy2(k) + Zdxy2(k) + Zsxy2(k));
    denomr = real(Zcxy2(k)*(Ztxy2(k) + Zdxy2(k) + Zsxy2(k)) + 2*Zdxy2(k)*(Ztxy2(k) + Zsxy2(k)));
    denomi = imag(Zcxy2(k)*(Ztxy2(k) + Zdxy2(k) + Zsxy2(k)) + 2*Zdxy2(k)*(Ztxy2(k) + Zsxy2(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy2(k)*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*Zsxy2(k) ) /
 ( (Zlcurr(k)+Zcxy2(k))*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*(Ztxy2(k)+Zsxy2(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy2(k)*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*Zsxy2(k) ) /
 ( (Zlcurr(k)+Zcxy2(k))*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*(Ztxy2(k)+Zsxy2(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr2 = Zlcurr;
effcurrmag2 = effcurrmag;
effcurrphase2 = effcurrphase;
```

```
%%% skin 3
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy3(k) ; Zdxy3(k) ; Zsxy3(k) ; Zcxy3(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy3(k) + Zdxy3(k) + Zsxy3(k));
    tdsi = imag(Ztxy3(k) + Zdxy3(k) + Zsxy3(k));
    denomr = real(Zcxy3(k)*(Ztxy3(k) + Zdxy3(k) + Zsxy3(k)) + 2*Zdxy3(k)*(Ztxy3(k) + Zsxy3(k)));
    denomi = imag(Zcxy3(k)*(Ztxy3(k) + Zdxy3(k) + Zsxy3(k)) + 2*Zdxy3(k)*(Ztxy3(k) + Zsxy3(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy3(k)*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*Zsxy3(k) ) /
 ( (Zlcurr(k)+Zcxy3(k))*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*(Ztxy3(k)+Zsxy3(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy3(k)*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*Zsxy3(k) ) /
 ( (Zlcurr(k)+Zcxy3(k))*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*(Ztxy3(k)+Zsxy3(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr3 = Zlcurr;
effcurrmag3 = effcurrmag;
effcurrphase3 = effcurrphase;


%%% skin 4
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy4(k) ; Zdxy4(k) ; Zsxy4(k) ; Zcxy4(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy4(k) + Zdxy4(k) + Zsxy4(k));
    tdsi = imag(Ztxy4(k) + Zdxy4(k) + Zsxy4(k));
    denomr = real(Zcxy4(k)*(Ztxy4(k) + Zdxy4(k) + Zsxy4(k)) + 2*Zdxy4(k)*(Ztxy4(k) + Zsxy4(k)));
    denomi = imag(Zcxy4(k)*(Ztxy4(k) + Zdxy4(k) + Zsxy4(k)) + 2*Zdxy4(k)*(Ztxy4(k) + Zsxy4(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy4(k)*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*Zsxy4(k) ) /
 ( (Zlcurr(k)+Zcxy4(k))*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*(Ztxy4(k)+Zsxy4(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy4(k)*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*Zsxy4(k) ) /
 ( (Zlcurr(k)+Zcxy4(k))*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*(Ztxy4(k)+Zsxy4(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr4 = Zlcurr;
effcurrmag4 = effcurrmag;
effcurrphase4 = effcurrphase;


%%% skin 5
```

```
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy5(k) ; Zdxy5(k) ; Zsxy5(k) ; Zcxy5(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy5(k) + Zdxy5(k) + Zsxy5(k));
    tdsi = imag(Ztxy5(k) + Zdxy5(k) + Zsxy5(k));
    denomr = real(Zcxy5(k)*(Ztxy5(k) + Zdxy5(k) + Zsxy5(k)) + 2*Zdxy5(k)*(Ztxy5(k) + Zsxy5(k)));
    denomi = imag(Zcxy5(k)*(Ztxy5(k) + Zdxy5(k) + Zsxy5(k)) + 2*Zdxy5(k)*(Ztxy5(k) + Zsxy5(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy5(k)*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*Zsxy5(k) ) /
 ( (Zlcurr(k)+Zcxy5(k))*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*(Ztxy5(k)+Zsxy5(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy5(k)*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*Zsxy5(k) ) /
 ( (Zlcurr(k)+Zcxy5(k))*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*(Ztxy5(k)+Zsxy5(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr5 = Zlcurr;
effcurrmag5 = effcurrmag;
effcurrphase5 = effcurrphase;


%%% skin 6
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy6(k) ; Zdxy6(k) ; Zsxy6(k) ; Zcxy6(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy6(k) + Zdxy6(k) + Zsxy6(k));
    tdsi = imag(Ztxy6(k) + Zdxy6(k) + Zsxy6(k));
    denomr = real(Zcxy6(k)*(Ztxy6(k) + Zdxy6(k) + Zsxy6(k)) + 2*Zdxy6(k)*(Ztxy6(k) + Zsxy6(k)));
    denomi = imag(Zcxy6(k)*(Ztxy6(k) + Zdxy6(k) + Zsxy6(k)) + 2*Zdxy6(k)*(Ztxy6(k) + Zsxy6(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy6(k)*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*Zsxy6(k) ) /
 ( (Zlcurr(k)+Zcxy6(k))*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*(Ztxy6(k)+Zsxy6(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy6(k)*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*Zsxy6(k) ) /
 ( (Zlcurr(k)+Zcxy6(k))*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*(Ztxy6(k)+Zsxy6(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr6 = Zlcurr;
effcurrmag6 = effcurrmag;
effcurrphase6 = effcurrphase;


%%% skin 7
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy7(k) ; Zdxy7(k) ; Zsxy7(k) ; Zcxy7(k) ];
    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy7(k) + Zdxy7(k) + Zsxy7(k));
    tdsi = imag(Ztxy7(k) + Zdxy7(k) + Zsxy7(k));
    denomr = real(Zcxy7(k)*(Ztxy7(k) + Zdxy7(k) + Zsxy7(k)) + 2*Zdxy7(k)*(Ztxy7(k) + Zsxy7(k)));
    denomi = imag(Zcxy7(k)*(Ztxy7(k) + Zdxy7(k) + Zsxy7(k)) + 2*Zdxy7(k)*(Ztxy7(k) + Zsxy7(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy7(k)*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*Zsxy7(k) ) /
 ( (Zlcurr(k)+Zcxy7(k))*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*(Ztxy7(k)+Zsxy7(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy7(k)*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*Zsxy7(k) ) /
 ( (Zlcurr(k)+Zcxy7(k))*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*(Ztxy7(k)+Zsxy7(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr7 = Zlcurr;
effcurrmag7 = effcurrmag;
effcurrphase7 = effcurrphase;


%%% skin 8
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy8(k) ; Zdxy8(k) ; Zsxy8(k) ; Zcxy8(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy8(k) + Zdxy8(k) + Zsxy8(k));
    tdsi = imag(Ztxy8(k) + Zdxy8(k) + Zsxy8(k));
    denomr = real(Zcxy8(k)*(Ztxy8(k) + Zdxy8(k) + Zsxy8(k)) + 2*Zdxy8(k)*(Ztxy8(k) + Zsxy8(k)));
    denomi = imag(Zcxy8(k)*(Ztxy8(k) + Zdxy8(k) + Zsxy8(k)) + 2*Zdxy8(k)*(Ztxy8(k) + Zsxy8(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy8(k)*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*Zsxy8(k) ) /
 ( (Zlcurr(k)+Zcxy8(k))*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*(Ztxy8(k)+Zsxy8(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy8(k)*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*Zsxy8(k) ) /
 ( (Zlcurr(k)+Zcxy8(k))*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*(Ztxy8(k)+Zsxy8(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr8 = Zlcurr;
effcurrmag8 = effcurrmag;
effcurrphase8 = effcurrphase;


%%% skin 9
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy9(k) ; Zdxy9(k) ; Zsxy9(k) ; Zcxy9(k) ];

    % use derivative = 0 condition to find optimal Zli (assuming Zlr = 0)
    tdsr = real(Ztxy9(k) + Zdxy9(k) + Zsxy9(k));
    tdsi = imag(Ztxy9(k) + Zdxy9(k) + Zsxy9(k));
    denomr = real(Zcxy9(k)*(Ztxy9(k) + Zdxy9(k) + Zsxy9(k)) + 2*Zdxy9(k)*(Ztxy9(k) + Zsxy9(k)));
    denomi = imag(Zcxy9(k)*(Ztxy9(k) + Zdxy9(k) + Zsxy9(k)) + 2*Zdxy9(k)*(Ztxy9(k) + Zsxy9(k)));
    Zli = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zlcurr = 1i*Zli;

        i3i1 = ( Zcxy9(k)*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*Zsxy9(k) ) /
 ( (Zlcurr(k)+Zcxy9(k))*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*(Ztxy9(k)+Zsxy9(k)) );

        effcurrmag = abs( i3i1 );
        effcurrphase = (180/pi)*atan( imag(i3i1) / real(i3i1) );
    else
        Zlcurr = [ Zlcurr 1i*Zli ];

        i3i1 = ( Zcxy9(k)*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*Zsxy9(k) ) /
 ( (Zlcurr(k)+Zcxy9(k))*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*(Ztxy9(k)+Zsxy9(k)) );

        effcurrmag = [ effcurrmag abs( i3i1 ) ];
        effcurrphase = [ effcurrphase (180/pi)*atan( imag(i3i1) / real(i3i1) ) ];
    end
end

Zlcurr9 = Zlcurr;
effcurrmag9 = effcurrmag;
effcurrphase9 = effcurrphase;


figure(10)
plot(freq10,effcurrmag1,'r',freq10,effcurrmag2,'y',freq10,effcurrmag3,'g',freq10,effcurrmag4,'b',
freq10,effcurrmag5,'m',freq10,effcurrmag6,'k',freq1,effcurrmag7,'r:',freq1,effcurrmag8,'b:',freq1,effcurrmag9,'k:')
title('Theoretical Maximum Current Transfer Efficiency')
xlabel('Frequency (Hz)')
ylabel('Efficiency (I3/I1)')
```

APPENDIX P

MATLAB CODE FOR CALCULATING THEORETICAL MAXIMUM VOLTAGE TRANSFER EFFICIENCY

```
% VoltageTransferOptimization.m

% Find input impedance that optimizes voltage transfer given previously
% calculated model parameters.  Assume load impedance approaches infinity
% (open circuit) and real part of input impedance is zero

% only use skin1 - skin6 for the VNA calculations
% all measurements are from abdominal skin
%
% skin1 = 6 mm
% skin2 = 5 mm
% skin3 = 4-5 mm
% skin4 = 7 mm
% skin5 = 4-5 mm
% skin6 = 7 mm
% skin7 = dead scalp, no thickness measurement
% skin8 = live scalp, no thickness measurement, looks like 7+ mm from pictures
% skin9 = dead ear, ~3+ mm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%% skin 1
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy1(k) ; Zdxy1(k) ; Zsxy1(k) ; Zcxy1(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy1(k) + Zdxy1(k) + Zsxy1(k));
    tdsi = imag(Ztxy1(k) + Zdxy1(k) + Zsxy1(k));
    denomr = real(2*Zsxy1(k)*(Ztxy1(k) + Zdxy1(k)));
    denomi = imag(2*Zsxy1(k)*(Ztxy1(k) + Zdxy1(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy1(k)*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*Zsxy1(k) ) /
 ( (Zivolt(k)+Zcxy1(k))*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zsxy1(k)*(Ztxy1(k)+Zdxy1(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy1(k)*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zdxy1(k)*Zsxy1(k) ) /
 ( (Zivolt(k)+Zcxy1(k))*(Ztxy1(k)+Zdxy1(k)+Zsxy1(k)) + 2*Zsxy1(k)*(Ztxy1(k)+Zdxy1(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt1 = Zivolt;
effvoltmag1 = effvoltmag;
effvoltphase1 = effvoltphase;



%%% skin 2
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy2(k) ; Zdxy2(k) ; Zsxy2(k) ; Zcxy2(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy2(k) + Zdxy2(k) + Zsxy2(k));
    tdsi = imag(Ztxy2(k) + Zdxy2(k) + Zsxy2(k));
    denomr = real(2*Zsxy2(k)*(Ztxy2(k) + Zdxy2(k)));
    denomi = imag(2*Zsxy2(k)*(Ztxy2(k) + Zdxy2(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy2(k)*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*Zsxy2(k) ) /
 ( (Zivolt(k)+Zcxy2(k))*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zsxy2(k)*(Ztxy2(k)+Zdxy2(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy2(k)*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zdxy2(k)*Zsxy2(k) ) /
 ( (Zivolt(k)+Zcxy2(k))*(Ztxy2(k)+Zdxy2(k)+Zsxy2(k)) + 2*Zsxy2(k)*(Ztxy2(k)+Zdxy2(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt2 = Zivolt;
effvoltmag2 = effvoltmag;
effvoltphase2 = effvoltphase;



%%% skin 3
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy3(k) ; Zdxy3(k) ; Zsxy3(k) ; Zcxy3(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy3(k) + Zdxy3(k) + Zsxy3(k));
    tdsi = imag(Ztxy3(k) + Zdxy3(k) + Zsxy3(k));
    denomr = real(2*Zsxy3(k)*(Ztxy3(k) + Zdxy3(k)));
    denomi = imag(2*Zsxy3(k)*(Ztxy3(k) + Zdxy3(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy3(k)*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*Zsxy3(k) ) /
 ( (Zivolt(k)+Zcxy3(k))*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zsxy3(k)*(Ztxy3(k)+Zdxy3(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy3(k)*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zdxy3(k)*Zsxy3(k) ) /
 ( (Zivolt(k)+Zcxy3(k))*(Ztxy3(k)+Zdxy3(k)+Zsxy3(k)) + 2*Zsxy3(k)*(Ztxy3(k)+Zdxy3(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
```

```
end

Zivolt3 = Zivolt;
effvoltmag3 = effvoltmag;
effvoltphase3 = effvoltphase;



%%% skin 4
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy4(k) ; Zdxy4(k) ; Zsxy4(k) ; Zcxy4(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy4(k) + Zdxy4(k) + Zsxy4(k));
    tdsi = imag(Ztxy4(k) + Zdxy4(k) + Zsxy4(k));
    denomr = real(2*Zsxy4(k)*(Ztxy4(k) + Zdxy4(k)));
    denomi = imag(2*Zsxy4(k)*(Ztxy4(k) + Zdxy4(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy4(k)*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*Zsxy4(k) ) /
 ( (Zivolt(k)+Zcxy4(k))*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zsxy4(k)*(Ztxy4(k)+Zdxy4(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy4(k)*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zdxy4(k)*Zsxy4(k) ) /
 ( (Zivolt(k)+Zcxy4(k))*(Ztxy4(k)+Zdxy4(k)+Zsxy4(k)) + 2*Zsxy4(k)*(Ztxy4(k)+Zdxy4(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt4 = Zivolt;
effvoltmag4 = effvoltmag;
effvoltphase4 = effvoltphase;



%%% skin 5
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy5(k) ; Zdxy5(k) ; Zsxy5(k) ; Zcxy5(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy5(k) + Zdxy5(k) + Zsxy5(k));
    tdsi = imag(Ztxy5(k) + Zdxy5(k) + Zsxy5(k));
    denomr = real(2*Zsxy5(k)*(Ztxy5(k) + Zdxy5(k)));
    denomi = imag(2*Zsxy5(k)*(Ztxy5(k) + Zdxy5(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy5(k)*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*Zsxy5(k) ) /
 ( (Zivolt(k)+Zcxy5(k))*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zsxy5(k)*(Ztxy5(k)+Zdxy5(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy5(k)*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zdxy5(k)*Zsxy5(k) ) /
 ( (Zivolt(k)+Zcxy5(k))*(Ztxy5(k)+Zdxy5(k)+Zsxy5(k)) + 2*Zsxy5(k)*(Ztxy5(k)+Zdxy5(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end
```

```
Zivolt5 = Zivolt;
effvoltmag5 = effvoltmag;
effvoltphase5 = effvoltphase;



%%% skin 6
for k=1:length(freq10)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy6(k) ; Zdxy6(k) ; Zsxy6(k) ; Zcxy6(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy6(k) + Zdxy6(k) + Zsxy6(k));
    tdsi = imag(Ztxy6(k) + Zdxy6(k) + Zsxy6(k));
    denomr = real(2*Zsxy6(k)*(Ztxy6(k) + Zdxy6(k)));
    denomi = imag(2*Zsxy6(k)*(Ztxy6(k) + Zdxy6(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy6(k)*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*Zsxy6(k) ) /
 ( (Zivolt(k)+Zcxy6(k))*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zsxy6(k)*(Ztxy6(k)+Zdxy6(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy6(k)*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zdxy6(k)*Zsxy6(k) ) /
 ( (Zivolt(k)+Zcxy6(k))*(Ztxy6(k)+Zdxy6(k)+Zsxy6(k)) + 2*Zsxy6(k)*(Ztxy6(k)+Zdxy6(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt6 = Zivolt;
effvoltmag6 = effvoltmag;
effvoltphase6 = effvoltphase;



%%% skin 7
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy7(k) ; Zdxy7(k) ; Zsxy7(k) ; Zcxy7(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy7(k) + Zdxy7(k) + Zsxy7(k));
    tdsi = imag(Ztxy7(k) + Zdxy7(k) + Zsxy7(k));
    denomr = real(2*Zsxy7(k)*(Ztxy7(k) + Zdxy7(k)));
    denomi = imag(2*Zsxy7(k)*(Ztxy7(k) + Zdxy7(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy7(k)*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*Zsxy7(k) ) /
 ( (Zivolt(k)+Zcxy7(k))*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zsxy7(k)*(Ztxy7(k)+Zdxy7(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy7(k)*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zdxy7(k)*Zsxy7(k) ) /
 ( (Zivolt(k)+Zcxy7(k))*(Ztxy7(k)+Zdxy7(k)+Zsxy7(k)) + 2*Zsxy7(k)*(Ztxy7(k)+Zdxy7(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt7 = Zivolt;
effvoltmag7 = effvoltmag;
```

```
effvoltphase7 = effvoltphase;



%%% skin 8
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy8(k) ; Zdxy8(k) ; Zsxy8(k) ; Zcxy8(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy8(k) + Zdxy8(k) + Zsxy8(k));
    tdsi = imag(Ztxy8(k) + Zdxy8(k) + Zsxy8(k));
    denomr = real(2*Zsxy8(k)*(Ztxy8(k) + Zdxy8(k)));
    denomi = imag(2*Zsxy8(k)*(Ztxy8(k) + Zdxy8(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy8(k)*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*Zsxy8(k) ) /
 ( (Zivolt(k)+Zcxy8(k))*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zsxy8(k)*(Ztxy8(k)+Zdxy8(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy8(k)*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zdxy8(k)*Zsxy8(k) ) /
 ( (Zivolt(k)+Zcxy8(k))*(Ztxy8(k)+Zdxy8(k)+Zsxy8(k)) + 2*Zsxy8(k)*(Ztxy8(k)+Zdxy8(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt8 = Zivolt;
effvoltmag8 = effvoltmag;
effvoltphase8 = effvoltphase;



%%% skin 9
for k=1:length(freq1)

    % calculated parameters at the indexed frequency
    Z = [ Ztxy9(k) ; Zdxy9(k) ; Zsxy9(k) ; Zcxy9(k) ];

    % use derivative = 0 condition to find optimal Zii (assuming Zir = 0)
    tdsr = real(Ztxy9(k) + Zdxy9(k) + Zsxy9(k));
    tdsi = imag(Ztxy9(k) + Zdxy9(k) + Zsxy9(k));
    denomr = real(2*Zsxy9(k)*(Ztxy9(k) + Zdxy9(k)));
    denomi = imag(2*Zsxy9(k)*(Ztxy9(k) + Zdxy9(k)));
    Zii = ( tdsi*denomr - tdsr*denomi ) / ( tdsr^2 + tdsi^2 );

    % record value for this frequency
    if (k==1)
        Zivolt = 1i*Zii;

        vlvs = ( Zcxy9(k)*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*Zsxy9(k) ) /
 ( (Zivolt(k)+Zcxy9(k))*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zsxy9(k)*(Ztxy9(k)+Zdxy9(k)) );

        effvoltmag = abs( vlvs );
        effvoltphase = (180/pi)*atan( imag(vlvs) / real(vlvs) );
    else
        Zivolt = [ Zivolt 1i*Zii ];

        vlvs = ( Zcxy9(k)*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zdxy9(k)*Zsxy9(k) ) /
 ( (Zivolt(k)+Zcxy9(k))*(Ztxy9(k)+Zdxy9(k)+Zsxy9(k)) + 2*Zsxy9(k)*(Ztxy9(k)+Zdxy9(k)) );

        effvoltmag = [ effvoltmag abs( vlvs ) ];
        effvoltphase = [ effvoltphase (180/pi)*atan( imag(vlvs) / real(vlvs) ) ];
    end
end

Zivolt9 = Zivolt;
effvoltmag9 = effvoltmag;
effvoltphase9 = effvoltphase;
```

```
figure(11)
plot(freq10,effvoltmag1,'r',freq10,effvoltmag2,'y',freq10,effvoltmag3,'g',freq10,effvoltmag4,'b',
freq10,effvoltmag5,'m',freq10,effvoltmag6,'k',freq1,effvoltmag7,'r:',freq1,effvoltmag8,'b:',freq1,effvoltmag9,'k:')
title('Theoretical Maximum Voltage Transfer Efficiency')
xlabel('Frequency (Hz)')
ylabel('Efficiency (Vl/V)')
```

BIBLIOGRAPHY

[1] Braingate clinical trials. URL: http://www.cyberkinetics.com.

[2] Statistical information from centers for disease control and prevention (cdc), department of health and human services. URL: http://www.cdc.gov.

[3] Ieee standard for safety levels with respect to human exposure to radio frequency electromagnetic fields, 3 khz to 300 ghz. Technical Report C95.1, IEEE, New York, NY, Apr 2006. Sponsor: IEEE International Committee on Electromagnetic Safety (SCC39).

[4] R. Andersen. Encoding of intention and spatial location in the posterior parietal cortex. *Cereb. Cortex*, 5:457–469, 1995.

[5] N. Balabanian. *Network Synthesis*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1958.

[6] A. Beric, P. J. Kelly, A. Rezai, D. Sterio, A. Mogilner, M. Zonenshayn, and B. Kopell. Complications of deep brain stimulation surgery. *Stereotact. Funct. Neurosurg.*, 77(1-4):73–78, 2001.

[7] W. Boucsein. *Electrodermal Activity*. Plenum Press, New York, 1992.

[8] Q. Chen, S. C. Wong, C. K. Tse, and X. Ruan. Analysis, design, and control of a transcutaneous power regulator for artificial hearts. *IEEE Trans. Biomed. Circ. Sys.*, 3(1):23–31, Feb 2009.

[9] Y. A. Chizmadzhev, A. V. Indenbom, P. I. Kuzmin, S. V. Galichenko, J. C. Weaver, and R. O. Potts. Electrical properties of skin at moderate voltages: contribution of appendageal macropores. *Biophys. J.*, 74:843–856, 1998.

[10] A. F. Coston and J. K-J. Li. Transdermal drug delivery: a comparative analysis of skin impedance models and parameters. In *IEEE EMBS '03*, pages 2982–2985, Sep 2003.

[11] G. W. Crile, H. R. Hosmer, and A. F. Rowland. The electrical conductivity of animal tissues under normal and pathological conditions. *Amer. J. Physiol.*, 60:59–106, 1922.

[12] A. Dario and G. Tomei. Management of spasticity in multiple sclerosis by intrathecal baclofen. *Acta Neurochir. Suppl.*, 97(Pt 1):189–192, 2007.

[13] E. Decker and C. Millsaps. Rechargeable battery cycle life issues. In *Battery Power Products & Technology*, Mar 2001.

[14] S. J. Dorgan and R. B. Reilly. A model for human skin impedance during surface functional neuromuscular stimulation. *IEEE Trans. Rehab. Eng.*, 7(3):341–348, 1999.

[15] L. Williams et al (Microsoft Corporation). Method and apparatus for transmitting power and data using the human body. U. S. Patent No. 6,754,472, Jun 2004.

[16] D. M. Ferreira, C. S. Silva, and M. N. Souza. Electrical impedance model for evaluation of skin irritation in rabbits and humans. *Skin Research and Technology*, 13:259–267, 2007.

[17] S. Gabriel, R. W. Lau, and C. Gabriel. The dielectric properties of biological tissues: II. measurements in the frequency range 10 hz to 20 ghz. *Phys. Med. Biol.*, 41:2251–2269, 1996.

[18] L. A. Geddes, K. S. Foster, J. Reilly, W. D. Voorhees, J. D. Bourland, T. Ragheb, and N. E. Fearnot. The rectification properties of an electrode-electrolyte interface operated at high sinusoidal current density. *IEEE Trans. Biomed. Eng.*, 34(9):669–672, Sep 1987.

[19] M. Ghovanloo, K. J. Otto, D. R. Kipke, and K. Najafi. In vitro and in vivo testing of a wireless multichannel stimulating telemetry microsystem. In *IEEE Int. Conf. EMBS*, pages 4294–4297, 2004.

[20] B. Godin and E. Touitou. Transdermal skin delivery: predictions for humans from in vivo, ex vivo and animal models. *Adv. Drug Delivery Reviews*, 59:1152–1161, 2007.

[21] K. Goto, T. Nakagawa, O. Nakamura, and S. Kawata. An implantable power supply with an optically rechargeable lithium battery. *IEEE Trans. On Biomed. Engr.*, 48(7):830–833, Jul 2001.

[22] G. M. Gray and H. J. Yardley. Lipid compositions of cells isolated from pig, human, and rat epidermis. *J. Lipid Research*, 16:434–440, 1975.

[23] S. Grimnes and . G. Martinsen. *Bioimpedance & Bioelectricity Basics*. Academic Press, San Diego, CA, 2000.

[24] S. Grimnes and . G. Martinsen. Cole electrical impedance model - a critique and an alternative. *IEEE Trans. Biomed. Eng.*, 52(1):132–135, Jan 2005.

[25] D. A. Groves and V. J. Brown. Vagal nerve stimulation: a review of its applications and potential mechanisms that mediate its clinical effects. *Neurosci. Biobehav. Rev.*, 29(3):493–500, May 2005.

[26] S. A. Hackworth, M. Sun, and R. J. Sclabassi. A prototype volume conduction platform for implantable devices. In *IEEE 33rd Annual Northeast Bioeng. Conf.*, pages 124–125, 2007.

[27] S. A. Hackworth, M. Sun, and R. J. Sclabassi. Experimental analysis of a voltage multiplier for wirelessly recharging an implantable devices. In *34th Annual Northeast Bioeng. Conf.*, 2008.

[28] M. I. Hariz. Complications of deep brain stimulation surgery. *Mov. Disord.*, 17(Suppl 3):S162–S166, 2002.

[29] W. J. Heetderks. Rf powering of millimeter- and submillimeter-sized neural prosthetic implants. *IEEE Trans. Biomed. Engr.*, 35(5):323–327, May 1988.

[30] K. W. Horch and G. S. Dhillon. *Neuroprosthetics: Theory and Practice.* World Scientific, River Edge, NJ, 2004.

[31] T. B. Hunter, M. T. Yoshino, R. B. Dzioba, R. A. Light, and W. G. Berger. Medical devices of the head, neck, and spine. *Radiographics*, 24(1):257–285, Jan-Feb 2004.

[32] K. Inoue, K. Shiba, E. Shu, K. Koshiji, K. Tsukahara, T. Oh-umi, T. Masuzawa, E. Tatsumi, Y. Taenaka, and H. Takano. Transcutaneous optical telemetry system with infrared laser diode. *ASAIO J.*, 44:841–844, 1998.

[33] U. Jacobi, M. Kaiser, R. Toll, S. Mangelsdorf, H. Audring, N. Otberg, W. Sterry, and J. Lademann. Porcine ear skin: an in vitro model for human skin. *Skin Research and Tech.*, 13:19–24, 2007.

[34] A. Jadoul, J. Bouwstra, and V. Prat. Effects of iontophoresis and electroporation on the stratum corneum. review of the biophysical studies. *Adv. Drug Delivery Reviews*, 35:89–105, 1999.

[35] K.-M. Jager, D. H. McQueen, I. A. Tchmutin, N. G. Ryvkina, and M. Kluppel. Electron transport and ac electrical properties of carbon black polymer composites. *J. Phys. D: Appl. Phys.*, 34:2699–2707, 2001.

[36] C. Joint, D. Nandi, S. Parkin, R. Gregory, and T. Aziz. Hardware-related problems of deep brain stimulation. *Mov. Disord.*, 17(Suppl 3):S175–S180, 2002.

[37] H. Kato and T. Ichida. Development of an agar phantom adaptable for simulation of various tissues in the range of 5-40 MHz. *Phys. Med. Biol.*, 32:221–226, Feb 1987.

[38] P. R. Kennedy, R. A. E. Bakay, M. M. Moore, K. Adams, and J. Goldwaithe. Direct control of a computer from the human central nervous system. *IEEE Trans. Rehabil. Eng.*, 8:198–202, 2000.

[39] D. B. Koch, S. Staller, K. Jaax, and E. Martin. Bioengineering solutions for hearing loss and related disorders. *Otolaryngol. Clin. North Am.*, 38(2):255–272, Apr 2005.

[40] D. Kondziolka, D. Whiting, A. Germanwala, and M. Oh. Hardware-related complications after placement of thalamic deep brain stimulator systems. *Stereotact. Funct. Neurosurg.*, 79(3-4):228–233, 2002.

[41] K. Kontturi and L. Murtomki. Impedance spectroscopy in human skin. a refined model. *Pharma. Research*, 11(9):1355–1357, 1994.

[42] M. Kosel and T. E. Schlaepfer. Beyond the treatment of epilepsy: new applications of vagus nerve stimulation in psychiatry. *CNS Spectr.*, 8(7):515–521, Jul 2003.

[43] E. S. Krishnamurthy. Treatment of depression in patients with epilepsy: problems, pitfalls, and some solutions. *Epilepsy Behav.*, 4(Suppl 3):S46–S54, Oct 2003.

[44] E. S. Kuh and D. O. Pederson. *Principles of Circuit Synthesis.* McGraw-Hill, New York, 1959.

[45] A. N. K. Lau and L. L. Miller. Release of the neurotransmitters glutamate and gamma-aminobutyric acid from an electrode. catalysis of slow redox propagation through a polymer film. *J. Am. Chem. Soc.*, 105:5278–5284, 1983.

[46] J. C. Lawler, M. G. Davis, and E. C. Griffith. The electrical impedance of the surface sheath of skin and deep tissues. *J. Invest. Dermatol.*, 34:301–308, 1960.

[47] D. L. Li, R. J. Sclabassi, and M. Sun. Bioinspired electric power delivery antenna through volume conduction. In *31st Annual Northeast Bioeng. Conf.*, pages 71–72, Apr 2005.

[48] Y. Li, K. G. Neoh, and E. T. Kang. Controlled release of heparin from polypyrrole-poly(vinyl alcohol) assembly by electrical stimulation. *Journal of Biomedical Materials Research Part A*, 73A(2):171–181, 2005.

[49] D. P. Lindsey, E. L. McKee, M. L. Hull, and S. M. Howell. A new technique for transmission of signals from implantable transducers. *IEEE Trans. Biomed. Engr.*, 45:614–619, 1998.

[50] W. Liu, M. Sivaprakasam, G. Wang, J. Granacki M. Zhou, J. LaCoss, and J. Wills. Implantable biomimetic microelectronic systems design. *IEEE Engr. in Medicine and Biology Magazine*, 24(5):66–74, Sep-Oct 2005.

[51] G. E. Loeb, R. A. Peck, and J. Martyniuk. Toward the ultimate metal microelectrode. *J. Neurosci. Meth.*, 63:175–183, 1995.

[52] H. O. Luders, editor. *Deep Brain Stimulation and Epilepsy.* Taylor & Francis, Independence, 2004.

[53] K. E. Lyons and R. Pahwa. Deep brain stimulation and tremor. *Neurotherapeutics*, 5(2):331–338, Apr 2008.

[54] R. S. Mackay. *Bio-medical telemetry: sensing and transmitting biological information from animals and man.* Wiley, New York, 2nd edition, 1998.

[55] C. R. Maiorana and H. Staecker. Advances in inner ear gene therapy: exploring cochlear protection and regeneration. *Curr. Opin. Otolaryngol. Head Neck Surg.*, 13(5):308–312, Oct 2005.

[56] E. T. McAdams and J. Jossinet. A physical interpretation of Schwan's limit current of linearity. *Ann. Biomed. Eng.*, 20:307–319, 1992.

[57] E. T. McAdams and J. Jossinet. The detection of the onset of electrode-electrolyte interface impedance nonlinearity: a theoretical study. *IEEE Trans. Biomed. Eng.*, 41(5):498–500, May 1994.

[58] E. L. McKee, D. P. Lindsey, M. L. Hull, and S. M. Howell. Telemetry system for monitoring anterior cruciate ligament graft forces in vivo. *Med. Biol. Eng. Comput.*, 36:330–336, 1998.

[59] Medtronic, Inc. *Kinetra Dual Program Neurostimulator for Deep Brain Stimulation*, 2003.

[60] Medtronic, Inc. *Soletra Neurostimulator for Deep Brain Stimulation*, 2003.

[61] N. A. Monteiro-Riviere, A. O. Inman, and J. E. Riviere. Identification of the pathway of iontophoretic drug delivery: light and ultrastructural studies using mercuric chloride in pigs. *Pharma. Research*, 11(2):251–256, 1994.

[62] K. S. Moon, H. D. Choi, A. K. Lee, K. Y. Cho, H. G. Yoon, and K. S. Suh. Dielectric properties of epoxy-dielectrics-carbon black composite for phantom materials at radio frequencies. *J. Appl. Poly. Sci.*, 77:1294–1302, 2000.

[63] M. Mowafy and R. G. Cassens. Microscopic structure of pig skin. *J. Animal Sci.*, 41(5):1281–1290, 1975.

[64] K. Murakawa, M. Kobayashi, O. Nakamura, and S. Kawata. A wireless near-infrared energy system for medical implants. *IEEE Engr. in Med. and Biol. Magazine*, 18(6):70–72, Nov-Dec 1999.

[65] N. M. Neihart and R. R. Harrison. A low-power fm transmitter for use in neural recording applications. In *IEEE Int. Conf. EMBS*, pages 2117–2120, 2004.

[66] M. A. L. Nicolelis. Action from thoughts. *Nature*, 409:403–407, 2001.

[67] M. Y. Oh, A. Abosch, S. H. Kim, A. E. Lang, and A. M. Lozano. Long-term hardware-related complications of deep brain stimulation. *Neurosurgery*, 50(6):1268–1274, 2002. discussion 1274-1276.

[68] T. F. Oostendorp, J. Debelke, and D. F. Stegeman. The conductivity of the human skull: results of in vivo and in vitro measurements. *IEEE Trans. Biomed. Engr.*, 47(11):1487–1492, Dec 2000.

[69] C-S. Poon and T. T. C. Choy. Frequency dispersions of human skin dielectrics. *Biophys. J.*, 34:135–147, Apr 1981.

[70] M. R. Prausnitz. The effects of electric current applied to skin: a review for transdermal drug delivery. *Adv. Drug Delivery Reviews*, 18:395–425, 1996.

[71] M. Pyo and J. R. Reynolds. Electrochemically stimulated adenosine 5'-triphosphate (atp) release through redox switching of conducting polypyrrole films and bilayers. *Chem. Mater.*, 8:128–133, 1996.

[72] B. D. Ratner, A. S. Hoffman, F. J. Schoen, and J. E. Lemons. *Biomaterials Science: An Introduction to Materials in Medicine.* Elsevier Academic Press, San Diego, CA, 2nd edition, 2004.

[73] J. Ray, D. Proops, I. Donaldson, C. Fielden, and H. Cooper. Explantation and reimplantation of cochlear implants. *Cochlear Implants International*, 5(4):160–167, 2004.

[74] I. Richard and P. Menei. Intrathecal baclofen in the treatment of spasticity, dystonia and vegetative disorders. *Acta Neurochir. Suppl.*, 97(Pt 1):213–218, 2007.

[75] L.E. Riley, S. A. Hackworth, C. Henry, M. Sun, R. J. Sclabassi, and D. Hirsch. Design of a phantom head for the in vitro testing of implantable devices. In *IEEE 33rd Annual Northeast Bioeng. Conf.*, pages 296–297, 2007.

[76] P. A. Roche, M. Sun, and R. J. Sclabassi. Signal multiplexing and modulation for volume conduction communication. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 157–160, Mar 2005.

[77] L. Rucker and A. Lossinsky. Percutaneous connectors. In *30th Neural Prosthesis Workshop, NINDS, NINCD, NIH*, pages 12–14, Oct 1999.

[78] H. P. Schwan. Linear and nonlinear electrode polarization and biological materials. *Ann. Biomed. Eng.*, 20:269–288, 1992.

[79] A. B. Schwartz. Cortical neural prosthetics. *Annu. Rev. Neurosci.*, 27:487–507, 2004.

[80] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue. Instant neural control of a movement signal. *Nature*, 416(6877):141–142, 2002.

[81] R. F. Smith, B. K. Rutt, and D. W. Holdsworth. Anthropomorphic carotid bifurcation phantom for mri applications. *J. Mag. Res. Imag.*, 10:533–544, 1999.

[82] O. Soykan. Power sources for implantable medical devices. *Business Briefing: Med. Dev. Manufact. & Tech.*, pages 76–79, 2002.

[83] M. Sun, G. A. Justin, P. A. Roche, J. Zhao, B. L. Wessel, Y. Zhang, and R. J. Sclabassi. Passing data and supplying power to neural implants. *IEEE Eng. in Med. and Biol. Magazine*, 25(5):39–46, Sep-Oct 2006.

[84] M. Sun, D. L. Li, P. A. Roche, J. Zhao, B. L. Wessel, and R. J. Sclabassi. Volume conduction for communication and power delivery. In *NIH Neural Interfaces Workshop*, Sep 2005. (Abstract).

[85] M. Sun, W. Liang, B. L. Wessel, Q. Liu, P. A. Roche, M. Mickle, and R. J. Sclabassi. Reciprocity of human tissue and its application to implantable devices for data communication. In *First International Conf. on Medical Implants*, Jul 2003. (Abstract).

[86] M. Sun, W. Liang, B. L. Wessel, Q. Liu, P. A. Roche, M. Mickle, and R. J. Sclabassi. A super low power implantable antenna for data transmission between implantable devices and computers. In *First International Conf. on Medical Implants*, Jul 2003. (Abstract).

[87] M. Sun, Q. Liu, W. Liang, B. L. Wessel, P. A. Roche, M. Mickle, and R. J. Sclabassi. Application of the reciprocity theorem to volume conduction based data communication systems between implantable devices and computers. In *IEEE EMBS '03*, volume 4, pages 3352–3355, 2003.

[88] M. Sun, Q. Liu, W. Liang, B. L. Wessel, P. A. Roche, M. Mickle, and R. J. Sclabassi. A volume conduction antenna for implantable devices. In *IEEE EMBS '03*, volume 4, pages 3356–3359, 2003.

[89] M. Sun, M. Mickle, W. Liang, Q. Liu, and R. J. Sclabassi. Data communication between brain implants and computer. *IEEE Trans on Neural Sys. and Rehab. Engr.*, 11(2):189–192, 2003.

[90] M. Sun, B. L. Wessel, P. A. Roche, J. Zhao, and R. J. Sclabassi. Computer simulation of volume conduction based data communication channel for neuroprosthetic devices. In *IEEE EMBS Special Topic Conf. on Neural Engineering*, pages 426–429, Mar 2005.

[91] Z. Tang, R. J. Sclabassi, C. Sun, S. A. Hackworth, J. Zhao, X. T. Cui, and M. Sun. Transcutaneous battery recharging by volume conduction and its circuit modeling. In *IEEE EMBS '06*, volume 1, pages 644–647, 2006.

[92] Z. Tang, R. J. Sclabassi, C. Sun, J. Zhao, S. A. Hackworth, and M. Sun. Circuit model of battery recharging by volume conduction. In *IEEE 32nd Annual Northeast Bioeng. Conf.*, pages 125–126, 2006.

[93] D. M. Taylor, T. S. I. Helms, and A. B. Schwartz. Direct cortical control of 3d neuroprosthetic devices. *Science*, 296:1829–1832, 2002.

[94] B. C. Thompson, S. E. Moulton, J. Ding, R. Richardson, A. Cameron, S. O'Leary, and G. Wallace. Optimising the incorporation and release of a neurotrophic factor using conducting polypyrrole. *Journal of Controlled Release*, 116(3):285–294, 2006.

[95] P. R. Troyk and M. A. Schwan. Closed-loop class e transcutaneous power and data link for microimplants. *IEEE Trans. Biomed. Engr.*, 39(6):589–599, 1992.

[96] I. W. H. M. Craane van Hinsberg, J. C. Verhoef, F. Spies, J. A. Bouwstra, G. S. Gooris, H. E. Junginger, and H. E. Boddé. Electroperturbation of the human skin barrier in vitro: II. effects on stratum corneum lipid ordering and ultrastructure. *Microsc. Res. Tech.*, 37:200–213, 1997.

[97] J. Volkmann. Deep brain stimulation for the treatment of parkinson's disease. *J. Clin. Neurophysiol.*, 21(1):6–17, Jan-Feb 2004.

[98] R. Wadhwa, C. F. Lagenaur, and X. T. Cui. Electrochemically controlled release of dexamethasone from conducting polymer polypyrrole coated electrode. *Journal of Controlled Release*, 110:531–541, 2006.

[99] X. Wang, X. Gu, C. Yuan, A. Chen, P. Zhang, T. Zhang, J. Yao, F. Chen, and G. Chen. Evaluation of biocompatibility of polypyrrole in vitro and in vivo. *J. Biomed. Mat. Research Part A*, 68A(3):411–422, 2003.

[100] J. D. Weiland, D. J. Anderson, and M. S. Humayun. In vitro electrical properties for iridium oxide versus titanium nitride stimulating electrodes. *IEEE Trans. Biomed. Engr.*, 49(12):1574–1579, Dec 2002.

[101] B. L. Wessel, P. A. Roche, M. Sun, and R. J. Sclabassi. Optimization of an implantable volume conduction antenna. In *IEEE EMBS '04*, pages 4111–4114, 2004.

[102] B. L. Wessel, R. J. Sclabassi, P. A. Roche, and M. Sun. Analytical and numerical optimization of an implantable volume conduction antenna. In *30th Annual Northeast Bioeng. Conf.*, pages 29–30, Apr 2004.

[103] K.D. Wise. Silicon microsystems for neuroscience and neural prostheses. *IEEE Engr. in Medicine and Biology Magazine*, 24(5):22–29, Sep-Oct 2005.

[104] K.D. Wise, D. J. Anderson, J. F. Hetke, D. R. Kipke, and K. Najafi. Wireless implantable microsystems: High-density electronic interfaces to the nervous system. *Proc. of the IEEE*, 92:76–97, 2004.

[105] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G Pfurtscheller, and T.M. Vaughan. Braincomputer interfaces for communication and control. *Clin. Neurophysiol.*, 113:767–791, 2002.

[106] T. Yamamoto and Y. Yamamoto. Electrical properties of the epidermal stratum corneum. *Med. Biol. Eng.*, 14:151–158, 1976.

[107] B. Ziaie, S. C. Rose, M. D. Mardin, and K. Najafi. A self-oscillating detuning-intensive class-e transmitter for implantable microsystems. *IEEE Trans. Biomed. Engr.*, 48:397–400, 2001.