2006-01-17

A Framework for an Implantable Wireless Pressure and Volume Sensor Focusing on the Diagnosis and Treatment of Shunt Failure in Hydrocephalus Patients

Donald Monte Wichern
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Electrical and Computer Engineering Commons

A FRAMEWORK FOR AN IMPLANTABLE WIRELESS PRESSURE AND

VOLUME SENSOR FOCUSING ON THE DIAGNOSIS AND

TREATMENT OF SHUNT FAILURE IN

HYDROCEPHALUS PATIENTS

by

Donald M. Wichern

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Brigham Young University

April 2006

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Donald M. Wichern

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

_____   _____
Date   Mark L. Manwaring, Chair

_____   _____
Date   Travis E. Oliphant

_____   _____
Date   David J. Comer

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Donald M. Wichern in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

_____   _____
Date   Mark L. Manwaring
                                                         Chair, Graduate Committee

Accepted for the Department

                                                         _____
                                                         Michael A. Jensen
                                                         Graduate Coordinator

Accepted for the College

                                                         _____
                                                         Alan R. Parkinson
                                                         Dean, Ira A. Fulton College of Engineering
                                                         and Technology

ABSTRACT

A FRAMEWORK FOR AN IMPLANTABLE WIRELESS PRESSURE AND

VOLUME SENSOR FOCUSING ON THE DIAGNOSIS AND

TREATMENT OF SHUNT FAILURE IN

HYDROCEPHALUS PATIENTS

Donald M. Wichern

Department of Electrical and Computer Engineering

Master of Science

The framework for a permanently implantable wireless compliance sensor was developed and validated using laboratory experiments. The proposed sensor would measure the intracranial pressure and fluid volume in the brain and return this information to a monitoring device. The designed sensor received power remotely from the monitoring device negating the need for an implanted power source. Impedance measurement estimation techniques were suggested, studied and applied to the compliance sensing system. A new impedance measurement technique, accounting for multiple variability in the domain, was developed. An extensive simulation environment was designed and used to develop the laboratory experiments and hardware. The laboratory experiments validated the theory and simulation, proving the concept. Future work was suggested including the next step in the design process and possible research directions for subsequent theses.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the years, a basic problem in medicine has been accurate and timely diagnosis of disease in order to provide quick and effective treatment. Both doctors and patients desire a decrease in the number of diagnoses mistakes and an increase in treatment response which has led to an enormous amount of research into smaller, faster, more powerful, more accurate and more useful sensing systems capable of augmenting physicians control over the treatment process. This desire is especially apparent in the neurosurgery field, where the delicate nature of the brain motivates the development of extremely sensitive diagnosis equipment maximizing information outflow while minimizing interaction with the brain. Hydrocephalus is one disease that has had specialty sensors created just for solving problems with its treatments. When a patient suffers from this disease, a shunt, or drainage tube, is placed in the brain. The tube contains a valve that is pressure sensitive, it opens when the pressure in the brain exceeds a certain threshold. The shunt and valve are permanently implanted in the patient and are intended to be a permanent solution. Unfortunately, over time, the shunts tend to fail. The problem is usually caused by one of two scenarios, either the shunt opening in the brain becomes clogged by scar tissue, or the valve controlling fluid outflow fails. In either of these situations, the physician must diagnose the cause of the failure and determine the best course of action to rectify the problem. Often, the diagnosis process requires that the physician implant additional sensors in to the patient's brain. Avoiding or eliminating the additional surgery is the motivation behind the research discussed in this paper.

The additional sensors implanted into the patients brain usually consist of a volumetric and pressure device which lets the physician know the absolute pressure in the brain as well as the brain compliance, a measure of brain health. This paper develops a combined pressure and volume sensing system with the intent that it be refined into a permanently implantable, remotely powered device integrated into the shunt. This type of sensor would allow swifter diagnosis by the physician and eliminate the additional surgeries required to place the integrated sensors.

This paper begins by explaining the disease known as hydrocephalus, covering current diagnosis techniques and exploring current research into permanently implantable sensors. It continues by developing the theory behind the combined pressure and volume sensing device capable of returning the brain compliance and proposing two estimation techniques, either of which could be used to extract information from the sensor. The paper describes the experiments validating the idea, compares the measurement techniques and suggests the types and sensitivities of the sensors necessary to finish the project. It also details the hardware constructed and proposes the next steps in creating the finished system. It finishes by analyzing its results and looking forward to future research.

The key contributions and advancements presented in this paper include a new technique for measuring simultaneously changing impedances across a changing wireless boundary. Generally, with knowledge of the terminal characteristics, it is impossible to calculate the value of two unknown impedances in a system. The technique developed in this thesis uses prior information of the system to develop a method for determining unknown system impedances using measurements taken from the terminals of a monitoring system inductively coupled to the changing impedance values.

Chapter 2

Problem Statement

Effective and timely medical treatment relies on accurate diagnosis by physicians. Physicians desire safer, quicker and more accurate diagnostic tools to aid them in disease treatment. All diagnostic tools require some type of sensing device to sample real world data and import it into a computing environment for processing.

2.1 Sensors and Sensing

Doctors, scientists, engineers and researchers spend a large portion of their time collecting and analyzing real-world data using electronic sensors and computing systems. Sensors sample some event, convert it to an electronic representation and send it to a computer for analysis, storage and display. One area in which sensing has proven especially crucial is the medical field. Medical sensors vary in capability, but from simple temperature and heart-rate monitors to complicated EKG readings sensors accurately measure pressure, temperature, particle concentrations and flow rates. Physicians rely on sensors to help them diagnose and treat ailments and disease. This paper focuses on developing a sensor to help neurosurgeons treat hydrocephalus.

2.2 An Introduction to the Disease of Hydrocephalus

Hydrocephalus, or water on the brain, is a disease caused by an abnormal buildup of cerebrospinal fluid (CSF) in the ventricles of the brain. The excess fluid increases pressure and can compress and damage the brain. Hydrocephalus can arise before birth or any time afterward. It may be caused by any of a number of factors including birth defects, hemorrhaging in the brain, infection, meningitis, tumor, or

(a)   (b)
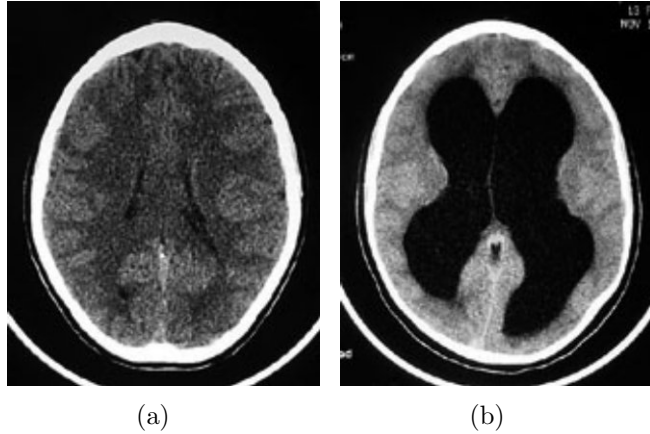
Figure 2.1: Two images of the brain showing (a), healthy brain tissue with slit-like vetricles, and (b), compressed brain tissue with swollen, malformed ventricles resulting from acute hydrocephalus.

head injury. Most forms of hydrocephalus are the result of obstructed CSF flow in the ventricular system. Figures 2.1(a) and 2.1(b) show images of a normal, healthy brain and a brain suffering from hydrocephalus. In Figure 2.1(b), the ventricles are enlarged and asymmetrical indicating an abnormal CSF buildup.

Normally CSF is in constant circulation and has a number of important functions: it acts as a protective cushion against injury for both the brain and spinal cord, contains nutrients and proteins necessary for normal brain function and carries waste products away from the surrounding tissues. The brain is a balanced system and more is not necessarily better. The increased pressure buildup due to hydrocephalus pushes the brain tissue outward compressing it against the skull causing nausea, headaches, blurred vision and seizures. In extreme instances, the pressure buildup can tear connective tissue in the brain resulting in brain damage or death. Often, before tearing occurs, the pressure in the brain is greater than the blood pressure of the patient which prevents the heart from pumping blood into the brain resulting in brain coma or death. In young children, with developing brains and soft craniums, it can cause severe retardation and significantly deform the skull as well.
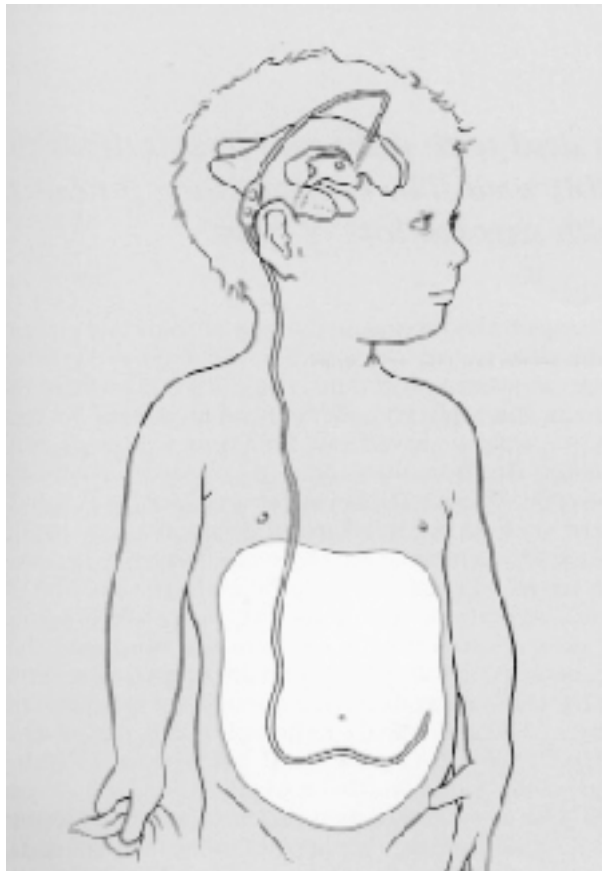
Figure 2.2: Ventriculoperitoneal (VP) shunt and reservoir system installed in a patient with hydrocephalus. The shunt runs subdermally from the cranium to the abdominal cavity.

2.3 Shunts and Reservoirs

To avoid the damage caused by hydrocephalus, a neurosurgeon installs a shunt to drain the excess CSF into the patient's abdominal region. Figure 2.2 shows a shunt running from the brain ventricle, to a reservoir, and then off down to the stomach. The shunt reservoir contains a pressure valve that automatically opens when the ICP exceeds a set value. As the patient ages, the pressure valve is adjusted so that it requires greater and greater pressure to open, as a result, the brain learns how to handle the excess CSF and is eventually weaned off the shunt system.

Unfortunately, shunts tend to fail over time. The most common type of shunt failure occurs in the brain ventricle when scar tissue grows over the shunt opening. Excess CSF builds up and the patient experiences nausea, headaches and dizziness.

The second most common type of failure occurs when the valve drains too much CSF resulting in an unacceptably low pressure. The symptoms, in this case, are the same as those for increased pressure, namely, nausea, headache and dizziness.

External symptoms do not help physicians diagnose shunt problems. Diagnosis requires an additional measurement giving the physician enough data to make an accurate judgment.

2.4   ICP and Brain Compliance as Metrics for Brain Health

Intracranial pressure (ICP) measurements are a solid metric for physicians helping them determine how well a brain is doing. ICP provides an absolute, unambiguous pressure reading informing the physician about shunt operation and any problems it may have developed.

In addition to using an absolute ICP reading, many physicians also use brain compliance as a metric for determining the health of a brain. Brain compliance, or simply compliance, is the relationship between ICP pulse pressure (ICP_{pp}) and the mean ICP (ICP_m) [1]. As the heart pumps blood into the brain, the additional pressure caused by incoming blood increases ICP creating an ICP_pp. The additional pressure spreads through the brain either raising or lowering the overall ICP. This profusion is easily measured by measuring water profusion, or volume, in the brain.

Figure 2.3 shows two common pulsitile waveforms representing brain compliance. The first is a healthy patient with normal ICP, the second is a patient with greatly increased ICP caused by inverting the patient on a tilt table. Notice the relationship between the pulsitile waveforms in the figures. In one, the brain pulse waveforms are well formed, similar in shape and easy to interpret, in the other, the waveforms are malformed and the venous waveform lags the pulse waveform by a noticable gap.

Many physicians and researchers use brain compliance in conjunction with ICP when diagnosing causes behind shunt failures [1], [2], [3], [4].

| (a) Healthy brain | (b) Increased ICP |
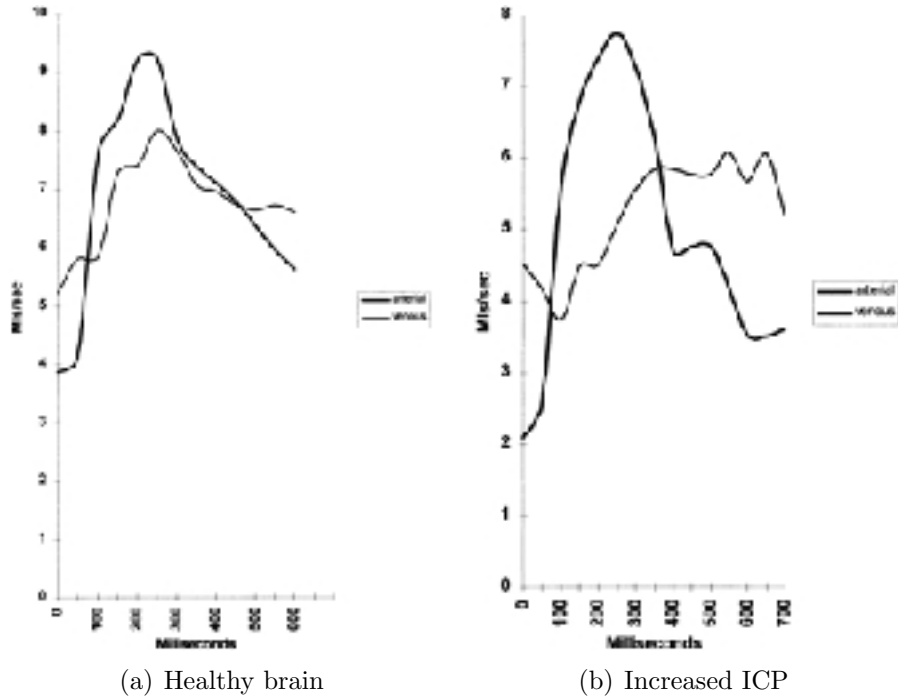
Figure 2.3: Illustration of the compliance in a healthy brain compared to the compliance of a brain with abnormal ICP.

2.5   Current Implantable Sensors

Patients suffering from neurological diseases such as hydrocephalus often need their ICP monitored on a regular basis. The common ICP monitoring approach involves drilling a hole in the patient's skull, inserting a pressure transducer into a brain ventricle, securing the sensor with a bolt and clamp and connecting the external wire to a monitoring device. This approach offers reliability and accurate calibration, and it is easy use, but it does present some risks and problems, namely, repeated surgeries, infection and lack of mobility.

2.6   Extension to Wireless Sensing Application

A logical extension, using an implanted permanent wireless ICP sensor offers all the same advantages and eliminates the risks and problems. By eliminating the disadvantages of the traditional ICP monitoring sensor, the wireless ICP sensor offers

advantages in diagnostic response time, out-patient or remote monitoring and patient safety [5].

A number of researchers are working on implantable sensors and they have focused specifically on implantable ICP sensors [6] [7] [8]. All of these implantable sensors exist in two domains, an analog information gathering domain and a digital information transfer domain. Most sensors attempt to move as quickly as possible from the analog domain to the digital domain by incorporating as much digital electronics as possible into the implant [9] [10].

Most implantable ICP sensors attempt to return an absolute pressure reading. Unfortunately, over time implanted absolute sensors experience a loss in accuracy called drift and they require calibration (another surgery) to correct this drift. Much research focuses on developing sensors immune to long term drift [11]. Fortunately, using brain compliance does not require an absolute pressure reading. The brain compliance, or stiffness, gives the physician information on the ICP independent of an absolute pressure reading because the technique uses a differential input offered by the relationship between ICP_{pp} and ICP_m.

A compelling argument exists for the development of an ICP sensor system that provides a pressure reading close to absolute pressure and a compliance reading that allows the attending physician to effectively diagnosis brain health in hydrocephalus patients. The design would avoid the increasing complexity of contemporary implantable sensors by minimizing the number of components in the implant and address the problems associated with calibration drift by providing a brain compliance reading in addition to an absolute pressure reading. The sensor could be integrated into a permanent shunt for long-term usage.

2.7 Conclusion

Collecting clinical evidence of shunt dysfunction usually involves implanting some sort of ICP sensor into the brain, taking readings, determining point of failure and fixing the problem. The goal of this research is to eliminate the repeated surgeries necessary when implanting an ICP sensor in order to diagnose the problem. By inte-

grating a pressure sensor and compliance meter into an existing shunt, the secondary surgery is eliminated producing fewer surgeries and quicker diagnosis. Both provide advantages to doctor and patient, especially in high-risk or time-critical situations.

The purpose of the research presented in this thesis is to develop a sensing and monitoring system for an implanted compliance meter. The meter must collect both pressure and volume information from the brain and transmit that data back to a monitoring device for analysis and display. This thesis develops the monitoring system and uses it to return information from a model of the compliance sensor. It explains the operating theory of the system, details simulation and experimental results, discusses the system's advantages and disadvantages and suggests areas for further research and study. In the course of designing the monitoring system, it also develops a new method of measuring impedances accounting for the issue of simultaneously changing impedances over an unstable inductive link.

Chapter 3

Theory of Operation

This chapter develops the theory and mathematical background behind the compliance sensing and monitoring system discussed in the previous chapters. It covers the sensor types comprising the system and their operation, wireless inductive coupling and inductor characteristics, reflected impedance and its connection to the sensor, an impedance measurement technique and estimating circuit parameters from impedance measurements.

3.1 The Proposed Implant and Monitoring System

The wireless compliance sensor consists of a monitoring system and a sensing system. Figure 3.1 shows an overview of the entire system. The monitoring device powers and communicates with the implanted sensor. It powers the sensor using inductive coupling between its coil and the sensor coil. It communicates with the sensor by monitoring the impedance it 'sees' looking into its coil. As the sensor detects changes in compliance, it communicates these changes by modulating the impedance it reflects back to the monitoring system.

The sensor consists of a minimal number of components and is constructed as simply as possible. One of the purposes of the research is to make the sensor as cheap and small as possible. As a result, all of the data processing and interpretation takes place in the monitoring system. The monitoring system is much more complicated and a majority of the theory developed around the compliance monitoring system focuses on its operation.
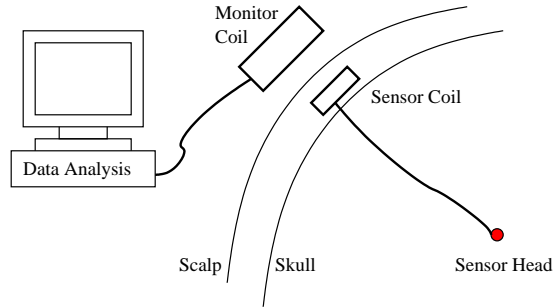
Figure 3.1: Overview of the compliance sensor showing the arrangement of the implant and monitoring systems.

3.2 The Sensors

The purpose of this research is to develop a method of measuring brain compliance using a permanently implanted device. In order to measure compliance, the implanted device must measure the ratio of fluid pressure to fluid volume in the brain. These two measurements require two sensors, one to measure pressure and one to measure volume. The compliance system uses a resistive strain gauge to measure pressure and a capacitive water sensor to measure volume.

3.2.1 The Resistive Model

Currently the most common method for measuring ICP involves implanting a resistive strain gauge in the brain and connecting it to a driving circuit to calculate the absolute ICP. One of the more popular strain gauges on the market today has a base resistance set near 1 kΩ with a 0.3 Ω change in resistance per millimeter mercury. It uses simple DC amplification to read the sensor. The compliance sensor suggested in this paper bases its strain gauge around a 50 Ω resistance and uses RF analysis to read the sensor. The 50 Ω base was chosen because of the traditional use in impedance matching in antenna and communication networks. The strain gauge is simply a device that changes its resistance based on the deformation caused by external pressure.
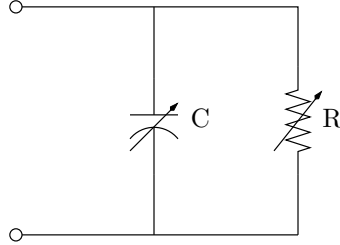
Figure 3.2: The equivalent circuit for the resistive fluid pressure and capacitive fluid volume sensors.

3.2.2 The Capacitive Model

Two oppositely charged parallel plates create a capacitor, but this is not the only valid capacitive arrangement. Two oppositely charged parallel wires can also form a capacitor. A capacitive water sensor is created when two parallel wires are placed in a wet environment. As the saturation of the medium changes, the capacitance between the wires changes. Two parallel wires embedded in the brain act as a volume sensor for CSF in the brain. As the brain engorges with fluid, the capacitance of the wires changes creating a measurable effect.

3.2.3 The Compliance Sensor Model

Figure 3.2 shows the equivalent circuit of the compliance sensor used in subsequent analysis. The model consists of a variable resistive element representing the contribution of the strain gauge in parallel with a variable capacitive element which represents the contribution of the water sensor.

3.3 Mutual Inductance

Permanent implanted sensors commonly communicate to the outside world using some type of inductive link. Mutual inductance is the phenomenon observed when alternating current and magnetic fields in a coil create or induce similar alternating currents and magnetic fields in a nearby coil. The brain compliance sensor system uses two coils, one in the implanted sensor and one in the monitor device. Figure 3.3 shows the sensors connected to the sensor-side inductor as well as a second inductor representing the monitor-side coil.
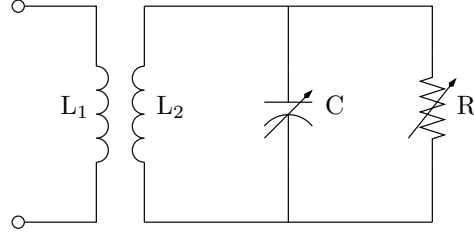
Figure 3.3: The water and pressure sensor circuit attached to the sensor side coil in proximity to the monitor side coil

3.3.1 Spiral Pancake Inductors

An easy way to manufacture consistently sized pancake inductors is to fabricate them on printed circuit boards (PCBs). Geometries and sizes suggested by [12] were a starting point for the spiral inductors manufactured and tested during this research. The relatively low quality factor (Q) expected from a PCB pancake inductor forces any meaningful system modeling equations to use a non-ideal model for inductive components. Specifically, $Z_L = j\omega L$ in any equations will be modeled by the addition of a series resistor representing real resistive losses in the coil,

$$Z_L = R + j\omega L. \tag{3.1}$$

3.3.2 The Sensor and Monitor Inductor Models

PCB inductors have a relatively high resistance to inductance ratio. It follows that modeling them requires more than simply assuming an ideal inductor. Both the sensor and monitor coils in the compliance meter are better represented by an inductor in series with a resistor modeling the device parameters. Figure 3.4 shows the updated sensor and coil system when the inductors from Figure 3.3 are replaced by models better representing the actual inductor impedances.

3.4 Reflected Impedance

The resistive and capacitive sensing elements implanted in the patient will communicate pressure and volume information by varying the impedances measured across them. Accurate impedance measurements are a crucial element in acquiring the information provided by the sensors. This section derives a mathematical model

14
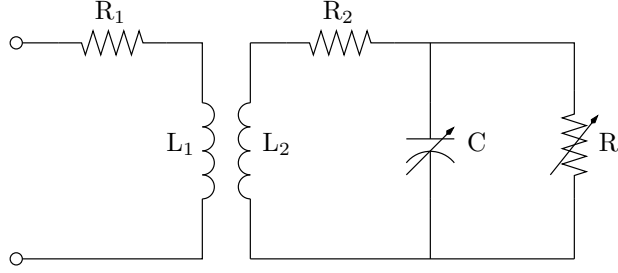
Figure 3.4: The water and pressure sensor equivalent circuit with an attached sensor-side coil adjusted to reflect an actual inductor model.
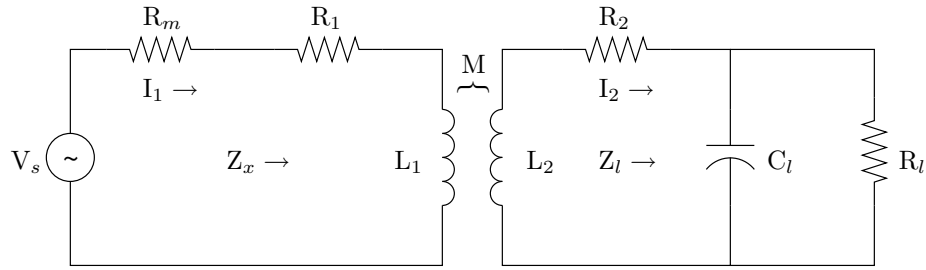
Figure 3.5: Equivalent circuit of the compliance sensing and monitoring system. M represents the inductive coupling between L_1 and L_2.

based on the equivalent circuit for the compliance sensing and monitoring system from Figure 3.4. Figure 3.5 illustrates a model showing a generalized description of what this derivation calculates, the impedance of load, which equals the impedance of the inductively coupled capacitive and resistive sensors reflected back to the monitoring system plus the impedance of the monitor-side coil. M, in Figure 3.5, represents the inductive coupling between L_1 and L_2 and is equal to $k\sqrt{L_1 L_2}$.

By Ohm's Law, impedance is equal to voltage divided by current. In order to calculate the impedance Z_x, expressions for V_s and the loop currents I_1 and I_2 must be derived. To begin, the phasor equivalents for all complex devices in the system are computed. The phasor equivalents are used because the driving voltage, V_s, is a sinusoidal signal allowing for this type of analysis. The reasons for sinusoids become apparent during the derivation. The phasor equivalents are

$$L_1 \quad = \quad j\omega L_1,$$

15

$$M = j\omega M,$$

$$L_2 = j\omega L_2,$$

$$C_l = \frac{1}{j\omega C_l}.$$

Next, the voltages around each loop are summed. The voltages around the first loop sum to

$$V_s = (R_m + R_1 + j\omega L_1)I_1 - j\omega M I_2 \tag{3.2}$$

and around the second loop the voltages sum to

$$0 = -j\omega M I_1 + (R_2 + j\omega L_2 + Z_l)I_2. \tag{3.3}$$

Z_l in 3.3 equals the parallel combination of R_l and C_l. For ease in notation let

$$Z_{11} = R_m + R_1 + j\omega L_1 \tag{3.4}$$

and

$$Z_{22} = R_2 + j\omega L_2 + Z_l. \tag{3.5}$$

Substituting 3.4 and 3.5 into 3.2 and 3.3 produces simpler equations to use when solving for the mesh currents I_1 and I_2. The modified mesh equations are

$$V_s = Z_{11}I_1 - j\omega M I_2 \tag{3.6}$$

and

$$j\omega M I_1 = Z_{22}I_2. \tag{3.7}$$

Solving for I_1 and I_2 gives

$$I_1 = V_s \frac{Z_{22}}{Z_{11}Z_{22} + \omega^2 M^2} \tag{3.8}$$

and

$$I_2 = \frac{j\omega M}{Z_{22}} I_1 = V_s \frac{j\omega M}{Z_{11}Z_{22} + \omega^2 M^2}. \tag{3.9}$$

The overall impedance of the circuit Z_{total} is simply the voltage of the source divided by the current entering the device,

$$Z_{total} = \frac{V_s}{I_1} = Z_{11} + \frac{\omega^2 M^2}{Z_{22}}. \tag{3.10}$$

The impedances of the coupled sensor elements and monitor side coil can now be calculated as the difference between Z_{total} and the source resistance R_m,

$$Z_x = Z_{total} - R_m = R_1 + j\omega L_1 + \frac{\omega^2 M^2}{R_2 + j\omega L_2 + Z_l}, \qquad (3.11)$$

giving an expression for Z_x containing only the impedances represented by the sensor elements and monitor coil. The third term of 3.11 represents the impedance of the sensor system reflected to the monitor side of the inductively coupled system.

3.5 Impedance Measurement

One common technique for measuring impedance across a range of frequencies uses a DIP meter to calculate the current and power consumptions of at load. While this technique provides an accurate measure of the magnitude of the impedance, it does not provide any information about the phase. A technique suggested by [15] uses circuit relationships and estimation theory to calculate both the magnitude and phase of the unknown impedance.

3.5.1 A New Impedance Bridge

Ohm's law states that understanding an impedance requires knowledge of the voltage and current characteristics of the device under test. The impedance measurement technique described here discusses a novel and relatively simple adaptation of a traditional impedance measurement device known as an impedance bridge [15].

Traditional impedance bridges work in one of two ways, in the first, the impedance being measured is compared against an impedance under control by the device. The voltages and currents across both devices are measured and the device changes the impedance it controls until the voltages and currents across both the known and unknown devices are the same. At this point the known impedance matches the unknown impedance and the unknown values are provided by the device. In the second impedance bridge, the device modulates the current flowing through the known impedance until the voltages across the devices match. When they match, the voltage and current across the known impedance give information about the unknown impedance.
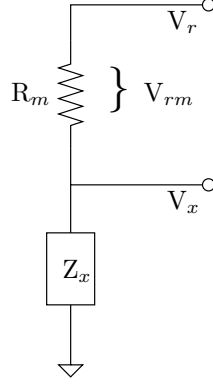
Figure 3.6: A circuit demonstrating the setup of a novel impedance bridge measurement device.

The impedance bridge in Figure 3.6 uses a completely different technique to solve for the impedance. It does not modulate the voltage and current over the unknown device or the known impedance, instead, it measures the input voltage V_r and the impedance modulated voltage V_x and calculates the load impedance using the equations resulting from the following analysis.

First, the voltage across the reference resistor is defined as

$$V_{rm} = V_r - V_x \qquad (3.12)$$

where V_{rm} is assumed to equal an arbitrary sine wave with amplitude A and frequency ω_o, that is,

$$V_{rm} = A \sin \omega_o t. \qquad (3.13)$$

V_x is defined to be

$$V_x = B \sin(\omega_o t + \phi), \qquad (3.14)$$

a sine wave related to V_{rm} by the same frequency but with differing amplitude and phase introduced by the load impedance Z_x. Using the identity $\sin(a + b) = \sin a \cos b + \cos a \sin b$, V_x can be rewritten as

$$
\begin{aligned}
V_x &= B \left(\sin \omega_o t \cos \phi + \cos \omega_o t \sin \phi \right) \\
&= B \sin \omega_o t \cos \phi + B \cos \omega_o t \sin \phi.
\end{aligned}
\qquad (3.15)
$$

18

Let W_1 and W_2 be two weighting variables. Let

$$W_1 = \frac{B}{A} \cos \phi \tag{3.16}$$

and

$$W_2 = \frac{B}{A} \sin \phi. \tag{3.17}$$

Substituting 3.16 and 3.17 back into 3.15 gives

$$V_x = W_1 A \sin \omega_o t + W_2 A \cos \omega_o t. \tag{3.18}$$

This expression for V_x is mathematically related to the expression for V_{rm} in 3.13. It shows that V_x is a weighted, phase shifted version of V_{rm}.

With expressions for V_{rm} and V_x, the next step is to calculate an expression for Z_x. Kirchhoff's current law forces the current flowing through the reference resistor to equal the current flowing through the unknown impedance. First the equations for the currents are written as

$$I_{rm} = \frac{V_{rm}}{R_m} = \frac{V_x}{Z_x} = I_x. \tag{3.19}$$

Next, the equation is solved for Z_x giving

$$Z_x = R_m \frac{V_x}{V_{rm}}, \tag{3.20}$$

and then the assumptions from 3.13 and 3.14 are substituted in to give

$$Z_x = R_m \frac{B \sin(\omega_o t + \phi)}{A \sin(\omega_o t)}. \tag{3.21}$$

Now, using a phasor transformation, Z_x is rewritten to be

$$Z_x = R_m \frac{B}{A} \angle \phi. \tag{3.22}$$

The real and imaginary parts of Z_x are separated by converting from polar to rectangular notation and writing

$$Z_{real} = R_m \frac{B}{A} \cos \phi \tag{3.23}$$

and

$$Z_{imag} = R_m \frac{B}{A} \sin \phi. \tag{3.24}$$
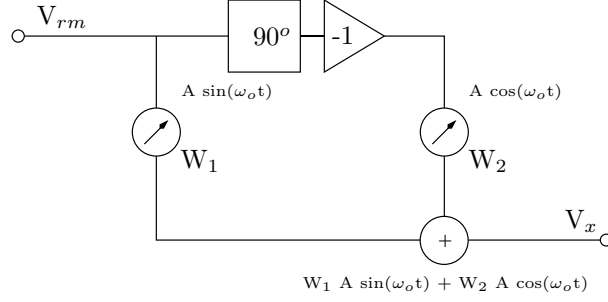
Figure 3.7: A system equation describing the process of discovering the unknown weights W_1 and W_2.

Substituting 3.16 and 3.17 into 3.23 and 3.24 and rewriting Z_x using standard rectangular coordinate notation gives

$$Z_x = R_m W_1 + j R_m W_2, \tag{3.25}$$

an expression for Z_x in terms of the known resistance, R_m, and the unknown weights. Solving 3.25 for W_1 and W_2 will give an expression for the impedance Z_x.

Knowing the input V_{rm} and the output V_x as well as the relationship given in 3.18, W_1 and W_2 can be computed using one of several algorithms. Figure 3.7 illustrates a fairly straightforward method. The system input, $V_{rm} = A \sin \omega_o t$ is delayed and inverted to create $A \cos \omega_o t$. The two signals are respectively multiplied by W_1 and W_2 and then added together. This new signal equals V_x. The system is described by

$$V_x(t_k) = W_1 A \sin \omega_o t_k + W_2 A \cos \omega_o t_k. \tag{3.26}$$

It is much easier to find a solution to the system after representing it as a matrix. Equation 3.26 is rewritten as

$$\begin{bmatrix} A \sin \omega_o t_0 & A \cos \omega_o t_0 \\ A \sin \omega_o t_1 & A \cos \omega_o t_1 \\ A \sin \omega_o t_2 & A \cos \omega_o t_2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \begin{bmatrix} V_x(t_0) \\ V_x(t_1) \\ V_x(t_3) \\ \vdots \end{bmatrix}, \tag{3.27}$$

which is simplified to give

$$\mathbf{A}\mathbf{w} = \mathbf{y}. \tag{3.28}$$

20

$\mathbf{A}$ is an n by 2 matrix whose first column contains samples of V_{rm} at times t and whose second column holds an inverted, 90° delayed copy of V_{rm}. $\mathbf{W}$ is a 2 by 1 vector containing W_1 and W_2 and $\mathbf{y}$ is a column vector holding samples of V_x at times t. Finding the unknown weights requires solving 3.27 for $\mathbf{w}$. Using the pseudo-inverse to solve 3.28, $\mathbf{w}$ can be written as

$$\mathbf{w} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{y}, \tag{3.29}$$

where $\mathbf{A}^H$ is the conjugate transpose of $\mathbf{A}$ [16]. Using 3.29, $\mathbf{w}$ can be directly computed from $\mathbf{A}$ and $\mathbf{y}$. Substituting the weights back into 3.25 produces a numerical expression for Z_x.

3.5.2 Application to the Compliance Monitoring System

Computing the entire load impedance Z_x only solves one-half of the problem. Applying the impedance measurement to the compliance sensor requires relating the impedance of the entire system to the load impedance represented by the parallel combination of R_l and C_l. Knowing the topology of the circuit model in Figure 3.5 enables calculation of the load parameters, however, the parameters are difficult to directly compute. The following section describes the difficulty involved and also suggests two methods of estimating the unknown circuit values.

3.6 Deriving Circuit Parameters from Impedance Measurements

The derivation of the reflected impedance in 3.11 provides a complex equation for Z_x containing three unknown variables, the coefficient of inductive coupling (M), the resistive load (R_l) and capacitive load (C_l). For a single frequency 3.11 provides information for two equations and three unknowns. It is impossible to find a unique solution to this type of under determined equation. Luckily, more data can be garnered by measuring the impedance at additional frequencies. Each additional measurement provides more information and helps drive the system towards a unique solution.

3.7 Noisy Measurements

Even after collecting more data, brute force data crunching does not solve the problem. Any measurement of real-world data contains some level of noise. In complex calculations, such as those used to determine M, R_l and C_l, noise must be taken into consideration or its effect will accrue and significantly affect any results. The estimation techniques described in this section use some form of noise minimization technique, either least squares (LS) or total least squares (TLS), to attempt to filter out unwanted variations caused by imperfect measurements.

3.8 Using Re-parameterization to Calculate R_l, C_l and M

The TLS algorithm solves linear systems of equations. In the case of the compliance sensor, it places restrictions on the device operations. For correct operation, no component in the system may exhibit frequency dependent operation, or if it does, the frequency dependency must be separable from the component value. Also, the system of equations fed into the TLS estimator must be linear. The reason no devices in the system are allowed to have frequency dependencies is because impedance measurements are taken at multiple frequencies and in order to find a solution, the capacitance or resistance at each frequency must be the same [17].

3.8.1 Deriving the Estimator Equation

The impedance bridge returns an expression for Z_x in rectangular form. In order to infer the load values, 3.11 must be split into its real and imaginary parts and compared to 3.25. First, Z_l in 3.11 is replaced by the parallel combination of R_l and C_l,

$$Z_x = R_1 + j\omega L_1 + \omega^2 M^2 \frac{1}{R_2 + j\omega L_2 + R_l || C_l}. \tag{3.30}$$

Next, Z_x is replaced by $R_z + jI_z$, a variable representing the real and imaginary parts of the measured impedance giving

$$R_z + jI_z = R_1 + j\omega L_1 + \omega^2 M^2 \frac{1}{R_2 + j\omega L_2 + R_l || C_l}. \tag{3.31}$$

Subtracting $R_1 + j\omega L_1$ from both sides gives

$$R_z - R_1 + jI_z - j\omega L_1 = \omega^2 M^2 \frac{1}{R_2 + j\omega L_2 + R_l||C_l}, \tag{3.32}$$

which is inverted to give

$$\frac{1}{R_z - R_1 + jI_z - j\omega L_1} = \frac{1}{\omega^2 M^2} \left(R_2 + j\omega L_2 + R_l||C_l \right). \tag{3.33}$$

Now the term

$$\frac{1}{R_z - R_1 + jI_z - j\omega L_1} \tag{3.34}$$

is replaced by a new complex variable representing the admittance of the system. Replacing 3.34 with $R_c + jI_c$ gives

$$R_c + jI_c = \frac{1}{\omega^2 M^2} \left(R_2 + j\omega L_2 + R_l||C_l. \right) \tag{3.35}$$

Both sides of the equation are multiplied by $\omega^2 M^2$ and $R_l||C_l$ is expanded into its real and imaginary components. The resulting equation is

$$\omega^2 R_c M^2 + j\omega^2 I_c M_2 = R_2 + j\omega L_2 + \frac{\frac{1}{R_l} - j\omega C_l}{\frac{1}{R_l^2} + \omega^2 C_l^2}. \tag{3.36}$$

The term $\frac{1}{R_l^2} + \omega^2 C_l^2$ is multiplied to both sides and the entire equation is separated into two parts, one for the real components and one for the imaginary components. The real components are represented by the equation

$$\omega^2 R_c \frac{M^2}{R_l^2} + \omega^4 R_c M^2 C_l^2 = R_2 \frac{1}{R_l^2} + \omega^2 R_2 C_l^2 + \frac{1}{R_1}, \tag{3.37}$$

and the imaginary components are represented by the equation

$$\omega^2 I_c \frac{M^2}{R_l^2} + \omega^4 I_c M^2 C_l^2 = \omega L_2 \frac{1}{R_l^2} + \omega^3 L_2 C_l^2 - \omega C_l. \tag{3.38}$$

The next step is to re-parameterize 3.37 and 3.38 by replacing the unknown values M, R_l and C_l. For the re-parameterization, $\mathbf{x}$ is a vector whose components are

$$x_1 = \frac{M^2}{R_l^2},$$
$$x_2 = M^2 C_l^2,$$

$$
\begin{aligned}
x_3 &= \frac{1}{R_l^2}, \\
x_4 &= C_l^2, \\
x_5 &= \frac{1}{R_l}, \\
x_6 &= C_l,
\end{aligned}
\tag{3.39}
$$

and is then substituted into 3.37 and 3.38 to produce

$$
\omega^2 R_c x_1 + \omega^4 R_c x_2 = R_2 x_3 + \omega^2 R_2 x_4 + x_5
\tag{3.40}
$$

and

$$
\omega^2 I_c x_1 + \omega^4 I_c x_2 = \omega L_2 x_3 + \omega^3 L_2 x_4 - \omega x_6.
\tag{3.41}
$$

These are rewritten in matrix form to better describe the system,

$$
\mathbf{Ax} =
\begin{bmatrix}
\omega^2 R_c & \omega^4 R_c & -R_2 & -\omega^2 R_2 & -1 & 0 \\
\omega^2 I_c & \omega^4 I_c & -\omega L_2 & -\omega^3 L_2 & 0 & -\omega
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
= \mathbf{0}.
\tag{3.42}
$$

The system described by 3.42 is under-determined, meaning there are more unknowns than equations, and homogeneous, meaning the solution vector $\mathbf{b}$ contains only zero values. This type of system has multiple solutions. Once the equation becomes over-determined, as a result of numerous frequency measurements, the only solution calculable will be $\mathbf{x} = \mathbf{0}$, which is obviously not the correct answer. A viable solution is to create two estimators, one assuming a known C_l and the other assuming a known R_l and to simultaneously estimate parameter values until an intersection is found.

3.8.2 An Estimator Assuming Fixed C_l

When creating an estimator assuming a fixed C_l, the only unknowns in the system are M and R_l. Equations 3.37 and 3.38 are re-parameterized using M and R_l

24

as the only unknown values. Let $\mathbf{x}$ equal a vector having entries

$$
\begin{aligned}
x_1 &= \frac{M^2}{R_l^2}, \\
x_2 &= M^2, \\
x_3 &= \frac{1}{R_l^2}, \\
x_4 &= \frac{1}{R_l},
\end{aligned}
$$

(3.43)

which are substituted into 3.37 and 3.38 to give

$$
\omega^2 R_c x_1 + \omega^4 R_c C_l^2 x_2 = R_2 x_3 + \omega^2 R_2 C_l^2 + x_4
$$

(3.44)

and

$$
\omega^2 I_c x_1 + \omega^4 I_c C_l^2 x_2 = \omega L_2 x_3 + \omega^3 L_2 C_l^2 - \omega C_l.
$$

(3.45)

The equations are rewritten as the following matrices to give

$$
\mathbf{Ax} = \begin{bmatrix} \omega^2 R_c & \omega^4 R_c C_l^2 & -R_2 & -1 \\ \omega^2 I_c & \omega^4 I_c C_l^2 & -\omega L_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \omega^2 R_2 C_l^2 \\ \omega^3 L_2 C_l^2 - \omega C_l \end{bmatrix} = \mathbf{b}.
$$

(3.46)

Now the equation can be solved using TLS to find an estimate for $\mathbf{x}$. Section 3.8.4 covers the TLS solution and explains how the fixed C_l estimator incorporates into the process.

3.8.3   An Estimator Assuming Fixed R_l

When creating an estimator assuming a fixed R_l, the only unknowns in the system are M and C_l. The equations 3.37 and 3.38 are re-parameterized using M and C_l as the only unknown values. Let $\mathbf{x}$ equal a vector having entries

$$
\begin{aligned}
x_1 &= M^2, \\
x_2 &= M^2 C_l^2, \\
x_3 &= C_l^2, \\
x_4 &= C_l,
\end{aligned}
$$

(3.47)

which are substituted into 3.37 and 3.38 to give

$$\omega^2 R_c \frac{1}{R_l^2} x_1 + \omega^4 R_c x_2 = R_2 \frac{1}{R_l^2} + \omega^2 R_2 x_3 + \frac{1}{R_1} \tag{3.48}$$

and

$$\omega^2 I_c \frac{1}{R_l^2} x_1 + \omega^4 I_c x_2 = \omega L_2 \frac{1}{R_l^2} + \omega^3 L_2 x_3 - \omega x_4. \tag{3.49}$$

The equations are rewritten as matrices to give

$$\mathbf{Ax} = \begin{bmatrix} \frac{\omega^2 R_c}{R_l^2} & \omega^4 R_c & -\omega^2 R_2 & 0 \\ \frac{\omega^2 I_c}{R_l^2} & \omega^4 I_c & -\omega^3 L_2 & \omega \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{R_2}{R_l^2} + \frac{1}{R_1} \\ \frac{\omega L_2}{R_l^2} \end{bmatrix} = \mathbf{b}. \tag{3.50}$$

Now the matrix can be solved using TLS to find an estimate for $\mathbf{x}$. The next section covers the TLS solution and explains how the fixed R_l estimator incorporates into the process.

3.8.4   TLS Solution

The TLS solution is covered in great detail in [18] and is not derived here, but the pertinent solutions are shown. The TLS solves for $\mathbf{x}$ using the singular value decomposition (SVD) of the matrix

$$\mathbf{C} = [\mathbf{A} \ \mathbf{b}], \tag{3.51}$$

where $\mathbf{C}$ is the column concatenation of $\mathbf{A}$ and $\mathbf{b}$. The SVD is a diagonalization of a non-square matrix and has the form

$$\mathrm{svd}(\mathbf{C}) = \mathbf{U\Sigma V}^h. \tag{3.52}$$

The solution is simply

$$\mathbf{x} = -\frac{\mathbf{V}(1 : \mathrm{end} - 1, \mathrm{end})}{\mathbf{V}(\mathrm{end}, \mathrm{end})}. \tag{3.53}$$

If $\mathbf{V}(\mathrm{end}, \mathrm{end})$ is zero, no TLS solution exists for $\mathbf{A}$ and $\mathbf{b}$.

26
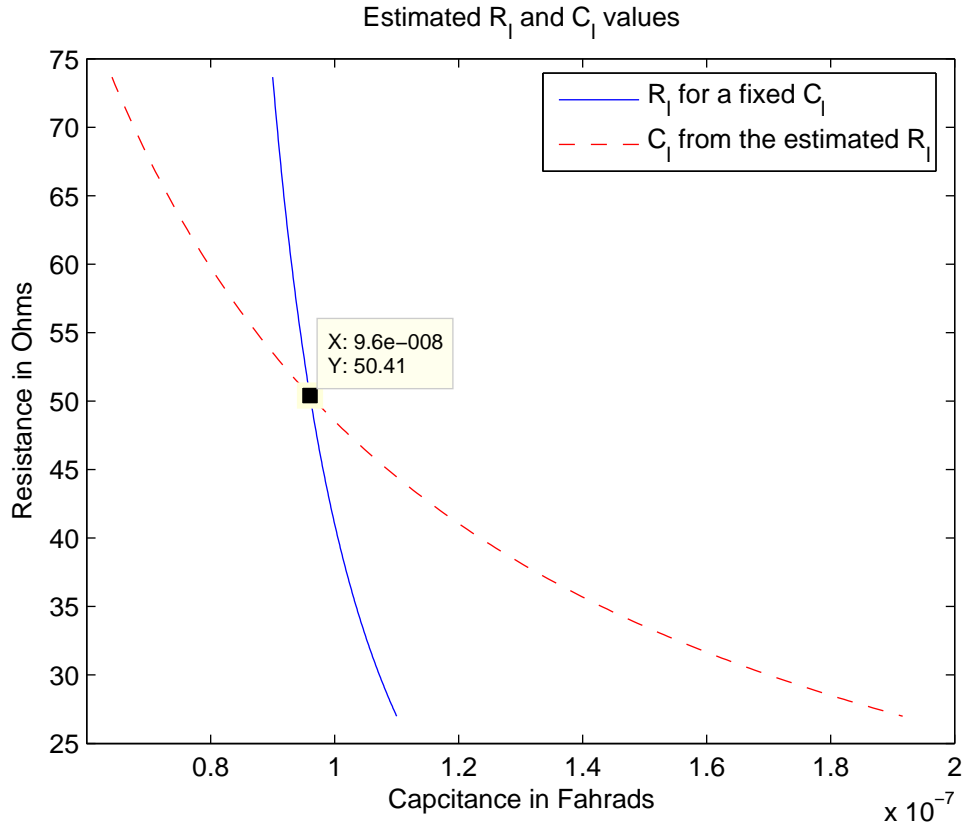
Figure 3.8: Two curves representing various estimates for R_l and C_l. The point of intersection represents the actual values of R_l and C_l.

3.8.5 Finding Estimates using the TLS Solution

With estimators for R_l and C_l, the next step is finding the best estimate for each. First, a number of values for R_l are estimated using assumed values for C_l. Next, the estimated R_l values are taken and used to estimate a new set of C_l values. The resulting curves are graphed and used to find the intersection of the data sets. The values at the intersection point represent the unknown circuit parameters in the compliance sensor. Figure 3.8 shows two curve estimates and the intersection point between them.

3.9 Using Careful Derivation to Minimize Noise

The derivations in sections 3.8.2 and 3.8.3 make no attempt to minimize the noise present in Z_x during linearization of 3.11. A prime example is present in 3.34

27

when the real and imaginary parts of Z_x are inverted to create a new complex number. When a complex number Z is inverted, the numerator and denominator are multiplied by the complex conjugate of Z and the result is

$$\frac{1}{Z} = \frac{Z^*}{|Z|^2}.$$ (3.54)

The new complex number was divided by the magnitude squared of the original number. Now assume that Z is some signal measured with noise present. The magnitude of this signal squared emphasizes the noise in the signal in all subsequent calculations.

The derivation in this section attempts to minimize noise in the signal by trying to avoid or minimize any multiplicative or exponential operations on the the measured input Z_x. In Chapter 4 the estimators developed in this section will be compared against the estimators of sections 3.8.2 and 3.8.3 in order to determine which type of estimator best solves for M, R_l, and C_l.

3.9.1 Deriving the Estimator Equation

Instead of splitting 3.11, which is repeated here for reference, into its real and imaginary parts, this estimator works with complex values in all its computations. Equation 3.11 is

$$Z_x = R_1 + j\omega L_1 + \omega^2 M^2 \frac{1}{R_2 + j\omega L_2 + Z_l}.$$ (3.55)

First, $R_1 + j\omega L_1$ and $R_2 + j\omega L_2$ are replaced by Z_{L1} and Z_{L2} giving

$$Z_x = Z_{L1} + \omega^2 M^2 \frac{1}{Z_{L2} + Z_l}.$$ (3.56)

Next, the fractional portions are cleared to give

$$(Z_x - Z_{L1})(Z_{L2} + Z_l) = \omega^2 M^2.$$ (3.57)

Treating Z_l as the unknown value, the equation is rewritten using the same matrix form used previously,

$$\mathbf{Ax} = (Z_x - Z_{L1})Z_l = (Z_{L1} - Z_x)Z_{L2} + \omega^2 M^2 = \mathbf{b} + c,$$ (3.58)

with the small addition of a constant representing the contribution $\omega^2 M^2$.

28

In the estimators developed in sections 3.8.2 and 3.8.3, one of the values C_l or R_l was assumed to be fixed while the other was estimated. Similarly, any solution to 3.58 will assume a value for M when calculating R_l and C_l.

It is important to note the form of $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{b}$. This is most easily illustrated by expanding the equations for a few values of ω. Remember that Z_x is a vector whose entries are measurements at different values of ω, thus, using an abbreviated notation,

$$
\begin{aligned}
a_1 x_1 &= b_1 + c, \\
a_2 x_2 &= b_2 + c, \\
&\vdots \\
a_m x_m &= b_m + c,
\end{aligned}
\tag{3.59}
$$

which implies that

$$
\mathbf{A} = \mathrm{diag}\begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix}, \quad
\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad
\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.
\tag{3.60}
$$

These equations can be solved by using the LS pseudo-inverse derived by [16] and used in Section 3.5.1 to solve for Z_x, namely,

$$
\mathbf{x} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{b}.
\tag{3.61}
$$

This solution assumes that c is a known constant. This is not the case. C is equal to $\omega^2 M^2$ and M is one of the unknowns in the system. Solving 3.61 requires a series of guesses over M to be fed into the equation and then checking the resulting $\mathbf{x}$ vector, which contains Z_l, to see if the guess was "good". The next chapter develops the estimator described in this section and produces an algorithm that checks the accuracy of the guesses returned from 3.61.

3.10   Conclusion

This chapter developed the theory and mathematical background behind the compliance sensing and monitoring system discussed in the previous chapters. It cov-

ered the sensor types comprising the system and their operation, inductive coupling and inductor characteristics, reflected impedance and its connection to the sensor, an impedance measurement technique and estimating circuit parameters from impedance measurements. It provided a solution to the parameter estimation problem and developed a graphical evaluation technique. The next chapter uses the theory provided here to design and develop the compliance monitoring system.

Chapter 4

Experimentation

This chapter describes the simulations and experiments validating the compliance sensor design. It covers the impedance bridge and parameter estimator Matlab models, discusses the considerations of micro-strip PCB design, describes the circuits and layouts of the compliance sensor and monitor, and explains the data collection techniques used during the experiment specifically focusing on the impedance bridge verification and calibration tests and the labview automation driving the system. The chapter finishes by analyzing the collected data and identifying key system characteristics such as statistical means and standard deviations.

4.1 Simulation

Accurate system simulation helps verify proper system operation. Often, problems in experiments can be found and fixed using the simulation. This section describes the simulation environment for the compliance sensor.

4.1.1 Matlab Model for the Impedance Bridge

The impedance bridge simulation models the bridge using a matlab script. The entire script is found in Appendix A.2. This section covers the highlights of the script and its verification.

Figure 4.1 shows the impedance bridge setup. In the script, the unknown impedance is simulated using the circuit equation derived in Section 3.4. The equation is repeated here for reference,

$$Z_x = R_1 + j\omega L_1 + \frac{\omega^2 M^2}{R_2 + j\omega L_2 + \frac{1}{\frac{1}{R_l} + j\omega C_l}}. \tag{4.1}$$

Table 4.1: Excerpt from `bridge.m` showing the creation of the simulation inputs V_r and V_x.

```
% Create the input and impedance modulated output
t_ = 0:1e-8:10e-6;
Vr = cos( w(k) * t_ - pi/2 );
[x,y] = pol2cart( -pi/2, 1 );
Vrs = x + i*y;
Vxs = Vrs * Zx(k)/(Zx(k) + Rm);
[a,r] = cart2pol( real(Vxs), imag(Vxs) );
Vx = r * cos( w(k) * t_ + a );
```
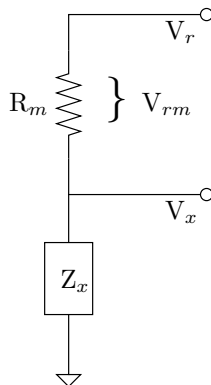

Figure 4.1: The impedance bridge setup for simulation and experimentation. V_r represents the reference signal, V_x represents the impedance modulated signal and V_{rm} is the difference between V_r and V_x representing the signal across the reference resistor.

The simulation script creates an input signal V_r and modulates it with Z_x to produce V_x. The modulation takes place in the phasor domain and is shown in Table 4.1.

The signal is re-sampled at a sampling rate equal to forty times the signal frequency. The reason for this is the creation of the signal $V_{rm} = W_1 A \sin \omega_o t + W_2 A \cos \omega_o t$ described in Section 3.5.1. Digitally delaying the signal exactly $\pi/2$ radians is easier when the delay equals an integer. In this case, the delay equals ten samples. The script finishes by calculating the impedance using the matrix pseudo-inverse described in 3.5.1. This process is excerpted from the script and displayed in Table 4.2.

Table 4.2: Excerpt from `bridge.m` showing the calculation of **W**, the unknown weights, using the pseudo-inverse of **A**.

```
% Calculate the voltage across the reference resistor
Vrm = Vr - Vx;
% Build the bridge estimator, A(:,1) is A sin wt and A(:,2) is
% A cos wt
A = [];
A(:,1) = Vrm(qu+1:length(Vrm));
A(:,2) = -1.*Vrm(1:length(Vrm)-qu);
% Calculate the bridge weigths, we can use the pseudo inverse
% or the TLS to find the solution.
W = inv( A' * A ) * A' * Vx(qu+1:length(Vx))';
% Build the estimated impedance array
Ze(k) = Rm * ( W(1) + i*W(2) );
```

Verifying the simulation is simple, comparing the impedance bridge output Z_e to the input Z_x shows that, in the absence of noise and analog to digital conversion error, the bridge exactly calculates the impedance of Z_x, making the difference between it and the error equal to

$$Z_e - Z_x = 0. \tag{4.2}$$

4.1.2 Matlab Model for the Parameter Estimator

The parameter estimator simulation models the estimator using a matlab script. The entire script is found in Appendix A.3. Portions of the script are displayed in tables 4.3 and 4.4 for clarity in explanation. The purpose of the estimator is to discover the values of the load resistance R_l and capacitance C_l. It does this by constructing two small estimators, one for R_l and one for C_l and using them together.

The simulation operates by first it creating a sample impedance Z_x using 4.1, just like the impedance bridge simulation. It then creates two estimators, one for a fixed C_l and one for a fixed R_l, and stores them in two sub-functions, `fixc` and `fixr` . Both functions are very similar and will be discussed together.

The functions begin by building the **A** and **b** matrices described in Sections 3.8.2 and 3.8.3. Table 4.3 displays the code for the function `fixc`. The functions then

33

Table 4.3: Excerpt from `estimator.m` showing the construction of the fixed capacitance estimator.

```
% --- from fixc
% calculate the estimation matrices Ax = b
A = [ ( w.^2 .* Rc ) ( w.^4 .* Rc * Cl^2 ) ...
      repmat( -R2, length(w), 1 ) -ones( length(w), 1 ) ...
      ( w.^2 .* Ic ) ( w.^4 .* Ic * Cl^2 ) ...
      ( -w * L2 ) zeros( length(w), 1 ) ];
A = reshape( A', 4, 2*length(w) )';
b = [ ( w.^2 .* R2 .* Cl^2) ( w.^3 * L2 * Cl^2 - w .* Cl ) ];
b = reshape( b', 2*length(w), 1 );

% use total least squares to calculate the answer x
C = [A b];
[u,s,v] = svd(C);
x = - v(1:end-1,end)/v(end,end);
```
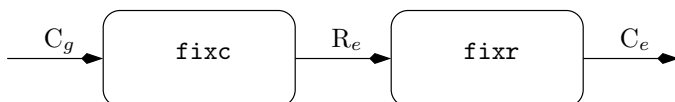

Figure 4.2: Flowchart of the estimator iteration process. C_g is a guess for C_l, R_e is an estimate of R_l and C_e is an estimate of C_l

compute the solution vector $\mathbf{x}$ using the TLS algorithm. `Fixr` returns

$$\mathbf{x} = \begin{bmatrix} M^2 & M^2 C_l^2 & C_l^2 & C_l \end{bmatrix} \tag{4.3}$$

and `fixc` returns

$$\mathbf{x} = \begin{bmatrix} \dfrac{M^2}{R_l^2} & M^2 & \dfrac{1}{R_l^2} & \dfrac{1}{R_l} \end{bmatrix}. \tag{4.4}$$

The script next makes an array of guesses for C_l and stores them in C_g as shown in Table 4.4 The term between the colons in C_g determines the accuracy of the guesses. In this case, C_l will be calculated to the nearest 0.1 nF. The smaller this number, the greater the accuracy of the system and the longer the script takes to run.

The script feeds its guesses for C_l into the fixed capacitance estimator producing estimates for R_l called R_e. It then takes these estimates and uses them as inputs into the fixed resistance estimator producing a set of estimates for C_l called C_e. This

Table 4.4: Excerpt from `estimator.m` showing the final calculation of the estimated solutions R_l and C_l

```
% 'Guesses' for Cl used to estimate Rl
Cg = [80e-9:1e-10:120e-9];
% The answer is the point where the guess is closest to the estimate
plot( Cg, Re, 'b', Ce, Re, 'r' );
[m,k] = min( abs( Cg - Ce ) );
r = Re(k);
c1 = Ce(k);
c2 = Cg(k);
```

process is illustrated in Figure 4.2. The final step is calculating the intercept point between C_e and C_g. Table 4.4 shows the matlab code that calculates the minimum distance between C_g and C_e. The script returns both the guess and estimate for C_l. The intersection between C_g and C_e is best shown graphically. Figure 3.8 shows the plotted curves and identifies the intersection point. The simulation exactly guesses circuit parameter values for any Z_x.

4.2 Compliance Sensor Devices and Construction

The compliance sensor system consists of two parts, the sensor implanted in the patient and the monitor coupled to the sensor through an inductive link. For the experiment, both devices were fabricated on PCBs with spiral inductors. Section 3.3.1 discusses the considerations behind pancake inductors wired on PCBs. The following two sections describe the circuit layouts for the sensor and monitor and list the board parameters.

4.2.1 Sensor Circuit and Layout

Figure 4.3 shows the very simple schematic for the compliance sensor. The capacitor and resistor values are displayed in Table 4.5. The circuit is designed so that an adjustable capacitance and resistance device can be attached. Either the sensors or a device mimicking sensor operation can be used to test the board.
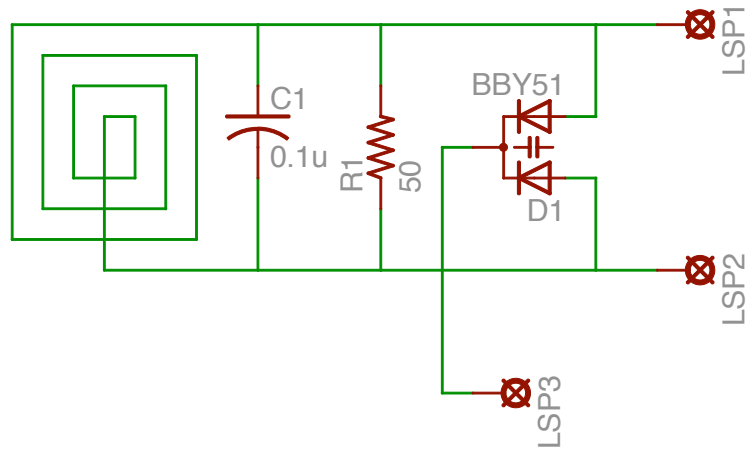
Figure 4.3: The compliance sensor schematic designed using the Eagle PCB design software.
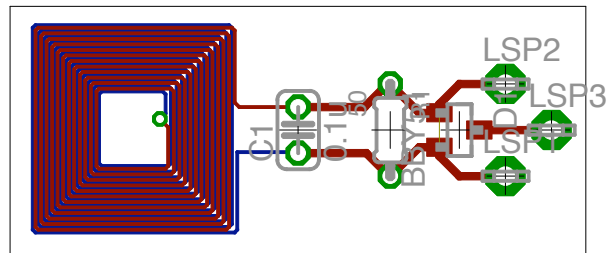
Figure 4.4: The compliance sensor board layout created using the Eagle PCB design software.

Table 4.5: Compliance sensor component values as measured at DC values before board construction.

Component	Value
L	3.1 Ω and 4.5 μH
R	48.7 Ω
C	0.1 μF
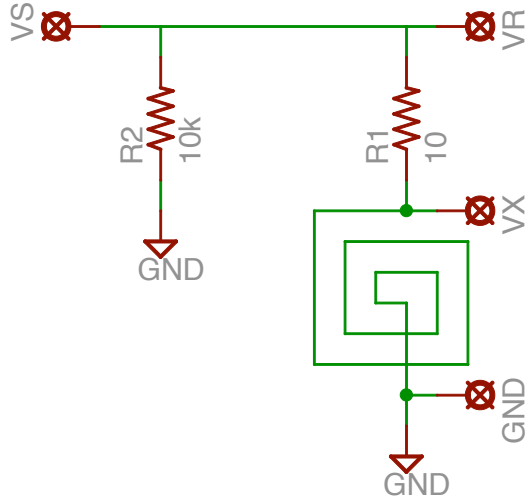
Figure 4.5: The compliance monitor schematic designed using the Eagle PCB design software.

Table 4.6: Compliance monitor component values as measured at DC levels before board construction.

Component	Value
L	3.1 Ω and 4.5 μH
R_m	9.8 Ω
R	10 $k\Omega$

Figure 4.4 shows the spiral inductor and other component locations on the PCB. After fabrication the trace resistances were measured using a fluke multimeter. The board component values are listed in Table 4.5.

4.2.2 Monitor Circuit and Layout

Figure 4.5 shows the simple schematic for the compliance monitor. The monitor only needs a reference resistor and PCB coil. The extra resistor from input to ground is there for a ground reference for the input signal. The test points in the circuit are locations where ADCs can capture V_r and V_x for further processing using the impedance bridge algorithm. Table 4.6 lists the monitor component values measured before board construction.
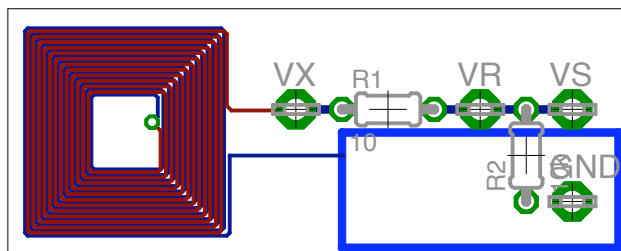
Figure 4.6: The compliance monitor board layout created using the Eagle PCB design software.

Figure 4.6 shows the spiral inductor and other component locations on the PCB. After fabrication the trace resistances and inductances were measured using a fluke multimeter. The board component values are listed in Table 4.6

Both the monitor and sensor PCBs were designed in eagle PCB. The board and schematic files are include in the appendices along with larger images of each. The boards were fabricated by Advanced Circuits, a PCB manufacturer based in Colorado.

4.3   Test Equipment

With the circuits and simulation frameworks complete, data can be collected and analyzed. For the purpose of data collection and device operation verification an Agilent signal generator and Tektronix oscilloscope were used for waveform generation and analog to digital conversion. The signal generator allowed precise frequency and waveform selection. For these tests it produced a sine wave between two and three hundred kilo-hertz with a five volt peak to peak amplitude. The oscilloscope sampled at an extremely high data rate, over five hundred samples per second. Using Tektronix's communication interface, between ten and fifteen periods were collected from the oscilloscope and transmitted back to the computer for analysis.

4.4   Impedance Bridge Software Verification

After designing the impedance bridge simulation and setting up the test equipment, the next step was to measure impedances with the impedance bridge and verify

Table 4.7: Excerpt from `bridge.m` showing the low-pass filter used to remove unwanted high frequency oscillations from the sampled data.

```
% The normalized cuttoff frequency is the cutoff frequency normalized
% to the nyquist frequency
wn = wc/nq;
% Construct the filter
b = fir1( 20, wn );
a = 1;
% Filter the data
vr = filter( b, a, vr_(:,2) );
vx = filter( b, a, vx_(:,2) );
```

the software operation. The first test was constructed on a breadboard and no considerations were taken to minimize signal interference from noise or any other sources before the oscilloscope captured the data. As a result, the data contained some high frequency oscillations that were interfering with the results. This noise was caused by both the lengths and construction of the cable running from the signal generator to the device and by the sampling noise present in the oscilloscope. The impedance bridge algorithm from Section 4.1.1 was modified to filter out the noisy data. The filtering is shown shown in Table 4.7. The cutoff frequency was chosen to be 500 kHz, well above the frequency of the driving waveform, but still low enough to remove the unwanted jitter.

Figure 4.7 shows the raw unfiltered V_r and V_x data collected by the oscilloscope compared to the filtered data used in the rest of the impedance calculations. The filter removed a lot of high frequency noise and caused the impedance measurements to be much more accurate and consistent.

Figure 4.8 shows the schematic for the simple voltage divider scheme. Both the reference resistor and load resistor were measured at 9.8 Ω before being placed in the arrangement in the figure. V_s was the driving waveform supplied by the signal generator.

The second test used the same breadboard configuration as the first test, but replaced the load resistor with a load inductor. Figure 4.9 shows the schematic for this test. The reference resistor had the same value from before and the inductor was
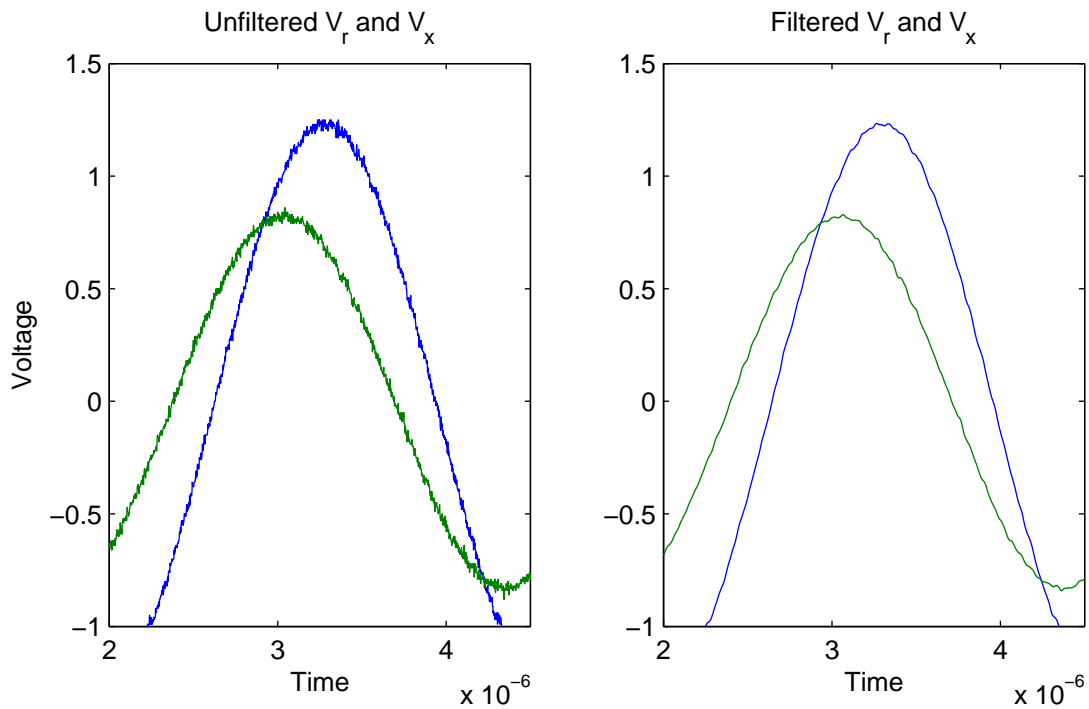
Figure 4.7: Raw V_r and V_x data collected by the oscilloscope plotted next to the filtered versions of V_r and V_x used in subsequent impedance analysis

$$V_s = \sin(\omega_o t)$$
$$\omega_o = 2\pi\ 30\text{kHz}$$
$$R_m = 9.8\ \Omega$$
$$R_l = 9.8\ \Omega$$
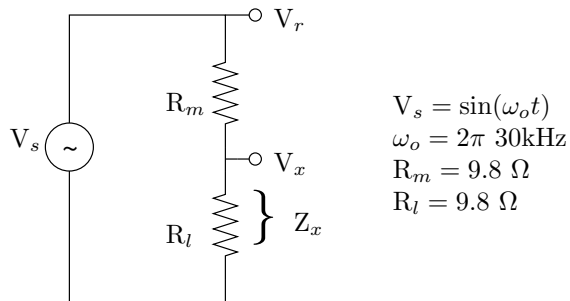
Figure 4.8: The first verification test measured the impedance of a purely resistive load.

40
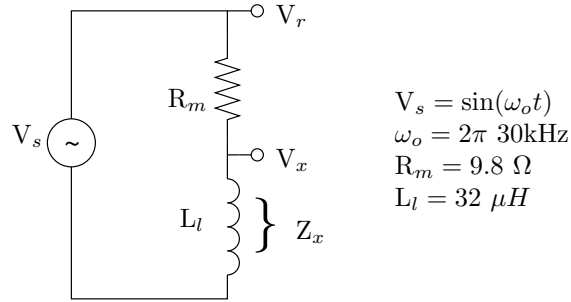
Figure 4.9: The second verification test measured the impedance of a purely inductive load.
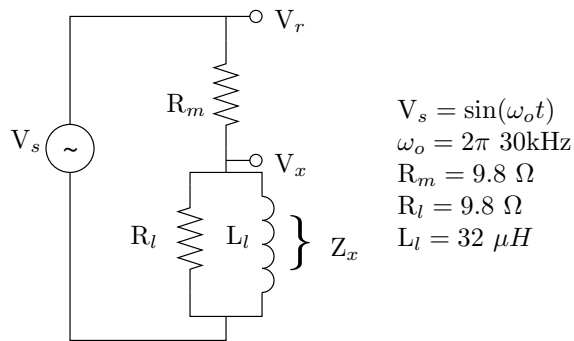
Figure 4.10: The third verification test measured the impedance of a complex load consisting of the parallel combination of the resistor and inductor from the first two tests.

measured to be 32 μH. The circuit was driven by V_s, the same waveform used for the first test.

The third test changed the configuration slightly placing the resistor form the first test and inductor from the second test in parallel. Figure 4.10 shows the schematic for this test. The reference resistor was still the same from the previous two tests and this time the parallel combination of the components was not measured beforehand, however, they were the same components used before so their individual measurements were still valid. The driving waveform was still the same.

Each test calculated forty impedances evenly spaced between 30 and 40 kHz. The data was collected by the labview automation system described in Section 4.5 and analyzed by the script `analysis_inductor` in Appendix A.7. The following discus-

sion summarizes the analysis results. It shows that the impedance bridge accurately measured impedance and accounted for device characteristics normally ignored by other measurement equipment.

In the first test, the device under test was a single resistor. Resistors are not frequency dependent and the measured resistance should be the same at any frequency. The mean and standard deviation of the data were calculated using Matlab and are listed in 4.5.

$$
\begin{aligned}
\mu_R &= 9.728 + j5.75e^{-2} \ \Omega \\
\sigma_R &= 1.42e^{-2} \ \Omega
\end{aligned}
\tag{4.5}
$$

In the second test, the device under test was a single inductor. Inductors are frequency dependent devices and so the impedance measurement at each frequency is different. Also, an inductor is modeled as an ideal inductor in series with an ideal resistor. With these considerations in mind the impedance of the inductor is $Z_l = R + j\omega L$. This means the measured impedance should have a small constant series resistance and a constant inductance once the effect of ω is divided out. The analysis script accounted for these and computed the statistics for the series resistance and complex inductance.

$$
\begin{aligned}
\mu_R &= 1.125 \ \Omega \\
\sigma_R &= 1.79e^{-2} \ \Omega \\
\mu_L &= 32.588 \ \mu H \\
\sigma_L &= 4.42e{-8} \ H
\end{aligned}
\tag{4.6}
$$

The results from these first impedance bridge tests show that the bridge consistently measures impedance over a range of frequencies. The third test helps solidify this result by comparing the impedance of the parallel combination shown in Figure 4.10 with the computed parallel combination of the resistor and inductor measured in Figures 4.8 and 4.9. A plot of the two impedances is shown in Figure 4.11. The statistics of the measured and computed calculations are listed in 4.7.

42
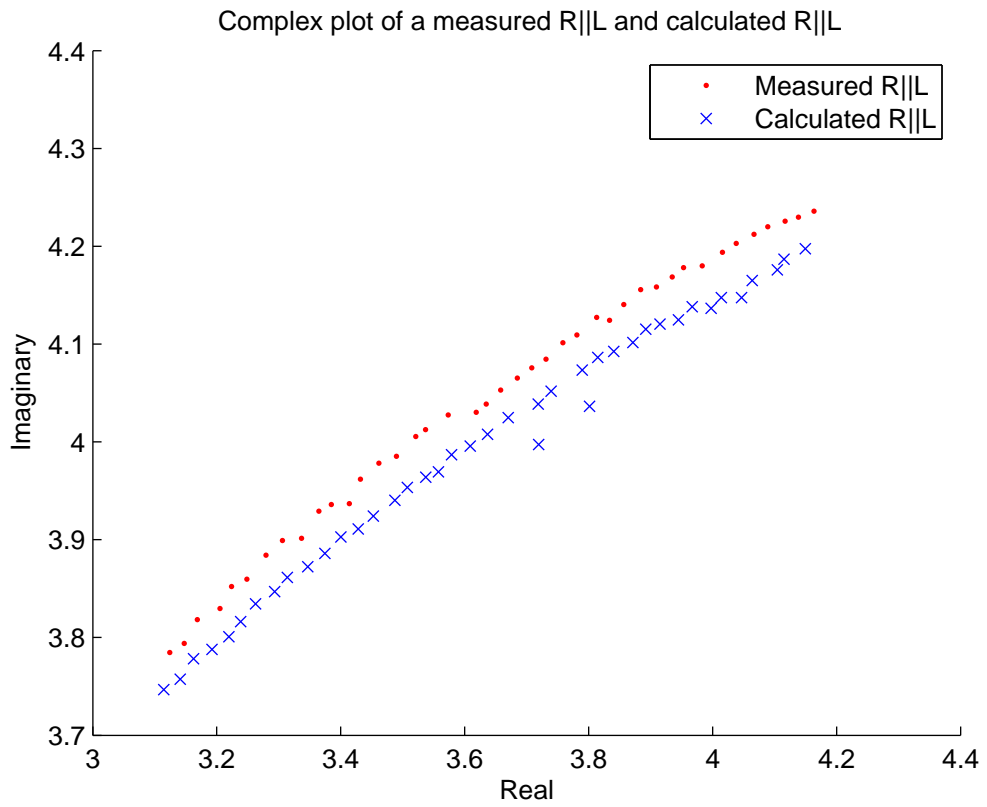
Figure 4.11: Measured parallel combination of R and L compared against calculated parallel combination of R and L

$$
\begin{aligned}
\mu_{R \text{ measured}} &= 8.13 \ \Omega \\
\sigma_{R \text{ measured}} &= 0.22 \ \Omega \\
\mu_{L \text{ measured}} &= 33.41 \ \mu H \\
\sigma_{L \text{ measured}} &= 1.7e^{-7} \ H \\
\mu_{R \text{ calculated}} &= 8.03 \ \Omega \\
\sigma_{R \text{ calculated}} &= 0.22 \ \Omega \\
\mu_{L \text{ calculated}} &= 33.25 \ \mu H \\
\sigma_{L \text{ calculated}} &= 1.8e^{-7} \ H
\end{aligned}
\tag{4.7}
$$

As the figure and statistics show, the measured and calculated data is similar and verifies the impedance bridge operation. Differences between the resistances and inductance in the different tests are present because the final test assumes L is an ideal inductor for the purposes of calculating the parallel resistance and inductance. The equations accounting for the non-ideal inductor are much more complicated and will not be solved here, but it is important to note the reason the computed parallel resistance is different from the single resistor measurements.

4.5 Labview Automation

Labview controls the data collection process by sending the appropriate waveform information to the agilent signal generator, adjusting the Tektronix oscilloscope settings and capturing waveform data from the scope. It also performs the data processing using a matlab script server to run the impedance bridge script from 3.5.1 as well as the estimator script developed in 3.8.2 and 3.8.3. Labview stores the data it collects in a comma separated values (CSV) text file in preparation for further analysis by Matlab.

4.6 Compliance Meter Data Collection and Analysis

Analysis of the Matlab simulation shows a point of interest in the impedance waveform between 200 and 300 kHz. Figure 4.12 shows a larger frequency sweep of
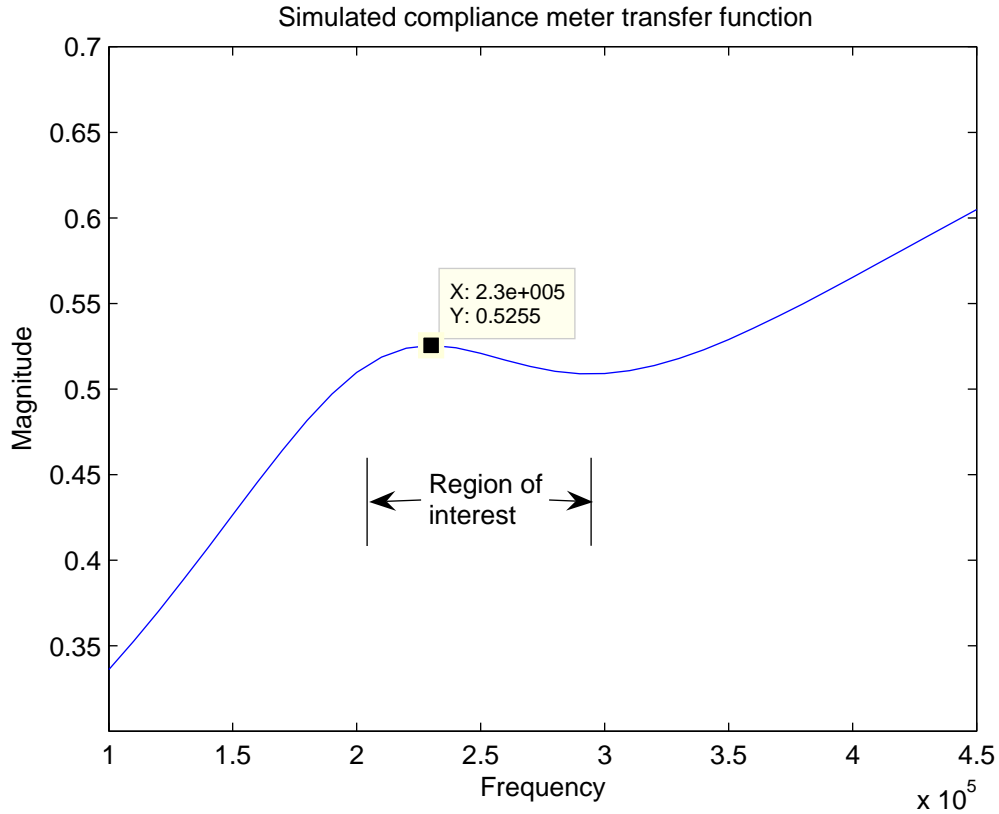
Figure 4.12: The magnitude of the impedance of the simulated compliance system emphasizing the region of interest between 200 kHz and 300 kHz.

the simulation highlighting the irregularity. This "hump" is caused by the reflected resonant impedance of the load capacitance and second inductively coupled coil. Refer back to Figure 3.5 to see a circuit diagram of the sensor. The continued rise after the resonant point is the magnitude of the monitor side coil overtaking or swamping out the magnitude of the reflected impedance. It seems apparent that more information is available around the point of interest than at any other point in the frequency sweep. All data collected for these tests sweeps between 200 and 300 kHz by 10 kHz steps giving eleven data points for analysis.

4.6.1 Calibration

The first step in testing the complete system was to calibrate the impedance meter by calculating the characteristics of the monitor-side coil without any loading.
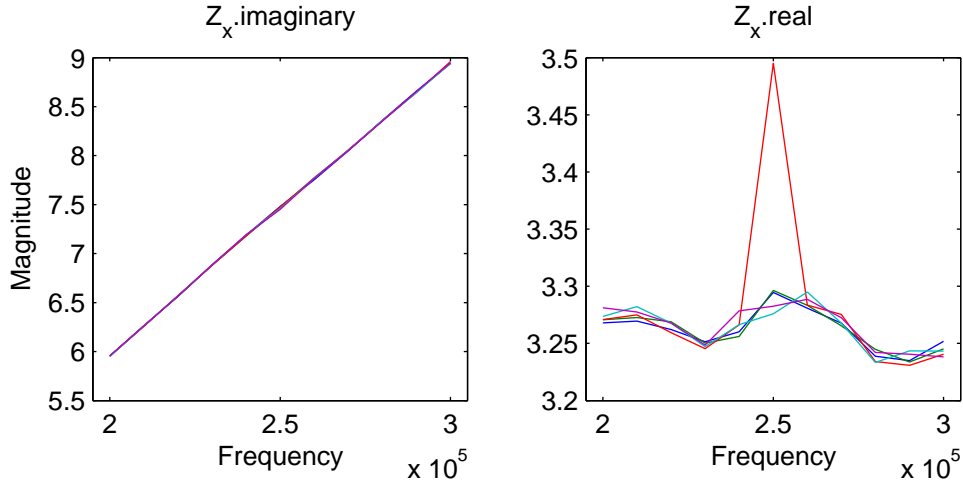
Figure 4.13: The magnitude of the real and imaginary portions of Z_x when measuring the impedance of the compliance meter in the absence of any load.

The unloaded coil was modeled by an ideal resistor in series with an ideal inductor and so the impedance had the form $Z = R + j\omega L$. The labview `zbridge.vi` script was used to run five frequency sweeps of the unloaded coil and capture the resulting data. The data was then analyzed by plotting the measured impedance, calculating the values for the resistance and inductance and generating the means and standard deviations for each component.

Figure 4.13 shows plots of the real and imaginary parts of the impedance. Each plot contains five frequency sweeps of eleven steps between 200 and 300 kHz. The plot of the imaginary component of Z_x reveals that the impedance of the inductor is extremely regular for each frequency sweep, the lines are almost indistinguishable. Calculating the actual inductor values and computing their statistics shows their stability.

$$\mu_L = 4.7508 \ \mu H$$
$$\sigma_L = 8.24 \ nH \tag{4.8}$$

The plot of the real component of Z_x shows the resistance of the inductor is not as stable as its inductance. One reason for this is that the inductance of the coil is in the mid-range of the values the impedance bridge can accurately measure

46

while the resistance of the coil is nearing the lower limit of the resistance that can be accurately measured. This means that small variations in resistance show up more readily than small variations in inductance. The statistics of the resistance show that while the values do fluctuate, they still have a decent standard deviation, meaning the resitance is calculated to be accurate within three-hundredths of an ohm which implies a high level of confidence that the measurement is correct.

$$
\begin{aligned}
\mu_R &= 3.26 \ \Omega \\
\sigma_R &= 3.6e^{-2} \ \Omega
\end{aligned}
\tag{4.9}
$$

The results from 4.8 and 4.9 are the calibrated monitor-side coil characteristics. They will be used as the values for both the monitor and sensor-side coil in the estimation computations.

4.6.2 Load Measurements

The labview test performed a frequency sweep of the sensor system and produced estimated outputs for R_l and C_l. During the course of the test, no changes were made to either the sensor or monitor other than the changing stimulus provided by the test apparatus. The estimator was configured to guess C_l within 10 pF. The data points were stored in a CSV file and matlab was used to analyze the results.

The estimates are plotted in Figure 4.14 and exhibit interesting characteristics worthy of discussion. The plot of the estimated capacitance clearly shows that a few of the data points noticeably deviate from the others. Comparing the deviant capacitance estimations to their corresponding resistance estimations reveals a similar irregularity in the resistance estimates. These points could be indicative of measurement error during data capture and could be removed from the data set without damaging the authenticity of the test. The data will be analyzed once with these points included and then repeated with the points removed in order to compare the results.
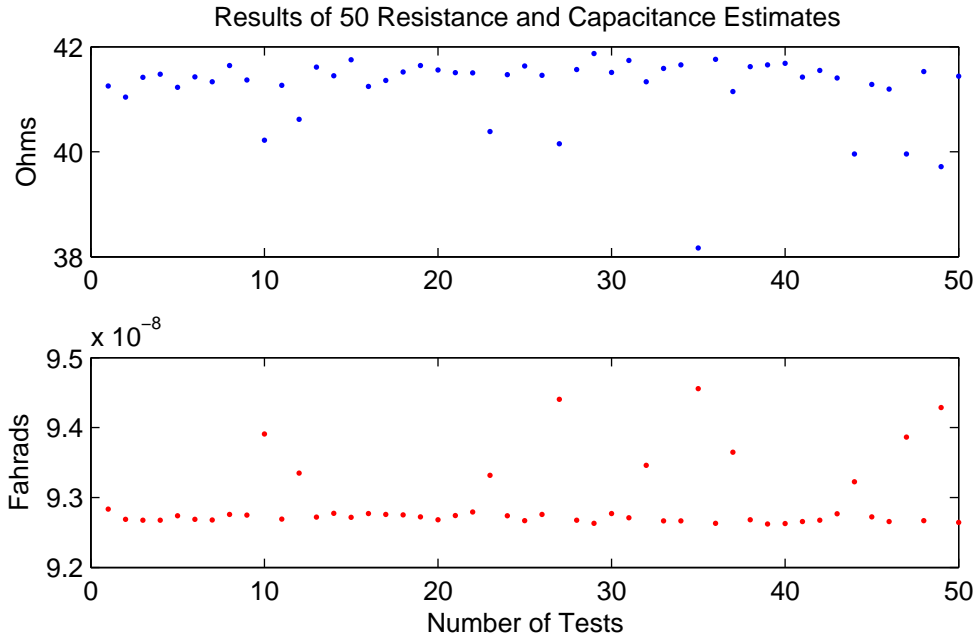
Figure 4.14: Results of 50 estimations of R_l and C_l without any load changes over the course of the test.

The mean and standard deviation of the original estimates displayed in Figure 4.14 were calculated to be

$$
\begin{aligned}
\mu_R &= 41.22 \ \Omega \\
\sigma_R &= 0.67 \ \Omega
\end{aligned}
\tag{4.10}
$$

for R_l. For C_l they were calculated to be

$$
\begin{aligned}
\mu_C &= 92.9 \ nF \\
\sigma_C &= .49 \ nF.
\end{aligned}
\tag{4.11}
$$

Figure 4.15 shows the estimates for R_l and C_l with the deviant estimates removed. Calculating the statistics for these modified estimates produced

$$
\begin{aligned}
\mu_R &= 41.49 \ \Omega \\
\sigma_R &= 0.17 \ \Omega
\end{aligned}
\tag{4.12}
$$

for R_l and

$$
\mu_C = 92.7 \ nF
$$

48
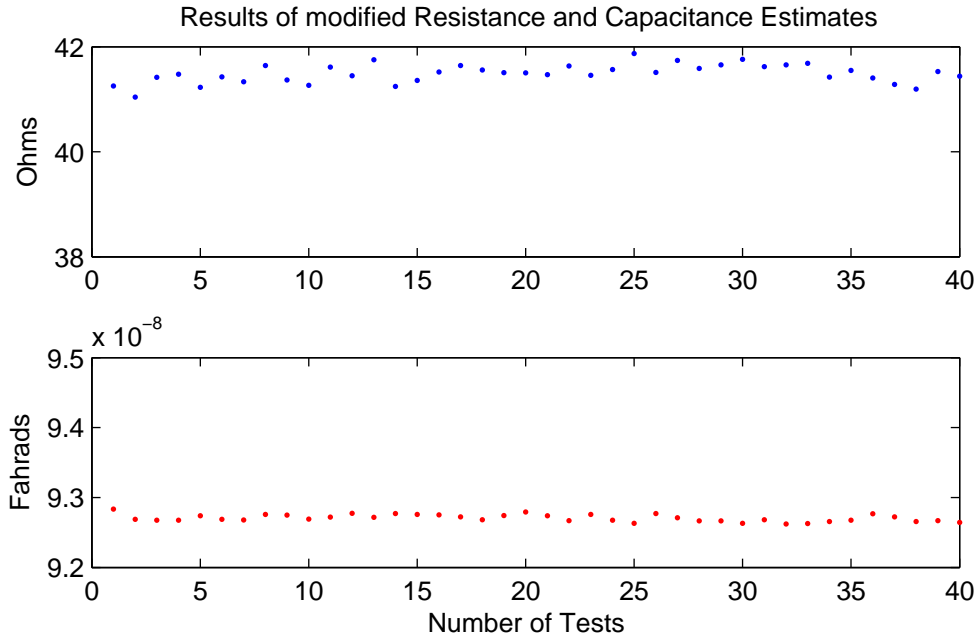
Figure 4.15: The results of the estimations of R_l and C_l after removing the deviant data points.

$$\sigma_C = 51.4 \ pF \tag{4.13}$$

for C_l.

As expected, the statistics of the modified data are much better than those of the unmodified data.

The compliance system, as it stands, does not provide any information about why a number of the data points deviate so significantly from the majority of the data points. Searching for this reason provided the insight leading to the discussion in Section 4.7.

4.7   A Second Analysis

This section uses the same data collected for the previous compliance meter load measurements, but it differs in the estimation technique employed. Section 4.7.1 explains the new parameter and details its matlab script. Section 4.7.2 simulates the estimator from Section 4.1.2 and the estimator from Section 4.7.1 and compares their

Table 4.8: Excerpt from `newest.m` showing the function declaration, input variables and iterator setup. The iterator selects values of M for the estimator.

```
% New estimator function for the impedance bridge compliance meter
function [r,c,k] = newest( f, Zx, kg );
...
% set up the guesses over k
mg = kg*sqrt(L1*L2);
...
```

responses to noisy input data. Finally, Section 4.7.3 analyzes the data collected for the previous tests and compares the results.

4.7.1 Estimation without Re-Parametrization

One problem in the estimation technique developed in sections 3.8.2 and 3.8.3 is that they multiply the effect of any noise in the system because of the way they were derived. The estimator described here attempts to avoid any noise amplification resulting from similar manipulations. Section 3.9.1 details the mathematical theory driving the estimator and this section presents its implementation, both for simulated and actual data analysis.

The values available for any type of data analysis are contained in the impedance measurements collected by the labview automation system. That system returns vectors of impedance measurements and the frequencies at which the measurements were taken. It follows that the inputs to the new estimator should be Z_x and f. Table 4.8 shows the function heading for the estimator. The third input variable, k_g, is a vector used to calculate m_g, the structure holding the "guesses" for M.

Table 4.9 shows how the function constructs as much of **A** and **b** as possible before looping over the guesses for M. Vectors of resistance and capacitance values are created for each guess and the mean and standard deviation of each vector are saved for later use. The final step in the function is to select the point of minimum variance in the estimate and return that value for R and C. The value returned for k represents the coefficient of inductive coupling between the sensor and monitor coils.

Table 4.9: Excerpt from `newest.m` showing the estimator and iteration method used to pick the best value of M.

```
% Construct as much of the A and B matrices as possible
A = diag( Zx - R1 - j*w*L1 );
H = inv( A' * A ) * A';
b = ( R1 + j*w*L1 - Zx ).*( R2 + j*w*L2 );

% Iterate over M
for i=1:length(mg)
    Zl = H * ( b + w.^2*mg(i)^2 );
    res = 1./real(1./Zl);
    cap = imag(1./Zl)./w;
    % Save the statistics
    mr(i) = mean(res); mc(i) = mean(cap);
    sr(i) = std(res);  sc(i) = std(cap);
end

[m,i] = min( sr );
r = mr(i);
c = mc(i);
k = kg(i);
```

The only way to ensure that `newest.m` actually works is to show that the point of minimum standard deviation is the point that most accurately selects R_l, C_l and M. Figure 4.16 displays a graph of standard deviation versus number of iterations over k_g. Notice the sharp descent towards the the minimum value which, in every tested case, corresponds to the correct values for the sensor components.

If, for the first measurement, k_g is constructed with a relatively low accuracy over a large range, the inner loop of the estimator runs much faster than if high accuracy is attempted immediately. The best method is to use a broad, low accuracy sweep to pinpoint about where k is located and then sweep again with high accuracy over a small area. Once k is found it can be assumed that the rest of the measurements have a similar coupling coefficient.

4.7.2 Comparison Using Noise Simulation

As expected, both estimators return exact values for R_l and C_l when stimulated by signals without added noise. This section explores the behavior of the estimators in the presence of noise. Section 4.1.2 explains the script file `bridge.m`

51

x 10^{-9} Standard Deviation in the calculated C$_l$
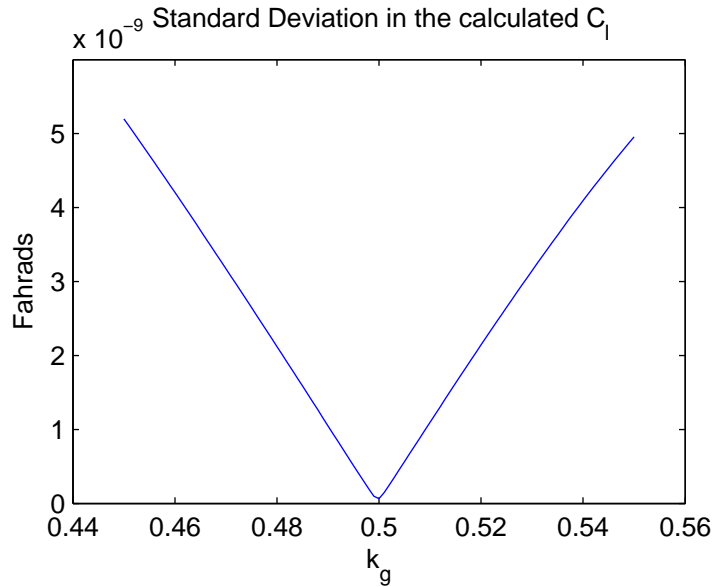
Figure 4.16: Standard deviation of C_l for various guesses of M. The obvious minimum value supports the use of the new estimator load calculations.

Table 4.10: Excerpt from `simZx.m` showing the differences between it at the original impedance bridge simulation script `bridge.m`.

```
% simZx simulates Zx at various frequencies and adds a
% noise component
function [f,Zx] = simZx( noise );
...
% add noise to the 'measured' inputs
n = noise * ( rand( 1, length(t_) ) - 0.5 );
Vrn = Vr + n;
Vxn = Vx + n;
```

which simulates the values collected by the impedance bridge. For noise analysis, it is slightly modified to reflect the effect of noise in the measurements. The new file is called `simZx.m` and the pertinent differences between it and `bridge.m` are displayed in Table 4.10.

The input variable `noise` adjusts the noise level of the signal. The noise is added from a normal distribution to simulate the assumed white noise present in the measured signal. For these tests the noise was set to 0.05.
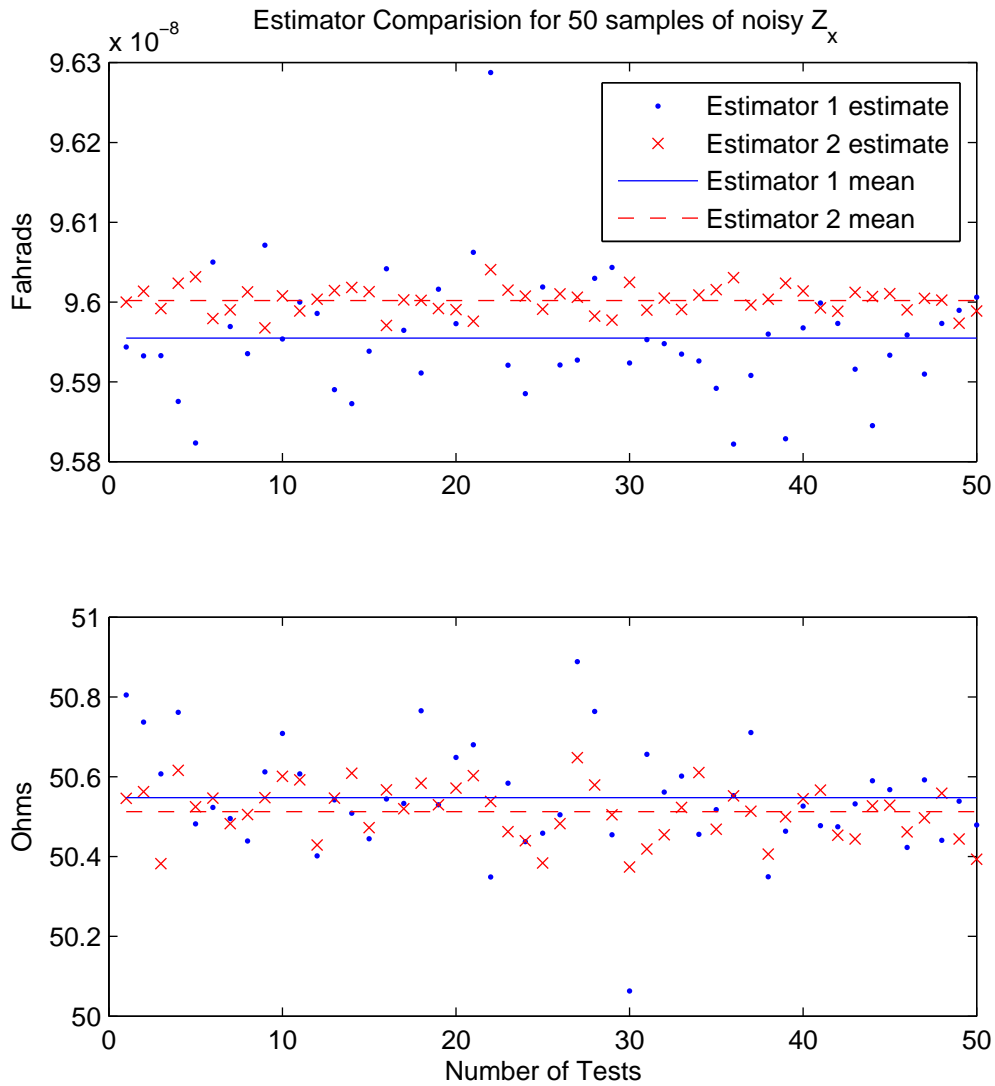
Figure 4.17: A 50 sample comparison between the estimators showing estimated and mean values for R_l and C_l.

Script `compare` creates an array of 50 noisy Z_x measurements and then estimates R_l and C_l with both estimators. Figure 4.17 displays the results of the estimates for each of the 50 tests. The dots represent estimates made with the original estimator and the new estimator is represented with the letter 'x'. The solid line represents the mean of the estimates made with the original estimator and the dashed line is the mean of the new estimator.

Table 4.11: Estimator comparison results showing significant improvement using the new estimation technique.

Script	Mean C_l	Std C_l	Mean R_l	Std R_l
estimator	95.95 nF	70.6 pF	50.54 Ω	0.137 Ω
newest	96.00 nF	16.3 pF	50.51 Ω	0.068 Ω

Table 4.11 displays the results of the noise test. Notice the difference in standard deviation between the original estimator and the new estimator. The new one performs significantly better, around an order of magnitude, than the original one. Hopefully this improvement will also be evident in the actual data analysis.

4.7.3 Impedance Testing with the New Estimator

The new estimator outperforms the original in simulation. This section runs the new estimator on the data collected earlier and displays the results. It then compares the statistics of each estimator. Figure 4.18 shows the estimates for R_l and C_l as well as k, the coefficient of coupling for each measurement. The statistics of the measurements are

$$\mu_R = 31.62 \ \Omega$$
$$\sigma_R = 0.18 \ \Omega \tag{4.14}$$

for R_l and

$$\mu_C = 90.7 \ nF$$
$$\sigma_C = 0.11 \ nF \tag{4.15}$$

for C_l. Comparing these to the statistics calculated by the original estimator shows that this one performs better.

4.8 Results

From the results in 4.12 and 4.13, it is clear that in order for the system to accurately represent changes in resistance and capacitance the strain gauge must change its value $\geq 0.2 \ \Omega$ per desired change in pressure and the water sensor must
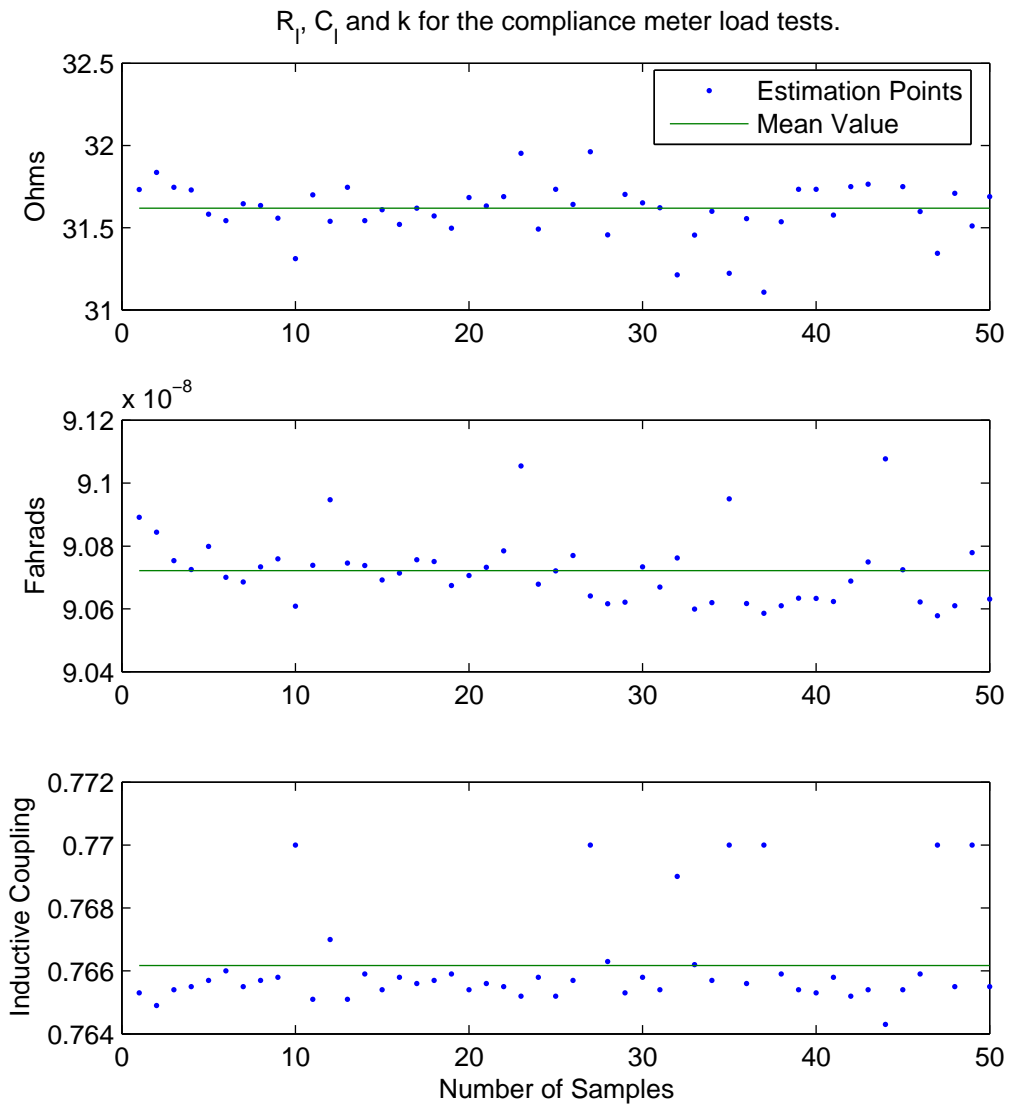
Figure 4.18: Plots of R_l, C_l and k created by the new estimator after analyzing the data collected for the compliance meter load test.

change its value ≥ 0.2 nF per desired change in volume. The implications of these results are discussed more fully in the conclusions.

Chapter 5

Conclusion

This thesis presents the framwork for a permanently implantable wireless sensor capable of returning information about the compliance of a patient's brain. The presented framework consists of an extensive simulation environment modeling the interaction of the compliance sensor with its monitoring device, a laboratory prototype of the sensing system and sufficient analysis to demonstrate the veracity and usefulness of the concept. The thesis develops a unique technique for impedance measurement allowing simultaneous capture of multiple changing impedances through a single interface. This chapter summarizes the design and results, addresses the usability of the system and suggests directions the research can take.

5.1 A Compliance Sensor

The sensor consisted of two sensing elements, a resistive strain gauge and a capacitive water sensor. The sensors were connected in parallel to a pancake inductor. The sensing system was inductively coupled to a monitoring system through the small pancake inductor which was fabricated on a printed circuit board. The monitoring system used a signal generator to drive the impedance bridge and an oscilloscope to capture waveform information. A labview script automated the data collection process. A PC analyzed the data using a series of Matlab scripts and determined the sensor values.

The system is a simulation and laboratory prototype that will be useful in continued design and development of the compliance meter. Using the simulation,

various additional measurement and analysis configurations could also be created from the results of this research.

5.2   Usability of the System

The system calculates load resistances down to 0.2 of an ohm values and load capacitances down to 100 nano-fahrads. These sensitivities inform future researchers about the types of sensors they may use during further development. These sensitivites raise the question of how sensitive does the sensor need to be in order to be useful. In order to be useful, the sensor must be able to return viable information about the compliance of the brain. This information is communicated through the relationship between the pressure and volume pulsitile waveforms. Sensors with these sensitivities connected to the system developed in this thesis, would return viable information about the compliance of the brain.

5.3   Uniqueness of the Impedance Measurements

The research presented in this thesis is unique and original in its approach to measuring multiple variability in a domain. The measurement of two simultaneously varying impedances, in this case a resistance and a capacitance, across a wireless boundary with an independently changing coupling is an exciting step forward in measurement techniques. The ability to collect and characterize device operation in this manner could open up a new realm of complex, implantable sensors and allow doctors, scientists and engineers to create smaller, more useful sensing devices.

5.4   Future Research

The next researcher should acquire sensors matching the specifications and sensitivites listed in the paper and build a custom monitoring system mirroring the current laboratory setup. It should be autonomous, meaning it is not tethered to a desktop computer. Future research directions could include looking into faster signal acquisition, noise reduction techniques to increase the accuracy of the measurements, hardware development, sensor selection, clinical testing and application to other venues where measuring two sensors simultaneously would be useful.

Appendix

Appendix A

Matlab Scripts

A.1 Internal Component Values

```
% ----------------------------------------------------------------------
% script: setComponentValues.m
% author: Don Wichern
% date:   9 Dec 2005
% desc:   Sets the global component values for use in simulation and
%         experimentation
% ----------------------------------------------------------------------


Rm = 9.8;
R1 = 3.26;
R2 = 3.26;
L1 = 4.75e-6;
L2 = 4.75e-6;
k = 0.5;
M = k * sqrt( L1 * L2 );
Cl = 96e-9;
Rl = 50.5;
```

A.2 Impedance Bridge Simulation

```
function [f,Zx] = simZx( noise );
% ----------------------------------------------------------------------
```

```
% script: simZx.m

% author: Don Wichern

% date:   9 Dec 2005

% desc:   Simulates the complex impedance of the sensor load reflected to

%         the  monitor coil.  Noise is the amount of white noise added to

%         the system.  Typical values are 0.01 - 0.05.

% -------------------------------------------------------------------------


setComponentValues


f = 2e5:5000:3e5;

f = f';

w = 2*pi*f;


% Actual bridge impedances

Zx = R1 + i*w*L1 + w.^2 .* M^2 .* ...

    ( 1 ./ ( R2 + i*w*L2 + 1 ./ (1/Rl + i*w*Cl) ) );


% Estimated bridge impedances

Zn = zeros(length(f),1);

Zf = zeros(length(f),1);


for k = 1:length(f)


    % Create the input and impedance modulated output

    t = 0:1e-8:40e-6;

    Vr = cos( w(k) * t - pi/2 );

    [x,y] = pol2cart( -pi/2, 1 );

    Vrs = x + i*y;

    Vxs = Vrs * Zx(k)/(Zx(k) + Rm);
```

```matlab
[a,r] = cart2pol( real(Vxs), imag(Vxs) );
Vx = r * cos( w(k) * t + a );


% add noise to the 'measured inputs
n = noise * ( rand( 1, length(t) ) - 0.5 );
Vrn = Vr + n;
Vxn = Vx + n;


% Low pass filter the data and replot
% Get the sampling period from the data file
st = t(2) - t(1);
% The sampling frequency is the inverse of the sampling period
sr = 1/st;
% The nyquist frequency is 1/2 the sampling frequency
nq = sr/2;
% Pick a cuttoff frequency
wc = 1e6;
% The normalized cuttoff frequency is the cutoff frequency normalized
% to the nyquist frequency
wn = wc/nq;
% Construct the filter
b = fir1( 20, wn );
a = 1;
% Filter the data
Vrf = filter( b, a, Vrn );
Vxf = filter( b, a, Vxn );


% resample based on the selected frequency
qu = 80; % number of samples in a quarter period
sr = 1/f(k)/4/qu; % sampling rate
```

```
ts = 0:sr:40e-6; % time index for resampling

% resampling routines

Vrn = spline( t, Vrn, ts );

Vxn = spline( t, Vxn, ts );

Vrf = spline( t, Vrf, ts );

Vxf = spline( t, Vxf, ts );


% Calculate the voltage across the reference resistor

Vrm = Vrn - Vxn;


% Build the bridge estimator

A = zeros(length(Vrm)-qu,2);

A(:,1) = Vrm(qu+1:length(Vrm));

A(:,2) = -1.*Vrm(1:length(Vrm)-qu);

y = Vxn(qu+1:length(Vxn))';


% Calculate the bridge weigths, we can use the pseudo inverse

% or the TLS to find the solution.

W = inv( A' * A ) * A' * y;

%    C = [A y];

%    [u,s,v] = svd(C);

%    W = - v(1:end-1,end)/v(end,end);


% Build the estimated impedance array

Zn(k) = Rm * ( W(1) + i*W(2) );


% Calculate the voltage across the reference resistor

Vrm = Vrf - Vxf;


% Build the bridge estimator
```

64

```
      A = zeros(length(Vrm)-qu,2);

      A(:,1) = Vrm(qu+1:length(Vrm));

      A(:,2) = -1.*Vrm(1:length(Vrm)-qu);

      y = Vxf(qu+1:length(Vxf))';


      % Calculate the bridge weigths, we can use the pseudo inverse

      % or the TLS to find the solution.

      W = inv( A' * A ) * A' * y;

%      C = [A y];

%      [u,s,v] = svd(C);

%      W = - v(1:end-1,end)/v(end,end);


      % Build the estimated impedance array

      Zf(k) = Rm * ( W(1) + i*W(2) );


end


% subplot(211);

% plot(f,abs(Zx),'.b',f,abs(Zn),'.r',f,abs(Zf),'.g');

% subplot(212);

% plot(f,angle(Zx),'.b',f,angle(Zn),'.r',f,angle(Zf),'.g');


Zx = Zn;
```

A.3   Estimator Simuation—Re-parameterized Equation

```
function [r,c] = estimator( f, Zx );

% -------------------------------------------------------------------------

% script: estimator.m

% author: Don Wichern

% date:   9 Dec 2005
```

```
% desc:    This script estimates the values of the capacitor and resistor in
%          an inductively coupled load.  As inputs it takes a an impedance
%          measured across the stimulating inductor as well as the
%          frequencies those impedances were calculated at.  It outputs the
%          resistor and capacitor value it estimates.
% -------------------------------------------------------------------------

% --- Measured circuit parameters
% Global declarations
global w
global L2
global R2
global Rc
global Ic
% Actual values
R1 = 3.26;
L1 = 4.75e-6;
L2 = 4.75e-6;
R2 = 3.26;
% --- Estimated parameters
% M  - always estimated
% Cl - assumed while estimating Rl
% Rl - assumed while estimating Cl

% --- Inputs
% f  - an array of frequencies
% Zx - an array of impedances measured at frequencies f
w = 2*pi*f;
% Make sure w and Zx are a column vectors
s = size(w);
```

```matlab
if s(1) < s(2)

    w = w';

end

s = size(Zx);

if s(1) < s(2)

    Zx = Zx.';

end


% --- Estimate Rl and Cl using the measured impedances

Zc = 1./ ( Zx - ( R1 + i*w*L1 ) );

Rc = real( Zc );

Ic = imag( Zc );


% --- Estimations

% These are all in the same file because it made it easier to port over to

% labview

% 'Guesses' for Cl used to estimate Rl

Cg = [90e-9:1e-10:110e-9];

% 'Estimates' for Cl and Rl

Ce = zeros(1,length(Cg));

Re = zeros(1,length(Cg));

Mer = zeros(1,length(Cg));

Mec = zeros(1,length(Cg));


% Calculate the estimates by selecting a search window of capacitances 'Cg'

% and using them to calculate a bunch of estimates for the resistance 'Re',

% then use the estimated resistances to check the capacitances 'Ce'.  The

% intersection, or place where the capacitances are the closest, is the

% answer.  Eventually, this section will take into account previous guesses

% in order to narrow the search window.
```

```matlab
for k = 1:length(Cg)
    x = fixc( Cg(k) );
    Re(k) = 1/x(4);
    Mec(k) = x(2);
    x = fixr( Re(k) );
    Ce(k) = x(4);
    Mer(k) = x(1);
end


% The answer is the point where the guess is closest to the estimate
%figure(1); plot( Cg, Re, 'b', Ce, Re, 'r--' );


[m,k] = min( abs( Cg - Ce ) );
r = Re(k);
c = Ce(k);
m1 = Mer(k);
m2 = Mec(k);


% --- Functions ------------------------------------------------------------
% --- fixc --- calculates Rl and M assuming a fixed Cl
function x = fixc( Cl );


% import the globals
global w
global L2
global R2
global Rc
global Ic


% calculate the estimation matricies Ax = b
```

```
A = [ ( w.^2 .* Rc ) ( w.^4 .* Rc * Cl^2 ) ...
    repmat( -R2, length(w), 1 ) -ones( length(w), 1 ) ...
    ( w.^2 .* Ic ) ( w.^4 .* Ic * Cl^2 ) ...
    ( -w * L2 ) zeros( length(w), 1 ) ];
A = reshape( A', 4, 2*length(w) )';
b = [ ( w.^2 .* R2 .* Cl^2) ( w.^3 * L2 * Cl^2 - w .* Cl ) ];
b = reshape( b', 2*length(w), 1 );


% Values stored in x
% x = [M^2/Rl^2; M^2; 1/Rl^2; 1/Rl ]


% use total least squares to calculate the answer
C = [A b];
[u,s,v] = svd(C);
x = -v(1:end-1,end)/v(end,end);


% --- fixr --- calculates Cl and M assuming a fixed Rl
function x = fixr( Rl );


%import globals
global w
global L2
global R2
global Rc
global Ic


% calculate the estimation matricies Ax = b
A = [ ( w.^2 .* Rc * 1/Rl^2 ) ( w.^4 .* Rc ) ...
    ( -w.^2 * R2 ) zeros( length(w), 1 ) ...
    ( w.^2 .* Ic * 1/Rl^2 ) ( w.^4 .* Ic ) ...
```

```
        ( -w.^3 * L2 ) ( w ) ];

A = reshape( A', 4, 2*length(w) )';

b = [ repmat( R2 * 1/Rl^2 + 1/Rl, length(w), 1 ) ( w * L2 * 1/Rl^2 ) ];

b = reshape( b', 2*length(w), 1 );


% Values stored in x

% x = [M^2; M^2*Cl^2; Cl^2; Cl ]


% use total least squares to calculate the answer x

C = [A b];

[u,s,v] = svd(C);

x = - v(1:end-1,end)/v(end,end);
```

A.4   Estimator Simulation—No Noise Equation

```
function [r,c,k] = newest( f, Zx, kg );

% -------------------------------------------------------------------------

% script: newest.m

% author: Don Wichern

% date:   9 Dec 2005

% desc:   This is the second estimator developed for the project.  It

%         attempts to minimize noise interference by avoiding complex

%         operations on the noise input data.

% -------------------------------------------------------------------------


% Set up the system circuit parameters.

setComponentValues

% convert frequency to radians

%[f,Zx] = simZx( 0.01 );

w = 2*pi*f;
```

70

```matlab
% set up the guesses over k
mg = kg*sqrt(L1*L2);
m = length(mg);
mr = zeros(m,1);
mc = zeros(m,1);
sr = zeros(m,1);
sc = zeros(m,1);


A = diag( Zx - R1 - j*w*L1 );
b = (R1 + j*w*L1 - Zx).*(R2 + j*w*L2);


for n=1:length(mg)
    % Use TLS to calculate the parameter values
    [u,s,v] = svd( [A (b + w.^2*mg(n)^2)] );
    Zl = -v(1:end-1,end)/v(end,end);
    res = 1./real(1./Zl);
    cap = imag(1./Zl)./w;
    % Save the statistics of the calculated values
    mr(n) = mean(res);
    mc(n) = mean(cap);
    sr(n) = std(res);
    sc(n) = std(cap);
end


% Use the statistics of the calculated values to choose the best guess
[m,n] = min( sr );
r = mr(n);
c = mc(n);
k = kg(n);
```

A.5 Simulation Comparision

```
% ----------------------------------------------------------------------
% script: compare.m
% author: Don Wichern
% date:   9 Dec 2005
% desc:   Compare the estimator against the new estimator
% ----------------------------------------------------------------------


m = 1;


r1 = zeros(m,1);
c1 = zeros(m,1);
r2 = zeros(m,1);
c2 = zeros(m,1);
k = zeros(m,1);


kg = [0.45:0.001:0.55];
for n=1:m
    [f,Zx] = simZx( 0.02 );
    [r1(n),c1(n)] = estimator( f, Zx );
    [r2(n),c2(n),k(n)] = newest( f, Zx, kg );
end


mr1 = mean(r1); mr2 = mean(r2);
sr1 = std(r1);  sr2 = std(r2);
mc1 = mean(c1); mc2 = mean(c2);
sc1 = std(c1);  sc2 = std(c2);


d = 1:m;
vmc2 = repmat( mc2, m, 1 );
```

```
vmc1 = repmat( mc1, m, 1 );
subplot(211); plot(d,c1,'.b',d,c2,'xr', d,vmc1,'b',d,vmc2,'r--');
vmr1 = repmat( mr1, m, 1 );
vmr2 = repmat( mr2, m, 1 );
subplot(212); plot(d,r1,'.b',d,r2,'xr', d,vmr1,'b',d,vmr2,'r--');
```

A.6   Import Bridge Data from Labview

```
function [f,Zx] = zbridge( fname );
% ------------------------------------------------------------------------
% script: zbridge.m
% author: Don Wichern
% date:   9 Dec 2005
% desc:   This file parses the data from zbridge.vi returning a vector of
%         frequencies and an array of impedances where each row correspons
%         to a frequency in f.
% ------------------------------------------------------------------------

d = importdata( fname, ',' );
% Store the frequency
f = d(:,1);
% Build an array of impedance measurements
[m,n] = size(d);
n = (n-1)/2;
Zx = zeros( m, n );
for k=1:n
    Zx(:,k) = d(:,2*k) + i*d(:,2*k+1);
end
```

A.7   Inductor Analysis

```
% ------------------------------------------------------------------------
```

```matlab
% script: analysis_inductor.m
% author: Don Wichern
% date:   9 Dec 2005
% desc:   This file analyzes impedance of the monitor side inductor coil.
%         It calculates the value of the inductor and the magnitude of the
%         series resistance.
% --------------------------------------------------------------------------


[f, Zx] = zbridge( './data/zbridge_5pt_l.csv' );


% Create an array of frequencies for inductor calculations
[m,n] = size(Zx);
fm = repmat(f,1,n);


% Calculate the inductances in Zx
L = imag(Zx)/2/pi./fm;
L = reshape(L,m*n,1);
% Calculate the resistances in Zx
R = real(Zx);
R = reshape(R,m*n,1);


% Calculate the statistics of L and R
st = [ mean(R) std(R); mean(L) std(L) ];
%
% subplot(121); plot(f,imag(Zx))
% ylabel( 'Magnitude')
% xlabel( 'Frequency' )
% title( 'Z_x.imaginary' )
% axis( [1.95e5 3.05e5 5.5 9] );
% subplot(122); plot(f,real(Zx))
```

```
% xlabel( 'Frequency' )
% axis( [1.95e5 3.05e5 3.2 3.5] );
% title( 'Z_x.real' )
```

A.8 Load Analysis—Re-parameterizing Estimator

```
% -------------------------------------------------------------------------
% script: analysis_load.m
% author: Don Wichern
% date:   9 Dec 2005
% desc:   This file analyzes the impedance of the compliance meter system
%         and calculates the estimates for load resistance and load
%         capacitance.  It also computes some system statistics useful for
%         validation arguments
% -------------------------------------------------------------------------


[f, Zx] = zbridge( './data/zbridge_50pt_3.csv' );


% Create data structures to hold the estimates
[m,n] = size(Zx);
Re = zeros(n,1);
Ce = zeros(n,1);
Cg = zeros(n,1);
% Calculate the estimates
for k=1:n
    [Re(k),Ce(k),Cg(k)] = estimator( f, Zx(:,k ) );
end
% Calculate the statistics
st = [ mean(Re) std(Re); mean(Ce) std(Ce);  mean(Cg) std(Cg) ];


figure(1)
```

```
subplot(211); plot( Re, '.b' );

ylabel( 'Ohms' );

title( 'Results of 50 Resistance and Capacitance Estimates' );

subplot(212); plot( Ce, '.r' );

ylabel( 'Fahrads' );

xlabel( 'Number of Tests' );
```

A.9   Load Analysis—Low Noise Estimator

```
% ------------------------------------------------------------------------

% script: test_newest.m

% author: Don Wichern

% date:   9 Dec 2005

% desc:   Verifies the new estimator using data collected from the

%         compliance meter.

% ------------------------------------------------------------------------

[f,Zx] = zbridge( './data/zbridge_50pt_3.csv' );


kg = [0.3:0.001:0.8];


[m,n] = size(Zx);

r = zeros(n,1);

c = zeros(n,1);

k = zeros(n,1);


for i=1:n

    [r(i),c(i),k(i)] = newest( f, Zx(:,i), kg );

end


n = [1:50];

subplot(311); plot(n,r,'.',n,repmat(mean(r),1,50));
```

```
subplot(312); plot(n,c,'.',n,repmat(mean(c),1,50));
subplot(313); plot(n,k,'.',n,repmat(mean(k),1,50));
```

Bibliography

[1] M. Aboy, J. NcNames, D. Cuesta-Frau, W. Wakeland, T. Thong, S. Lai, and B. Goldstein, "Significance of intracranial pressure pulse morphology in pediatric traumatic brain injury," *EMBC*, 2003.

[2] M. Bergsneider, I. Yang, X. Hu, D. L. McArthur, S. W. Cook, and W. J. Boscardin, "Relationship between valve opening pressure, body position, and intracranial pressure in normal pressure hydrocephalus: Paradigm for selection of programmable valve pressure setting," *Neurosurgery*, vol. 55(4)), pp. 851–859, Oct. 2004.

[3] S. Sood, S. D. Ham, and A. I. Canady, "Current treatment of hydrocephalus," *Neurosurger Quarterly*, vol. 11(1), pp. 36–44, Mar. 2001.

[4] I. Piper, C. F. Contant, and G. Citerio, "Brain monitoring with information techonology. the "brain-it" group experience," *Minerva Anestesiologica*, vol. 66, pp. 17–21, 2000.

[5] G.-Z. Yang, "Future advances in mobile integration of implanted devices for cardiac care," in *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, vol. 3, (Chicago, IL), pp. 1946–1949, 23–28 July 2000.

[6] M. L. Manwaring, V. D. Malbaša, and K. L. Manwaring, "Remote monitoring of intracranial pressure," in *Annals of the Academy of Studencia*, (Studencia, Yugoslavia), pp. 77–80, 7th International Inter University Scientific Meeting Academy of Studencia – Telemedicine, 20 Apr. 2001.

[7] Q. Huang and M. Oberle, "A 0.5-mw passive telemetry ic for biomedical applications," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 937–946, July 1998.

[8] J. A. V. Arx and K. Najafi, "A wireless single-chip telemetry-powered neural stimiulation system," in *IEEE International Solid-State Circuits Conference*, 1999.

[9] G. Gudnason and E. Bruun, *CMOS Circuit Design for RF Sensors*. Kluwer Academic Publishers, 2002.

[10] S.-Y. Lee, S.-C. Lee, and J. Chen, "Vlsi implemenation of wireless power and data transmission circuits for micro-stimulator," *VLSI Technology, Systems, and Applications, 2003 International Symposium on*, pp. 164–167, 6–8 Oct. 2003.

[11] H. J. Yoon, J. M. Jung, J. S. Jeong, and S. S. Yang, "Micro devices for a cerebrospinal fluid (csf) shunt system," *Sensors and Acutators*, vol. 110, pp. 68–76, 2004.

[12] R. N. Simons, D. G. Hall, and F. A. Miranda, "Spiral chip implantable radiator and printed loop external receptor for rf telemetry in bio-sensor systems," *Radio and Wireless Conference, 2004 IEEE*, pp. 203–206, 19–22 September 2004.

[13] J. W. Nilsson and S. A. Riedel, *Electric Circuits*. Addison-Wesley, 5 ed., 1996.

[14] D. J. Comer and D. T. Comer, *Advanced Electronic Circuit Design*. John Wiley & Sons, Inc., 2003.

[15] G. R. Steber, "A low cost automatic impedance bridge," *QST devoted entirely to Amateur Radio*, pp. 36–39, October 2005.

[16] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 3 ed., 2002.

[17] H. H. Asada and K. J. Cho, "A recursive frequency tracking method for passive telemetry sensors," Tech. Rep. 3-3, MIT Home Automation and Healthcare Consortium, March 2002.

[18] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 1 ed., 2000.