The Electrical Properties of Interfacial Double Layers

A thesis

submitted in partial fulfilment

of the requirements for the Degree

of

Doctor of Philosophy

at the

University of Waikato

by

Mark Hedley Jones

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

University of Waikato
2015

Abstract

When solids and liquids are brought together, interfacial double-layers are likely to form. They are too small to feel or see so their presence goes mostly unnoticed at the macroscopic level. A double layer is essentially a cluster of ions and/or charged molecules which are drawn from the body of a liquid to the surface of a solid. They are responsible for stabilising some of our most important fluids – blood, milk, paints, and inks. Without the protection of double-layers, these mixtures clump and lose their fluidity.

This thesis examines both electricity generation from, and the electrical impedance of, interfacial double layers.

Interfacial double-layers represent the underlying theme of this work, which is broken into two parts. In part I, double layers are used as a means of converting fluid-mechanical energy into electrical energy. My application for this is an energy harvester that could power electronic water meters. Domestic water meters are typically installed where electrical connection is not feasible. Harvesting energy at the meter may make electronic metering a feasible long-term solution. My findings show that double layer based energy harvesters are not efficient enough for this application *yet*. However, recent literature on the subject suggests large gains in efficiency may be possible using more exotic materials. Such gains would allow a compact harvester to generate enough energy to operate an electronic meter with wireless transmitter.

Part II models the electrical impedance of electrodes submerged in electrolytes. Double-layers contribute to the electrical impedance between solid-fluid interfaces. This work is important to designers of medical implants. Engineers use solutions of saline to mimic the environment experienced by their implants once implanted. This provides a way to test implant electronics without putting a patient at risk. A way of characterising the interface between electrodes and an electrolyte is to model it mathematically. An elec-

trical model of an electrode-electrolyte interface was recently developed by my supervisor, Jonathan Scott. I use that model to compare electrodes placed in solutions of saline to those placed in a living animal. Measurements of the two show that no one concentration of saline matches the situation inside a live spinal cavity. I then create a low-cost electrolyte test solution that better matches the impedance measured in a living animal's spinal cavity.

Acknowledgements

Thanks to Jonathan Scott (my chief supervisor), Steve Newcombe (Waikato University's award winning glass blower) and Peter Single (the senior electrical engineer at Saluda Medical) for their time, resources and patience. Thank you to my second supervisor, Marcus Wilson, for checking up every so often and help with proof-reading. Thanks also go to the University of Waikato for funding the first three years of this work with a Waikato Doctoral Scholarship. The support of my partner (Sarah) and my mother (Gina) has kept me on track during my research. Lastly, thank you to everyone who has contributed to open-source projects, especially those part of:

- The Linux kernel and GNU tools,
- Gnome desktop environment,
- Inkscape vector drawing software,
- Gimp image manipulation program,
- The Arch Linux distribution,
- TEX and its derivative LATEX,
- Python,
- ngSpice.

Work done throughout this thesis has relied heavily on these tools.

Contents

Chapter 1
Introduction

Is it possible to harvest energy from water without moving parts? What is the electrical impedance between electrodes in an electrolyte solution? Although seemingly unrelated, the answer to both lies in behaviour that occurs where liquids comes into contact with solids. That behaviour is the formation of arranged layers of liquid against the solid surface, called a double layer. This thesis is separated into two parts, each addressing one of the two questions above related to double layers. Part I studies double layers on insulating solids as a means of energy conversion. A number of double layer based power harvesters are fabricated and their output is measured. Converting fluid energy into electrical using double layers would allow for a "no moving parts" or "solid-state" energy harvester. Such a harvester could potentially outlast a mechanically based equivalents (due to reduced wear on moving components) and be cheaper to produce (owing to a lower component count). One application of particular interest is smart metering of domestic water usage. Part II models the electrical impedance between two electrodes when submerged in an electrolyte. Double layers play a large role in this impedance as they dictate the concentration of ions at the electrode's surface. Measurement of interface impedance allows for direct comparison between a range of environments into which electrodes are placed. This is important when designing an implant that will be inserted into a person. Before introducing background material on interfacial double layers, my motivation for doing this work is discussed. This is followed by a statement of originality and an outline of the structure of this thesis.

Motivation

My research began with the question "is it possible to harvest energy from water without moving parts?" The motivation to answer this question lay in the idea of building an energy harvester to power an electronic water meter.

Doing this without the moving parts of more traditional mechanisms, such as turbines, should increase the harvester's life-span and be generally more robust. I started by looking at three possible harvesting mechanisms:

- piezoelectric oscillators,
- electrostatic generators, and
- streaming potential cells

The piezoelectric oscillator was the equivalent of a water whistle with a vibrational energy harvester attached. The electrostatic generator was a version of Lord Kelvin's Electrostatic Generator with a harvesting application [1]. And the streaming potential cell was a mystery at the time. We knew geologists used streaming potentials to measure underground water flow. The only thing we knew about the mechanism was that forcing water through something somehow generated a voltage. Learning about that mechanism and answering the following questions started me on the path that became this thesis.

1. Where does streaming voltage come from?
2. What role does the geometry of a streaming device play?
3. Could you change the materials to get more voltage?

After experimentation and energy budgeting, I eventually concluded that streaming cell harvesters are not yet practical. Low conversion efficiency, a susceptibility to clogging and the need for high manufacturing tolerances make them unsuited for domestic water metering. However, this research allowed me to gain a working knowledge of interfacial double layers.

During my doctoral studies my supervisor, Jonathan Scott, took a sabbatical at Saluda Medical in Sydney. At the time, Saluda were developing a medical implant for spinal cord stimulation. Jonathan and Saluda's senior electronic engineer developed an electrical model of the impedance presented by electrodes immersed in a solution of saline. That model uses electrical components to simulate the electrical impedance between an electrode and an electrolyte. This means it can be entered into electrical simulation software and used to simulate an implanted electrode. Much of the behaviour the model simulates is due to double layers. Saluda's engineers use a dilute solution of phosphate buffered saline to approximate human spinal cavities into which their electrodes are implanted. They do not know how good this

approximation is, but it was the most appropriate mixture they had. The alternative was to embed an electrode in a live animal and measure the response - that is also what they do. Live animal experiments are costly and how they differ from solutions of saline is still unknown. The interface model is the starting point for the second phase of my research, which characterises the interface between an electrode and biological solutions. I have leveraged my understanding of interfacial double layers from part I to understand how the model worked, and use it correctly.

Statement of Originality

The work contained in this thesis is my own except where otherwise acknowledged.

Publications Arising From This Work

- Jones, M.H. & Scott, J. (2014). *Scaling of Electrode-Electrolyte Interface Model Parameters In Phosphate Buffered Saline.* Published in IEEE Transactions on Biomedical Circuits and Systems, Issue 99.
- Jones, M.H. & Scott, J. (2014). *Feasibility of Harvesting Power to Run a Domestic Water Meter Using Streaming Cell Technology.* In proceedings of the 21st Electronics New Zealand Conference, ENZCON 2014, Waikato University, Hamilton, New Zealand.
- Jones, M.H. & Scott, J.B. (2011). *The energy efficiency of 8-bit low-power microcontrollers.* In Proceedings of the 18th Electronics New Zealand Conference, ENZCON 2011, Massey University, Palmerston North, 21-22 November 2011, pp. 87-90.

Thesis Outline

This thesis is broken into two parts. Part I is concerned with energy harvesting with double layers, specifically by the use of streaming cells. Part II measures and models the impedance of an interface, specifically those between implant electrodes. Put simply, part I deals with double layers on insulating surfaces, and part II deals with double layers on conductive surfaces.

The next chapter (chapter 2) contains background material on double layers, including their formation and a breakdown of their structure. The topics of streaming cells and impedance modelling are introduced in that

chapter. Then, part I begins by looking at streaming cells for the purpose of running an energy harvesting water meter. It starts at chapter 3 with a brief mathematical analysis, followed by streaming cell fabrication, and then measurements of their ability to harvest energy. Chapter 4 estimates the amount of energy that would be available to a streaming cell energy harvester. Chapter 5 looks at the amount of energy needed to run microprocessors and wireless transmitters. This concludes with an estimate of the amount of energy required to run an electronic water meter. To conclude part I, chapter 6 combines the data obtained and the feasibility of streaming cell energy harvesting for electronic water metering is discussed.

The second part of the thesis (part II) starts with an overview of the electrode interface model (chapter 7). Chapter 8 deals with measurement of the various model parameters in both phosphate buffered saline (section 8.1) and inside a live sheep's spinal cavity (section 8.2). Finally, chapter 9 presents work on the creation of a mixture designed to better represent the environment inside sheep spine compared to phosphate buffered saline (PBS).

Chapter 2
Background

Double Layers

Modelling and measuring the electrical properties of interfacial double layers draws on both electronic and chemical concepts. Those with an electronic background are unlikely to be familiar with double layers. This section provides background on what a double layer is and how one is formed, beginning with a discussion of liquids.

Formation

Double-layers are organised layers of liquid that are comprised of two distinct layers. Because double layers are a property of liquids, and most common liquids are water based, the properties of water is an appropriate place to start. The following properties of water are not necessary for the formation of double layers, but knowing of them helps build a mental model of the system. At the microscopic scale, individual atoms and molecules within liquids interact with complexity. The density of atoms and molecules in water is extreme, 3.33×10^{22} H_2O molecules per millilitre. These molecules are polar, meaning that one side is negatively charged while the other appears positively charged, shown in fig. 2.1. This causes them to respond to electric fields by rotating, so as to minimise their potential energy in the field. Because of this, any ions present become surrounded by arranged volumes of water.

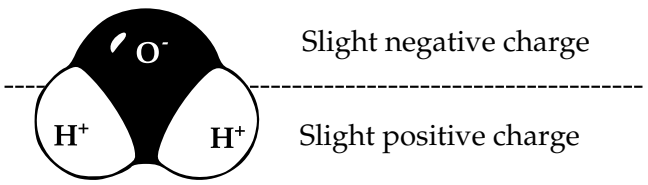


Figure 2.1: Graphic showing a representation of a water molecule

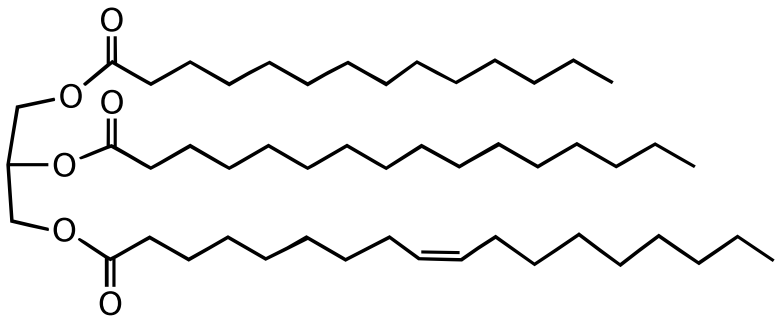


Figure 2.2: Graphic showing the structural formula of a butterfat molecule. Milk is made of butterfat molecules suspended in water. Double layers are responsible for keeping the butterfat molecules from clumping together.

For example, a positive ion will be surrounded by water molecules orientated such that their hydrogen atoms all point away from the ion. It is also possible for water to spontaneously disassociate from H_2O into a proton (H^+) and a hydroxide anion (OH^-).

To form a double layer, a liquid containing ions must meet a solid object with charged trapped at its surface. Once this happens, the ions within the liquid are drawn to, or repelled from, the solid's surface. The point where the two states of matter meet is called the interface. Those ions that have been drawn to the interface collect together to form a double layer. Double layers can occur when pure water is the liquid, because of its ability to disassociate, but mostly it is the ions from an electrolyte solution (one containing salts) that form the layer [2]. The double layer is simply the collection of ions drawn from a liquid to the surface of a solid.

'Solid' may refer to the walls of a container or particulates suspended in solution. When a particulate is suspended throughout a solution it is referred to as an emulsion. For example, milk is such an emulsion of butterfat in water. Figure 2.2 shows the structure of a butterfat molecule. A butterfat molecule is a long structure that can be approximated as being a solid. The stability of an emulsion, such as milk, depends on the strength of the double layers that encapsulate each particle. These double layers shield the molecules or particles from one another electrically. By shielding them from each other they are unable to collect and bond together. This shielding prevents milk from coagulating and turning to lumps.

A physical model

In the previous section, a brief explanation of what a double layer is and where they form was given. The anatomy of a double layer and some of its

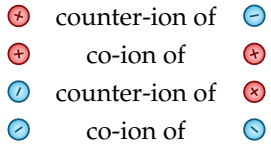


Figure 2.3: Graphic showing relationship between co-ion and counter-ions.

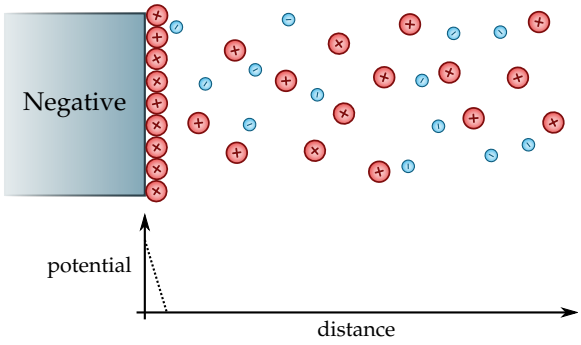


Figure 2.4: Diagram showing the structure of the Helmholtz layer.

properties will now be defined. When discussing ions, it is convenient to use the term 'co-ion' and 'counter-ion'. These terms refer to ions containing charge – like or opposite – in polarity to a second charge or body of charge. For example, if a negatively charged surface attracts positively charged ions then positive ions are counter-ions to the surface. Likewise, if a positively charged surface was to repel a positive ion then the positive ion is the co-ion. The terms are convenient because they remove the need to identify specific polarities during discussion. This relationship is shown in fig. 2.3.

Three models of the double layer have been put forward, beginning with the Helmholtz Model [3]. Helmholtz first proposed his parallel plate capacitor based model in 1879 [4]. His model consists of two layers of surface charge, one inside the solid and one in the liquid. Counter-ions sit in a *compact layer*, meaning that they are bound to the surface and are therefore immobile [5]. Figure 2.4 shows this as a row of tightly packed positive ions along the solid's surface. Past the layer of bound surface ions there is no effect from the charged surface of the solid. In essence, his model describes the interface as a single layer of ions held against the edge of a solid. The problem with this model is its inability to predict the layer's capacitance. Measured capacitance of double layers depends on the potential difference across the layer, and the concentration of ions in the solution [6]. Helmholtz's model does not account these dependencies, and therefore is not an accurate representation.

Later, Goüy and Chapman independently proposed that charge in the

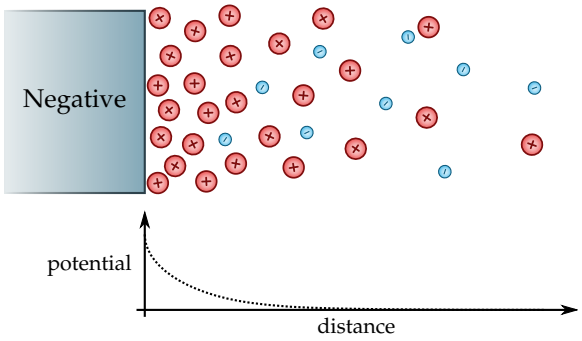


Figure 2.5: Diagram showing the structure of the Goüy-Chapman layer.

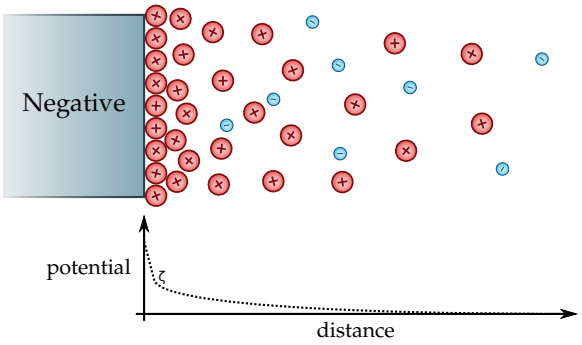


Figure 2.6: Diagram showing the structure of the Goüy-Chapman-Stern layer.

liquid phase may instead be held in a *diffuse layer* [7]. This meant that ions in the layer were not fixed and that the density of charge in the layer could vary. Figure 2.5 illustrates the concept by the lack of ions bound to the surface and the gradual decline in counter-ion concentration with distance from the surface. The Goüy-Chapmam Model accounts for the observed variation in capacitance by distributing charge in the liquid as a concentration gradient from the surface of the solid. The layer is free to change its concentration profile in response to applied voltages and ionic concentration. In the case of a higher voltage, the layer is pulled closer to the surface, becoming thinner. In the case of a higher electrolytic concentration, the layer is more concentrated with a higher charge density, again becoming thinner. The Goüy-Chapmam Model predicts the change in capacitance by growing or shrinking the size of the layer, but it still fails to predict the capacitance at high ionic concentrations. This is in part because it makes no account for the physical size of the ions in the electrolyte; it models them as point charges [6]. Ions in this model can get infinitely close to the surface regardless of their size. This becomes a problem at high ionic concentration where the surface becomes crowded.

In 1924, Otto Stern published his modified version of the Goüy-Chapmam

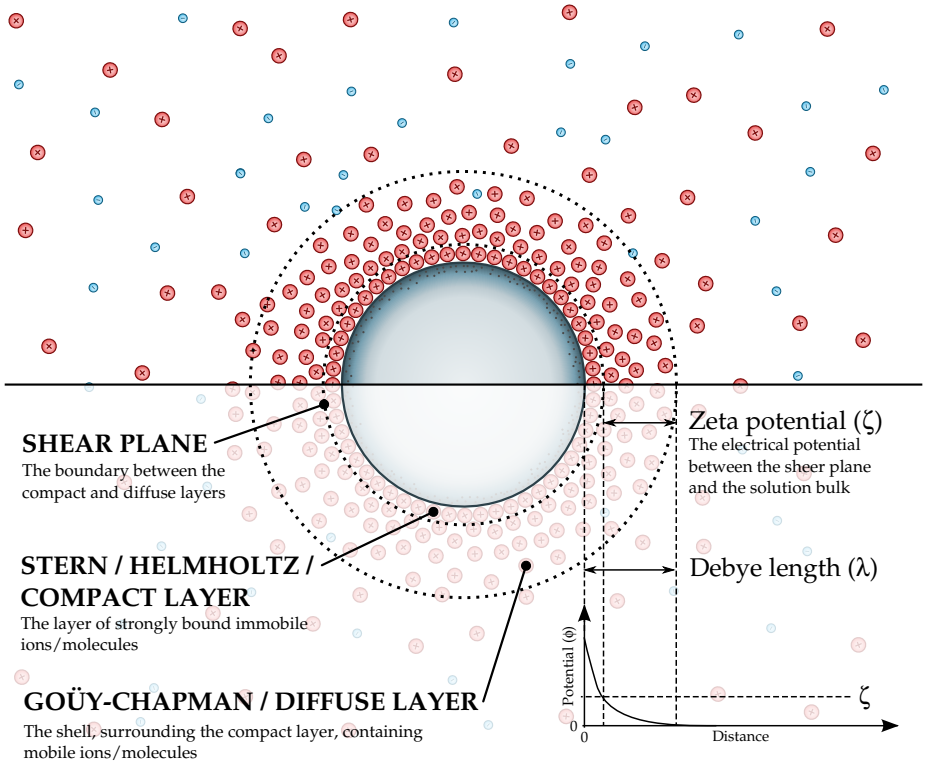


Figure 2.7: Diagram showing elements of a double-layer adsorbed to a negative sphere.

model [8]. This model, illustrated in figure 2.6, extends the Goüy-Chapmam model by setting the minimum distance an ion can get to the solid's surface. A consequence of this is that the compact layer as is seen in Helmholtz's model is reintroduced, but the model still allows for a concentration gradient exterior to this layer. It resembles the Helmholtz model at high ionic concentration but accounts for spread in the layer dimensions at lower concentrations. The Stern, sometimes referred to as the Goüy-Chapmam-Stern, model is a well accepted model of the interfacial double layer [9].

Structure

Figure 2.7 shows double layer organisation according to the currently accepted Goüy-Chapmam-Stern model. It shows the compact layer adsorbed to the surface of the suspended solid. In this layer the ions are immobile due to the electrical strength at the surface. Surrounding this layer is the diffuse layer. Ions here are still drawn to the solid, but not so strongly as to be immobile. The voltage in this layer (relative to the bulk of the solution) decays from that at the compact/diffuse layer boundary to zero within the solution bulk. The

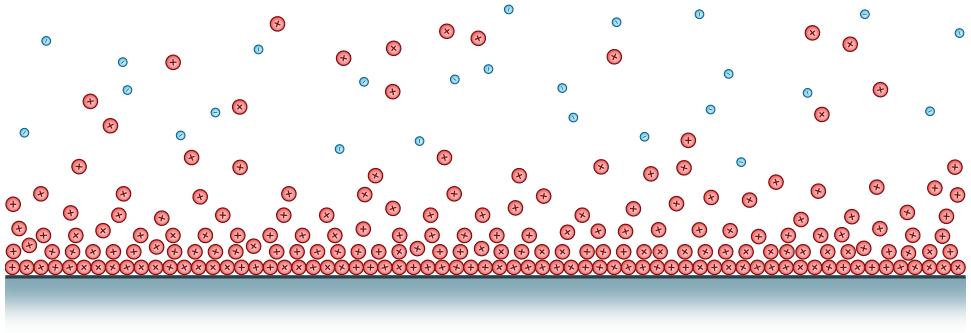


Figure 2.8: Diagram showing the formation of a double layer along a solid wall.

Zeta potential is the voltage between those two points, i.e., it is the voltage at the plane where the diffuse layer (Goüy-Chapmam layer) meets the compact layer (Helmholtz layer) - relative to the voltage in the bulk of the solution. The dividing plane between the two layers is called the shear plane. It represents the nearest distance from the surface at which the layer can move laterally. The shear plane is an important parameter with linear geometries, such as the inside of a pipe, as it represents the true no-slip boundary.

The thickness of a typical double layer is between $1-100\,\text{nm}$ [10], as defined by its Debye length [11]. The Debye length is the distance between the interface and the point in the liquid where the voltage is no longer affected by the charged interface. As mentioned in the previous section, this varies based on the ionic concentration of the solution and the charge at the solid's surface.

Streaming Cells

Consider a double layer that has formed along the edge of a perfectly flat surface. Figure 2.8 illustrates this situation, where the walls are negatively charged and therefore the counter-ions are positively charged. Counter-ions, separated from the bulk of the liquid, line the exterior of the wall. Although charge has been separated out from the bulk of the electrolyte solution, this does not lead to a usable form of electrical energy.

Basic science tells us that energy cannot be created or destroyed, it must be converted from one form to another. In this case, counter-ions are electrostatically bound to the interface and removing them requires work. Although the counter-ion density has been increased at the boundary, the charge is not free. Migration of charge to the walls ceases once the surface potential has been neutralised. Double layer formation takes work to undo and the process stops once the layer is formed. Generating electrical energy from the layer will require an additional form of energy to help isolate collected charge, in this

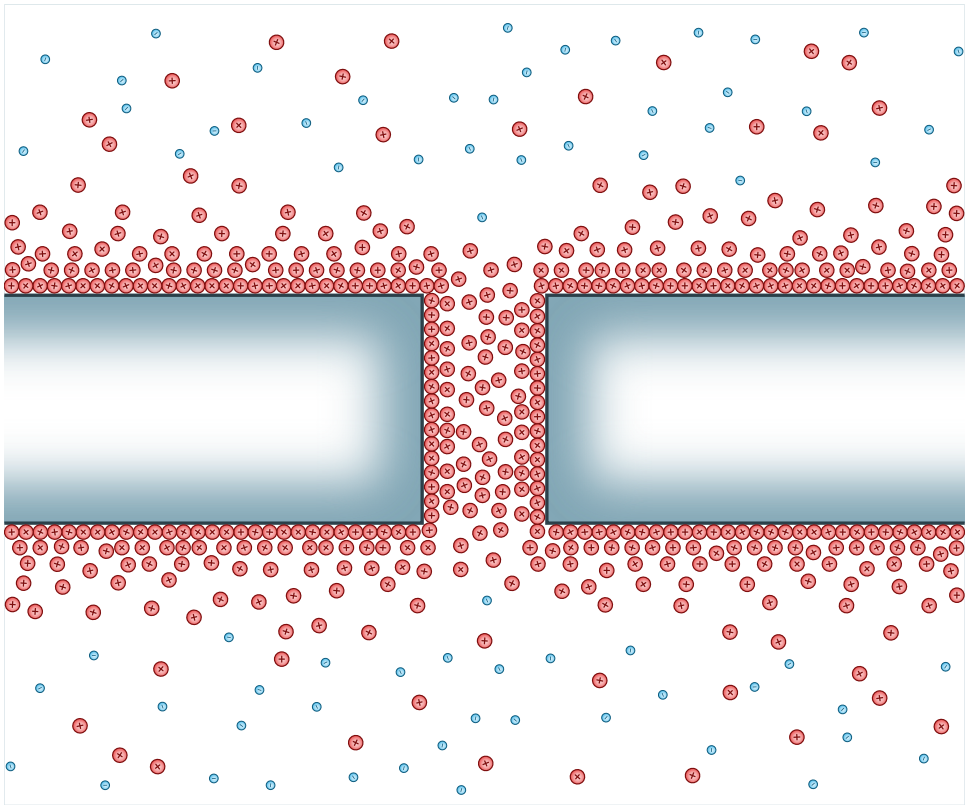


Figure 2.9: Diagram showing double layer formation within a streaming cell in a state of equilibrium.

case mechanical. Liquids pass mechanical power as a combination of pressure and flow, similar to the way electrical power is transferred by voltage and current. Forcing liquid through a harvester will cause a drop in pressure as the liquid passes through the harvesting mechanism. The mechanisms presented here use mechanical power to create an ionic imbalance between the bodies of water on each side of the harvesting mechanism. This means separating and isolating negative and positive ions from each other. Figure 2.9 shows another charged wall, but with the addition of a small channel. Notice that the channel contains no co-ions, it is exclusively occupied by counter-ions. The ratio of counter-ions to co-ions within the channel is controlled by the width of the channel. The narrower the channel, the less likely it is for co-ions to get inside. This channel is small enough that the layers overlap one another, repelling co-ions completely. The channel and the two separated bodies of liquid now form an energy harvester. Counter-ion rich fluid is transported across the channel by applying a pressure differential. As counter-ions exit the channel on the low-pressure side, new ions move to replenish the double layer on the high-pressure side. A diagram showing the channel geometry, but with

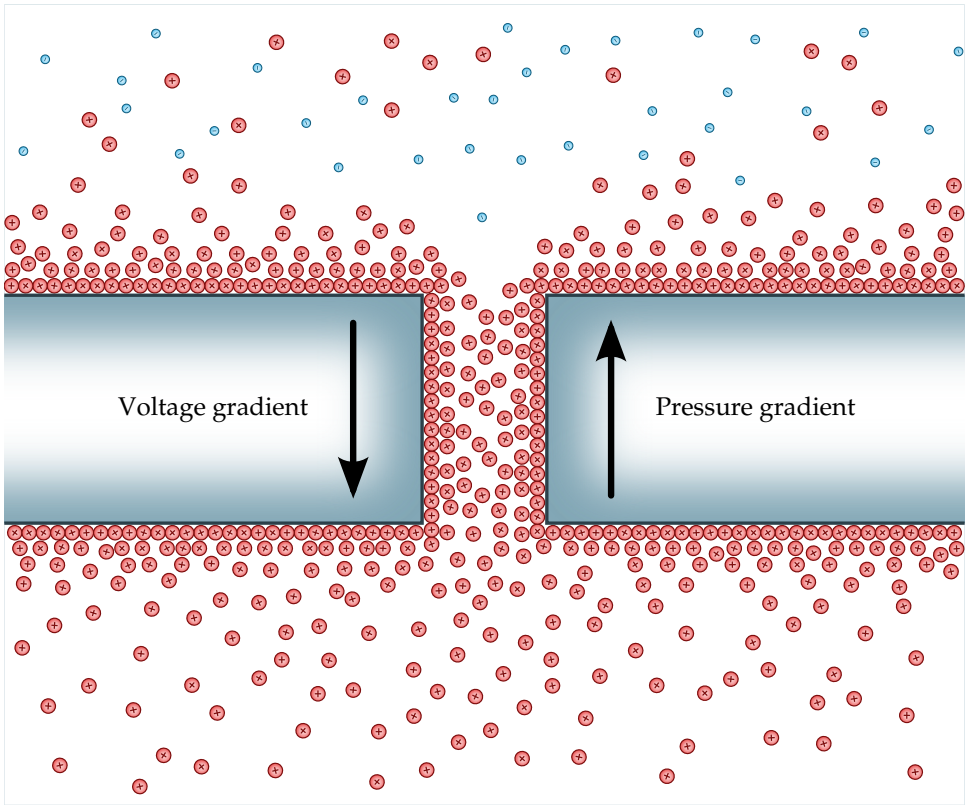


Figure 2.10: Diagram showing double layer formation within a streaming cell that has a pressure differential applied.

pressure applied and a voltage gradient generated is shown as figure 2.10. The two walls and channel can now be referred to as a streaming cell. Streaming cells are able to continuously separate ions from an electrolyte fluid. The voltage across a streaming cell increases as those ions are pumped through and collect on each side. This only works when the solid's surface has charge at its surface, necessary to create the double layer in the first place.

A channel can be created individually using a range of fabrication methods, such as chemical etching or using narrowly separated parallel plates. They can also be formed en masse by using porous materials such as glass or ceramics, where the pores themselves act as channels. Glass has the convenient property that it spontaneously obtains a negative surface charge when in contact with water, the requirement for double layer formation. This surface charge is caused by the deprotonation of surface silanol groups in glass ($SiOH \leftrightarrows SiO^- + H^+$) [12]. By immersing a glass channel in an electrolyte solution, the glass donates protons into the solution leaving its surface negatively charged. In turn, a positively charged double layer lines the channel's inner walls ready to be pumped through the channel. This means the glass

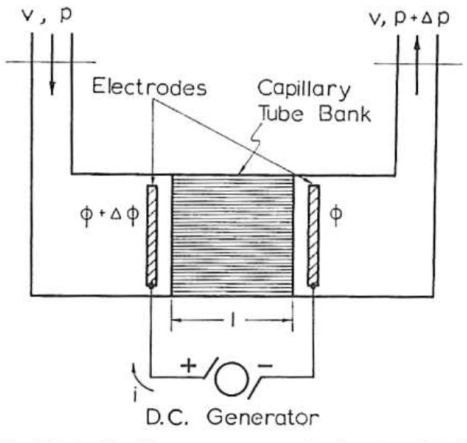


Figure 2.11: Image of Osterle's electrokinetic pumping cell, as taken from [13]

channels have a higher voltage on the low-pressure side and a lower voltage on the high-pressure side. The low pressure side is at the harvester's outlet, where the counter-ions (cations in this case) collect. The extra positive ions gives the liquid on that side a positive charge relative to the inlet side of the harvester.

The concept behind the device is relatively straight-forward, but the physical reality is complex. The diagrams presented here are simplified, having perfectly flat walls containing single atom ions carrying a single charge. No mention of molecules has been made, which increases the complexity further. Polar molecules such as water have positive and negative components offset in space. Although simplified, this discussion illustrates the how streaming cells work. Next, literature concerning the operation, design and improvements to streaming cell technology is presented and discussed.

Literature review

In 1964, a paper by Osterle gave an analysis of energy conversion from streaming cells, both for the purpose of pumping fluid or generating electrical power [13]. The cell he used consisted of fine capillary tubes stacked together to form a streaming cell. A diagram of that cell, in its pumping configuration, is reproduced here as fig. 2.11. Importantly, he shows that a streaming cell has the same conversion efficiency whether it is in a pumping mode, where electrical energy is supplied, or in a generating mode, where electrical energy is produced. This opens the body of relevant literature on the subject to include electrokinetic pumping devices. Based on his analysis, Osterle gives an illustrative example of a streaming cell producing electrical energy. He states that his tube bank having a volume of 100 cm^3 with 100 kPa of hydrostatic

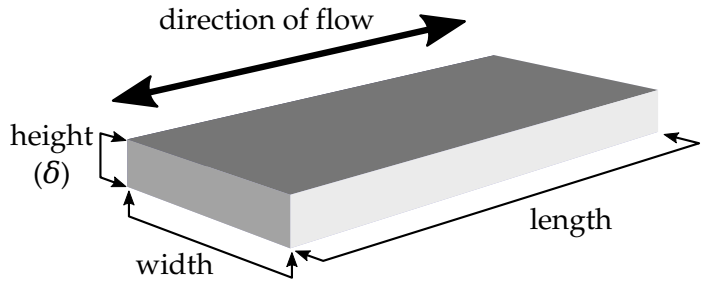


Figure 2.12: Diagram showing how the properties 'width', 'height', and 'length' refer to the geometry of a streaming cell or micro-channel in this thesis.

pressure applied would be capable of producing 0.49 W of electrical energy. This would require 125 W of pumping power to achieve, giving an energy conversion efficiency of 0.392 %.

Within the space of a year three papers related to properties of fluid flow in fine capillaries, such as those used by Osterle are presented. Burgreen and Nakache investigate the fluid flow when the capillaries are rectangular [14], and the efficiency of such capillaries when used to either generate electrical power or as a pump [15]. Their work develops the mathematics behind rectangular streaming cells and shows fine glass capillaries are equally efficient when used to generate electrical power or to induce liquid pumping. Rice and Whitehead make an analysis of fluid flow profiles that consider the effect of double layer interactions [16]. They show how the double layer affects the level to which liquid can permeate a material populated with cavities. Together these three papers mark the beginning of research into streaming cells.

There appears to be little published research into streaming cells since then until 2003, when a surge of papers related to optimal dimensions of streaming cells appear. An analysis relating energy conversion efficiency to the length of a streaming cell channel indicated that short cells are the most efficient [17]. Figure 2.12 shows the meaning of 'width', 'height', and 'length' when referring to streaming cells or micro-channels in this thesis. However, more recent work by Chang and Yang shows a decrease in conversion efficiency at maximum power when the channel length is low [18]. This work suggests there is an optimum channel length, which will also be dependent on the conductivity of the working fluid. Investigation by Daiguji et al. into the relationship between the Debye length of the double layer and streaming cell conversion efficiency found that a channel is most efficient when its height is twice that of the Debye length [19]. This corresponds to the point at which double layers formed within a cell begin to overlap with one another.

In 2005 a seminal paper by van der Heyden et al. reported on streaming cell measurements made in a single micro-channel 70 µm in height [20]. Many valuable contributions were detailed in this paper, namely they:

1. confirmed that reversing the polarity of surface potential reverses the direction of the streaming current.

2. found that the maximum conversion efficiency corresponded to channels where double layers begin to overlap. This confirms the relationship put forward by Daiguji et al. [19]

3. showed that boundary conditions involving constant surface potentials, used up to this point to model streaming, are inaccurate.

4. predicted a maximum energy conversion efficiency of $\sim$6 % for potassium chloride solutions of 1×10^{-5} mol in silica channels of height 145 nm.

Subsequent research by the same authors show that conversion efficiency is maximised at low salt concentrations [21]. They also predict an energy conversion efficiency of 12 % for streaming cells using electrolyte solutions containing lithium. Around the same time, Daiguji et al. publish work suggesting that in order to increase cell efficiency one may either reduce the channel height or decrease the ionic concentration of the working fluid [22]. This supports the work of van der Heyden et al. with respect to efficiency gains with the working fluids having low ionic concentrations.

In 2007, van der Heyden et al. publish a measured energy conversion of 3.2 % [23]. They suggest that the conversion efficiency of a channel is limited by a property termed 'Stern conductance'. The concept of Stern conductance is that the Stern layer (see fig. 2.7) itself provides a pathway for electrical conduction. This conduction turns the surface of the glass into an electrically conductive surface causing the cell to partially self-discharge. Stern conductance is often referred to simply as 'surface conductance'. Davidson and Xuan published a mathematical model shortly after confirming the role of Stern conductance on streaming cells, in particular those with low ionic strength [24]. They suggest that this is the reason for poor measured efficiencies in light of the much higher predicted values.

Most recently, the concept of hydrodynamic slip has been applied to streaming cells as a way of increasing conversion efficiency. Estimates from mathematical models predict conversion efficiencies between 30 % and 70 % [25–27]. Hydrodynamic slip refers to the ability of a fluid to move, or slip, perpendicular to a boundary (such as the wall of a pipe). Slip is advantageous to

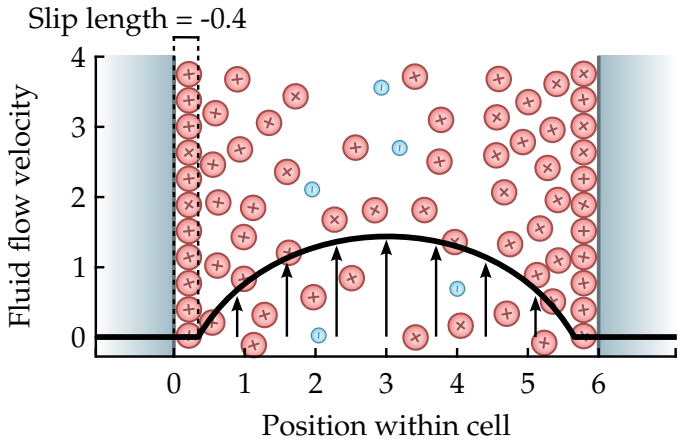


Figure 2.13: Diagram showing negative slip along a boundary. The direction of flow is upward. The parabola illustrates the flow profile, with the arrows indicating fluid velocity.

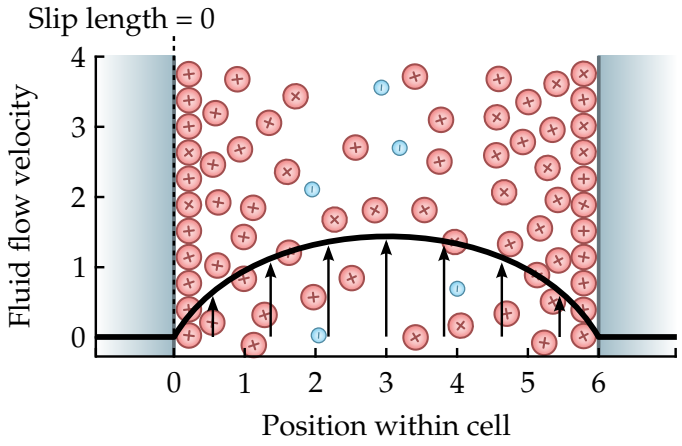


Figure 2.14: Diagram showing the no slip boundary condition along a boundary. The fluid does not move at the wall.

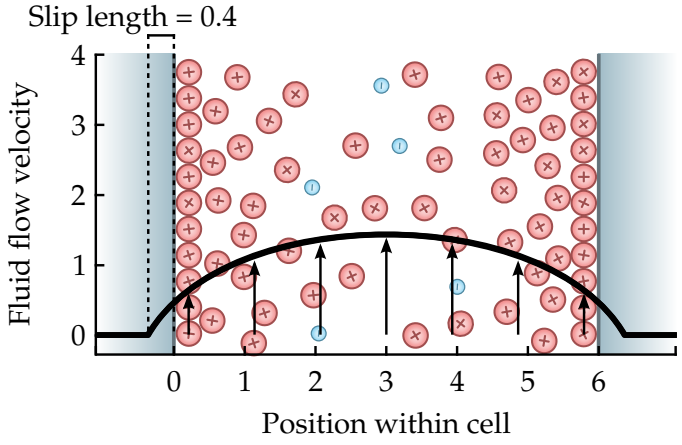


Figure 2.15: Diagram showing positive slip along a boundary. Fluid at the boundary moves relative to the wall.

streaming cells because it permits ions in the Stern layer to move relative to the wall. Hydrodynamic slip has an associated length (the slip-length), which refers to the distance between the point at which the parabolic flow profile drops to zero and the wall. Figure 2.14 illustrates a typical 'no-slip boundary condition' where the fluid at the solid/fluid boundary is static. This condition, along with viscosity, is responsible for the parabolic flow profile a fluid takes as it moves through pipes. As the highest counter-ion density is found at the boundary, where the flow is stationary, much of the charge a streaming cell is designed to pump sits dormant. Eijkel showed that a channel's zeta potential and its slip length are linked [28]. The general problem with slip-based mechanisms is that a high zeta potential is optimal for double layer formation, however it also promotes wetting. Wetting and hydrodynamic slip are related to each other by the strength of attraction between a liquid and a solid. To explain, the terms hydrophobic and hydrophilic are used to describe surfaces that repel and attract water. A hydrophobic surface has a low tendency to support water, i.e., water will bead and roll off a hydrophobic surface. Conversely, a hydrophilic surface is one that water *is* attracted to, causing a droplet to spread and stick to the surface. Hydrodynamic slip occurs when a channel's walls are hydrophobic, allowing water at the interface to slip along the boundary of the solid. High zeta potentials attract water to the solids surface due to electrostatic attraction and Eijkel's publication illustrates that the zeta potential and hydrodynamic slip are related. In order to improve the situation in steaming cells, a surface should be both non-wetting and hold a high surface charge. Conservative estimates place an efficiency of 40 % on cells having slip lengths tens of nanometres long, obtainable using carbon nanotubes and using solutions having low salt concentration.

Theoretical predictions of the efficiency of standard micro/nano-fluidic channels are 2% for pure water and 7% for sodium chloride [21]. However, measured conversion efficiencies as reported thus far are:

- "far less than 1 %" forcing potassium chloride through a porous glass plug having pores in the range $1 - 1.6\,\mu m$ [9].
- 0.01% by forcing tap water through porous glass with pore sizes from $10 – 16\,\mu m$ [17].
- 0.8% by forcing pure water through a ceramic rod populated with $6\,\mu m$ pores [29].
- 3% by forcing a sodium chloride solution through a $75\,nm$ by $50\,\mu m$ silica channel [23].

- 0.77% by forcing a sodium chloride solution through a 200 nm pore in an alumina membrane [30].
- 5% by forcing a sodium chloride solution through a 0.5 nm cylindrical pore in polyethylene terephthalate foil [31].

It is clear from the literature that there is significant progress to be made with respect to increasing the conversion efficiency of streaming cells. Techniques to induce hydrodynamic slip at the fluid-solid interface are predicted to increase this efficiency to 30-40% [26, 27], but progress in this area is dictated by advancements in materials science. Experimental results utilising slip enhanced channels have not yet been reported in the literature. Surface enhanced channels will not be investigated due to manufacturing difficulty, cost, and the level of scientific development required to make progress.

In summary, the finding that maximum conversion efficiency occurs at low ionic concentration supports the use of tap-water as a working fluid. Using glass as a substrate seems to be a suitable choice, which is both cost effective and easy to source. The dimensions of channels found in the literature suggest that fabricating trial cells is feasible with the equipment available. A conversion efficiency of 0.01 % should be achievable with the use of a porous glass plug when tap-water is used as the working fluid. This efficiency will be used a baseline to compare measured efficiency from fabricated cells. That concludes background material on streaming cells for energy harvesting.

Impedance Modelling

Part II studies the impedance of an interface between an electrolyte and an electrode. Most electronic components have well defined electrical impedances, or models describing those impedances. Circuit simulators, such as the popular SPICE, are designed to deal with branches of components whose currents are functions of voltage, or the integral or first differential of voltage. One complication with modelling double layers is that they are described by fractional-order functions. Fractional order functions can be differentiation or integral functions but are only partially applied. Fractional calculus is not mainstream mathematics, but does find use in niche areas such as describing double layer behaviour. For non-linear electronic components such as transistors and diodes, SPICE has models describing their behaviour built in. But SPICE has no way to deal with circuit elements described by fractional order differentiation or integration functions. The choice of SPICE as a means of simulating an electrode-electrolyte interface offers convenience to engineers

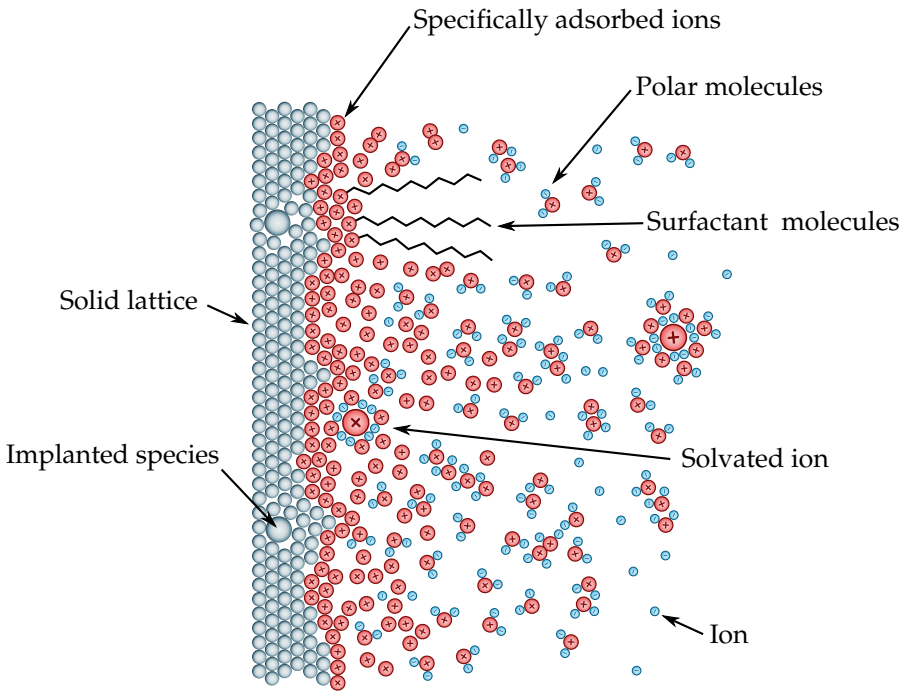


Figure 2.16: Diagram of a liquid-solid interface showing various types of molecular interactions, surface imperfections and polar molecules. This diagram is based on the work of Bard et al. [32]

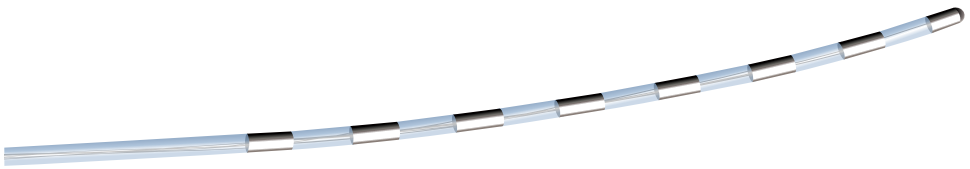


Figure 2.17: Drawing of the St. Jude Medical "Octrode". Such electrodes are used with spinal stimulation implants.

who can integrate the model into their designs. Other tools such as Python or Matlab would also be capable of simulating the interface's response, but SPICE is fast, free and well known by engineers. So how would it be possible to enter an electrode-electrolyte-electrode system into a SPICE circuit and simulate it? Part II of this thesis uses a model of an electrode-electrolyte interface to solve this problem for implant designers. That model is built up from combinations of basic components, which all have well known impedances. The model can then be entered into almost any circuit simulator. This means an electronic engineer can add it into their design simulate the electrical loading between electrodes in an electrolyte bath.

Illustrations of double layers were presented in section 2.1 showing how the interface is comprised of layers. A more realistic illustration of a solid-liquid layer is shown in fig. 2.16. It shows interactions between polar molecules, between polar molecules and ions, surfactants at the interface, an imperfect solid/liquid boundary, and the Stern layer. The complexity of interactions that happen at the interface are what make is so difficult to model. Complexity is not only limited to the electrolyte as the geometries of both the electrodes and the environment containing the electrolyte must be taken into consideration.

To the designers of medical implant devices, estimating the impedance of electrodes in an electrolyte is critical for safe stimulator design. Saluda Medical is an emerging company located in Sydney, Australia, in the business of designing high-tech spinal cord stimulator implants. Their engineers have designed a stimulator fit for human implantation, but have questions around the electrical impedance their implants will experience once implanted. Getting the value wrong means the implant would fail to deliver the desired stimulus current, or otherwise fail due the presence of an unexpected load. Naturally, Saluda want a way to determine that impedance before they put stimulators into human spinal cavities. This means modelling the electrode-electrolyte interface itself, which is a central idea in interface science. Part II of this thesis looks at ways of doing that using a model of the interface based on electronic components. Such a model is suitable for entry into common

circuit simulators, such as SPICE, that electronic engineers already use. The result for an engineer would be a black box circuit element that connects electrodes together inside an existing SPICE model. That black box would fill the missing gap between their electrodes in their electronic model, giving them a way to simulate the entire system during the design phase. Ultimately this could mean shorter design cycles with less animal testing saving both time and money.

Literature review

In 1899, Warburg presented his interpretation of the impedance between electrodes in an electrolyte [33]. That interpretation was that the electrode's response is dominated by the diffusion of active species through the electrolyte bulk. He described the situation as a resistor in parallel with a capacitor, a combination since referred to as a 'Warburg impedance' or 'Warburg element'. He draws the following relationships for the phase and capacitance of an electrode:

$$\phi = \pi/4 \tag{2.1}$$

$$C_W = k/f^{0.5} \tag{2.2}$$

where k depends on the metal, C_W is the Warburg capacitance, f is the applied frequency, and θ is the phase angle. The above relationships are drawn for electrodes with an infinitely small current density and does not account for DC behaviour. This concept of a Warburg element goes on to be used in many, if not most, electrode interface models published since. However, shortly after, Fricke showed that for low current densities the electrode capacitance and phase angle are described more accurately by:

$$\phi = m\pi/2 \tag{2.3}$$

$$W_W = k/\omega^m \tag{2.4}$$

where m and k depend on the type of metal used, with m being somewhere between 0.15 and 0.32 [34]. These relationships applied for stimuli in the frequency range of $100 - 3500$ Hz, but still does not account for the DC situation. In 1936 Murdock and Zimmerman publish empirical data in support of Fricke's phase relation [35]. Over the next fifty years, numerous studies of the electrode interface and its corresponding electrical equivalent circuit have been made [36–38]. However, no progress is made over that time toward modelling the DC response of an electrode interface. In 1975, a model put forward

by Geddes et. al adds resistance in parallel with the Warburg impedance to account for that DC response [39]. Seven years later, Onaral and Schwan measure the impedance of platinum electrodes in saline from 1 mHz-1 kHz [40]. They confirm Fricke's phase angle holds down to a frequency of 10 mHz, but comment on deviation below that. This suggests there is a limit to how well the traditional Warburg element accounts for low frequency signals. Like Geedes et. al they also use resistance in parallel with the Warburg element to account for the DC response of the interface. In their work they propose that a Cole–Cole relaxation model would describe the electrode interface behaviour over all frequencies. Lastly, Schwan showed that the value of interface capacitance and series resistance is not fixed [41]. Rather, it is dependent on the current density (the total current being passed by an electrode divided by its surface area).

Overall, the trend in electrode interface publications appears to be a gradual refinement of the initial model put forward by Warburg in 1899. Each modifies that model slightly to account for conditions outside of its predictive capability. The Warburg impedance is accurate when the current density is low, but fails to predict changes in capacitance and resistance of higher densities. A recent review of interface modelling indicates there is still no reliable circuit model of the electrode-electrolyte interface [4].

The electrode model used in part II is a simplified version of the Scott–Single electrode interface model [42]. It uses a constant phase element (CPE) in place of the Warburg element, which has the form:

$$Y(\omega) = Y_0(j\omega^n) \tag{2.5}$$

where Y_0 is an admittance constant, $\omega = \pi f$ and n determines the angle. When $n = 0$, it represents a resistor, when $n = 1$ it describes a capacitor, and when $n = 0.5$ it is the Warburg element. This allows the Scott-Single model to the interface with any phase angle between 0 and 90°. Based on the work of Morrison, the Scott-Single model implements the CPE as a series of resistor-capacitor branches [43]. This allows the model to be solved by simulation software working in the time-domain, such as SPICE. It is also unique within the literature for its use of a memristor and diode pair as a means of describing Faradaic current conduction, something thus far described by resistance.

Comparisons of permittivity and resistivity between biological tissues have been published [44], but comparisons between standard saline solutions and biological fluids have not been found.

Part I

Double Layers on Insulators: Harvesting Energy

In section 2.1, the topic of interfacial double layers were introduced. Then, in section 2.2, a way of utilising double layers to harvest energy - in a process called streaming - was studied. The possibility of using streaming cells as a means of powering electronic water meters is investigated. The following chapter (chapter 3) presents a brief mathematical analysis before a batch of cells are fabricated and measured. Measurements from those cells are used to determine their efficiency. Then in chapter 4 the applicability of streaming cells for use in water metering is discussed. This presents an estimation of water use in a typical New Zealand home that will be used to determine the amount of energy available to a harvester. Chapter 5 measures the energy consumption of low-power, 8-bit microcontrollers and wireless transmitters. The measured data is used to estimate the energy requirements of an electronic water meter with wireless transmitter. Finally, in chapter 6, the feasibility of using streaming cells as energy harvesters for water meters based on the previous measurements is discussed and conclusions are drawn.

Chapter 3
Streaming Cell Energy Harvesting

This chapter begins with a mathematical analysis of streaming cells and operating parameters. Then, in section 3.2, a variety of streaming cell designs are built. Early attempts at making streaming cells are presented, followed by more successful streaming cell designs. Ten streaming cells are made using that design with each having slightly different geometric dimensions. The electrical output and energy conversion efficiency of these cells is measured in section 3.3. Measurement results are discussed in section 3.5, followed by my concluding remarks.

General Analysis

A basic model of operation for a streaming cell is established. This determines what parameters are important when maximising a cell's output power.

Mathematics

Mathematical analysis of streaming cells provides a basic understanding of the parameters involved with their output and geometry. Rigorous mathematical analysis of streaming cell performance is extremely involved and is well detailed in the literature [45]. As aspects of a double layer's structure are still not fully understood, the mathematics behind them is still being developed. Computer simulation and mathematical models continue to shed light on ionic organisation at liquid-solid interfaces [46]. For that reason, I have not attempted to model a streaming cell physically. Instead, I piece together a relatively simple mathematical model quantifying important operating parameters.

Streaming voltage

Gu and Li derived the following equation relating the streaming voltage to the pressure applied across a streaming cell [47].

$$\frac{V_s}{\Delta P} = \frac{\varepsilon_r \, \varepsilon_0 \, \zeta}{\mu(\sigma + \frac{2}{\delta}\lambda)} \tag{3.1}$$

where:

V_s is streaming voltage,

ΔP is the hydrostatic pressure across the channel,

ϵ_r is the relative permittivity of the liquid,

ϵ_0 is the absolute permittivity of free space,

ζ is zeta potential,

μ is the fluid's viscosity,

σ is the fluid's bulk conductivity,

δ is the channel's height (refer to fig. 2.12),

λ is the channel's surface conductivity.

This equation is specific to parallel plate channels, of the type constructed in the following section. It requires that the width of the channel's cross-section is a minimum of twenty times larger than its height, which it is for the cells constructed here. Gu and Li use this equation as a means of finding the zeta potential and surface conductance by rearranging it into the following form:

$$\frac{\varepsilon_r \, \varepsilon_0 \, \Delta P}{\mu \, V_s \, \sigma} = \frac{1}{\zeta} + \left(\frac{2 \, \lambda}{\zeta \, \sigma}\right) \frac{1}{\delta} \tag{3.2}$$

Later, the streaming voltage of ten cells with the same dimensions of those used by Gu and Li will be measured. The left hand side of this equation will be plotted against the inverse of channel height to see if their results can be replicated. If successful, this will give a way of determining the zeta potential and surface conductance of the fabricated cells.

Streaming current

Gu and Li, also derive a similar equation for streaming current [47]. This equation, shown below, has been slightly rearranged to match the form of eq. (3.1)

$$\frac{I_s}{\Delta P} = \frac{\varepsilon_r \, \varepsilon_0 \, \zeta \, W \, \delta}{\mu \, L} \tag{3.3}$$

where:

W is the width of the channel,

L is the length of the channel.

This equation is similar to that given by Olthuis et al. for a porous plug, but has been derived specifically for rectangular channels [9].

Optimisation

Having a mathematical model of a streaming cell allows for optimisation of its operating parameters. The model shows that any load placed across a streaming cell is actually placed in parallel with that cell's internal resistance. Therefore, choosing a suitable load is an important design consideration. It is possible to optimise the cell's output for maximum power output, or maximum efficiency. So which is best suited to harvesting applications? The only time energy can be extracted from a streaming cell is when pressure developed across it because of liquid flowing through it. When power is available to harvest, it is beneficial to collect as much as possible – no matter how much is wasted. Any energy that we could not capture will be lost. In situations requiring maximum efficiency, the efficiency of the system approaches 100 % as the power delivered approaches 0 %. In situations requiring maximum power, the maximum achievable efficiency is 50 %. This means we can only harness half of the electrical power developed by a cell, at best.

Electrical model

Figure 3.1 depicts schematically how a streaming cell operates when connected to an external load. A model of this sort is commonly used to analyse the behaviour transistors. This particular model is based on the work of Olthuis et al. in [9], but has been modified slightly. Instead of showing the zeta potential (ζ) as the equivalent voltage source, it is shown here instead with the pressure applied (ΔP). This change was made because there is no way of controlling the zeta potential - it determined by the properties of the

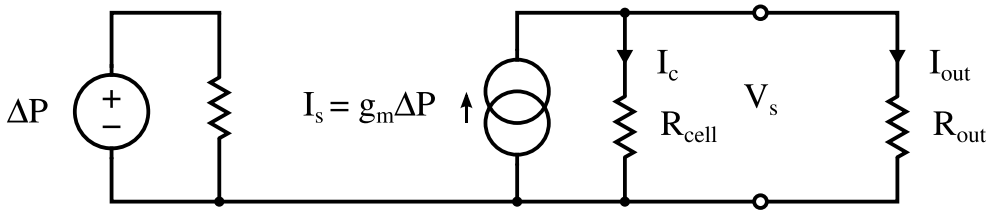


Figure 3.1: Schematic diagram of a streaming cell with connected load resistance.

particular interface. However, the amount of pressure developed across the cell is controllable, and from eq. (3.3) it is shown to be directly proportional to streaming current. In fact, the transconductance (g_m) for this model *is* eq. (3.3). The model aids analysis in that it shows the electrical configuration of an external load resistance (R_{out}). As shown, any load resistance placed across the cell is being placed in parallel with the internal electrical resistance of the cell. This will help to determine how best to optimise the cell in order to maximise its electrical output.

Optimising R_{out} for maximum power

Figure 3.1 shows that an electrical load placed across a streaming cell is actually placed in parallel with the cell's own internal resistance. The maximum power theorem states that only 50% of power is transferred when maximum power transfer is achieved. The condition of maximum power transfer occurs when the load resistance and the source resistance are equal. In the case of streaming cells, this means that the output of the cell is maximised when the load resistance is equal to the electrical resistance of the cell itself.

Optimising streaming cell parameters

Ohm's law and the power equation give:

$$\begin{aligned} P &= V \times I \\ V &= I \times R \\ \therefore P &= I^2 \times R \end{aligned} \tag{3.4}$$

For maximum power transfer the source and load resistances must be equal and therefore the amount of current in the load is equal to the amount of current lost within the cell. This means that half of the streaming current (I_s) runs through the load. Now take eq. (3.3), rearrange it for streaming current

(I_s), half it, and substituting it into eq. (3.4):

$$\begin{aligned}
\frac{I_s}{\Delta P} &= \frac{\varepsilon_r\,\varepsilon_0\,\zeta\,W\,\delta}{\mu\,L} \\
I_s &= \frac{\varepsilon_r\,\varepsilon_0\,\zeta\,W\,\delta\,\Delta P}{\mu\,L} \\
I_{out} &= \frac{\varepsilon_r\,\varepsilon_0\,\zeta\,W\,\delta\,\Delta P}{2\,\mu\,L} \\
P_{max} &= \left(\frac{\varepsilon_r\,\varepsilon_0\,\zeta\,W\,\delta\,\Delta P}{2\,\mu\,L}\right)^2 \times R \\
P_{max} &= \left(\frac{\varepsilon\,\zeta\,W\,\delta\,\Delta P}{\mu\,L}\right)^2 \times \frac{R}{4}
\end{aligned} \tag{3.5}$$

where P_{max} refers to the maximum output power, $\varepsilon = \varepsilon_0\,\varepsilon_r$ and $R = R_{cell} = R_{out}$.

To get a better feel for this equation, it helps to substitute in the parameters that affect R. From here we will refer to R as the internal electrical resistance of the cell. It also refers to the external resistance in the maximum power condition, but we are free to vary that to match the internal resistance.

Let us begin with the understanding that:

$$R \propto \frac{L}{A\,\sigma} \tag{3.6}$$

$$R_h \propto \frac{L\,\mu}{A} \tag{3.7}$$

where σ is the conductivity of the liquid and A is the cross-sectional area of the cell. The first eq. (3.6) states that the electrical resistance will increase with cell length and decrease with the cross-sectional area of the cell and the conductivity of the fluid. The second eq. (3.7) states that the fluid mechanical resistance will increase with the length of the cell and the viscosity of the fluid, and decrease with the cross-sectional area of the cell.

We can identify the presence of R_h in eq. (3.5), knowing that the cross-sectional area of the cell (A) is its width (W) times its height (δ). Starting with this equation we substitute eq. (3.6) in and rearrange to produce an

approximate relationship between pressure, cell length and area:

$$
\begin{aligned}
P_{max} &= \left(\frac{\varepsilon\,\zeta\,W\,\delta\,\Delta P}{\mu\,L}\right)^2 \times \frac{R}{4} \\
P_{max} &\propto \left(\frac{W\,\delta}{\mu\,L}\right)^2 \times (\varepsilon\,\zeta\,\Delta P)^2 \times \frac{L}{A\,\sigma} \times \frac{1}{4} \\
P_{max} &\propto \left(\frac{A}{\mu\,L}\right)^2 \times (\varepsilon\,\zeta\,\Delta P)^2 \times \frac{L}{A\,\sigma} \times \frac{1}{4} \\
P_{max} &\propto \frac{A^2}{L^2} \times \left(\frac{\varepsilon\,\zeta\,\Delta P}{\mu}\right)^2 \times \frac{L}{A} \times \frac{1}{4\,\sigma} \\
P_{max} &\propto \frac{A}{L} \times \left(\frac{\varepsilon\,\zeta\,\Delta P}{\mu}\right)^2 \times \frac{1}{4\,\sigma} \\
P_{max} &\propto \left(\frac{\varepsilon\,\zeta\,\Delta P}{\mu}\right)^2 \times \frac{A}{4\,L\,\sigma} \\
P_{max} &\propto \frac{\Delta P^2\,A}{L} \times \frac{\varepsilon^2\,\zeta^2}{4\,\mu^2\,\sigma}
\end{aligned}
\tag{3.8}
$$

This eq. (3.8) suggests that a cell with a small length, large area and high pressure is the best candidate for maximising power output. When using tap water we have no control over the remaining parameters, highlighting the importance of cell geometry and applied pressure.

Streaming Cell Fabrication

Building a streaming cell seemed a simple task at the outset. This section gives a brief overview of the work related to creating that first working streaming cell.

First streaming cells

The work that first sparked the interest of both my primary supervisor and I in streaming devices was that of Varga and Seymour [48]. In that paper it was reported that a device employing cavitation as a means of increasing the resistance between two bodies of water was capable of developing over 50 V across its ends. A diagram of the cavitation device is shown as fig. 3.2. Summer research student Jonathon McMullan recreated that device, shown as fig. 3.3, with the intention of reproducing results from that paper. The device was turned on a lathe from acrylic in two pieces and glued together. A brass rod (not shown) is inserted up the shaft (presented vertically in the image) into the flow of water. The rod had a narrow cylindrical tip, small enough to allow it to fit into the main flow pipe (presented horizontally in

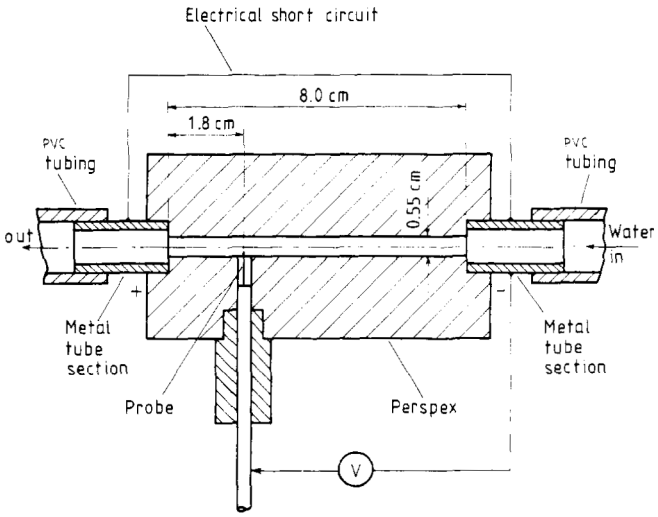


Figure 3.2: Diagram of a cavitation device, taken from [48], reported to be able to generate over 50 V across its ends by pumping water through it.

Figure 3.3: Photo of the first streaming device built. It has been constructed from acrylic and brass by summer research student Jonathon McMullan. It was built to recreate the streaming behaviour reported by Varga and Seymour, but failed to produce a measurable output.

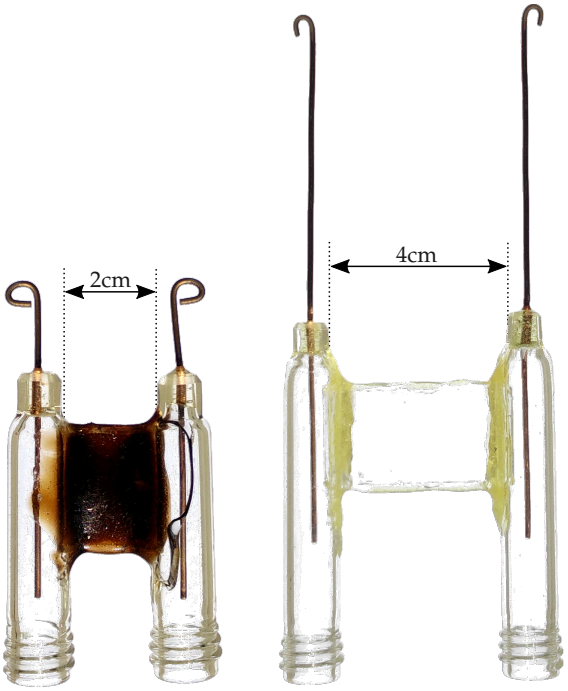


Figure 3.4: Photo showing two examples of a second design of streaming cell made entirely from glass.

the image), with a flat face ground into one side. Water was forced through the device (left to right in the image) and the rod caused cavitation of the water as it flowed past the inserted rod tip. Varga and Seymour measured a potential difference of over 50 V between the inserted rod and the brass end fittings using a flow rate of $0.0425\,\mathrm{l\,min^{-1}}$. That experiment – which tried the same flow rate, materials, liquids, and measurement setup – did not produce measurable streaming voltage. The following year, summer research student Wane Crump and I conducted experiments to determine the amount of charge that could be transported on water droplets. Research into energy harvesting with an electrostatic generator is presented in appendix A.

Later, other designs of streaming cells, namely those of Gu and Li [47], were found in the literature. Their design of streaming cell looked simple and easy to fabricate, so we attempted to replicate them. Employing Waikato University's glass-blower, Steve Newcombe, two streaming cells were fabricated entirely from soda-lime glass. Those streaming cells are shown in fig. 3.4. Each of the two channels were made by placing a 50 µm sheet of copper between the glass slides and then welding the slides together to seal the cell. The copper was etched away with acid once the cell was sealed. The cells were then glass welded to the side tubes that held the copper electrodes. By varying the length of the two channels (one of 2 cm, the other 4 cm) we

Item	Brand	Product details
Microscope slides	Sail Brand	JIA 7101WT - 26 x 76mm
Shims	Garlock	Colorplast - 50 μm,80 μm, 120 μm and 250 μm
Epoxy	Selleys	Araldite - Ultra Clear Resin
Pressure sensor	Honeywell	24PC15SMT - 0 – $\pm$15 PSI

Table 3.1: Materials used to construct the streaming potential cells

hoped to determine what role length played on channel output characteristics.

Both of the cells burst under the water pressure applied from lab taps. A crack along the right hand reservoir is visible on the 2 cm wide channel. Attempts were made to strengthen the channels by coating the joins with industrial glues, but were not successful.

Robust streaming cells

The two previous attempts to create streaming cells had failed. In the moments before failure, the all-glass streaming cells developed measurable voltage across their terminals. The developed voltage was enough to warrant further pursuit, but the design needed revising in order to increase their pressure handling capability. A new design was found that used epoxy resin and acrylic to contain the channels. This design appeared to be much more resilient to cracking. Aspects of that design were taken from a paper by Gu and Li [47].

Construction

Construction began by sectioning standard microscope slides into halves. This produced glass panels of approximately 26 mm $\times$ 38 mm $\times$ 1 mm. A single panel was then epoxied to an acrylic base plate, as is shown in Figure 3.5. Once set, plastic shims were cut to the required size, covered with a very thin layer of epoxy, and placed along the edges of the slide. The shims lined the sides of the glass panel such that they left a 1 cm gap through the centre. A second glass slide was then placed on top of the shims and epoxy resin used to seal the sides. Pressure was applied to the stack while the epoxy set to ensure the epoxy was distributed correctly and to control the channel height. A photo of the shims glued between the two slide halves is shown in Figure 3.6. Once set, each channel was examined under a microscope to determine the internal channel height. Each of the four corners were measured to ensure the internal dimensions remained rectangular once set. To finish, acrylic reservoirs where mounted over each end of the channel. These reservoirs facilitate the connection of fluid tubes and voltage probes to each end of the channel. The final assembly is shown in Figure 3.7. A full list of materials

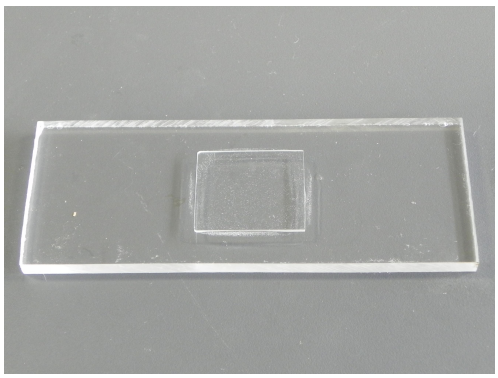


Figure 3.5: Photo showing half of a glass slide glued to acrylic base plate. Part of the process for constructing a streaming cell.

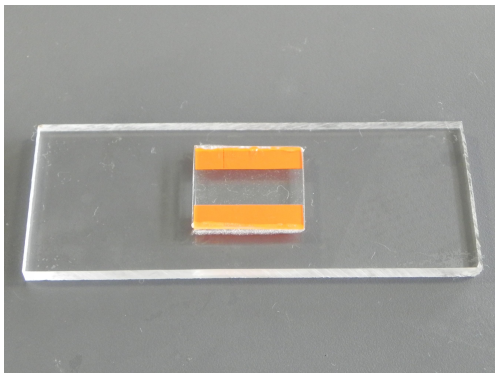


Figure 3.6: Photo showing plastic shims sandwiched between two slide halves. Part of the process for constructing a streaming cell.

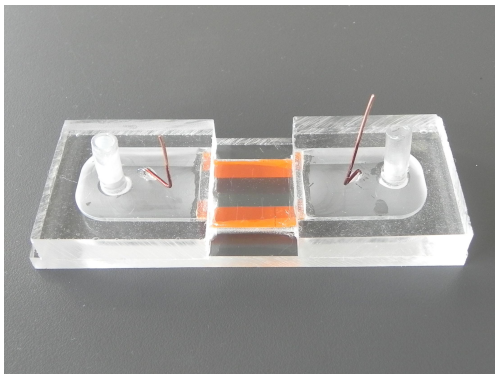


Figure 3.7: Photo showing final streaming cell assembly.

used to construct the channels is presented as Table 3.1. A total of ten channels were made and tested using this method.

Measurements

Of the papers describing experiments using streaming potential cells ([21,47, 49,50]), I chose that of Gu and Li to replicate [47]. Their method employs a simple cell design, discussed in the previous section, and a detailed description of the experimental procedure. This section details measurement of ten streaming cells based on those of Gu and Li.

Experimental setup

Measurement of each harvester was made in a laboratory using high-sensitivity measurement instruments and a lab water tap. Each cell's output power was measured with a precision source measurement unit (SMU) and applied pressure was monitored with a differential pressure sensor.

The Agilent E5270B is a mainframe system that holds banks of SMUs with connections to a GPIB computer interface. The Department of Engineering's E5270 contains three SMUs, each having the ability to measure currents as low as one femto-Ampere. The device uses separate 'force' and 'sense' connections to ensure the voltage/current being set is accurately controlled where they meet. Additionally, it uses tri-axial cables to minimise any interference from outside sources, which is important when measuring such low currents.

The input resistance to the E5270's measurement units are specified as 13 GΩ. It is essential to use such a high impedance measurement due to the high internal resistance of the cell. I later show that the internal electrical resistance of the cell is in the order of 5 MΩ, so the E5270's input impedance is roughly two thousand times larger. The internal resistance of a typical multimeter is 10 MΩ, too close to that of the cell and would therefore affect the measured output.

The pressure sensor used was a Honeywell 26PC SMT Series differential pressure sensor. It came as surface mount package, making it a cost effective solution, but delicate to set up. On its exterior are two ports to which rubber tubes are attached. Between those ports, internal to the sensor, sits a diaphragm. That diaphragm controls the resistance between two nodes in the sensor's bridge circuit (shown in figure 3.8). By applying 10 V DC between the 'Vcc' and 'GND' pins, the output voltage between outputs 'A' and 'B' correspond to the applied pressure. Measurement of that output voltage was done using the precision measurement mainframe.

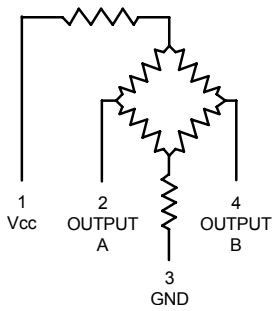


Figure 3.8: Schematic diagram of the differential pressure sensor bridge circuit (taken from [51])

CELL - A	Voltage/current at high-pressure side of cell
PRESS. - A	Output A of pressure sensor
PRESS. - B	Output B of pressure sensor
CELL - B	Voltage/current at low-pressure side of cell

Table 3.2: Definitions for labels used in fig. 3.9

The mainframe was controlled from a PC running Python scripts utilising the open source Python-vxi11 library [52]. This allows sweeping the amount of current drawn from the harvester while recording the corresponding voltage drop. This is the equivalent of varying the load resistance, which allows us to find the point of maximum power transfer.

Figure 3.9 shows the measurement setup as a diagram. It shows connection of the measurement mainframe, bench-top power supply, streaming cell, pressure sensor, and the lab tap. Table 3.2 provides details of the abbreviated electrical connection labels used in the figure 3.9.

Measurement issues

There were two issues with the measurement setup that may have impacted measurements. Firstly, the electrodes used were copper and are susceptible to polarisation by electrolysis. Secondly, the differential pressure sensor was only rated to 15 PSI (approximately 100 kPa), less than half the maximum pressure developed across the cell.

Electrolysis on at the electrode surface would cause the electrodes to polarise, resulting in a semi-permanent offset voltage appearing between electrodes. That offset voltage would be opposite in polarity to what is developed while cell was in operation. By reversing the flow of water through the cell, the polarisation should be reversed. The use of more suitable electrode materials would reduce this effect, for instance platinum black electrodes. Copper was used for the electrodes as it was cheap, easily obtainable, and

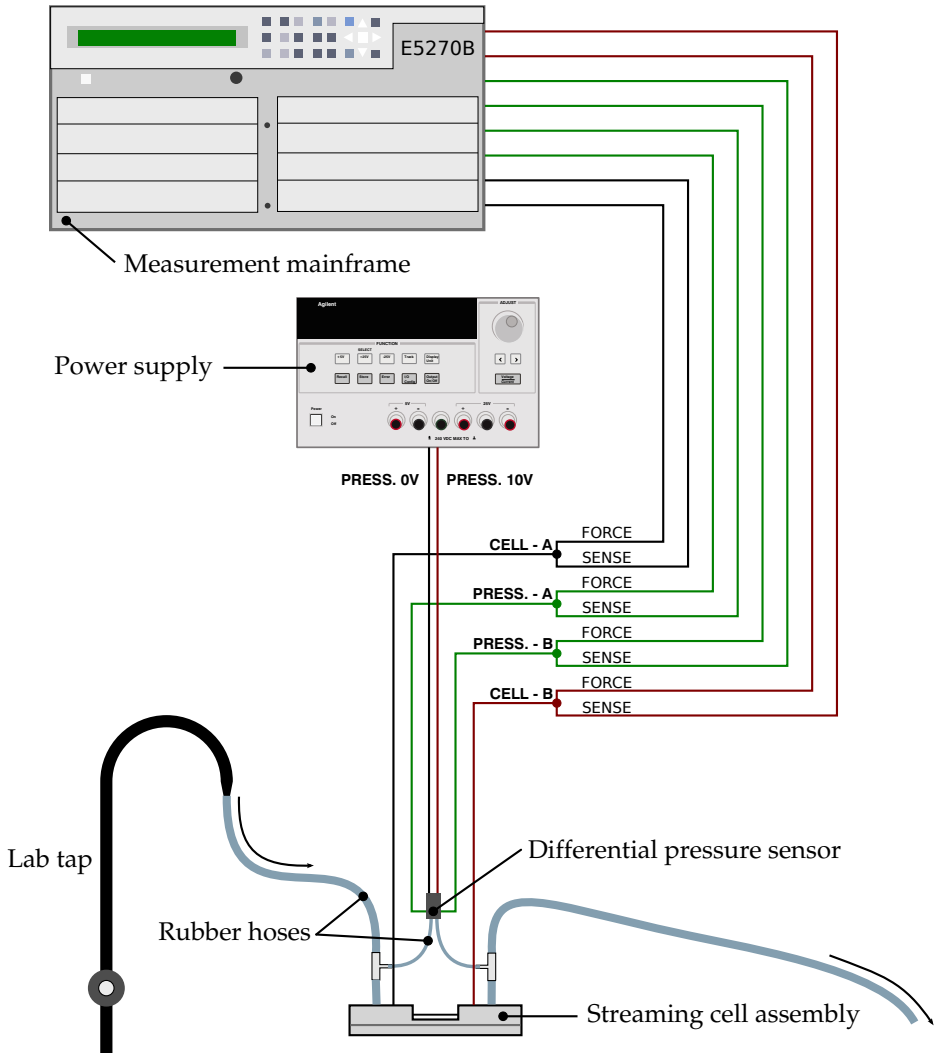


Figure 3.9: Diagram showing the measurement setup used to measure power output from streaming cell energy harvesters.

easy to work with. Offset voltages can be removed after measurement by subtraction. This was not a perfect solution, but the offset voltage could be measured and accounted for so was left as-is.

From measurement of the output *voltage* of the cells, the presented graphs and figures have been adjusted to remove the effects of electrolysis. This was done by adding an offset to the measured data to shift the y-intercept up to 0 V. This represents the situation where platinum black electrodes been used. As no absolute data is taken from these measurements, the y-intercept adjustment does not affect any subsequent predictions made about the cells. Only the gradient of the output relative to the pressure applied is used, and even then, only to select a suitable candidate cell for power measurement. *Most importantly*, no offsets have been applied to measurements of the cell power output.

Although the maximum rated pressure of sensor was 15 PSI (approximately 100 kPa), the sensor's output remained linear up to our maximum pressure of 40 PSI (275 kPa). I expect that exceeding the sensors specified pressure will result in a lower 'mean time to failure', but its output remained true. As a precaution, a tyre pressure gauge was used to roughly confirm the output of the sensor at the end of the cell measurements. This was a crude test, however its output matched that of the differential pressure sensor so was taken as an indication of accuracy.

Results

Results from streaming cell measurement are broken into two sub-sections. The first presents the output voltage of the ten cells in response to applied water pressure. From these measurements, the cell with the highest voltage/pressure ratio is found. The second sub-section shows the maximum power that can be harvested from that cell. These are the most important measurements as they reveal the energy conversion efficiency of the cells.

Streaming voltage versus pressure

Figure 3.10 shows adjusted results of streaming voltage measurements from each of the ten cells. A full set of figures, one cell per graph, can be found in appendix B as figs. B.1 to B.10. It represents the first successful measurements I had made of streaming cells. During these measurements three cells burst under pressure, two were dropped and subsequently shattered, and one suffered epoxy failure. No measurements of flow rate or output current were made during these early experiments. As a result they reveal very

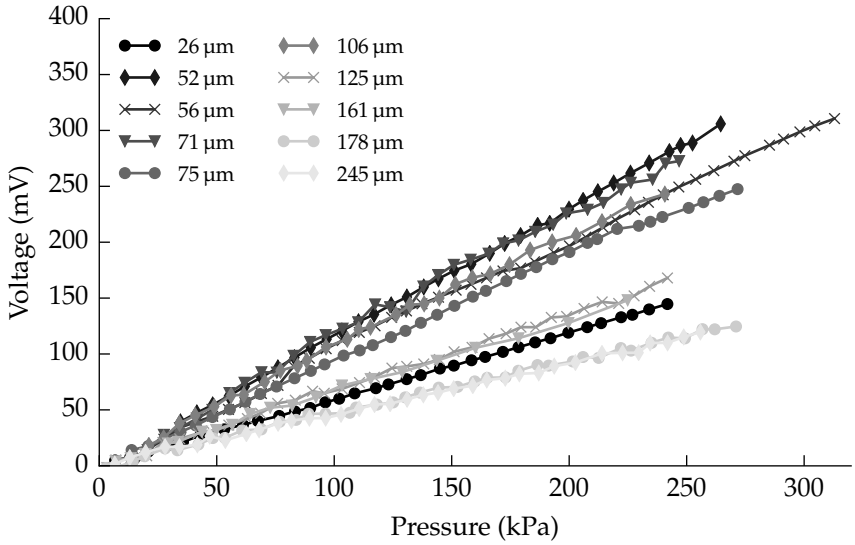


Figure 3.10: Plot showing adjusted streaming voltages versus applied pressure.

little about the efficiency of the cells themselves. We cannot determine either the mechanical energy put into the cells, nor the output power available. However, we can relate these measurements to those made by Gu and Li as will be shown and discussed shortly.

The cell having an internal height of 56 µm required an offset adjustment of 405 mV to remove its vertical offset. This indicates that the electrodes were highly polarised by the time measurements were made. This effect is especially evident in fig. 3.12, which shows raw measured data without vertical offset adjustment. Some of the traces exhibit a certain amount of 'jitter' in their pressure-to-voltage gradients. This is likely due to the time difference between adjacent measurement points. Measurement points were not taken monotonically, instead being extracted from a number of pressure cycles.

Figure 3.11 shows the streaming voltage to pressure gradients versus channel height. This data has been taken from the previous graph (fig. 3.10)) to show the response as a function of channel height.

Output power versus load resistance

Figure 3.13 shows the characteristic power curve of the 71 µm high streaming cell channel. It was chosen as it was the highest performing cell, in terms of voltage/pressure gradient, that had not yet ruptured. Pressure fluctuations near the end of the experiment are due to usage of the department's water system (e.g., a tap being turned on). Their effect is visible in both the streaming voltage and output power traces, highlighting the strong coupling to applied

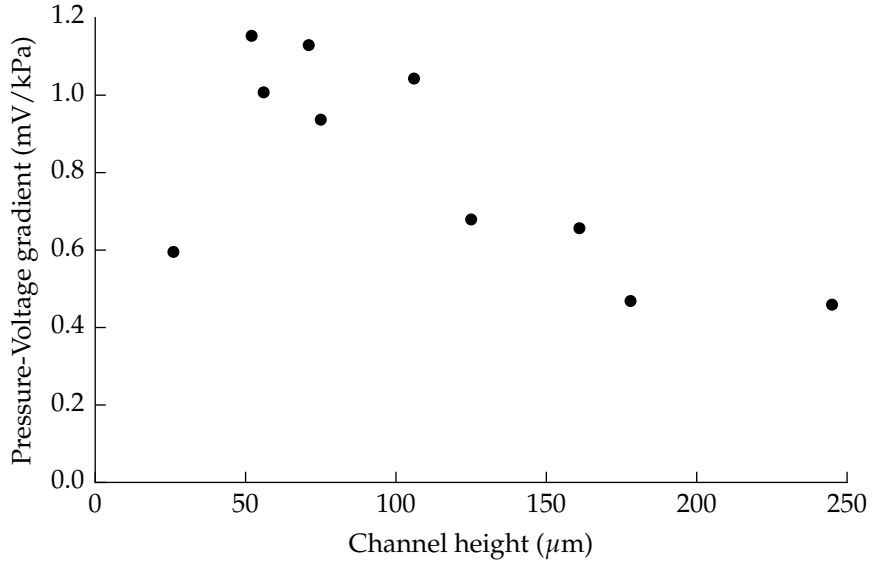


Figure 3.11: Scatter plot of voltage/pressure gradient versus channel height for each of the measured channels.

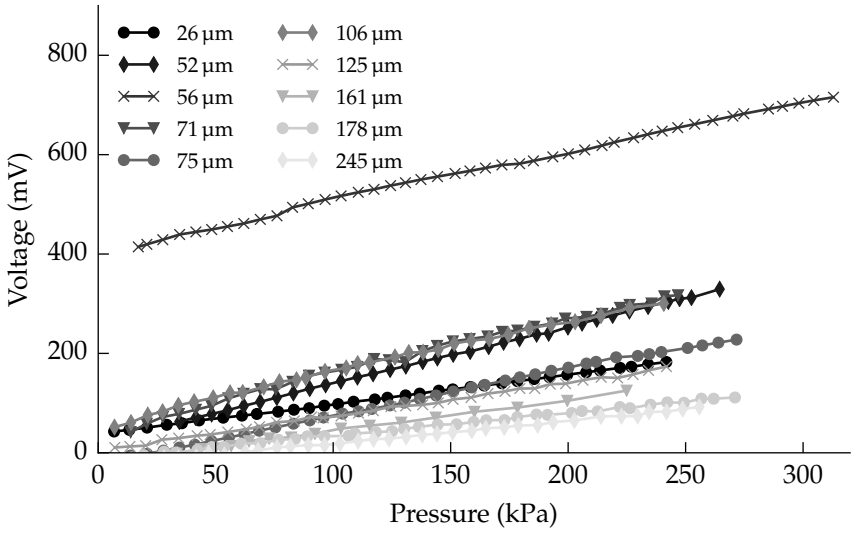


Figure 3.12: Plot showing unadjusted streaming voltages versus applied pressure.

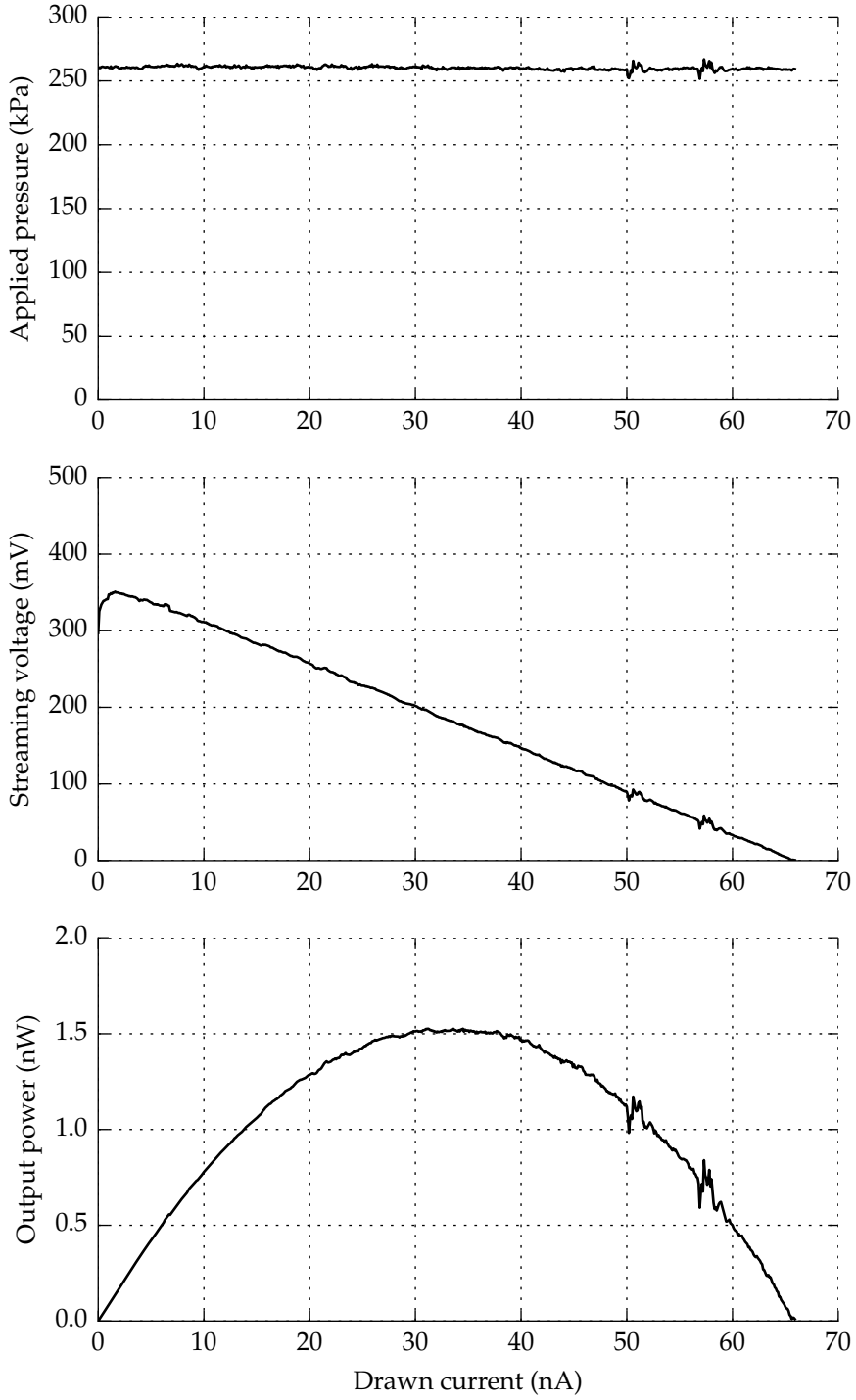


Figure 3.13: Graph showing output power versus effective load resistance for a 71 µm high channel at a pressure of 260 kPa.

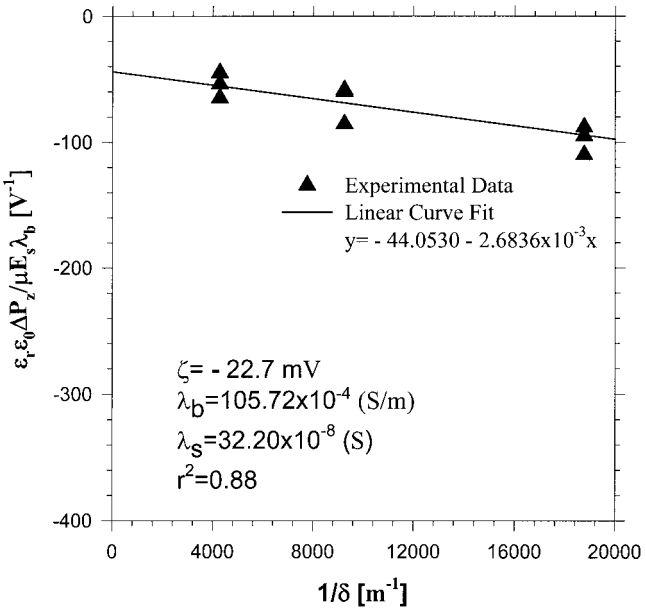


Figure 3.14: Plot showing measured data from Gu and Li's paper on streaming cells relating the streaming voltage and pressure differential to the channel height with distilled water as the working fluid [47].

pressure.

The maximum power delivered by the cell was 1.52 nW, corresponding to a current draw of 33.5 nA with a streaming voltage of 182 mV. Generating this power required 260 kPa of pressure, resulting in a flow rate of 2.05 ml s^{-1}. This equates to 539 mW of pumping power lost to the device and therefore an energy conversion efficiency of 2.8×10^{-9}

Discussion

Initial measurements of streaming voltage revealed that the output voltage is directly proportional to applied pressure. Containing pressures reaching 260 kPa within glass structures is difficult.

Comparing the streaming voltage measurements taken from each of the ten cells to the measurements made by Gu and Li yielded surprising results. In their paper [47], they determined the zeta potential (ζ) and surface conductivity (λ) by plotting measurements and fitting a linear equation to their data. The use the y-intercept of the resulting line to give the inverse zeta potential and the slope of the line gave information about the surface conductivity.

Their results for three streaming cells are shown here (taken from [47]) as figs. 3.14 and 3.15. The first (fig. 3.14) shows measurements when distilled water is used as the working fluid, the second (fig. 3.15) shows the measure-

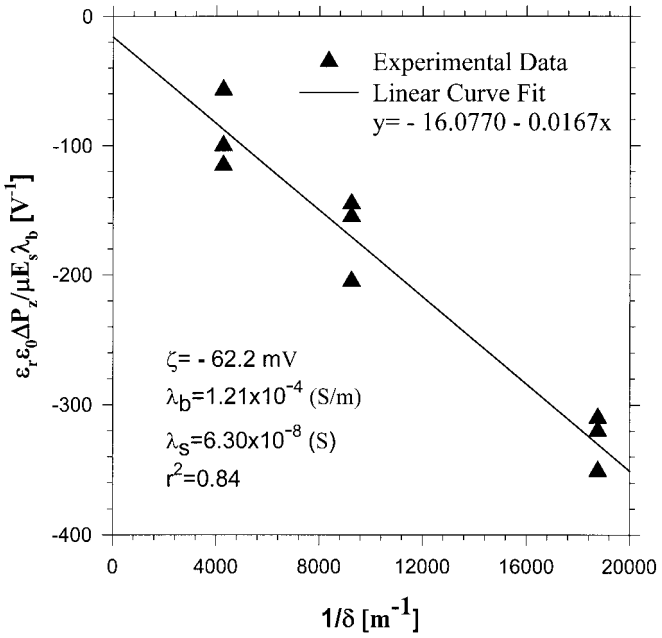


Figure 3.15: Plot showing measured data from Gu and Li's paper on streaming cells relating the streaming voltage and pressure differential to the channel height with a 1 mmol sodium chloride solution as the working fluid [47].

ments for a weak saline solution. It is interesting to note that they have what looks to be fairly linear data, although it is hard to tell with only three channel sizes.

By comparison, fig. 3.16 plots the same variables from measurements taken from the ten streaming cells fabricated here. In this graph, the following substitutions have been made: λ_b has been replaced with σ (both refer to the bulk conductivity of the solution), and E_s has been replaced with V_s, (both refer to the streaming potential). The response to variation of channel height is clearly non-linear. Their method of finding the zeta potential rests on the following rearrangement:

$$\frac{\varepsilon_r \varepsilon_0 \Delta P}{\mu V_s \sigma} = \frac{1}{\zeta} + \left(\frac{2\lambda}{\zeta\sigma}\right)\frac{1}{\delta} \tag{3.9}$$

where λ is the surface conductivity and δ is the channel height. So as the channel height (δ) tends to infinity, the left hand side tends toward the zeta potential (ζ). This notion seems counter intuitive since the zeta potential is defined at the plane of shear (as shown in fig. 2.7), relative to the solution bulk. The equation is stating that no-matter how far you separate the walls, the minimum voltage-pressure gradient you can get is still set by the zeta potential.

Measurement of the output power generated by the 71 µm streaming cell

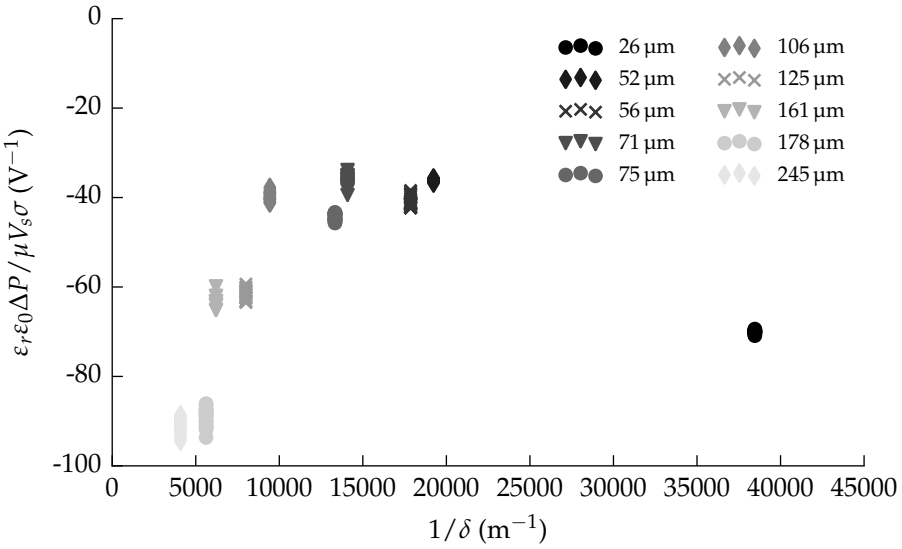


Figure 3.16: Scatter plot with results of streaming cell measurements in terms of those made by Gu and Li [47] (for comparison).

is promising. From this measurement the power transfer curve is evident. Referring back to the model presented as fig. 3.1, we can now calculate the channels internal electrical resistance (R_{cell}). We know from the graph that the maximum power transfer occurred at a current of 33.5 nA with a streaming voltage of 182 mV. Via Ohm's law this equates to a load resistance (R_{out}) of 5.43 MΩ, which from the maximum power theorem we know must be equal to the cell's internal resistance.

Concluding Remarks

Conversion of mechanical pumping into electrical energy can be done with narrowly separated plates of glass. The conversion efficiency seen here was low, much lower than reported in the literature. A channel 1 cm by 3 cm by 71 µm produced 1.5 nW under a pressure differential of 260 kPa. That took 359 mW of pumping power to produce, yielding an efficiency in the order of 1×10^{-9} Precision engineering, with regards to cell construction, will likely lead to greater efficiency. This is based on reports from the literature, where higher efficiency channels utilised much narrower channels.

Measurements of ten streaming cells were compared to the results published by Gu and Li. The linear relationship of Gu and Li between channel height and their plotted parameter could not be reproduced. Instead, results showed a highly non-linear relationship as shown in fig. 3.16. The gradient of measurement points within the range of channel heights measured by Gu

and Li is reversed. The reason for the discrepancy is not clear.

The streaming voltage was found to scale linearly with the applied pressure. This could potentially be useful as a means of sensing water flow rates. A dual purpose such as power sourcing and flow measurement lends itself well to water metering applications.

Chapter 4
Applicability to Water Metering

Water metering is becoming increasingly common throughout the world [53]. Sourcing and processing drinkable water is an expensive task. Cheap and reliable methods for reading water meters is important. As supplies of drinkable water become constrained, volumetric pricing will become increasingly common. Harvesting energy at the location of metering would eliminate the need for batteries. If energy harvesting could be done without moving parts, lower component wear should lead to increased service life versus mechanical meters. This chapter investigates the feasibility of using streaming cell technology as a means of powering electronic water meters.

Chapter 3 discussed electric power generation from streaming cells. A 2×10^{-9} conversion efficiency from water flow and pressure was demonstrated. Also, streaming voltage was found to be directly proportional to the pressure across a cell. Can that pressure dependence be used as a method to meter water consumption while generating power? Probably. However, questions like this are only relevant if the harvester is feasible. Its feasibility is dictated by whether or not it can provide enough energy. To find that out, the following questions must be answered:

1. What quantity of energy is there available to harvest?

2. What fraction can be harnessed?

3. How much power do we need?

The second question was answered in chapter 3 (0.0002 %), and the third will be answered in chapter 5. This chapter estimates the quantity of energy available for harvesting in a typical domestic setting.

Trends in Water Metering

In New Zealand - Auckland City, Tauranga City, Nelson City, Whangarei District, and the Tasman District have already implemented water metering [54]. For residents of Wellington, New Zealand's capital city, water metering is optional. In metered locations, meter readers must manually read the display of each meter, which is a long and laborious task.

Automatic meter reading systems (AMR) are an alternative method of collecting that data. Hamilton City Council is trialling such systems in remote areas in the hopes of adopting them for wide-spread use. There are two types of automatic meter reading systems: an external reader (with communication interface) that attaches to a compatible meter, or an all-in-one unit that meters and reports usage. These systems offer advantages separate from taking away the laborious job of meter reading. Increased billing frequency helps customers reduce their consumption by giving more frequent feedback. They remove the need to access the customer's property [53]. Electronic analysis of the meters readings provides an easy way to detect water leaks.

It is estimated that 10 % of post-meter water consumption is due to leakage in the residential sector [55]. Measuring night-time water flow is a convenient way of estimating flow due to leakage. Britton et al. show that communicating with customers whose homes showed signs of leakage that a night-time flow reduction of 89 % is achievable. In contrast, a control group's night-time usage increased by over 50 % during the same time-frame. The benefits of automatic water metering are not limited to billing, but improving the network as a whole.

Domestic water meters are typically installed at a property's boundary in a plastic box set into the ground. An installation typical for an Auckland residential area is shown in fig. 4.1. The meter is installed over five meters into the property from the road-side. It is not feasible to connect it to a source of power because of its location. No commercially available domestic energy harvesting water meters, suitable for burial, exist on the market as of January 2015.

A common configuration for wireless automatic meter reading is to have a reader/transmitter device that is separate to the meter itself. Such a device usually attaches to the meter's display or has a wire connecting it to the meter. Being detachable and tamper-proof means it must be powered by batteries. A commonly stated battery life for such units is ten years [56], close to a battery's shelf-life. We investigate the possibility of replacing these batteries with a streaming cell based energy harvester. A harvester removes the need

Figure 4.1: Photo showing a domestic water meter, and installation, (Kent PSMT 25mm) typical of an Auckland residential area.

Figure 4.2: Photo of a wireless transmitting module from Watercare NZ. The device attaches to a compatible water meter and contains its own battery [56].

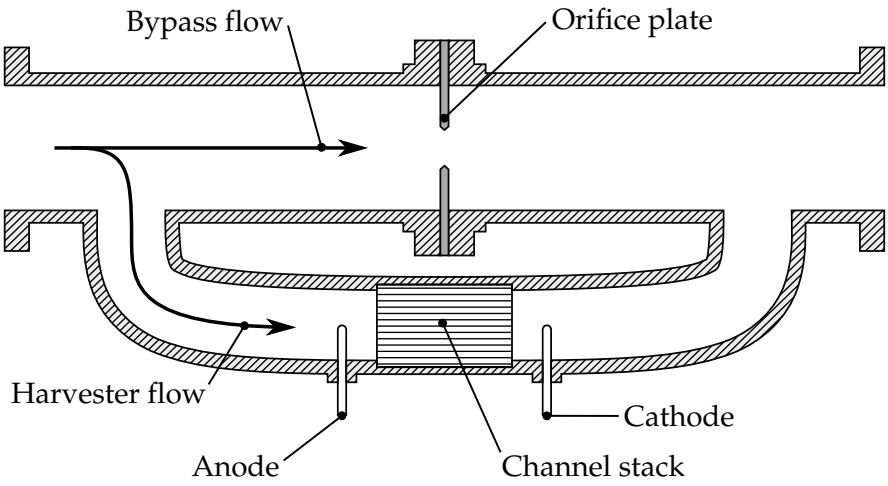


Figure 4.3: Diagram showing the intended design of streaming cell harvester suitable for domestic connection.

for batteries, but needs to be plumbed into the water feed. For this reason the resulting device would most likely replace the meter, as opposed to being an attachment.

Mechanical Design of a Harvester

In chapter 3, energy was converted between fluid-mechanical to electrical using a single channel. Harvesting for electronic water meters will require more energy than that channel could produce. There are multiple ways of increasing the output power of the channel design used in chapter 3: the channel can be widened (doubling the width will double the output power), or multiple channels can be stacked together (this multiplies the output by the number of channels formed). Scaling the harvester is not considered a problem, but the pressure drop it develops is. High fluid resistance is inevitable since practical efficiencies are only obtained when the internal dimensions are small.

To control the pressure drop across the harvester, the mechanical design shown in fig. 4.3 is proposed. It gives the capability of controlling the hydrodynamic resistance of the unit as a whole by means an orifice plate. The plate sits in the "main line" causing a pressure differential in proportion to the flow rate. An orifice plate with a hole equal in diameter to the main pipe causes no pressure differential as it causes no flow obstruction. Conversely, a plate without a hole forces all liquid through the harvester, causing the maximum pressure differential. Using an appropriate sized orifice plate, the customer will be unaware of the harvester's presence and a suitable pressure differential

Item	Measurement	Summer	Winter	Unit
Shower	Duration	6.6	7.0	minutes
	Volume	50.0	52.5	litres
	Flow	8.1	8.0	litres/minute
	Frequency	0.9	0.9	/person/day
Washing	Volume	122	123	litres
	Frequency	0.35	0.36	/person/day
Toilet	Volume	6.6	6.8	litres
	Frequency	4.9	4.5	/person/day

Table 4.1: Average usage characteristics for a shower, washing machine and toilet as obtained from Heinrich's water usage report [57].

will be developed. For the sake of analysis we assume the orifice plate will be sized to match the pressure loss of a mechanical meter. This assumption means the amount of harvestable energy is equal to the amount dissipated in a mechanical water meter. The following section quantifies the amount of energy a water meter dissipates over an average week in a typical Auckland home.

Quantifying Harvestable Energy

Using a bypass pipe with an orifice plate, the pressure drop across a streaming cell energy harvester can be controlled. The following calculations are based on the assumption that the streaming cell energy harvester will be set to collect the same amount of energy already lost inside a typical mechanical meter.

Heinrich monitored water consumption of 51 homes throughout Auckland in 2008 [57]. His report shows the majority of domestic water is consumed by the shower (30 %), washing machine (27 %) and toilet (20 %). Together these account for over 75 % of domestic water consumption. Data from table 4.1 was used to build a typical water usage profile. Heinrich published a similar report in 2007 that contained water flow profiles, the flow profile of the toilet has been taken from that report [58].

Figure 4.4 shows the flow rates for each of the three items considered (toilet, shower and washing machine). Volumes for each of the events, and flow profile of the toilet, match the measurements reported by Heinrich. Specifically, the total volumes for each are: 122l, 49.51 and 6.22l for the washing machine, shower and toilet respectively.

The Kent 25-PSMT series mechanical water meter is the most commonly installed water meter in the Auckland district [60]. Figure 4.5 shows the head-loss, or pressure differential, versus flow rate for the Kent PSM range of

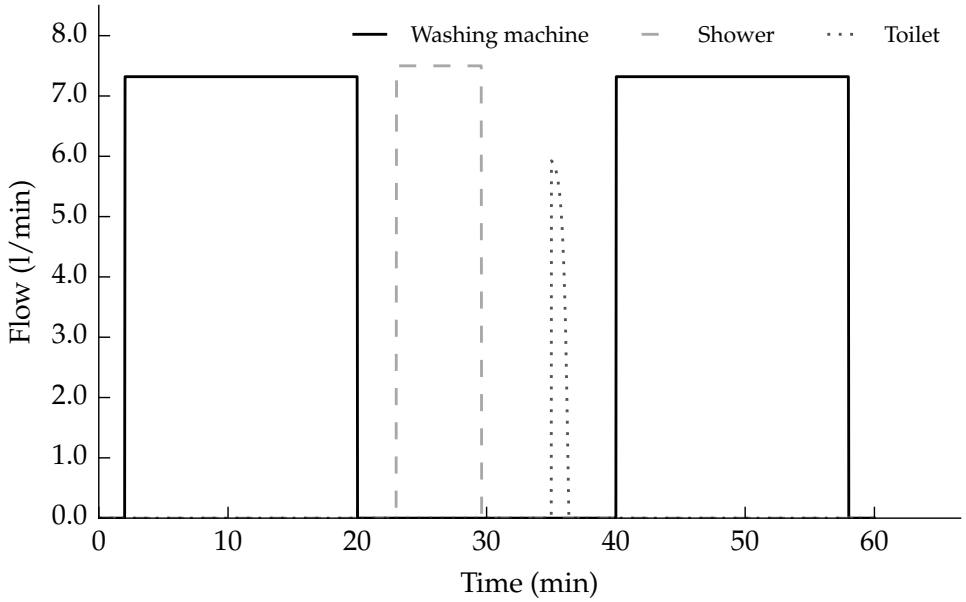


Figure 4.4: Graph showing a water consumption profile of constructed instances of washing machine use, a shower and a toilet flush. The washing machine's wash and rinse cycles are separated in time.

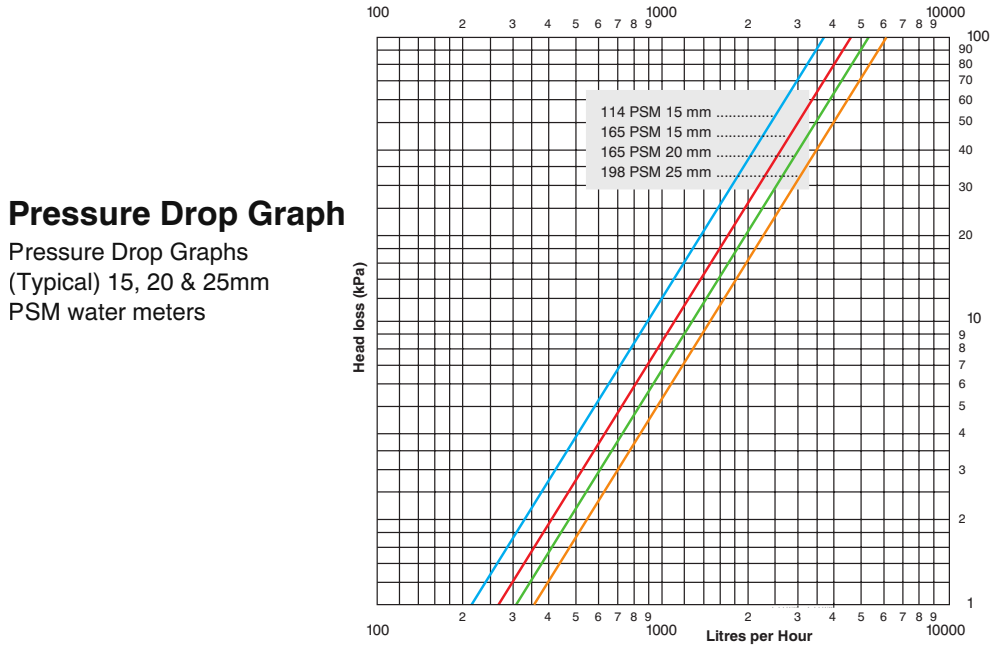


Figure 4.5: Log-log graph showing the pressure developed across the Kent PSM series mechanical water meters. Taken from [59].

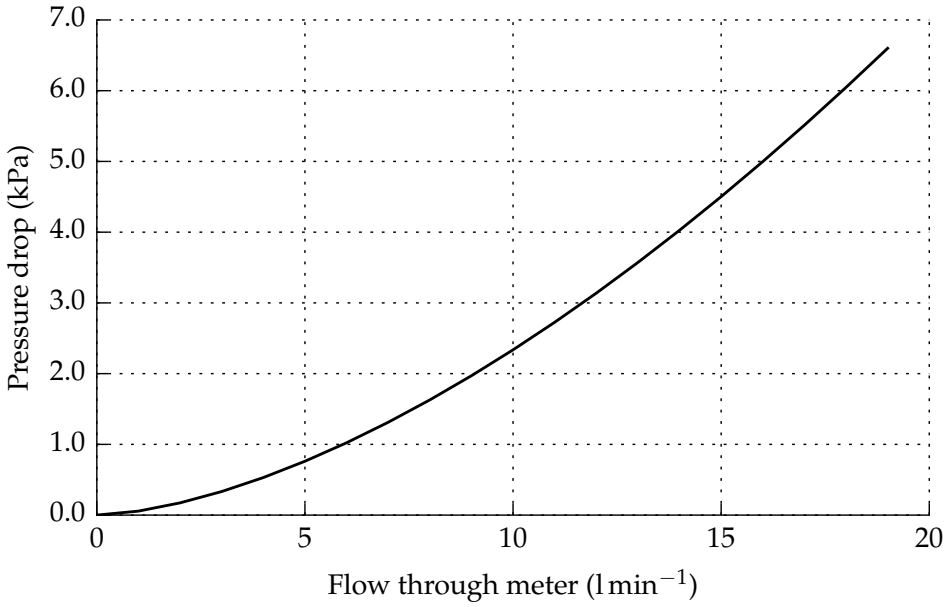


Figure 4.6: Graph showing fitted curve to the pressure loss graph presented as fig. 4.5.

meters. The following equation was created to describe the trace representing the 25 mm PSMT meter:

$$\Delta P = e^{3.725\,log(flow)-9.5} \tag{4.1}$$

Equation (4.1) is plotted in fig. 4.6 on linear scales.

Knowing the pressure differential as a function of flow provides a means of converting between flow and power dissipation. Like an electrical resistance, power dissipated by a fluid-flow obstruction is the product of the difference in driving force across the resistance and the flow through it. In this case the driving force is pressure and the flow is volumetric.

$$\begin{aligned} power &= pressure \cdot flow \\ Watt &= Pascal \cdot \frac{cubicmeter}{second} \\ \frac{kg \cdot m^2}{s^3} &= \frac{kg}{m \cdot s^2}\,\frac{m^3}{s} \end{aligned} \tag{4.2}$$

Equation (4.2) shows the units that will be used to determine power dissipation and ensures they balance. Running the profiles of fig. 4.4 through eq. (4.1), with relevant unit conversions, yields fig. 4.7. It shows the power dissipated within the water during each event. Integrating each trace with respect to time gives the total energy lost over the course of each. Those

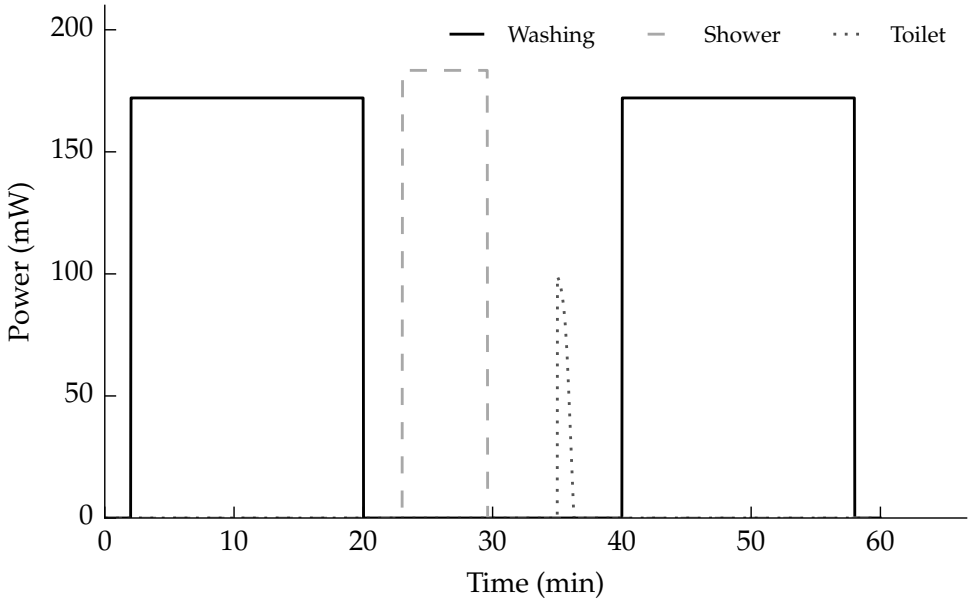


Figure 4.7: Graph of calculated power dissipation in a typical domestic mechanical water meter for each of the sample profile events.

Washing	172 J
Shower	72.6 J
Toilet	5.07 J

Table 4.2: Calculated energy dissipation within a mechanical water meter for a single washing machine cycle, shower and toilet use.

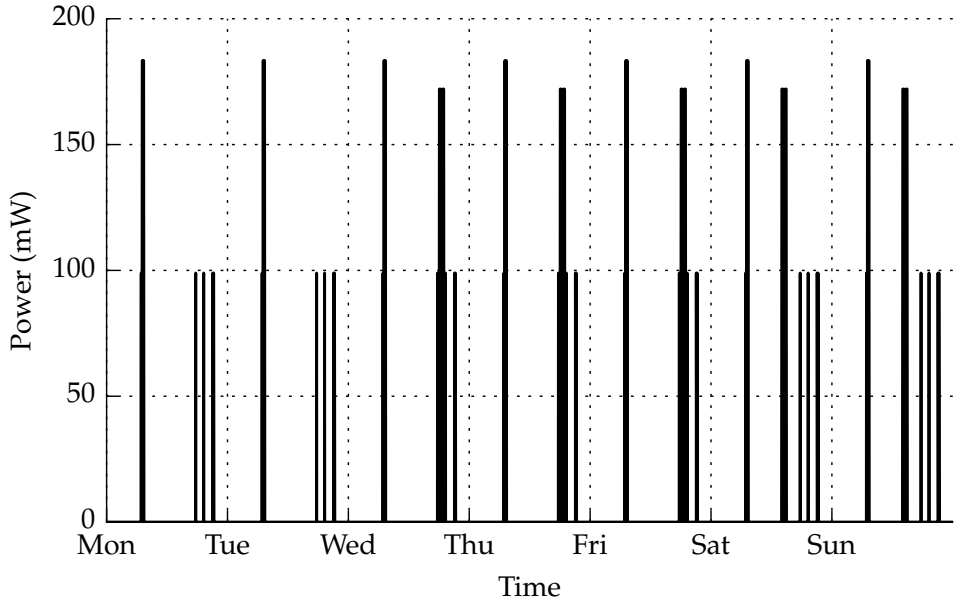


Figure 4.8: Graph of power dissipation within a mechanical water meter over a week for a two occupant dwelling.

Washing	860 J
Shower	1020 J
Toilet	283.9 J

Table 4.3: Calculated total energy dissipation over a period of one week within a mechanical water meter for typical use of a washing machine, shower and toilet.

energy figures are provided in table 4.2.

Average use frequencies for each event type are reported by Heinrech and are shown earlier in table 4.1. Using these figures, and by selecting appropriate times of day, an energy dissipation profile representative of one week was constructed. Five uses of a washing machine, fourteen showers, and fifty-six toilet flushes occur during this time. This profile is shown as fig. 4.8. The profile fits usage figures of a home having two occupants, although most homes have more than two occupants. A systematic bias toward underestimating typical water usage has been used where possible. This underestimation also occurs by not including water use by means other than washing machines, showers or toilets. The intention is that the feasibility of harvesting, if the results showed near a possibility, would be more robust in light of these biases. Table 4.3 combines the energy dissipation for each event type over a week. The total energy dissipated in the meter during such a week is 2.16 kJ. Daily energy available is expected to fluctuate due to sporadic use of the washing machine in most homes. Ignoring washing machine use, indicative of week-

day consumption, the quantity of harvestable energy is expected to be about 280 J per day.

Knowing the quantity of energy available to a harvester is a key factor determining its feasibility. The efficiency of converting energy into the electrical domain was measured in chapter 3 and was found to be 2×10^{-9} Based on that figure the measured cell would collect 560 nJ of energy per day from a two person home. Energy output that low is unlikely to be sufficient for automatic meter reading. The efficiency measured in chapter 3 was lower than others had suggested. It may still be possible to close the gap between what we can produce and what we need. The amount of energy required to run an electronic water meter is estimated next.

Chapter 5
Required Harvesting Performance

The amount of energy lost in a mechanical water meter has been estimated, as has the fraction of that energy which can be harnessed. Now, the amount of energy required to operate an electronic water meter is sought. This estimation will reveal how much further the cells built earlier would need to be improved to be viable.

Microcontrollers

Central to the operation of an electronic water meter is the microcontroller (MCU). The primary function of the microcontroller is to read and log the amount of water consumed by the meter. The programme contained on the controller will also decide when to transmit that data and monitor energy usage. It is therefore a key component and is expected to consume the majority of energy.

This chapter compares the power consumption and operational efficiencies of six low power MCUs deemed suitable for use in electronic metering applications. These microprocessors are low power, general purpose, 8-bit processors from Microchip, Atmel, and Freescale. Each of the microprocessors will have their energy consumption recorded while carrying out various functions over a range of supply voltages. Such measured functions are analogue-to-digital conversion, non-volatile memory writes, processing, and sleeping.

Selection of low power processors

The following processors were chosen from the three chosen manufacturers.

- Microchip PIC16F1827
- Microchip PIC16F688

	PIC16F1827	PIC16F887	PIC16F688	PIC12F675	ATtiny25V	ATtiny13V	MC9S08QG8
Vdd (min)	1.8	2.0	2.0V	2.0	1.8	1.8	1.8
Vdd (max)	5.5	5.5	5.5V	5.5	5.5	5.5	3.6
I (sleep)	30nA	50nA	50nA	1nA	100nA	<100nA	450nA
CLOCK (min)	31kHz	31kHz	31kHz	31kHz	16kHz	16kHz	1MHz
CLOCK (max)	32MHz	8MHz	8MHz	4MHz	16MHz	9MHz	10MHz
EEPROM	256B	256B	256B	128B	128B	64B	†
Serial	USI	USI	USI	–	USI	–	USI
USART	UART	UART	UART	–	–	–	–
ADC	10bit	10bit	10it	10bit	10bit	10bit	10bit

†Has 8,192 bytes of software programmable flash (16 pages of 512 bytes each).
‡Has 256 bytes of software programmable flash (4 pages of 64 bytes each).

Table 5.1: Feature comparison of benchmarked microprocessors.

- Microchip PIC12F675
- Atmel ATtiny25V
- Atmel ATtiny13V
- Freescale MC9S08QG8

A basic feature comparison of the MCU selection is shown in table 5.1.

Benchmarking power consumption

It is important to ensure that each processor is operated so as to minimise power consumption, which meant taking certain precautions. Spare pins were set as outputs and tied to Vdd with 10 kΩ resistors. Unused peripherals were disabled including any watchdog timers and brownout detection circuitry. To allow more accurate sleep current measurements, the chips were placed in a chip carrier with 10 kΩ resistors soldered between the general purpose pins and Vdd. The chip and carrier was washed in isopropyl alcohol and dried before being suspended by connections to the measurement device. This step minimises leakage current between the pins due to oils and dirt that may be transferred by touching or resting on a table. Measurements were carried out using the Agilent E5270B Precision Measurement Mainframe. This device has been used for most other measurement situations throughout this thesis for its high impedance inputs and measurement accuracy.

Sleep mode

A microprocessor in sleep mode is essentially powered off, the difference being that volatile data is preserved. In order to consume as little power as possible an MCU should spend as much time in sleep mode as possible. The power consumption while sleep states will determine a large part of the water meters overall energy requirements.

Figure 5.1 shows the amount of current consumed by each MCUs while in their deepest sleep states. Surprisingly, the PIC16F1827 consumes the most current in this state, almost one thousand times more than the specified sleep current of 30 nA [61]. The Freescale MC9S08QG8 consumed energy at an average of 11% higher than specified [62]. Both the Atmel ATtiny13V and ATtiny25V fell within their specification, [63, 64] respectively. The Microchip PIC12F675 fell within specification [65] and was clearly ahead in terms of minimum current draw during sleep.

There appears to be a trade-off between the two Atmel processors in the way of minimum power consumption and minimum response to Vdd. The

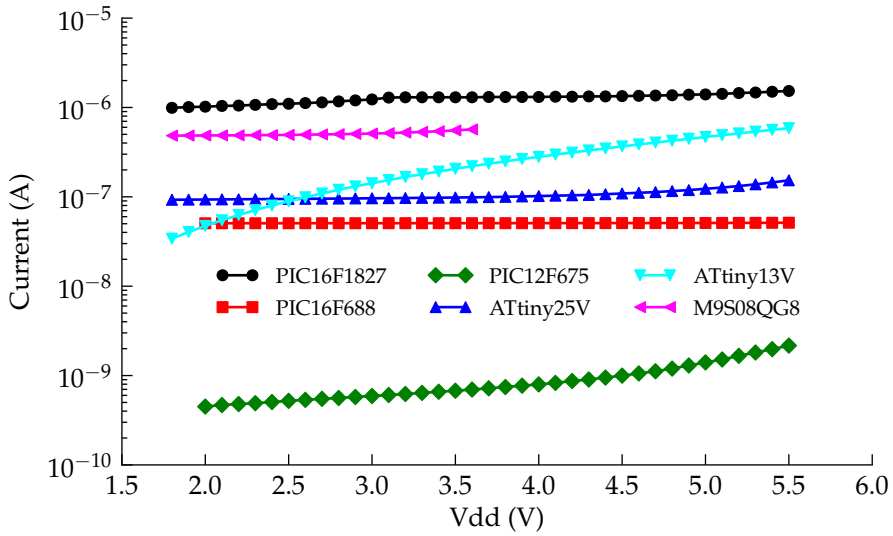


Figure 5.1: Graph showing current consumed by MCUs in sleep mode versus supply voltage.

	Instructions
PIC16F1827	53
PIC16F688	35
PIC12F675	35
ATtiny25V	120
ATtiny13V	120
MC9S08QG8	145

Table 5.2: Instruction-set size for each tested microprocessor.

ATtiny13V required approximately 2.7 times less power than the ATtiny25V at 1.8V, but above 2.5V the ATtiny25V draws less current.

As the PIC16F1827 was so far off its specified value, measurements were repeated numerous times using code written in both assembler and HI-TECH C. A total of five different processors were tried, all giving the same result. All steps outlined in the PIC16F1827's datasheet to reduce power consumption had been followed.

Disclaimer on processing

Measuring the amount of power required to process information is complicated. The way each chip carries out processing operations internally can differ from one another, even though all produce the same result.

To illustrate, algorithm 5.1 shows a simplified programme. To determine the programme's outcome the processor must first evaluate whether 'danger' is greater than or equal to five. Then it will either branch to the function 'fight'

Algorithm 5.1 Simple C-code representation of a branch instruction.

```
1        if (danger >= 5) flight();
2        else fight();
```

Algorithm 5.2 Pseudo machine-code representation of a branch instruction.

```
1        load 5 into register 001
2        load danger into register 002
3        branch-if-greater-or-equal 001 002 flight_call
4        call-subroutine fight
5        jump-to continue
6        [flight_call]
7        call-subroutine flight
8        [continue]
```

or continue on to execute the function 'flight'.

Algorithms 5.2 and 5.3 demonstrate two different ways of implementing 5.1 using pseudo machine-code. The decision of which to use is made by the compiler, which *should* take the instructional efficiency of the specific MCU into account. This is an overly simplistic example, but it illustrates that there are multiple paths leading to the same result. Importantly, not all of those paths require the same amount of effort on the processor's behalf. This means that the compiler's ability to optimise code efficiently plays a role in determining the overall performance of the chip. This also means that the programmer should not be concerned with instructional efficiency as the compiler should transform C-code into machine code that best suits the target MCU.

Another factor in processing efficiency comes down to the number of different instructions it is capable of. The list of instructions a processor is capable of is called its instruction set. Most 8-bit MCUs are based on reduced instruction set computing (RISC) architecture, as opposed to complex instruction set computing (CISC). When compared to a CISC based CPU, a RISC based chip

Algorithm 5.3 Pseudo machine-code representation of an alternative branch instruction.

```
1        load danger into register 001
2        subtract-from-register 001 5
3        branch-if-minus 001 fight_call
4        call-subroutine flight
5        jump-to continue
6        [fight_call]
7        call-subroutine fight
8        [continue]
```

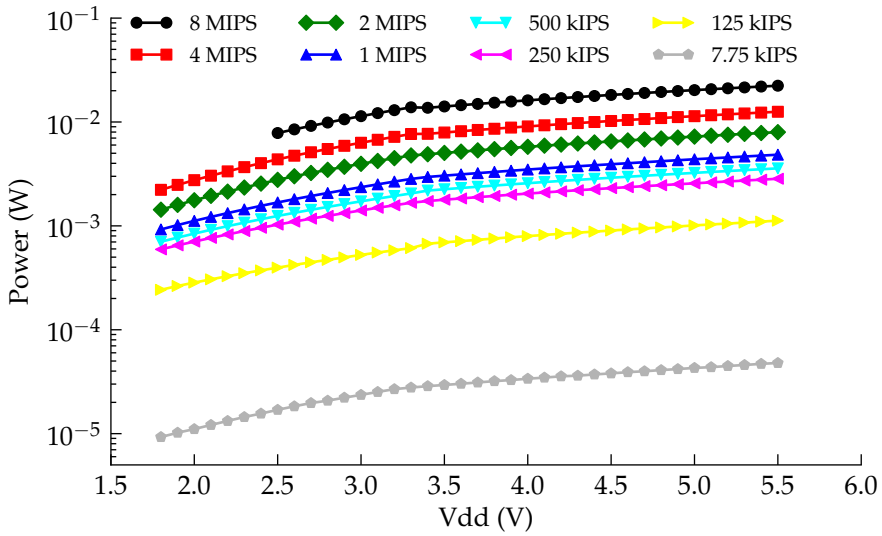


Figure 5.2: Graph showing power consumed by the PIC16F1827 while processing versus supply voltage.

is simpler and therefore usually cheaper to produce and simpler to program. However, "Instruction traces from CISC machines consistently show that few of the available instructions are used in most computing environments" [66], meaning that many of the extended operations in CISC designs are under-utilised. Processors with smaller instruction sets are capable of achieving the more complex operations by chaining multiple instructions together. This means that processors with smaller instruction sets may take longer to execute certain instructions. Finally, the frequency of a microprocessor isn't necessarily the frequency at which it performs operations, although sometimes it is. For instance, the Atmel and Freescale microprocessors perform one instruction per clock cycle, whereas the Microchip processors perform one instruction every four clock cycles.

Processing

Results in this section are expressed in terms of instructions per second (IPS). The Microchip PIC16F1827 displayed the lowest energy usage with 10 µA while clocking 7.75 kIPS (as shown in fig. 5.2). Microchip MCUs complete one instruction every four clock cycles, so the 7.75 kIPS actually corresponds to a standard clock frequency of 31 kHz.

Figure 5.3 shows that the PIC16F688 consumes less power than the PIC16F1827 at low voltages for the same instruction rates (except at 7.75kIPS). There appears to be a flatter response in power consumption with respect to Vdd in

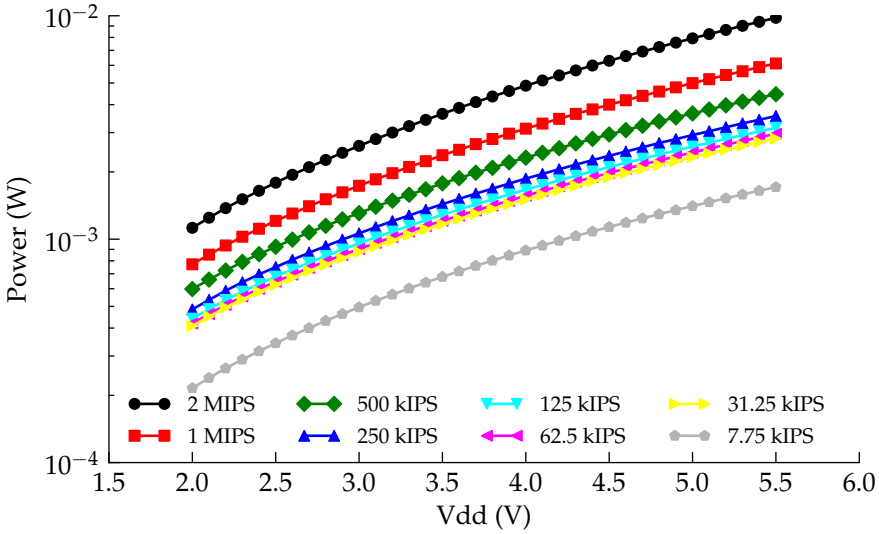


Figure 5.3: Graph showing power consumed by the PIC16F688 while processing versus supply voltage.

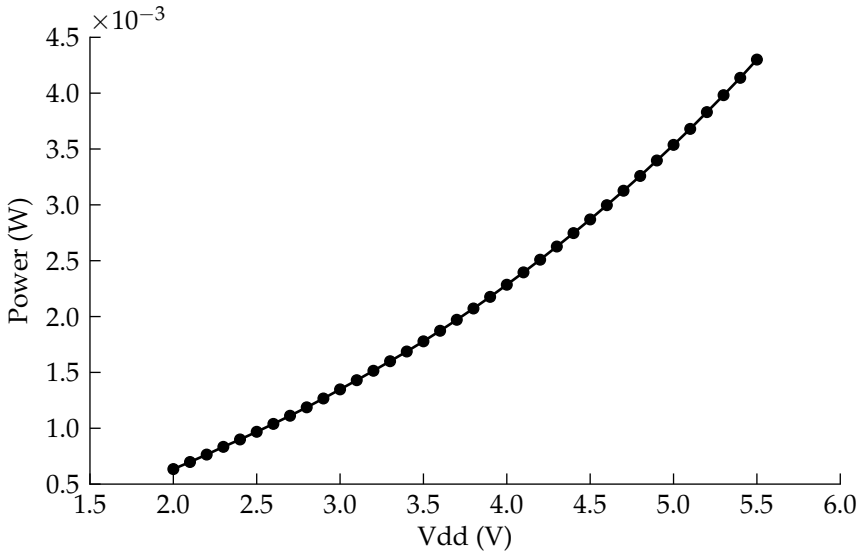


Figure 5.4: Graph showing power consumed by the PIC12F675 while processing versus supply voltage.

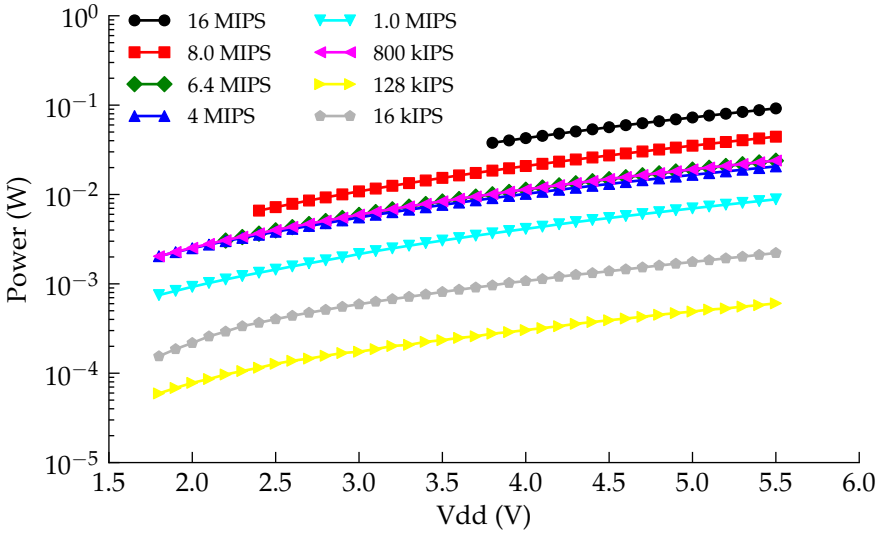


Figure 5.5: Graph showing power consumed by the ATtiny25V while processing versus supply voltage.

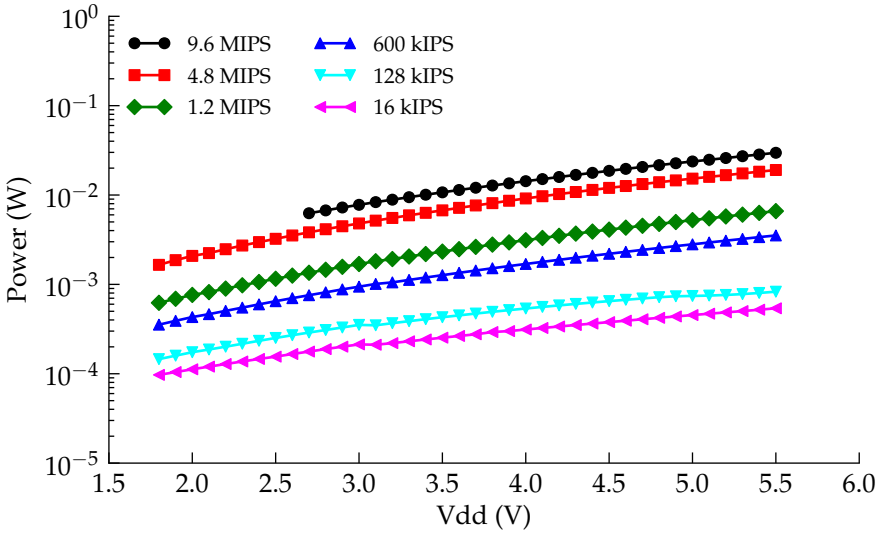


Figure 5.6: Graph showing power consumed by the ATtiny13V while processing versus supply voltage.

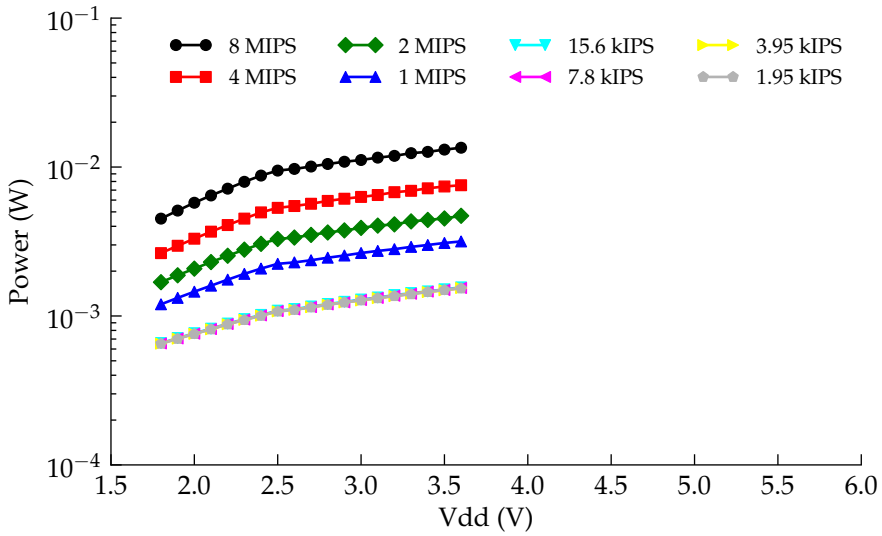


Figure 5.7: Graph showing power consumed by the MC9S08QG8 while processing versus supply voltage.

the PIC16F1827. Again, this appears to be a similar trade-off to what was mentioned earlier (in fig. 5.1) with the Atmel chips.

The PIC12F675 (fig. 5.4) used approximately the same power as the PIC16F688 (fig. 5.3) for its 1 MIPS trace. Figures 5.5 and 5.6 show both Atmel MCUs having similar requirements. The MC9S08QG8, although being able to clock the slowest, performed very poorly at low frequencies. At 1.95 kIPS it consumed approximately the same amount of power as the Microchip MCUs operating at 1 MIPS.

Overall, the PIC16F1827 gives the widest range of power consumption options, with the ATtiny25V offering similar performance options.

Joules of energy consumed per instruction cycle

A convenient, and more insightful, way to interpret the previous processing power consumption graphs is to calculate the energy spent per instruction performed. The energy cost of an instruction cycle can be calculated using equation 5.1.

$$E_i = \frac{I \times Vdd}{f_i} \tag{5.1}$$

where E_i is the number of joules consumed per instruction, I is the current draw, Vdd is the input voltage and f_i is the instruction frequency.

Figure 5.8 compares the most energy efficient operating conditions of each of the tested chips. What is most interesting about this graph is the

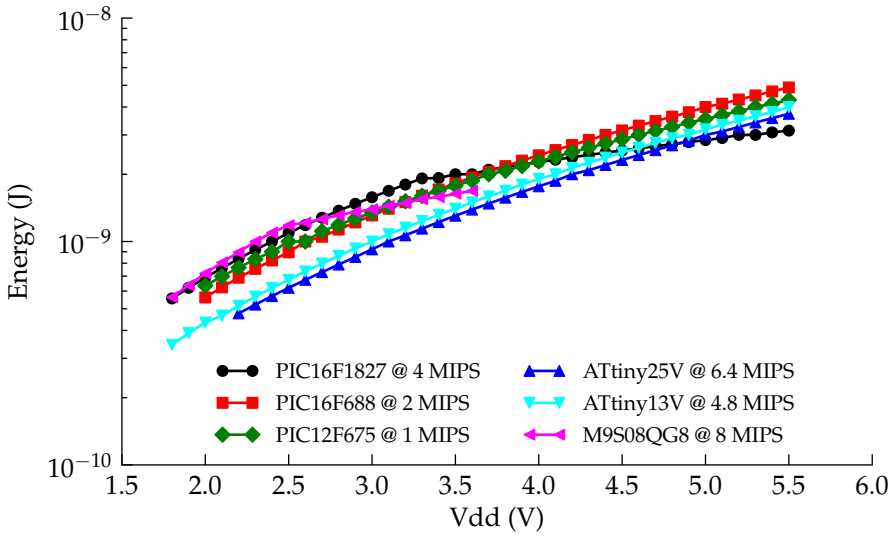


Figure 5.8: Graph showing instruction cycle energy consumption for each MCU versus supply voltage.

Algorithm 5.4 Benchmarking algorithm

```
1  unsigned short lfsr = 0xACE1u;
2  unsigned period = 0;
3  do {
4    lfsr = (lfsr >> 1) ^ (-(lfsr & 1u) & 0xB400u);
5    ++period;
6  } while(lfsr != 0xACE1u);
```

high degree of overlap. Also, the greater efficiencies occur at high operating frequencies. A simple rule of thumb for selecting the most power efficient operating frequency based on these results is to choose the highest frequency where the MCU can operate over its full input voltage (Vdd) range. For comparison, figure 5.9 shows the trade-off made when selecting a higher frequency, which is typical across the range of MCUs tested.

Instruction efficiency

Calculating the amount of energy consumed per instruction only shows part of processor efficiency. The amount of processing done per instruction is not taken into account in such measurements. Some MCUs have extra instructions that are designed to help speed up code execution by combining commonly used groups of instructions. To shed light on instructional efficiency, the number of instructions each of the processors takes to complete a benchmark function is found. This will allow for a more accurate representation of execution efficiency. The function used to benchmark each of the processors is

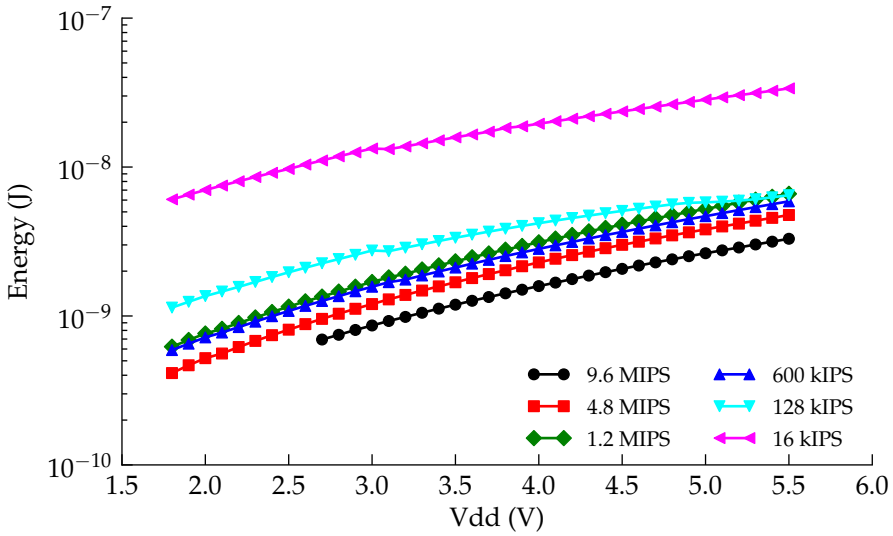


Figure 5.9: Graph showing instruction cycle energy consumption of the ATtiny13V versus supply voltage.

a linear feedback shift register based pseudo-random number generator [67]. It is well suited to an 8-bit microprocessor as it requires no complex math functions, uses little memory and has a well defined end. The code for this function is shown as algorithm 5.4. It starts with a 16-bit number and runs it through the linear feedback register in a tight loop until the initial value of the 16-bit feedback register is produced again. This function steps through every possible combination of bits possible in a 16-bit number (except for 0 and 65535) in a pseudo-random order before exiting the loop. The function combines the exclusive-OR (XOR), bit shifting, bitwise AND, increment a value and numerical comparison operations in a tight loop. The benchmarking function was compiled and run on each of the MCUs operating at a range of supported frequencies.

To determine the instructional efficiency, the code was set to toggle the state of a digital output pin. The toggle frequency of that pin was recorded using a Tektronix MSO 4054 oscilloscope. The number of instruction cycles each chip took to complete the benchmark was deduced by multiplying the time taken to complete the benchmark by the instruction cycle frequency. The results of the benchmark are shown in fig. 5.10. To calculate the number of instruction cycles taken by the Microchip family of processors, one quarter of the chip's operation frequency was used. This meant that the number of clock cycles consumed was four times higher. It is clear from fig. 5.10 that the Atmel (ATtiny25V and ATtiny13V) microprocessors are by far the most efficient microprocessor in terms of executing code using a minimum number

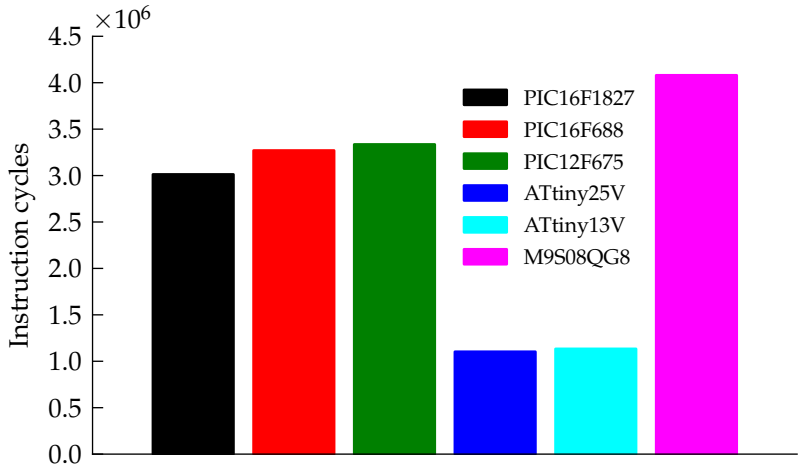


Figure 5.10: Graph showing the number of instruction cycles taken to complete a benchmark routine for each MCU.

of instructions of the selection. The reason for this is most probably due to the larger instruction set and higher compiler optimisation.

Non-volatile memory

In order for a microprocessor to keep information about its current state and recorded data in the event of power loss it must write to non-volatile memory. Non-volatile memory is implemented as either electrically erasable and programmable read only memory (EEPROM) or Flash memory. Flash memory is similar to EEPROM with the exception that it must be erased in large blocks, or pages, before it can be written to. All of the tested MCUs have on-board EEPROM with the exception of the MC9S08QG8 which has flash memory instead. Table 5.1 shows the amount of non-volatile memory space available on each of the chips.

The energy consumption of each of the chips with EEPROM memory during a 1-byte write operation is shown as fig. 5.11. A curious situation arose with the PIC12F675 where its calculated energy consumption was negative when operated below 4.7 V. It consumed *less* current while performing write operations than running through the same code loop without performing writes. The measurement was repeated several times and produced the same result. Those data points were excluded from the plot as they do not represent the true energy cost of writing to EEPROM. It is likely that while writing to EEPROM other parts of this chip are disabled or put to sleep.

In the case of the MC9S08QG8, which has Flash memory instead of EEPROM, the power consumption in the 'E + W' trace was calculated as $1/512^{th}$

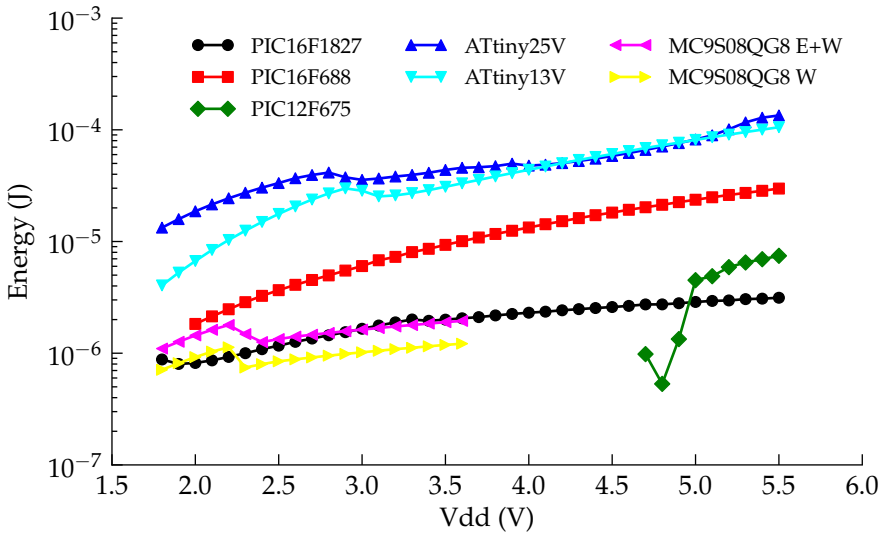


Figure 5.11: Graph showing energy consumed per non-volatile erase/write operation versus MCU supply voltage.

of consumed page erase energy consumption added to the energy cost of a single write operation. The trace labelled 'W' (magenta) shows the energy cost for a single write operation. In order for the MC9S08QG8 to perform a write operation, the destination byte must have already been pre-erased at an earlier point in time. This may be useful for power harvesting since the energy expensive page erase operation, which consumes an average of 302 µJ, can be performed when available energy is plentiful. These results show that the Microchip and Freescale microprocessors are the most energy efficient when writing to non-volatile memory.

Analog-to-digital conversion

The operation of an electronic water meter may require that analogue-to-digital conversions are made. Measuring the amount of energy consumed per conversion was done in much the same way as the previous tests. Each of the chips had similar converters feature-wise. Results from the measurements are presented as fig. 5.12.

Wireless Transmission

Because water meters are typically installed in remote areas where grid connection is unavailable, data must be collected via wireless interface. In Hamilton, a major utility provider has established a wireless mesh network between

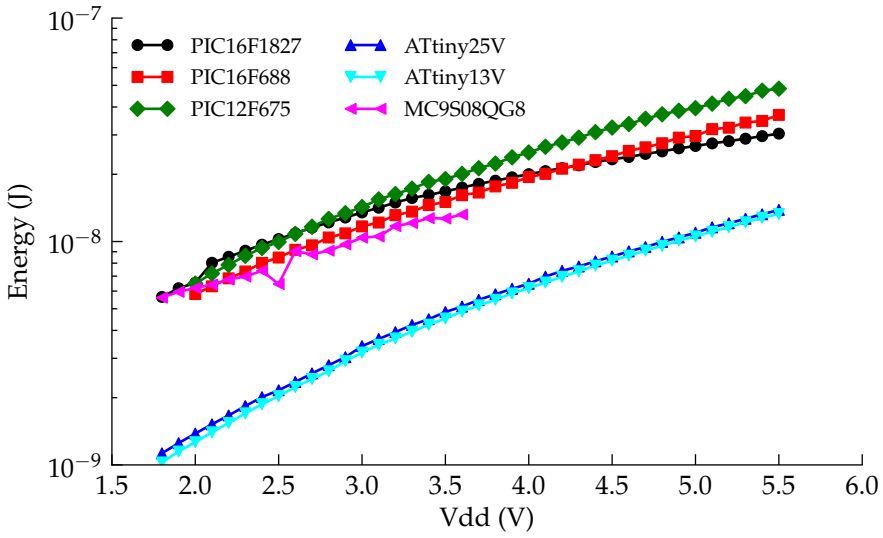


Figure 5.12: Graph showing the energy consumed per ADC measurement versus MCU supply voltage.

smart electricity meters installed in residential homes. That network utilises ZigBee wireless transceivers, making ZigBee an convenient choice for transmitting water metering data [68]. Two types of wireless transmitters were chosen for energy measurement, a HOPE RF RFM12B transceiver and a Digi International Xbee Series 2 transceiver. The power consumption versus time during a wake, send one hundred and sixty bytes, and power down cycle was captured for both transceivers. By integrating the area under this curve the total energy consumed per transmission is found. The transmitters were kept 1 m from their receivers with no obstructions between them. This represents ideal transmission conditions, something that our electronic water meter is very unlikely to encounter. The actual RF reception between a base station and installed meter will vary greatly between installations and weather conditions. For example, wet ground is likely to obstruct RF transmission due to the transmitter being shielded by a more conductive medium. Instead of trying to quantify the energy required in those situations, the best case was measured and an estimate of the worst case is estimated to be one hundred times larger. The transmit power can increase by a factor of 320 for the RFM module, however the transmit power only represents a proportion of total power usage. The modules must power up, start their internal oscillators, receive the data to be transmitted from the main processor and then transmit the data packets. I have crudely estimated that the difference between the lowest and the highest total power consumption, based on reception alone, will be a factor of 100.

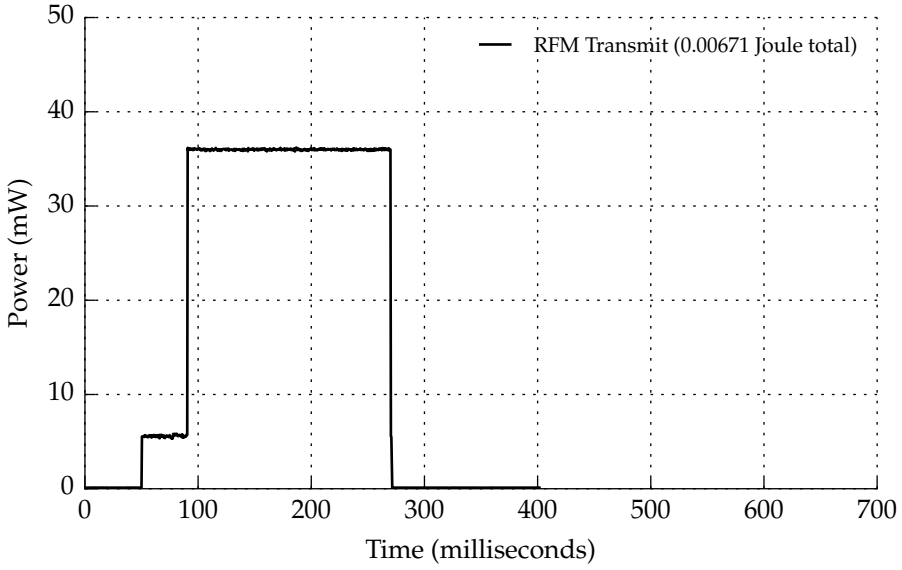


Figure 5.13: Graph of power draw from a HopeRF RFM12B transceiver module versus time during a power-up and transmit sequence.

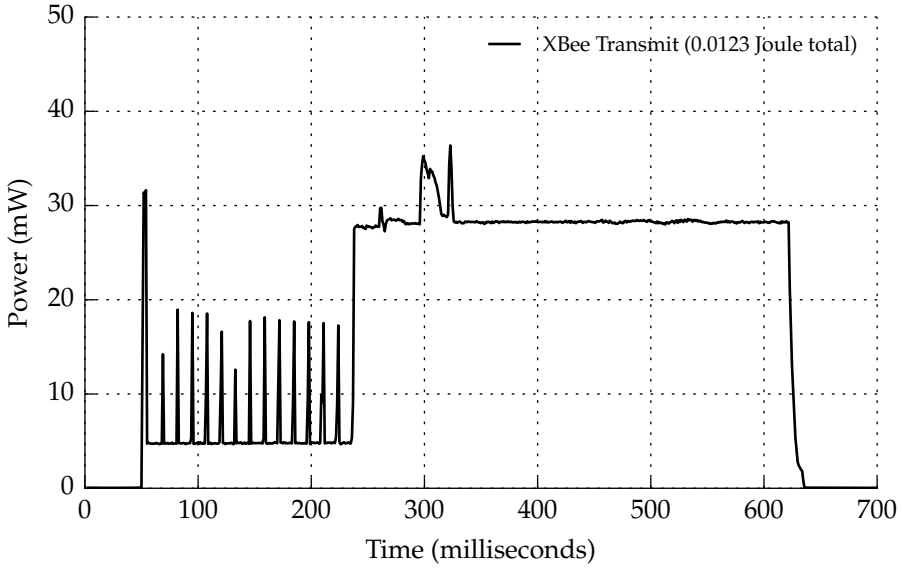


Figure 5.14: Graph of power draw from a Digi International Xbee Series 2 transceiver module versus time during a wake & transmit sequence.

The RFM12B operates with a carrier frequency of 433 MHz, a globally available and license-free frequency. It has been shown that frequencies in the 300 −-400 MHz range reduce the path loss in buried transmitter situations to a level that allows feasible communication [69]. Additionally, 433 MHz has the ability to penetrate concrete and water [70]. It has a maximum output power of 5 dBm (3.2 mW) at this frequency. The Xbee has a maximum output power of 0 dBm (1 mW) and operates at 2.4 GHz. Figures 5.13 and 5.14 show the captured power consumption during the tests. The RFM12B used about half as much energy sending the same data as the Xbee. Combined with its favourable communication frequency, the RFM12B is a sensible choice for the electronic water meter. In total the RFM consumed 6.71 mJ and the Xbee consumed 12.3 mJ. Adjusting these figures for the worst case (multiplying by one hundred) gives 671 mJ for the RFM12B and 1.23 J for the Xbee.

Final Estimate of Energy Requirements

A crude estimation of an electronic water meter's microprocessor event loop is as follows.

1. Sleep for 1 second
2. Execute 1000 instructions
3. Make 2 analogue conversions
4. Write 2 bytes to non-volatile memory

This would allow the microprocessor to watch the display of a mechanical water meter and store the readings. This loop would occupy approximately one second, so it will occur 86400 times per day. Every so often the collected data would need to be transmitted, a potentially costly exercise in terms of energy usage. On top of the previously stated event loop is a data transmission loop which would execute every six hours.

1. Execute 1000000 instructions (Data compression)
2. Power up and transmit 160 bytes of using RF transceiver
3. Write 10 bytes to non-volatile memory

Tables 5.3 and 5.4 combine the measurements and estimates from the previous sections with the event loop estimation. They show that approximately 12 J would be consumed per day by an electronic water meter. It was shown

Mode	Count	Unit	Energy
Sleep	1	seconds	97.4 nJ
Processing	1000	instructions	1.14 µJ
ADC	2	conversions	2.56 nJ
EEPROM	2	bytes	79.0 µJ
		Total (per day)	6.93 J

Table 5.3: Estimated daily energy expenditure for basic processing functionality

Mode	Count	Unit	Energy
Processing	1000000	instructions	1.14 mJ
Transmit	160	bytes	1.23 J
EEPROM	10	bytes	394 µJ
		Total (per day)	4.93 J

Table 5.4: Estimated daily energy expenditure required to transmit 160 characters every six hours

in section 4.3 that 280 J of energy is already dissipated in a mechanical water meter per day. This equates to a conversion efficiency of 4.28 %, which would be the minimum efficiency required to run the meter continuously.

Chapter 6
Conclusion

Estimates of energy availability have been made by calculating the amount of energy lost in a traditional water meter. Those estimates showed that for a typical New Zealand household approximately 280 J of harvestable energy is available per day. In the previous section, the amount of energy that would be needed to run an electronic water meter was estimated to be 12 J per day. This means that in order to power an electronic water meter a minimum conversion efficiency of 4.28 % is required. However, calculations from cells assembled in section 3.3 showed that readily obtainable conversion efficiencies are in the order of 2.8×10^{-9} Conversion efficiencies over 1 % have not been reported in the literature, and one paper suggests the theoretical maximum is 2 % [21]. For these reasons, the use of streaming cells as a method of energy harvesting from water and current materials is expected to be infeasible. The literature suggests there is room for improvement, to levels which would make streaming cell harvesters practical, but these gains are reliant on new nano-materials.

During the course of this research two issues came to light with regards to streaming cell harvesting. The first issue is a susceptibility to clogging. Having such narrow openings in a domestic water feed is likely to trap dirt and contaminants at the channel openings. This lowers the effective efficiency and will require periodic cleaning, lowering the benefit to utility companies. The second issue is the manufacturing precision required to create the channels. Parts manufactured with high precision are generally small, but a streaming cell would need precise dimensions and a large surface area. This could be overcome with the use of materials like porous glass. As a result of the low measured efficiency, streaming cells for the purpose of energy harvesting are not studied further. Part II looks at the electrical impedance of medical implant electrodes. The research here into double layers and the role they play in energy harvesting applications is directly applicable there.

Part II

Double Layers on Conductors: Electrical Impedance

For the engineers of medical implant devices, knowing the electrical impedance between electrodes is vital. Having a tool to simulate such impedances allows those designers to ensure fault free operation of potentially lethal devices. The most commonly used theoretical approach in electrochemical modelling is to construct an equivalent electrical circuit [71]. An electrical model of the interface impedance between electrode and electrolyte is presented in the proceeding chapter. That model was developed by my chief supervisor Jonathan Scott and Peter Single of Saluda Medical, Sydney. Validation of the model was made in a standard solution of saline, but details of how saline concentration affected the parameters were unknown. In part II of this thesis I take that model and extend its predictive capability to a range of salinities. Having such a model allows for easy comparison between different electrolytes and electrode geometries. Using that ability, I characterise the interface in an anaesthetised sheep's spinal cavity and compare the results to the various saline solutions measured in the lab. That comparison showed that the situation in live sheep is significantly different to that of standard saline solutions. Using the measurement methods developed, I then develop a mixture that closely matches the electrical impedance seen in sheep. This mixture now serves as an improved test solution for the engineers of medical implant devices.

Chapter 7
Interface Modelling

This chapter looks at each of the components within the interface model. The parameters that govern the behaviour of those components will be discussed, as will methods of measuring those parameters. The measurements and determined parameters will be presented in the next chapter (chapter 8).

The Scott-Single Interface Model

In 2013, Jonathan Scott and Peter Single published an electrical model of an implantable electrode array in saline [72]. That model simulates the electrical impedance an implanted electrode experiences once implanted into a human spinal cavity. It is general enough to use in any situation where electrodes are placed in an electrolyte. A simple case of such a situation is depicted in fig. 7.1. The model comes in two parts: an electrode interface, and the resistance of the electrolyte between interfaces. Figure 7.2 shows the electrical equivalent of connecting two electrodes through an electrolyte. It has two interfaces, both having their liquid sides joined electrically through the bulk resistivity of the electrolyte. The metal side of the interfaces is what the rest of the circuit (such

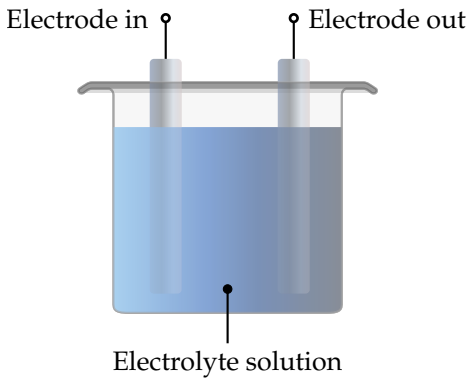


Figure 7.1: Diagram of two electrodes submerged in an electrolyte solution; which can modelled by the Scott-Single Interface Model.

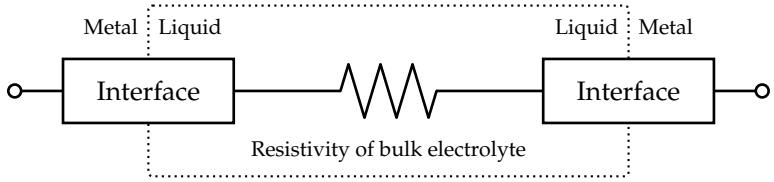


Figure 7.2: Connection diagram of two electrodes (with interfaces) connected together by the resistivity of an electrolyte solution.

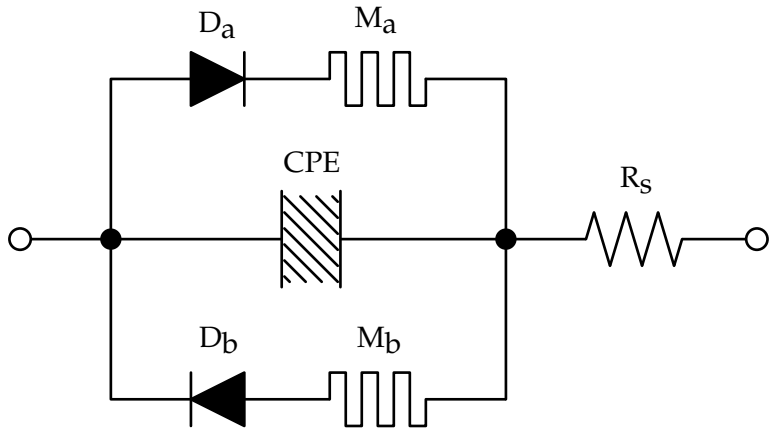


Figure 7.3: Electrical schematic of the electrode-electrolyte interface

as the implant electronics) connects to.

Figure 7.3 shows the electrode-electrolyte interface. This is what is inside each of the 'Interface' boxes shown in fig. 7.2). It is an electronic equivalent circuit of the transition between the metal of an electrode and the liquid of an electrolyte. The interface model used throughout this thesis is a simplified version of this in that the memristors (elements M_a and M_b) have been omitted.

The model has three parallel branches connected in series with a resistor. That resistor represents the non-reactive resistance of the interface itself, which all current passing through the interface experiences.

The two diode and memristor branches (top and bottom) mimic Faradaic conduction. Any current leaving the electrode and entering the electrolyte (or vise versa) must pass through one of these two branches. The direction of the diode on the branch determines whether it passes cathodic current or anodic current. Together they mimic the rate of a specific Faradaic reaction. As there is only one anodic/cathodic pair, the model simulates only one reaction (proceeding in either direction). For the purpose of impedance modelling in this thesis it is not important to know exactly what that specific reaction is. Since Faradaic currents are to be avoided in an implanted setting, determining their onset is critical. Therefore, it is important to know that those paths model the *first* (or lowest energy) reaction between the electrode and electrolyte.

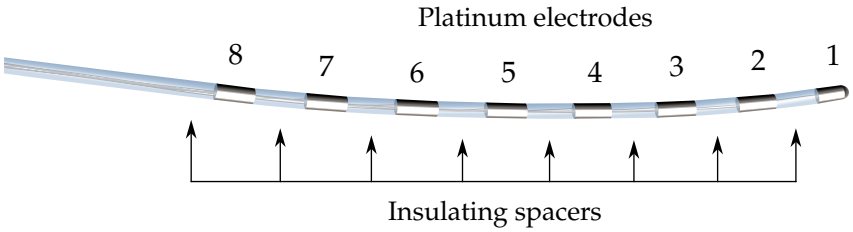


Figure 7.4: St. Jude Medical Octrode. An eight electrode array commonly used in spinal stimulation implants. The electrode numbering shown here will be used throughout this work.

Finally, at the centre of the model is the constant phase element (CPE). For the purpose of a light introduction assume that it is simply a capacitor. Any small signals can pass through the CPE, and therefore across the interface. It models the redistribution of ions around the electrode in response to a change in electrode voltage. A sudden change in voltage causes ions in the electrolyte to be repelled from, or attracted to, the electrode. Attracting or repelling ions *is* moving charge. Once the ions have redistributed, that movement of charge stops. This is the capacitive nature that the CPE models. Before moving on, consider that moving ions around requires energy (a volume, however small, of liquid has moved). This means a CPE does not behave entirely like a capacitor – it is lossy and has an impedance-frequency roll-off not equal to $-20\,\mathrm{dB/Decade}$. Mode detail on how this, and the other components, are modelled is presented in the following subsections.

Inter-electrode resistivity (resistor network)

Modelling the resistance between two electrodes in a fixed geometry situation is simply a matter of inserting an appropriately sized resistor between the two interfaces. The resistance is dependent on the electrolyte's conductivity, the combined surface area of the two electrodes and the distance between them. The St. Jude Medical Octrode is an eight electrode array commonly used in spinal cord stimulator implants. Its eight electrodes are made from platinum and are separated along the end of a lead by insulating material. An illustration of an Octrode is presented as fig. 7.4. Modelling such an electrode array requires a resistor network that connects all electrodes to one another. To be an adequate representation, the resistance between every possible combination of electrode pairs must match the measured value in an electrolyte.

Scott and Single created a resistor network for modelling the electrolyte conductivity based on the geometry of the electrodes and the resistivity of the

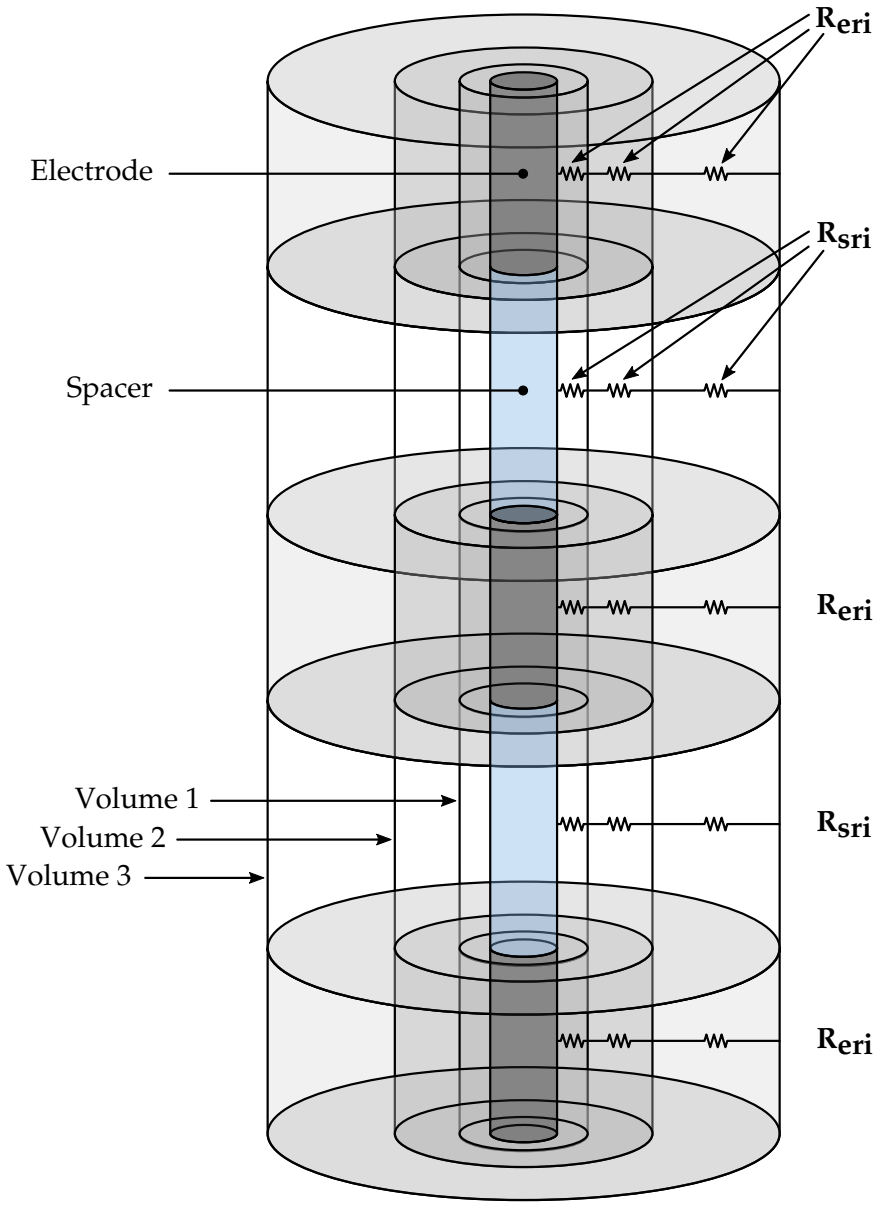


Figure 7.5: Diagram showing the first three radial volumes expanding from the surface of the electrode and insulating spacer. Each volume has a diameter twice that of the one inside it. These volumes are used mathematically to determine relationship between resistances in the resistor mesh.

electrolyte. By sectioning the surrounding liquid into cylindrical volumes they calculated the equivalent resistance between those volumes in both radial and longitudinal planes. The radii of the volumes double at each layer which correspond to a fixed radial resistance between each layer. There are two different radial resistances: one for the rings expanding from the insulating spacers, and one for those expanding from the electrode cylinders. The two layers alternate due to each electrode being separated by an insulating spacer. Figure 7.5 illustrates the idea on a subsection of the electrode array by showing three radial volumes surrounding three electrodes and two spacers. The longitudinal resistances quarter in size with each ring layer and after the last radial resistor each node is shorted together. The full mesh for the eight electrode array is five layers deep with three rows of padding at each terminating end, totalling two hundred and five resistors in total. Figure 7.6 shows the resistor network schematic. Further details of how the mesh geometry and resistor values were calculated can be found in [42].

The parameters that describe the resistor mesh are:

- R_{eri} - The initial resistance placed radially from an electrode.
- R_{sri} - The initial resistance placed radially from a spacer.
- R_{li} - The longitudinal resistance.
- Depth - Number of layers between the electrode/spacer and the common end node in the ladder.
- Padding - Number of additional spacing rows to be added to each end of the mesh.

Interface series resistance (resistor)

The series resistor at the right hand side of the model schematic (labelled R_S) represents the purely resistive component of the interface's impedance. As it is in series with all other components in the interface model, there is no way for charge to cross the interface without encountering this resistance. The parameter used to denote the interface's series resistance is:

- R_S – The series resistance of the interface

Polar/double-layer effects (constant phase element - CPE)

At the centre of the model is the constant phase element (CPE), or fractional pole capacitor. A CPE is a device that behaves like a cross between a capacitor

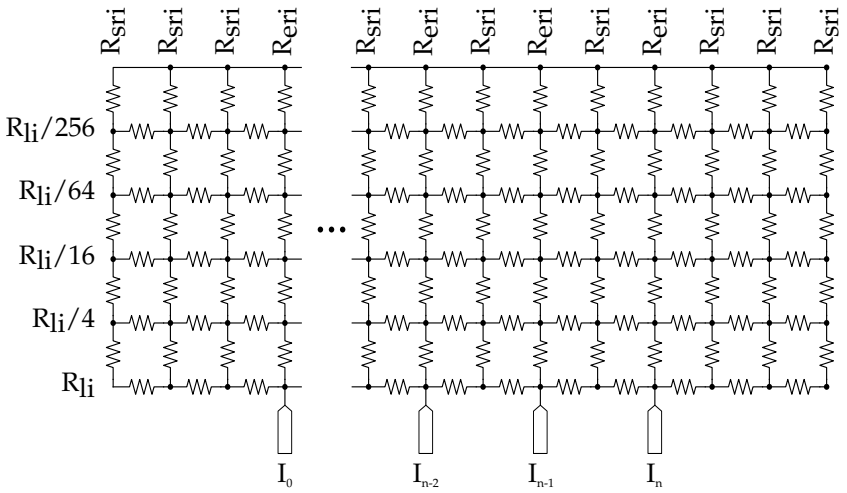


Figure 7.6: Resistor mesh used to model the electrical resistance between interface pairs. R_{li} is longitudinal resistance, R_{sri} and R_{eri} is the radial resistance for the spacers and electrodes respectively, and I is an interface.

and a resistor. They are primarily used to describe the capacitance of double layer interfaces, which is the function it serves in this model. It is capacitive in the sense that voltage leads current, but by an amount less than 90°. Mathematically, the 90° angle between voltage and current in a capacitor comes from:

$$I(t) = C \times \frac{dV(t)}{dt} \tag{7.1}$$

When $V(t)$ is a sine wave, this becomes

$$I(t) = C \times \frac{dSin(t)}{dt} \tag{7.2}$$

$$= C \times Cos(t) \tag{7.3}$$

Which describes how the current always 90° out of phase with voltage in a capacitor. A CPE on the other hand has a phase angle somewhere *between* 0 and $-90°$. This requires a fractional differentiation of eq. (7.1) such as:

$$I(t) = C\frac{d^n V(t)}{dt^n} \tag{7.4}$$

where n is a non-integer number. Partially applied differential is uncommon outside of pure mathematics. As a consequence of having a current/voltage relationship of less than 90°, a CPE's impedance magnitude decays at a rate lower than 20 dB/decade.

SPICE and other commonly used circuit simulators do not support fractional pole capacitors so entering one into the model will require building

it up from discrete components. In 1959, Morrison demonstrated a way of

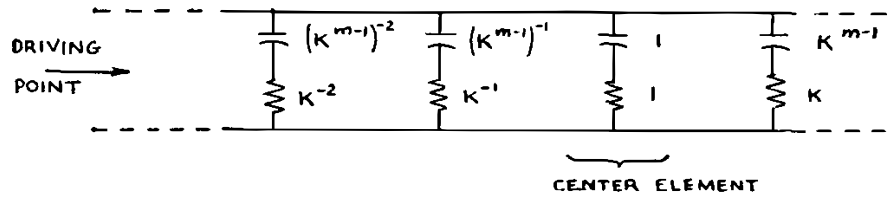


Figure 7.7: Ralph Morrison's implementation of a constant phase element using an infinite array of resistor-capacitor pairs (taken from Morrison's paper – [43]).

creating constant phase elements from an infinite array of resistor-capacitor (RC) pairs [43]. One of Morrison's implementations of a constant phase element is presented as fig. 7.7. In that implementation, each parallel branch has precisely chosen resistor and capacitor values such that when summed together the impedance magnitude versus frequency is a constant slope, i.e., the impedance does not flatten at a particular frequency as it would with a single RC pair. Creating any element comprised of an infinite number of sub-elements is not possible, however by selecting only those elements that contribute to the bandwidth of interest the result is the same within the selected frequency window. Using that method and selecting only RC pairs with a cut-off frequency in the range of 1 mHz to 1 MHz, a practical CPE can be created.

Figure 7.8 shows the individual contributions from each RC branch in an implementation of a CPE. Each grey trace represents a single RC branch within the CPE, as are shown in fig. 7.7. The value of the resistor in each branch is evident by the vertical position of the traces, visible at the right-hand side of the graph. The branches in this particular example have been spaced in the frequency domain at a density of three per decade, as marked by the black crosses. This means that per decade of frequency, there are three corner frequencies, each relating to an RC pair (so three branches for every decade of frequency response). Because each of there branches are in parallel, the total response of the CPE as a whole is the superposition of the response of each branch. That response is shown as the black trace on the graph. The critical observation is that the slope of the resulting trace is different to the slope of the capacitors in each of the individual branches. This allows the CPE to behave fractionally as a capacitor, being anywhere between resistive (horizontal response) and capacitive (20 dB/decade slope). Lines 186–189 of listing E.2 show how values for the resistors and capacitors within the CPE are calculated.

The CPE represents readily reversible reactions, polar reorientation, and

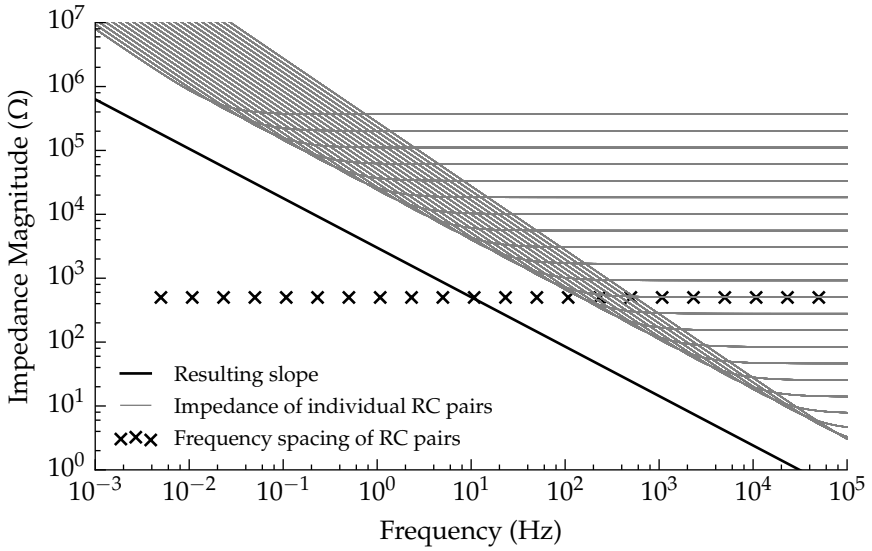


Figure 7.8: Graph showing how a non -20 dB per decade slope can be constructed from a series of -20 dB per decade low-pass filters

ionic repulsion and attraction between the electrode's surface and the electrolyte. It is capacitive in nature because each of these mechanisms store charge, which can be drawn back by reversing the applied electromotive force. Parameters used to describe the behaviour of the CPE are:

- m – Used to select resistor-capacitor pairs in each branch and ultimately determines the slope of the CPE's frequency response
- k – Number of R-C branches per decade of frequency. A higher branch density gives a better approximation at the cost of increased computation time.
- $|Z|$ @ 1 kHz – Sets the vertical position of the magnitude of CPE's frequency response at a known frequency

Faradaic reactions (diodes)

If the voltage placed across the interface is kept within certain limits, the CPE and series resistance (R_S) would be all that is necessary to accurately mimic a single electrode-electrolyte interface. But once the electric potential across the interface becomes high enough, Faradaic reactions will occur at the electrode's surface. Faradaic reactions are reactions involving charge transfer, adding ionised species to the electrolyte and often producing gas. Gas, or any new species, is disastrous in an implanted setting as this causes damage to the implant's host. Possible Faradaic reactions between saline and platinum

electrodes are:

$$Pt + H_2O \Leftrightarrow PtO + 2H^+ + 2e^- \tag{7.5}$$

$$PtO + H_2O \Leftrightarrow PtO_2 + 2H^+ + 2e^- \tag{7.6}$$

$$Pt + H^+ + e^- \Leftrightarrow Pt - H \tag{7.7}$$

$$Pt + H_2O + e^- \Leftrightarrow Pt - H + OH^- \tag{7.8}$$

The electrical current density through an electrode as a function of electrode overpotential and the cathodic and anodic reactions occurring at each electrode is given by:

$$i_{net} = i_0 \left\{ \frac{[O]_{(0,t)}}{[O]_\infty} e^{-\alpha_c n f \eta} - \frac{[R]_{(0,t)}}{[R]_\infty} e^{(1-\alpha_c) n f \eta} \right\} \tag{7.9}$$

This is the current-overpotential equation and is derived from the more general Butler-Volmer equation [72, 73]. In eq. (7.9), i_{net} is the net Faradaic current across the electrode-electrolyte interface, i_0 is the exchange current density, $[O]_{(0,t)}$ and $[R]_{(0,t)}$ are the oxidant and reductant concentrations at the electrode surface as a function of time, $[O]_\infty$ and $[R]_\infty$ are concentrations of reactant in the bulk electrolyte, α_c is the cathodic transfer coefficient (approximately 0.5), n is the number of moles of electrons per mole of reactant oxidised, f is Faraday's constant divided by the product of the gas constant and the absolute temperature (F/RT), and η is the electrode overpotential. This equation describes the forward and reverse electrical current through an electrode by separating the forward and reverse reactions: oxidisation and reduction. Taking a single half of the equation, either the reduction or oxidisation, yields an equation that is similar to that for the current through a diode. This observation was made by McAdams and utilised in the Scott-Single model [74]. The standard Ebers-Moll equation of a diode equation is:

$$I = i_0 \left(e^{V_D/nV_T} - 1 \right) \approx i_0 e^{V_D/nV_T} \tag{7.10}$$

where:

- I is the current through the diode,
- i_0 is the diode saturation current,
- V_D is the potential across the diode,
- n is the diode's ideality factor, and

- V_T is the thermal voltage (defined as the product of Boltzmann's constant and temperature divided by the charge on an electron).

The two parameters needed to describe the behaviour of the diode are i_0 and n, which will later be determined for the diodes in the model. The diodes themselves can not account for the relative abundance of reactants for the redox reactions ($[O]_{(0,t)}/[O]_\infty$ and $[R]_{(0,t)}/[R]_\infty$ from eq. (7.9)), this needs to be considered separately and is discussed next.

Chemical species depletion (memristors)

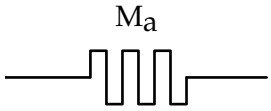


Figure 7.9: Electrical symbol of a memristor, as is used in the original electrode-electrolyte interface model

A memristor is a two port device that sets its resistance based on its own history. The resistance can either depend on the integral over time of the voltage placed across it or the total charge passed through the device [75]. Its name is a portmanteau of the word 'memory' and 'resistor' owing to its use of memory to set its resistance [76].

The memristive device models species depletion in the electrolyte as an increase in resistance in the diode/memristor branch in proportion to the integral of charge passed through the branch. As the specific Faradaic reaction proceeds, it consumes the reactants from the electrolyte bulk until eventually none is left. Increasing the resistance in series with a conducting diode has the effect of removing that diodes current path from the circuit, simulating the depletion of reactants of the modelled reaction.

Memristors were removed from the model used in this work as they added complexity that would yield little in the way of research outcomes. The diodes are only used to model the *onset* of Faradaic conduction, which is the most relevant parameter of the Faradaic modelling. Once these reactions begin, the electrode overpotential has been pushed too far and there is little to be gained from knowing how far the reaction can be run until the reactants have been depleted. In an implanted setting it is likely that the electrolyte will circulate throughout the body, bringing new reactants to the electrode's surface over time. Species depletion is likely to be a slow process, dependent on the electrolyte volume and species concentration.

Figure 7.10 shows the interface model used throughout the remainder of this thesis. Although it is slightly different to the Scott-Single interface model,

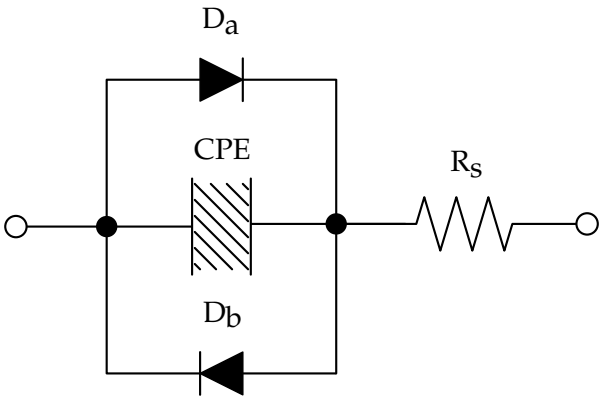


Figure 7.10: Electrical schematic of the electrode-electrolyte interface without memristors (as used throughout this thesis)

the other parameters are unaffected by the removal of these elements.

Phosphate Buffered Saline as an Electrolyte

The model has been fitted to phosphate buffered saline (PBS) because it was the closest artificial representation of human spinal fluid at the time of writing. Engineers at Saluda used a concentration one-tenth that of a standard solution of PBS as a test solution for their spinal implants. It was not understood how well the one-tenth concentration matched cerebrospinal fluid electrically, which is the main question this work sets out of answer. The ingredients used to make the stock solution of standard PBS are given in table 7.1 and the procedure for mixing up derivative solutions are:

1. Weigh out dry ingredients and combine in a large stock bottle.

2. Add 800 ml of distilled water and stir until all solids have dissolved.

3. Measure the pH and adjust to 7.4 by adding HCl.

4. Continue to add distilled water until the stock solution occupies a volume of 1000 ml.

Six bottles ranging in concentration from full strength (1.0X) to one-fortieth (0.025X) of the stock solution were created by dilution. Table 7.2 shows the volumetric ratios used to create those six concentrations of PBS, which are then used to fit the model parameters to.

Ingredient	Quantity	Unit
H_2O	1000	ml
NaCl	8.00	g
KCl	0.20	g
Na_2HPO_4	1.44	g
KH_2PO_4	0.24	g

Table 7.1: Ingredients used to produce one litre of stock solution of phosphate buffered saline.

Stock (ml)	Water (ml)	Final Concentration
700.0	0.0	1.00X
350.0	350.0	0.50X
175.0	525.0	0.25X
70.0	630.0	0.10X
35.0	665.0	0.05X
17.5	682.5	0.025X

Table 7.2: Final dilutions of stock to create six 700 ml solutions ranging from 0.025X to 1X standard PBS concentration.

Parameter Extraction Methods

The interface model has six parameters (plus five supporting parameters) that are used to set the behaviour of each component in the model. Finding suitable values for each parameter is essential to ensure the final model is a good representation of the system it mimics. Critical to finding those suitable values are the methods used to extract measurement data relating to different parts of the interface model. A divide-and-conquer approach is taken wherever possible so that parameter measurements for specific elements are isolated from other elements in the model. The following section describes those measurements and explains how they isolate the properties of each component.

Scott and Single found a set of parameters that described the Octrode in a one-tenth concentration solution of phosphate buffered saline (PBS). In this section I create six different solutions of PBS ranging in concentration from 1.0x to 0.025x the concentration of a standard PBS solution. I extend the Scott-Single model to work over a range of concentrations by expressing relevant parameters in terms of a dependent variable - PBS concentration.

Inter-electrode resistivity

In order to find the parameters of any elements within the electrode-electrolyte interface it is first necessary to find the inter-electrode resistances. The reason

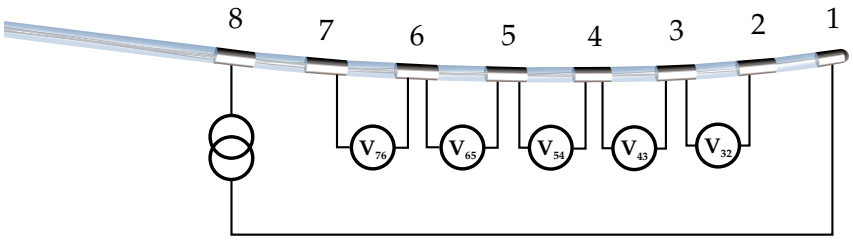


Figure 7.11: Schematic of trans-impedance measurements where electrodes eight and one are driven and the remainder are used in voltage differential measurements.

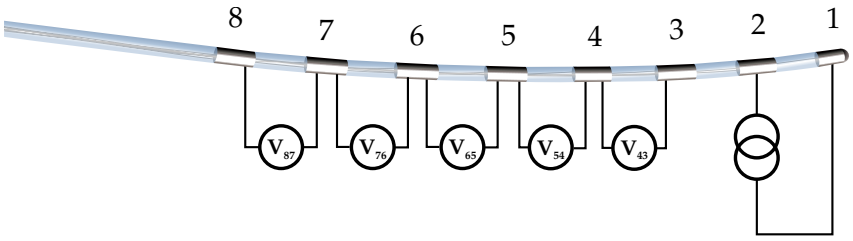


Figure 7.12: Schematic of trans-impedance measurements where electrodes two and one are driven and the remainder are used in voltage differential measurements.

is that no element within the interface can be measured without a resistive contribution being added by the electrolyte itself. To model the restive contribution from the electrolyte bulk, a resistor mesh is created that connects each electrode together. To determine the resistances used in that resistor mesh, I use the same method as was used by Scott and Single [42]. Once those resistances are accounted for, the behaviour of components in the interface can be calculated from measurements that include those inter-electrode resistances by subtracting out that expected contribution.

Scott and Single measure the trans-impedance of the eight electrodes submerged in the electrolyte solution. These transimpedance measurements pass a stimulus current between two electrodes while measuring voltage differentials between pars of non-stimulus carrying electrodes. The the current driven through a stimulus pair and the resulting voltage across the measured is turned into an impedance by division. The measurements are tabulated and used as a reference for the optimisation functions error function. Table 7.3 gives an example of what these tabulated measurements look like (taken from the sheep measurements presented later).

By measuring the voltage across pairs of non-driven electrodes using a suitably high impedance measurement, those measurements will correspond to the voltage difference in the electrolyte. For this method to work it is assumed that the current passing through each non-driven interface is zero, and therefore no voltage is dropped between the electrode's metal and the

Stimulus pair	Measured pair	Z (magnitude)	Z (phase)
1,2	3,4	51.4 Ω	175°
1,2	4,5	10.0 Ω	−167°
1,2	5,6	7.33 Ω	−164°
1,2	6,7	5.08 Ω	−163°
1,2	7,8	4.30 Ω	−165°
7,8	1,2	2.40 Ω	15.4°
7,8	2,3	2.98 Ω	18.3°
7,8	3,4	4.29 Ω	14.3°
7,8	4,5	10.9 Ω	8.97°
7,8	5,6	38.2 Ω	−2.64°

Table 7.3: Example of tabulated measured transimpedance results. Such values would then be fed into an optimisation routine to find appropriate values of resistance for the mesh.

Parameter	Determined from:
R_{eri}	optimised fit via SPICE simulation
R_{sri}	optimised fit via SPICE simulation
R_{li}	optimised fit via SPICE simulation
Padding	previous value of 3 rows used (from Scott & Single)
Depth	previous value of 5 layers used (from Scott & Single)

Table 7.4: Parameters determined by an optimised fit between a simulated mesh and transimpedance measurements.

electrolyte solution. It is for this reason that voltage differentials can not be measured on any of the stimulus electrodes. Figures 7.11 and 7.12 show the two measurement configurations used to collect trans-impedance data. Those trans-impedance results are recreated using a simulated mesh of resistors with fitted values to the three resistor parameters. Optimisation routines can find the three values which produce the mesh with closest match to the measured values. The source of parameter values for the mesh are given in table 7.4. A mesh with those values of padding and depth, and made to fit between eight electrodes, contains 205 resistors.

Constant phase element & series resistance

By accounting for the value of resistance seen between electrodes it is now possible to probe deeper into the interface model. Calculation of both the CPE and the series resistance (R_S) is made via impedance spectroscopy methods. It is possible to use frequency to separate the responses of the CPE from interface's series resistance. The impedance of the CPE dominates below 10 Hz,

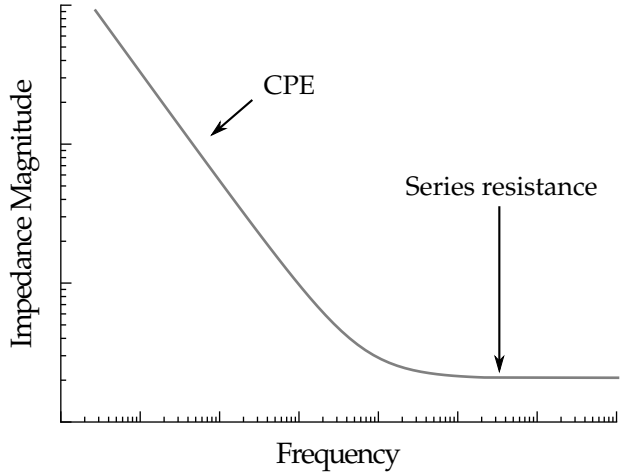


Figure 7.13: Schematic log-log plot of frequency vs impedance magnitude of a single interface and inter-electrode impedance. The response of the CPE and that of the total series resistance is separated in the frequency domain.

Parameter	Determined from:
k	previous value of 3 branches used (from Scott & Single)
m	determined from the slope of |Z| vs. frequency response
|Z| @ 1 Hz	impedance magnitude at 1 Hz
R_S	impedance at high frequency (10 kHz) end of the trace

Table 7.5: Parameters describing CPE behaviour and interface series resistance (R_S) as determined using impedance spectroscopy based measurements of electrode-electrolyte interface.

where its slope and magnitude can be determined. At higher frequencies, greater than 1 kHz, the series resistance of both the interface (R_S) and the previously determined inter-electrode resistance is evident. This separation of responses is illustrated in fig. 7.13. Subtracting the inter-electrode resistance, determined previously, from the measured resistance yields the interface's series resistance (R_S). Parameters for the CPE, such as slope and vertical position, are determined from the low frequency part of the trace where the slope is not disturbed by resistive behaviour. The parameter m determines the slope of the created CPE, but is not itself a direct measure of slope, i.e., it is used to pick values of resistance and capacitance in each branch in the CPE, which then determine the resulting slope. Parameters for the CPE and series resistance are summarised in table 7.5.

Faradaic current

Measurement of electrical currents associated with Faradaic reactions requires increasing the electrode overpotential until reactions at the electrode's surface

Parameter	Description
i_0	optimised fit of threshold voltage to measured curve
n	optimised fit of growth rate to measured curve

Table 7.6: Parameters determined from fitting diode parameters to measured response of Faradaic current.

begin. Those currents then increase exponentially with increasing electrode overpotential. Scott and Single used a triangular voltage stimulus as a means to identify the onset of Faradaic conduction at the interface. The triangular wave is equivalent to a constant ramp-up and ramp-down of voltage placed across the interface. Current flowing into a capacitor is given by:

$$I(t) = C \times \frac{dV(t)}{dt} \tag{7.11}$$

When $\frac{dV(t)}{dt}$ is a constant, as is the case for a linear ramp voltage stimulus, the current is also constant. We make the assumption here that fractional differentiation of eq. (7.11), as in the case of a CPE, will give the same relationship between voltage and current. By slowly ramping the electrode overpotential the current draw should be constant up to a point after which it becomes exponential. The voltage corresponding to the point at which the current draw becomes exponential will be used to determine the onset of Faradaic conduction. That point, together with the rate of growth, would then be used to fit values for i_0 and n. The parameters i_0 and n are the diode's saturation current and ideality factor respectively, as summarised in table 7.6. Problems arose with those measurements, discussed in the next chapter, which showed the behaviour of the CPE was less predictable than expected.

Optimisation

Optimisation routines are used to determine the appropriate parameter values from measurements of the interface's various components. Each optimisation is performed by Python scripts that utilise the open-source scientific package SciPy, which contains a range of optimisation functions. The optimisation functions require the user to write a 'residuals' function that accepts parameter values and returns an error. The optimiser, when executed, then repeatedly calls the residuals function with parameter values that minimise the error returned. A illustrative and heavily simplified optimisation routine follows

```
1 import scypy.optimize
2 import numpy as np
```

```
3  import math
4  import measured_data
5  import spice_simulator
6
7  measured = measured_data.get_transimpedance()
8
9  def simulate(r_eri, r_sri, r_li):
10   # Create a netlist suitable for feeding into ngSpice
11   netlist = spice_simulator.generate_resistorMesh(r_eri, r_sri, r_li)
12
13   # Simulate the circuit
14   results = spice_simulator.simulate_circuit(netlist)
15
16   # Return the transimpedance measurement values
17   return results
18
19 def residuals(resistances):
20   # Separate out passed parameter values
21   r_eri = resistances[0]
22   r_sri = resistances[1]
23   r_li = resistances[2]
24
25   # Run values through simulator
26   simulated = simulate(r_eri, r_sri, r_li)
27   error = 0.0
28   measurements = [
29     '12-23',
30     '12-34',
31     '12-45',
32     '12-56',
33     '12-78',
34     '18-23',
35     '18-34',
36     '18-56',
37     '18-67']
38   # Square the difference between measured and simulated values
39   for measurement in measurements:
40     difference = measured[measurement] - simulated[measurement]
41     error += math.pow(difference, 2)
42
43   # Return the sum
44   return error
45
46 # Starting values for optimisation
47 initial_guess = [100, 250, 1025]
48
49 # Call the scipy to minimise the error
50 solution = scipy.optimize.leastsq(residuals,
51                                   x0=initial_guess,
52                                   epsfcn=0.5)
53 print(solution)
```

Chapter 8
Interface Parameters

Details of components in the electrode-electrolyte interface model and methods of determining its parameters have been discussed. Focus will now move to measuring and fitting suitable parameter values to the model. Model parameters are determined for various concentrations of phosphate buffered saline (PBS), and then for comparison – in a living sheep's spinal cavity. The comparison will show whether a one-tenth concentration of PBS is in-fact a good substitute for cerebrospinal fluid (CSF), which it is assumed to be by medical implant engineers.

Phosphate Buffered Saline

Scott & Single fitted parameters of their model to a one-tenth concentration (0.1X) of a standard solution of PBS. A one-tenth concentration of PBS is a commonly used solution of buffered saline [42]. I measure and fit parameters not only to the one-tenth concentration, but to six concentrations spanning 0.025X to 1X the concentration of a standard buffered saline solution. For the model parameters that change with salinity, a fit is made using regression analysis to PBS concentration. Doing so provides a model that can be used to predict the impedance response of an electrode array submerged in a wider range of saline concentrations.

Each of the PBS measurements were made in 1000 ml glass bottles containing 700 ml of the PBS solution to be measured. Measurements were made in a temperature controlled environment set at 21° Celsius. All measurements were automated by the use of Python scripts running on a GNU/Linux based workstation. The scripts communicate with the instruments both to configure measurements and collect data. Each measurement set was repeated for each of the six solutions used. The six concentration of PBS that were measured are shown in table 8.1.

Concentration
1.00X
0.5X
0.25X
0.1X
0.05X
0.025X

Table 8.1: Six PBS concentrations used to fit model parameters to.

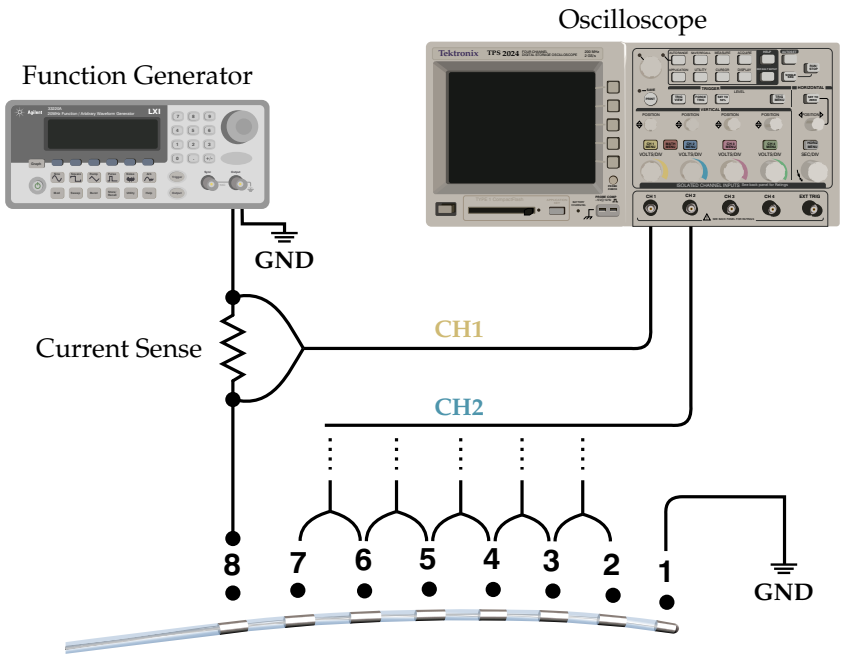


Figure 8.1: Illustration of one of two measurement configurations used to measure the electrode trans-impedances. Each of the electrode pairs were measured in sequence using the shown equipment.

Inter-electrode resistivity

With the electrode array immersed in a saline solution, a 10 kHz sinusoidal current having a peak amplitude of 500 µA was passed through the stimulus electrodes using an Agilent 33220A function generator. A shunt resistor inserted in series with the function generator allows measurement of current between electrodes. The differential voltage across a pair of non-stimulating electrodes and the voltage across the shunt resistor was measured using a Tektronix TPS 2024 oscilloscope. Figure 8.1 shows the measurement configuration used when electrodes one and eight are used as the stimulus electrodes. The second configuration has electrodes eight and seven as stimulus electrodes and the remaining electrode pairs are used to measure trans-impedance voltage differentials.

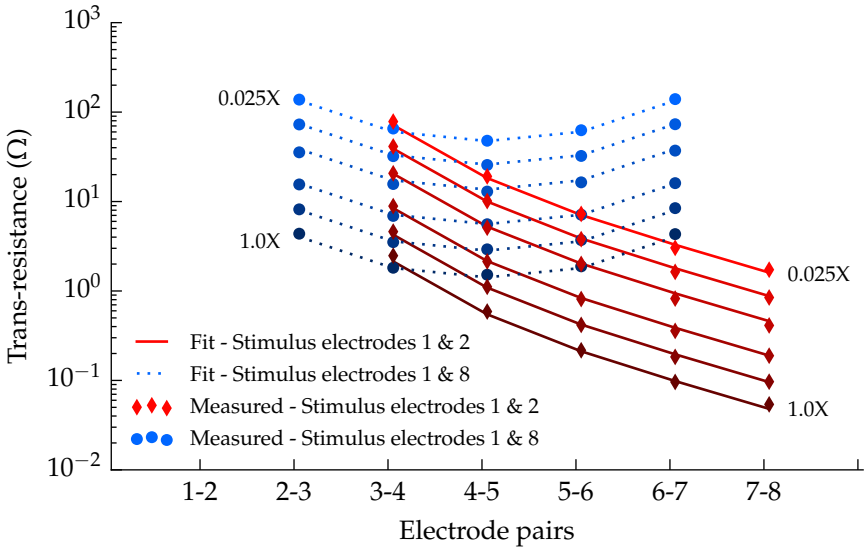


Figure 8.2: Measured and fitted values of trans-impedance for both measurement configurations. Voltage measurements are made between adjacent pairs of electrodes as current is pushed through the stimulus electrodes.

Parameter	Value
R_{eri} (Ω)	$0.407\ /\ \sigma$
R_{sri} (Ω)	$R_{eri} \cdot 3/4$
R_{li} (Ω)	$3.71\ /\ \sigma$
Depth (layers)	5
Padding (layers)	3

Table 8.2: Resistor mesh parameters for the electrode array in various concentrations of PBS. Electrolyte conductivity (σ) is expressed in units of S/cm.

The results of those measurements, in both configurations, are represented as markers in fig. 8.2. The graph serves to validate the parameter selection shown in table 8.2, where dotted lines show the result of feeding those values back through the SPICE model and markers show measured values. Red traces show trans-impedance measurements when electrodes one and two are used (adjacent tetrapolar measurements). Conversely, blue traces show trans-impedance measurements when electrodes one and eight are used (opposite tetrapolar measurements). Measurements were not made on electrodes that were used to carry stimulus. Each point was calculated by taking the voltage differential across a pair of electrodes (V_{diff}) and dividing by the stimulus current. Remember, the stimulus current was set to be around 500 µA peak, which was chosen so as to be well under the 'water window' where electrolysis occurs.

Values for R_{eri} and R_{li} were determined using a Python optimisation script

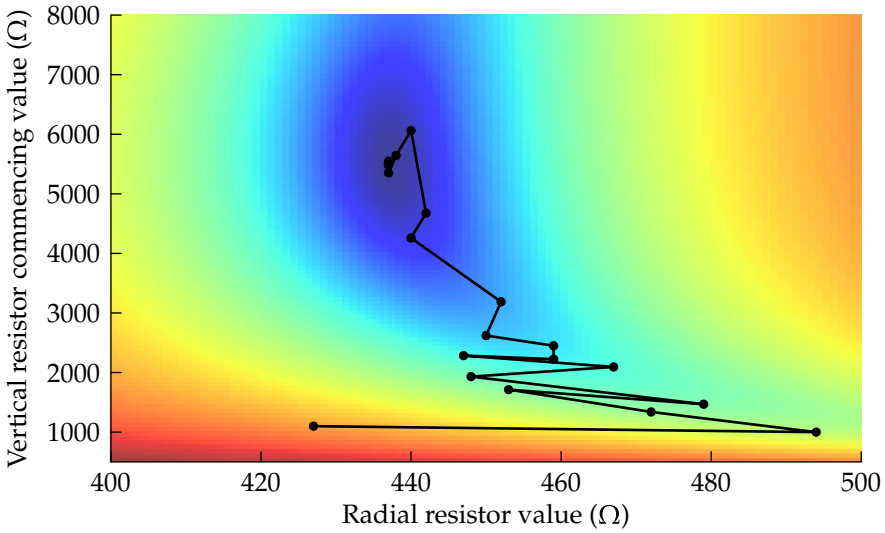


Figure 8.3: Graph showing the path taken by the optimisation script while finding resistor values that minimise error between the two resistor parameters and the measured trans-impedance results. Radial resistor refers to R_{eri} while vertical resistor refers to R_{li}. The heat-map background shows values of total error for all possible combinations of resistor values on the axes.

for each concentration of PBS. The optimisation script selects candidate values for R_{eri} and R_{li}, simulates the mesh using those values, and then calculates the equivalent trans-impedance values. The error between simulated trans-impedance values and measured values is calculated and the process repeats, selecting different values of R_{eri} and R_{li} to improve the fit. Figure 8.3 shows a progression of R_{eri} and R_{li} values chosen by the optimisation script while finding a pair that minimise error. The total error for a simulation was taken to be the sum of the squares of difference between the simulated and measured values, normalised to the measured value. The final values of R_{eri} and R_{li} that minimise the total error are shown in table 8.2. R_{sri} is a dependent variable, so is expressed in terms of R_{eri}, and the remaining parameters have been re-used from the work of Scott & Single. Figure 8.2 shows measurement results for each pair of non-stimulated pair of electrodes along with simulated results using the fitted parameters.

Constant phase element & series resistance

Measurement of both the CPE and the interface's series resistance was made using an impedance spectroscopy method. Those measurements were made by passing a sinusoidal current between electrodes two and seven of the electrode array. Use of the end electrodes (one and eight) was avoided as a

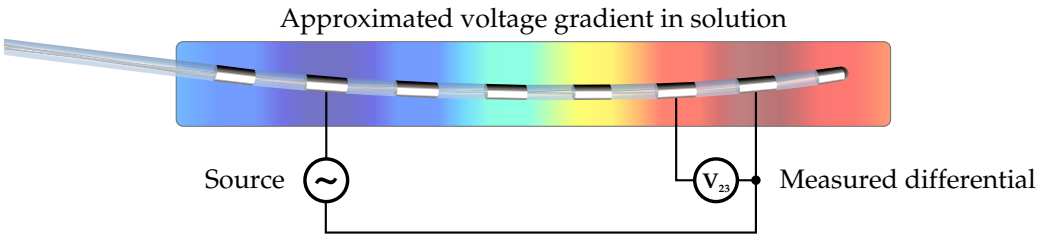


Figure 8.4: Illustrated voltage gradient in electrolyte solution at each electrode's surface when potential is applied across electrodes two and seven. Measurement of electrolyte voltage taken between electrodes 2 and 3.

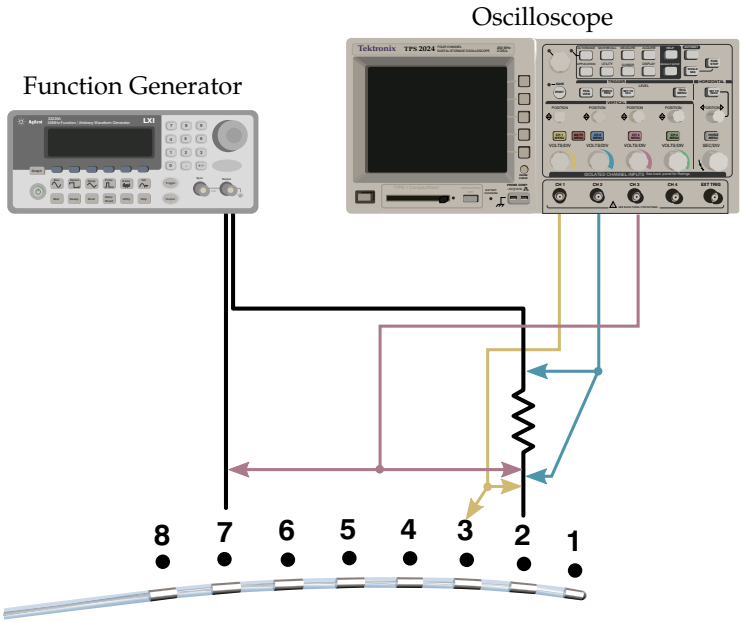


Figure 8.5: Diagram showing the measurement configuration used to measure the CPE response and interface series resistance.

precaution to reduce end effects resulting from the electrode's geometry. The sinusoidal voltage at the liquid side of the interface was taken as the voltage that appears at an adjacent electrode (electrode three) when a suitably high impedance measurement is made, this is illustrated in fig. 8.4. This measurement relies on the ability to make high impedance voltage measurements to minimise voltage drop across the electrode interface, for which the Tektronix TPS 2024 four channel oscilloscope was used again. This oscilloscope has floating channels, each having an input resistance of 10 MΩ when using 10X probes. The Agilent 33220A function generator was used again to generate the stimulus waveforms applied between electrodes two and eight. A current sense resistance of 10 kΩ was inserted in series with the waveform generator's output and was measured by the oscilloscope. By measuring the current through electrode two and the voltage across the interface (measured between

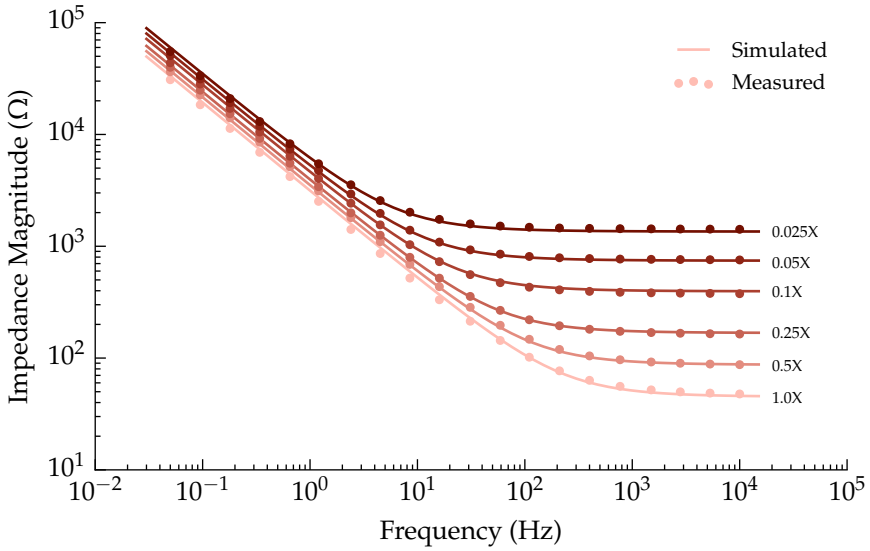


Figure 8.6: Impedance magnitude of both the measured interface response and the fitted response at each of the six concentrations of PBS.

electrodes two and three), the impedance of the interface is calculated. A diagram of the measurement setup is shown as fig. 8.5. For each of the six solutions, twenty frequencies (log-spaced) were sampled between 50 mHz and 10 kHz for the impedance measurements. At each frequency the stimulus waveform amplitude was re-adjusted to be 300 mV-peak as the interface's impedance changed.

The oscilloscope has an input resistance of 10 MΩ and capacitance of 12 pF. At low frequencies the interface impedance is high, in the order of 10 kΩ, and circuit loading from the oscilloscope's probes will affect measurement results. As there is still three orders of magnitude between the maximum resistance of the system and the input impedance of the measurement device, this effect is deemed negligible. The maximum frequency used to measure the interface is 10 kHz, for which the oscilloscope's capacitance will also have negligible effect.

Figures 8.6 and 8.7 show the calculated impedance magnitude and phase from measurements as markers and simulation results from fitted parameters as traces. Figure 8.8 shows the SPICE model used to simulate parameter values for the CPE and R_s. Final values were found by minimising the difference between the simulated response and the measured response using a Python script. For each set of parameter values in the optimisation, the script builds a SPICE circuit based on the values, simulates the circuit, calculates the interface impedance and compares the values to the measured results. The process is automated and runs until a minimum error between simulated and measured

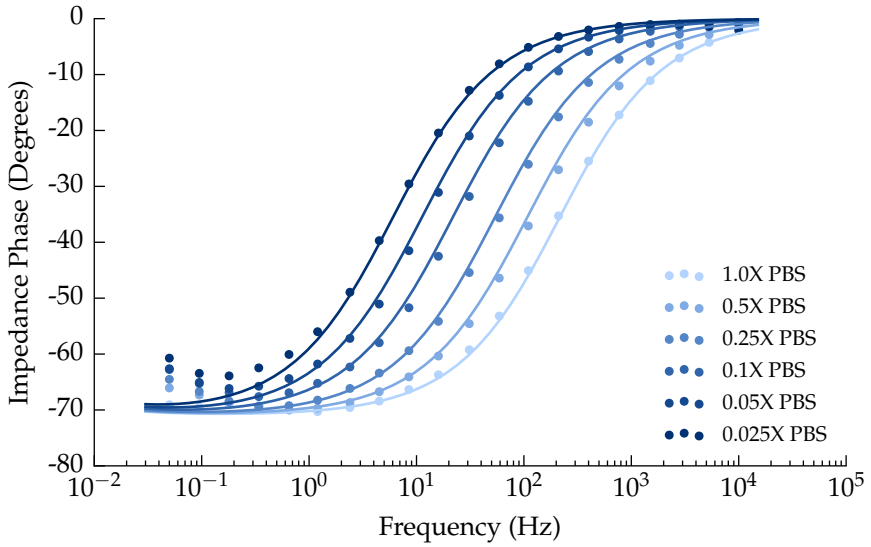


Figure 8.7: Impedance phase of both the measured interface response and the fitted response at each of the six concentrations of PBS.

results is found. Once found, the script exits and displays the final values of each parameter.

After parameter values are found for each concentration of PBS, another optimisation is made to fit relevant parameters to PBS concentration. The parameters that scale with concentration are the series resistance (R_S) and the CPE's impedance magnitude at 50 mHz. The final fit expresses these parameters as functions dependent on PBS concentration (equivalent to salinity). Individual parameter values for each concentration, along with the resulting fit, is shown in fig. 8.9. Measurement of the vertical position of the CPE's impedance magnitude trace was made at 50 mHz as opposed to the parameter's defined value of 1 Hz. This was done to avoid any effect from the series resistance interfering with the measured value. As the slope of the CPE is always the same (even between concentrations), the value can be easily converted back to the equivalent value at 1 Hz. Measured resistances at high frequencies include the inter-electrode resistance (R_{23}) which has been included in the plot, but is later subtracted to leave only R_S. The final parameters for the CPE and R_S are given in table 8.3.

Faradaic current

Using the same oscilloscope and function generator as the previous measurement, the oscilloscope was set to measure voltage between electrodes two and seven and the current through the current sense resistor (fig. 8.10). The func-

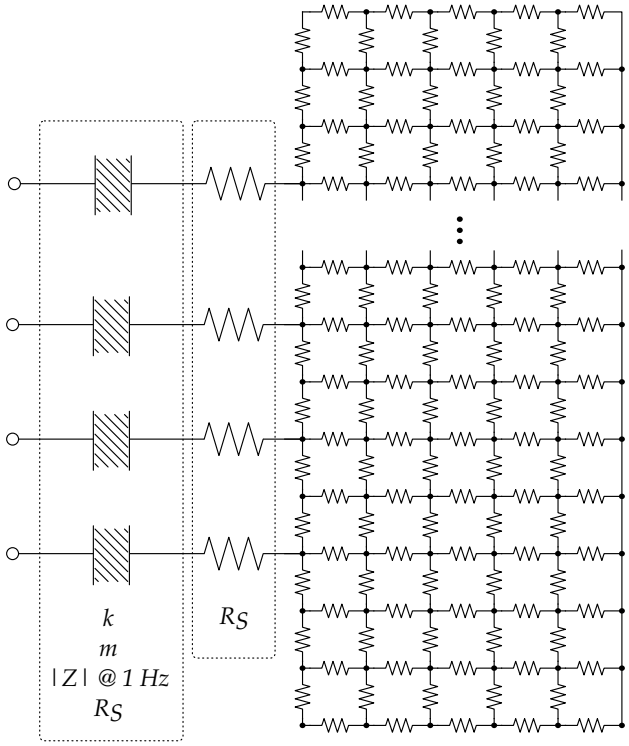


Figure 8.8: The SPICE model schematic used to find optimum values for parameters of the CPE and interface series resistance. Parameters for the resistor mesh are those determined previously.

Parameter	Value
m	1.34
k	1.773
|Z| @ 1 Hz (Ω)	$3284 \times concentration^{-0.158}$
R_S (Ω)	$13.38 \times concentration^{-0.8397}$

Table 8.3: CPE and R_s parameters. Concentration is relative to the stock solution of phosphate buffered saline.

tion generator was set to produce a triangle wave stimulus, or linear ramp, also between electrodes two and seven. Electrical current associated with Faradaic reactions rises exponentially after a threshold electrode overpotential. The point at which the electrical current draw begins to move exponentially with increasing voltage represents the onset of the associated reaction.

Figure 8.11 shows measured data where the Faradaic response is evident for each concentration. The repeatability of these measurements was low although care was taken to recreate the same conditions for each run. To try and improve the repeatability the following was tried:

- Maintaining a constant ambient temperature

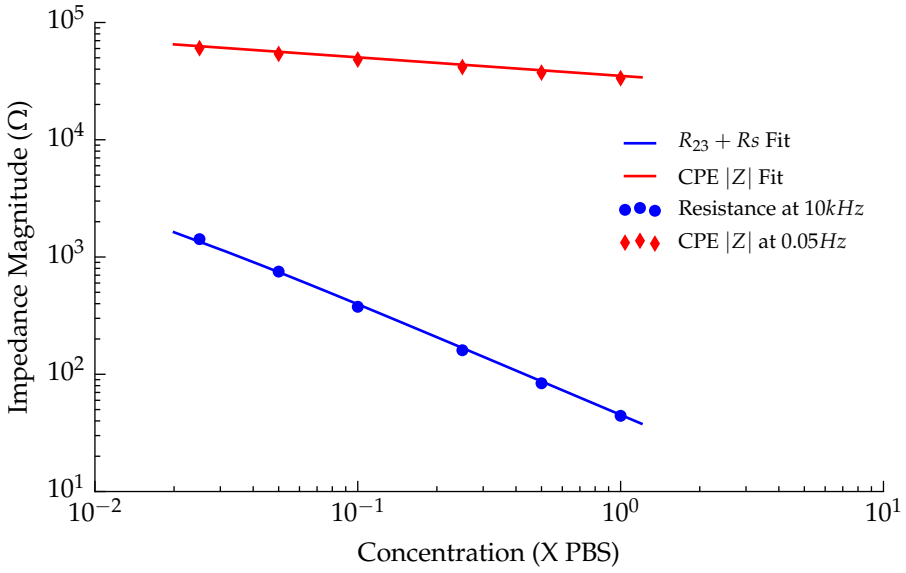


Figure 8.9: Plot showing fitted parameter values for the CPE impedance magnitude at 50 mHz and series resistance at each of the six concentrations of PBS (shown as markers). The solid trace shows the resulting fit between those values as a function of concentration.

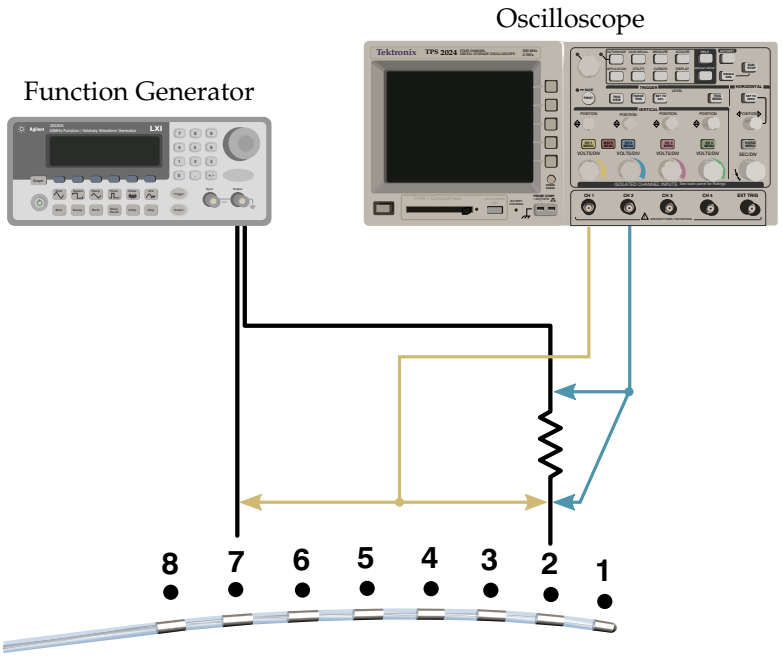


Figure 8.10: Illustration of the cyclic voltammetry measurement configuration used to measure the response of the interface when driven into Faradaic conduction mode.

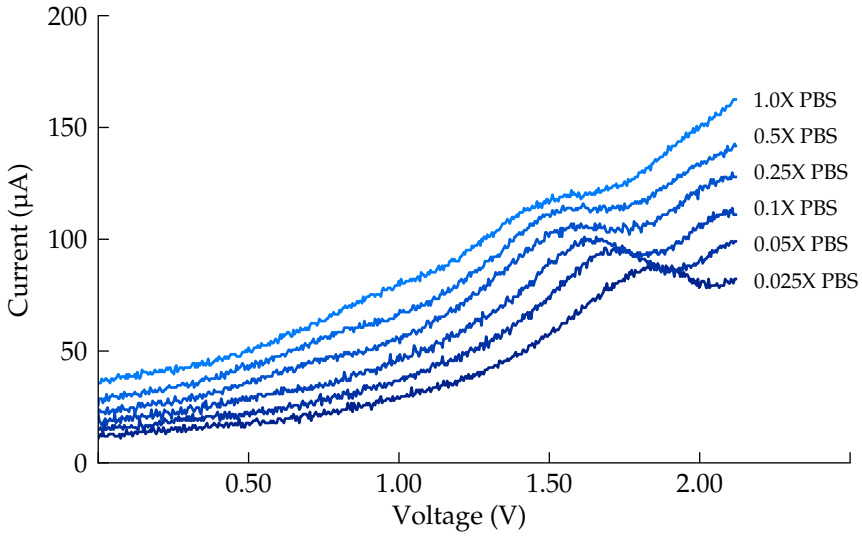


Figure 8.11: Graph showing measured Faradaic response of each concentration of PBS to a linearly increasing voltage between electrodes two and seven.

- Cleaning the electrodes between each measurement using isopropyl alcohol
- Keeping the electrolyte moving at a constant velocity using a motorised stirrer
- Allowing the system to settle for periods of two hours between measurements

These steps did reduce variation, but by no means removed it. Sweeping the voltage at $0.12\,\mathrm{V\,s^{-1}}$ was slow enough that results did not appear to be too distorted but fast enough that a measurement run could be completed quickly. Completing measurements quickly seemed important at the time as it was often the case that an artefact would show up during a measurement run, which initially appeared to have no obvious cause, and affect the remainder of the experiments. These artefacts would manifest themselves sometimes as a peak at a certain voltage, otherwise as distortions to the current/voltage trace. A key insight was realising that after the voltage across a pair of electrodes had been pushed into Faradaic region they then began to behave differently, even after being returned to lower stimulus voltages. In fig. 8.11 it is clear that each concentration has a different Faradaic response. This means that when the maximum voltage is applied to each of the solutions that the highest concentration is driven further into its Faradaic region than the rest. That in turn would create an artefact that would appear on the remaining traces

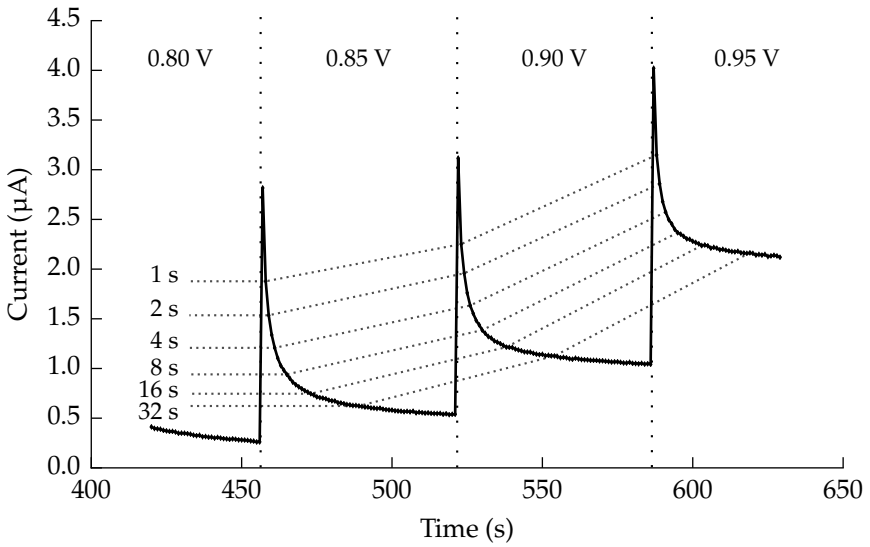


Figure 8.12: Graph showing the measured response of a pair of interfaces to multiple step responses. Vertical dotted lines indicate when in time the step occurred. Dotted traces show points in time after each response. Measurements are between electrodes two and seven on the Octrode submerged in 1X PBS.

(those of lower concentration), that would not have otherwise been there. The issue of artefact and dependence on sweep rate led me to find other ways of measuring Faradaic currents.

Step based Faradaic measurements

A revealing measurement came from the use of the Agilent E5270B precision measurement mainframe, the same instrument used to measure the streaming potential cells (part I). By increasing the voltage between the electrodes in discrete steps and recording the current over time it became clear that the CPE was having a large effect on the Faradaic measurements. Figure 8.12 shows three transitions in steps of 50 mV occurring 64 s apart. Dotted traces link measurements made a set time after each transition, with the delay times indicated to the left of the graph. So for example, the top trace represents data that would be obtained if the settling time after each step was one second. It shows that the longer the settling time is, the lower the measured current - even though the response is the same. This graph shows the effect the CPE is having on measurement results, as well as the duration of time necessary for the transient response to settle in most instances.

Subsequent measurements of CPE settling time show that a delay of 64 s between steps is adequate to allow the CPE voltage to settle. Those measurements are shown as fig. 8.13, with the 64 s window highlighted in

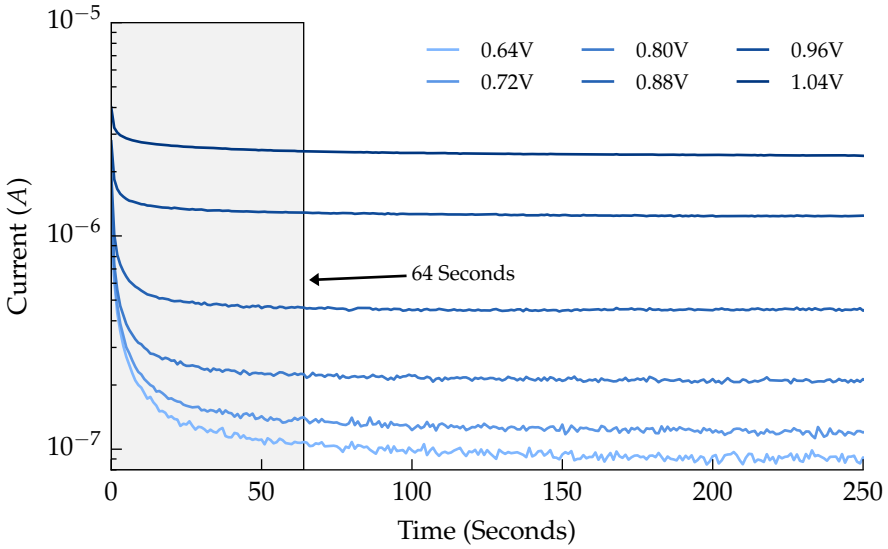


Figure 8.13: Graph showing CPE discharge curve after a step transition between each of voltage trace in increasing order. Measurements are between electrodes two and seven on the Octrode submerged in 1X PBS. A delay of 10 000 seconds elapsed between each step.

grey. The dependence of the capacitance upon voltage or current is clearly visible by comparing the 0.64 V trace to that of the 1.04 V. This variation is most likely the change in capacitance (and series resistance) that Schwan published in 1968 [41].

Figure 8.14 shows measurements of four concentrations of PBS overliad on top of one another. This graph reveals that not only does the capacitance vary with applied voltage, as was shown in fig. 8.13, but also with concentration of PBS. A consequence of this is that not waiting long enough to sample the current gives the impression that a higher concentration of PBS results in larger Faradaic currents. This is shown by the dotted trace that is sampled 10 s after each step, which I believe is representative of results obtained using cyclic voltammetry methods. Importantly – the settled current draw for each concentration is the same. Any separation between concentrations at the sixty-four second mark for each step appear to be unordered.

Successful measurement of Faradaic current

Figure 8.15 shows the collected measurements of the electrical current due to Faradaic reactions using the stepped measurement method. Spread in the measurements at low voltages is due to noise in the measurement samples. There are three important observations that can be made from this graph:

1. Saline concentration has no measurable impact upon the Faradaic reac-

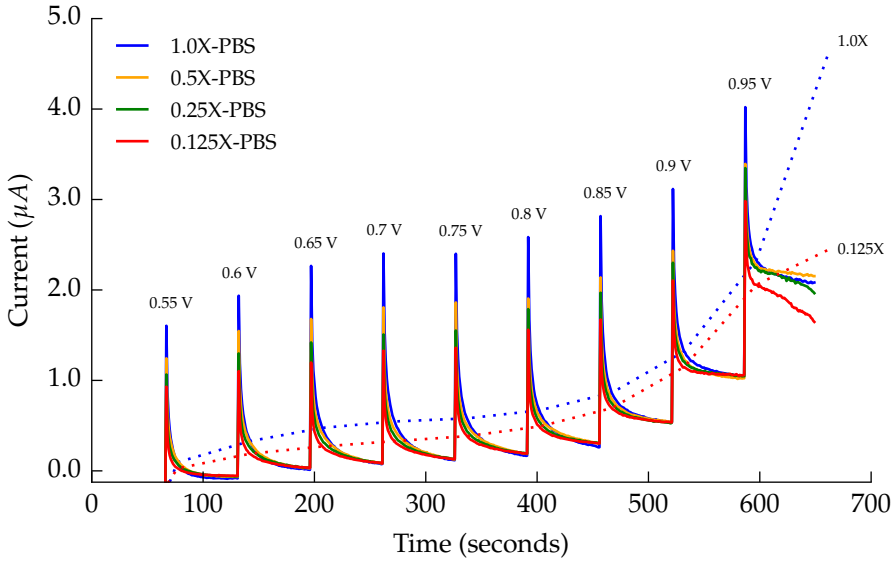


Figure 8.14: Graph showing measurements of four concentrations of PBS as each is stepped from 0.55 V to 0.95 V. Measurements are between electrodes two and seven on the Octrode. A delay of 64 seconds elapsed between each step. Dotted traces connect current measurements taken 10 s after each step.

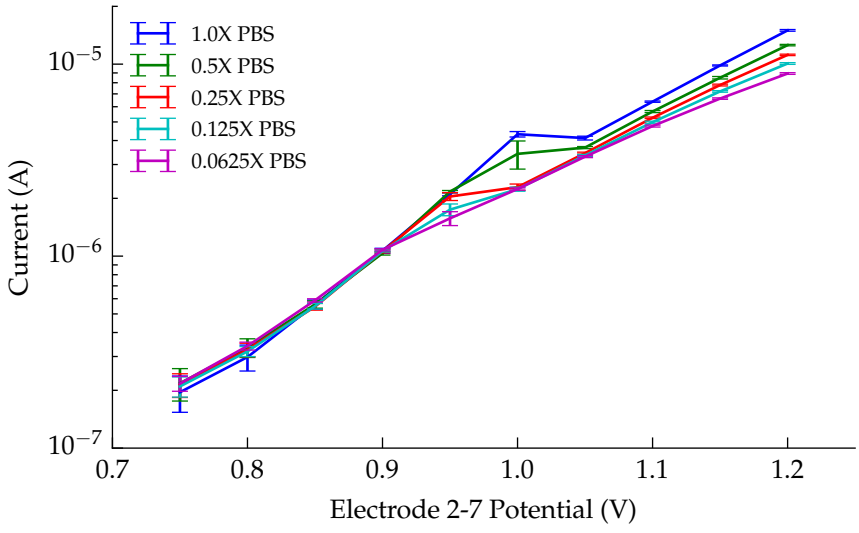


Figure 8.15: Graph showing the electrical current draw associated with Faradaic reactions versus applied electrode overpotential. Measurements used the stepped method with a wait time of 64 s between transitions. Vertical bars mark the standard deviation of the final forty measurements before the following step.

tion rate when the applied overpotential between a pair of electrodes is less than 0.9 V.

2. Faradaic current draw is directly related to saline concentration above 1.05 V.

3. Between 0.9 V and 1.05 V of overpotential (between electrodes), each trace transitions to a mode of saline concentration dependence. These transitions happen in order of increasing saline concentration.

It appears that the change in behaviour between 0.9 V and 1.05 V is due to a transition to diffusion-controlled conduction between electrodes. I hypothesise that below 0.9 V the charging of the CPE draws available ions to the electrode, creating a layer of high ionic concentration at the surface irrespective of that of the solution bulk. It is this layer that is consumed by the Faradaic reactions at a rate that increases exponentially with electrode overpotential. The effect of the bulk solution concentration while this layer exists is negligible until the point at which the layer is consumed faster than it can be replenished. At this point, and with increasing overpotential, Faradaic conduction is governed by diffusion of ions from the solution bulk into that layer. The rate at which those new ions diffuse into the layer is a function of the concentration, or abundance of ions available in the bulk. This explains the divergence of conduction with concentration between 0.9 V and 1.05 V and why there is no observable dependence on the bulk ion concentration beforehand. As Faradaic reactions are dangerous in an implanted setting, and therefore to be avoided, interest in Faradaic reactions lies in determining their onset. For the purpose of our model, it is sufficient to place a 0.9 V limit across a pair of electrodes and proceed on the basis that Faradaic conduction is not affected by the saline concentration.

Using the "step and wait" method to measure electrical currents associated with Faradaic conduction gave improved results, both in repeatability and expected response. Figures 8.16 and 8.17 show results using the 1.0X PBS solution, with other concentrations following the same pattern. In fig. 8.16 it can be seen that the simulated CPE does not follow the decay curve of the interface after each transition. Notice again how the capacitance is dependent on the electrode overpotential, but the CPE fails to capture that information. Final parameter values for the Faradaic currents (the diodes of the model) are given in table 8.4.

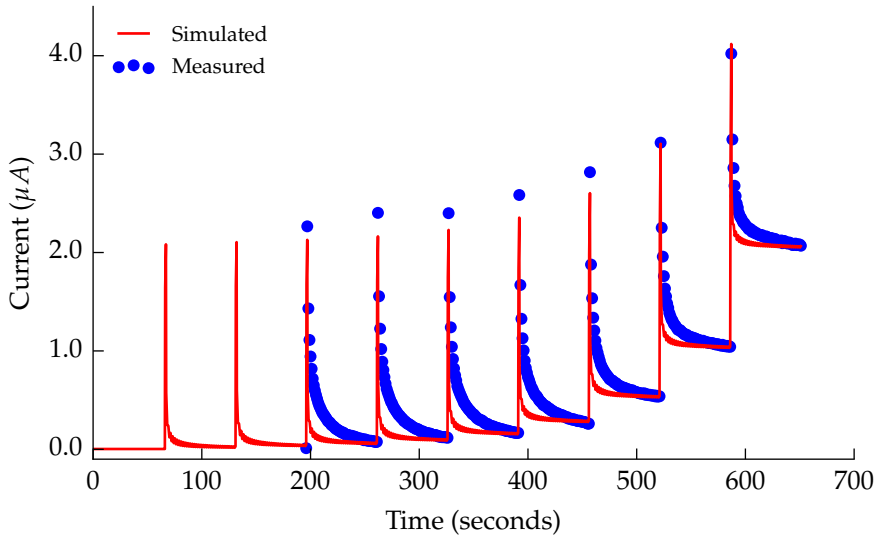


Figure 8.16: Graph comparing measured Faradaic response of a pair of interfaces (1.0X PBS) to the simulated response using fitted parameter values for i_0 and n. Each spike is a step in electrode overpotential, with the steps shown in the following graph.

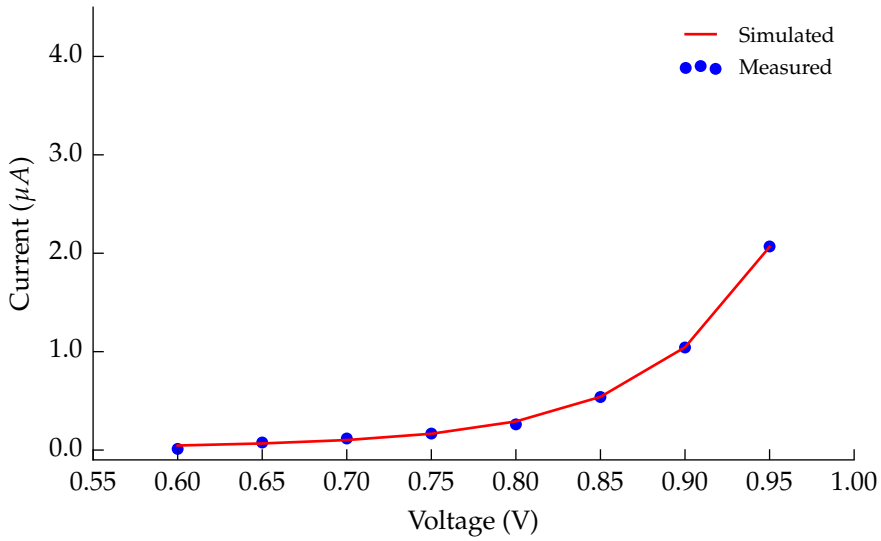


Figure 8.17: Graph comparing the measured settled electrical currents of Faradaic reactions (in 1.0X PBS) for a pair of interfaces to simulated final values using the fitted parameter values for i_0 and n.

Table 8.4: Faradaic parameters

Parameter	Value
$i0$	$2.757\,pA$
n	1.36

Table 8.5: Determined interface parameters for the St. Jude Medical Octrode in phosphate buffered saline. The parameter *concentration* refers to the dilution of PBS, e.g., 0.025 for a one-fortieth dilution to 1.0 for the stock PBS mixture.

Parameter	Value
$R_S\ (\Omega)$	$13.38 \times concentration^{-0.8397}$
m	1.34
k	1.773
$\lvert Z \rvert$ @ 1 Hz (Ω)	$3284 \times concentration^{-0.158}$
$i0$	$2.757\ pA$
n	1.36

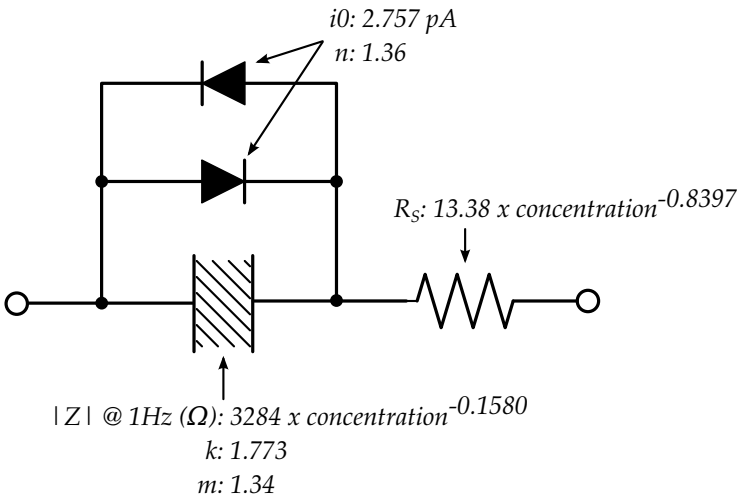


Figure 8.18: Schematic of the electrode-electrolyte interface including parameter values for platinum and buffered saline.

Final model

Parameter values for each of the model's components have been found. Collecting the parameters that describe the interface's impedance results in table 8.5. This table excludes the parameters of the resistor network as they do not describe the interface itself.

Epidural Insertion into Live Sheep

The previous section dealt with measuring and fitting numerical values to the electrode-interface parameters in various solutions of buffered saline. Phosphate buffered saline, specifically a 0.1X concentration of a standard solution, was used for electrode characterisation as it was believed to be a good substitute for cerebrospinal fluid. Electronic engineers at Saluda

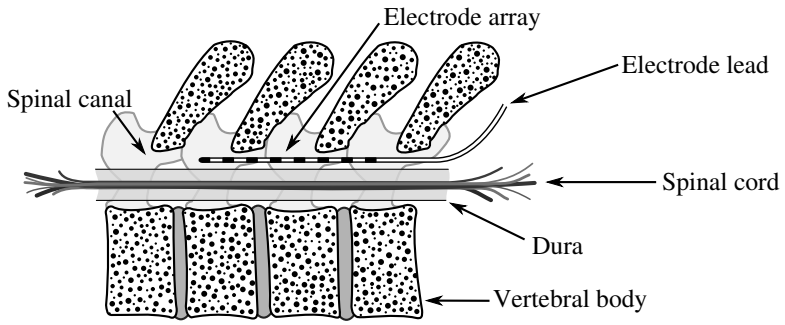


Figure 8.19: Diagram showing the positioning of the St. Jude Medical Octrode electrode array inside the sheep spinal cavity. The view is a cross-section of the spine with the dorsal side at the top.

Medical used these 0.1X PBS solutions to test their implant devices to make sure they were capable of driving and handling the impedance presented by the electrode-electrolyte interface and spinal cavity. Not knowing how closely the saline solutions resembled live biological spinal fluid they also tested their implants and electrodes in living sheep. A sheep's spinal canal is smaller than a human's, but large enough to insert an epidural electrode array, making them a relatively accessible means of in-vivo testing for medical applications. Geometrically they are similar enough for the sheep to be used as a test substitute for human spinal implant testing. Measurement in a live sheep's spinal canal still requires a lot of resources such as a surgeon, access to an operating theatre, equipment suitable for use in an operating theatre, ethical approval, and time. When experimenting with sheep, the sheep would be anaesthetised and kept alive for the duration of the experiments, which often last over twelve hours. A veterinary surgeon would prepare and monitor the sheep constantly during the experiments to ensure that it was fully anaesthetised and then euthanise the sheep at the end of testing.

These tests offered an opportunity for me to characterise the electrode-electrolyte interface inside a living mammal. This section repeats the measurements and parameter value determination of the previous section but this time inside a living sheep's spinal cavity. The same electrode as was used in the previous measurements (St. Jude Medical Octrode) was inserted into the spinal cavity of the sheep (just outside the dura) for each experiment, as shown in fig. 8.19.

Sheep used to gather experimental data here were provided by the Keams Facility at the Royal North Shore Hospital of Sydney under the Animal Care and Ethics Committee approval. Two sheep were used to gather measurement data, but inadequate electrode placement and a loss of spinal fluid from previous experiments meant that useful data was only obtained from the

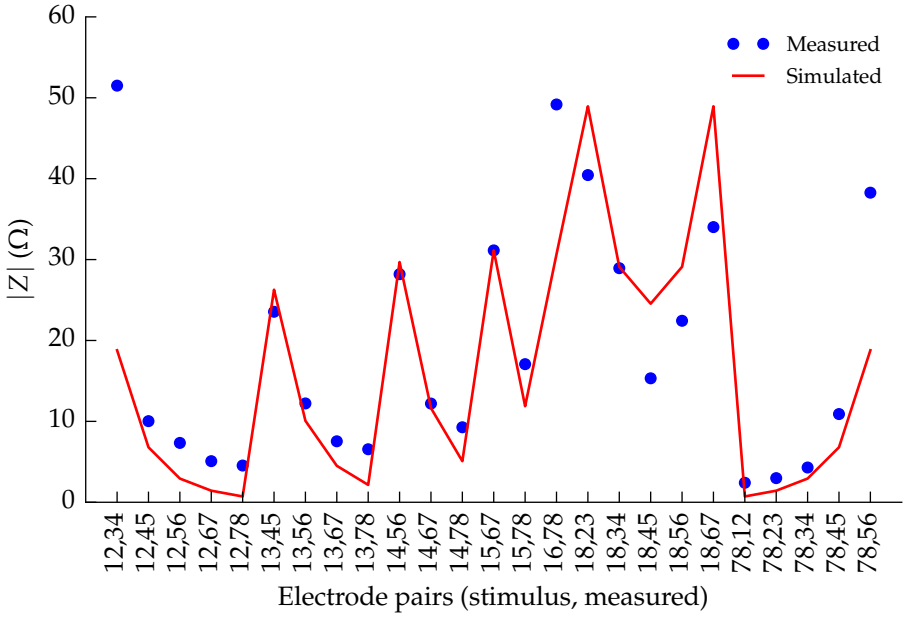


Figure 8.20: Graph showing measured and simulated trans-impedance magnitudes for twenty five combinations of stimulus-measure pairs of electrodes.

second sheep. These experiments complied with the Australian Code of Practice for the Care and Use of Animals for Scientific Purposes. In each case the sheep were injected with alfaxalone to induce anaesthesia and were then intubated and ventilated with an oxygen-air mixture containing isoflurane. During the course of experimentation the animals were monitored using electrocardiogram, arterial blood pressure, arterial saturation, and end-tidal (exhaled) carbon dioxide levels. All ethical considerations and procedures around animal testing were handled by Saluda Medical. Unless otherwise stated, measurement procedures and the equipment used in the hospital are the same as those used to measure the electrode-electrolyte response in PBS.

Inter-electrode resistivity

Trans-impedance measurements were the first to be made once the electrode array was inserted into the sheep's spinal canal. These measurements were more extensive than those made in saline as additional stimulus electrode pairs were used. The extra measurements were made with the hope that they may capture more information regarding the impedance structure of the surrounding spine geometry; mostly the bone.

Figures 8.20 and 8.21 show both the measured and simulated results for the impedance magnitude and phase response respectively. The magnitude measurements show that when stimulating between electrodes one and eight

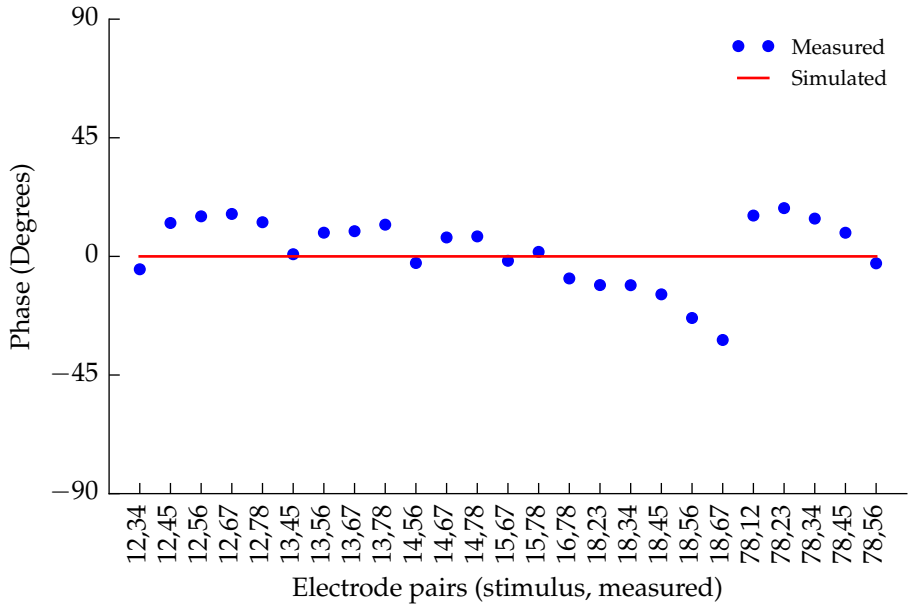


Figure 8.21: Graph showing measured and simulated trans-impedance phase for twenty five combinations of stimulus-measure pairs of electrodes.

and measuring on electrodes two and three that the impedance is approximately that of the one-tenth concentration of PBS (compared with results from fig. 8.2). This result appears to support the idea that a one-tenth concentration of a standard buffered saline solution is a good substitute for a spinal cavity. For the case where the stimulus is placed between electrode one and two and the impedance is measured between electrodes seven and eight, the comparison suggests a lower concentration of PBS than $1/10^{th}$. Swapping the stimulus and measure electrodes around gave different trans-impedance values, i.e., the point at 12,78 does not equal that of 78,12. This suggests that the electrode array shifted inside the cavity over the course of the measurements.

One important insight from these measurements is the phase response, as is shown in fig. 8.21. As much as 30 degrees of phase angle between the stimulus current and electrode voltage was observed when separation between the stimulus and measure pairs is at its maximum. This shows that the spinal cavity itself is a significantly reactive component. For comparison, the PBS solutions displayed no measurable reactance for all of the equivalent trans-impedance measurements. The decreasing phase angle of measurements using one and eight as stimulus electrodes is a result of the measured pair of electrodes being between the stimulus, i.e., it is a result of our electrode choices. There are other instances where the phase angle appears to drop below zero, but these are likely an artefact of the measurements themselves.

Parameter	Value
R_{eri} (Ω)	500
R_{sri} (Ω)	375
R_{li} (Ω)	176
Depth (layers)	5
Padding (layers)	3

Table 8.6: Determined resistor mesh parameters for an electrode array in a live sheep's spinal cavity.

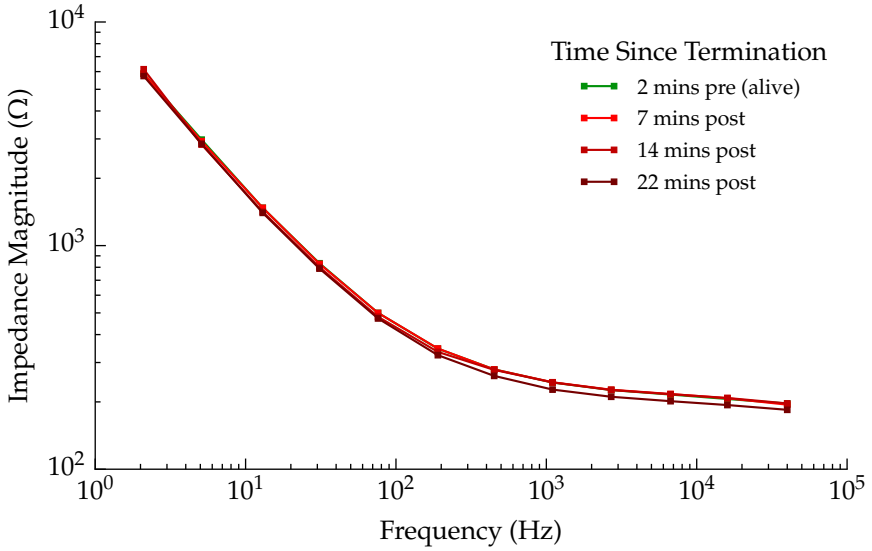


Figure 8.22: Graph showing measured CPE impedance magnitude response before and after termination.

Those situations only occur when the stimulus and measured electrodes are adjacent to one another. This means the impedance, and therefore signal-to-noise ratio, was at its lowest. Stimulus current was reduced for the in-vivo measurements to prevent muscle spasms.

The simulated results shown in figs. 8.20 and 8.21 (shown as the red trace) were calculated using the resistor mesh parameter values shown in table 8.6. Those values were determined using the same SciPy optimisation library for Python as was used to fit the values in PBS.

Constant phase & series resistance

Figures 8.22 and 8.23 show impedance magnitude and the phase response of the CPE at the interface between the electrode and the sheep's spinal cavity. Measurements were made over a thirty minute period starting two minutes before termination.

An important question I hoped to answer was whether the impedance

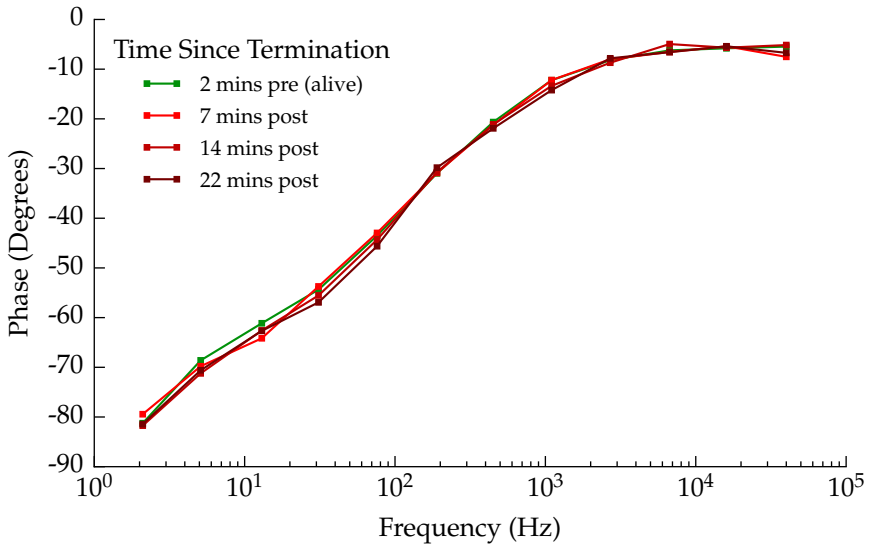


Figure 8.23: Graph showing measured CPE impedance phase response before and after termination.

response in live sheep would be any different to that of a dead sheep. Results indicate that there is practically no difference for at least thirty minutes after termination. Measurements should have been carried out over a longer time-frame after termination as it would likely take longer for the fluid composition to change. As the measurements required termination of the sheep, they had to be done after all other experiments had been completed. Since each sheep was shared between other research groups this meant that termination happened late (early the following morning) due to accumulated delays in previous experiments. Measuring the CPE response in the spinal cavity of a butchered sheep would offer a useful reference point for those measurements.

Figures 8.24 and 8.25 compare the average impedance (both magnitude and phase) of the previous graphs with the six concentrations of PBS used in the previous section. Simulated results from a numerical fit to the measured data appear as the red trace. At low frequencies, below 1 Hz, the simulated data deviates substantially from measured results. The cause for this is unclear, but in chapter 9 the opposite response appears when using unbuffered saline solutions - which may provide a clue. What is interesting is that the series resistance in sheep spine is similar to that of a 0.25X PBS solution, whereas the CPE behaves more like that of a concentration much lower than 0.025X. Based on this data, a one-tenth concentration of PBS does appear to make a reasonable trade-off between the CPE and series resistance parts, but the match between both is poor. In the following section the possibility of creating a solution that better matches these results will be explored.

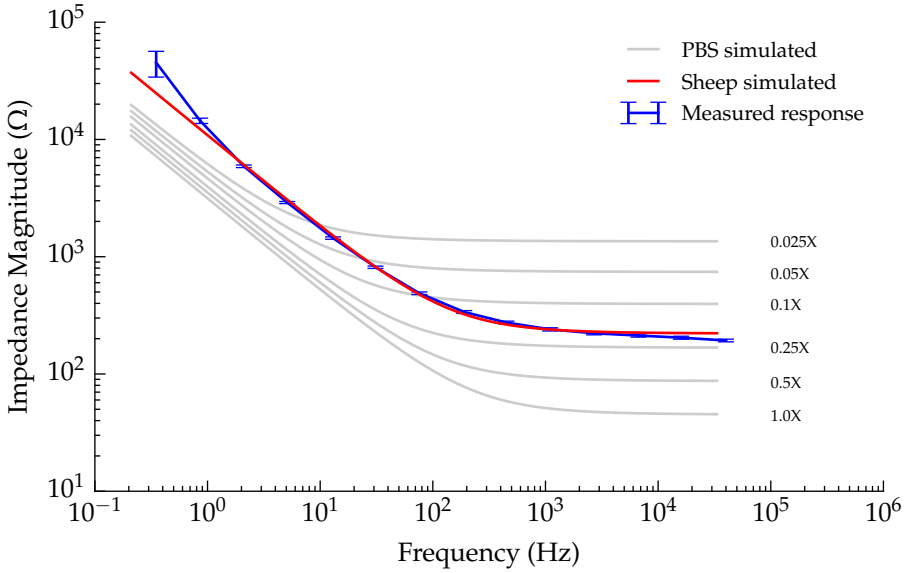


Figure 8.24: Graph showing average CPE response in live sheep compared to the six solutions of PBS, visible as the grey traces, and simulated response based on fitted parameters.

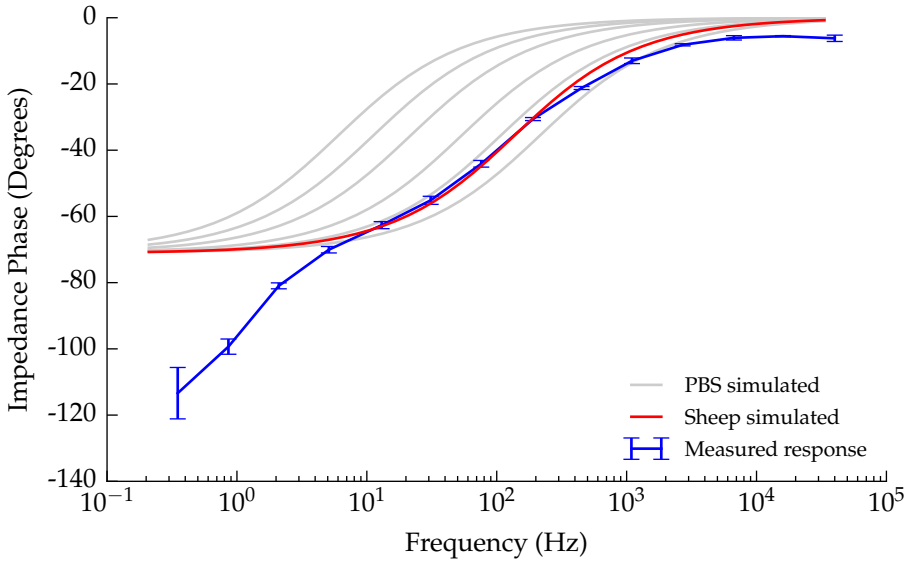


Figure 8.25: Graph showing average CPE response in a live sheep's spinal cavity compared to six concentrations of PBS, visible as grey traces, and simulated response based on fitted parameters.

Table 8.7: Determined interface parameters for the electrode array in a live sheep's spinal cavity.

Parameter	Value
R_S	126 Ω
m	1.34
k	1.77
|Z| @ 1 Hz (Ω)	11.3 kΩ
$i0$	Undetermined
n	Undetermined

Faradaic current

Faradaic measurements on the live animal were abandoned as they were deemed likely to cause muscle contractions by the veterinarian surgeon. Attempts were made to measure Faradaic response using a much lower stimulus current but no signal was evident above the noise so were discarded. These measurements may be possible if done long enough post-termination so as not to cause muscle movement.

Final model

Parameter values for the final sheep model are presented in table 8.7. Unfortunately, the diode/Faradaic parameters were not obtained on live sheep due to concerns of causing violent muscle contractions. Those measurements are likely to be of value to implant designers as they provide a reference for the beginnings of Faradaic conduction. Access to an recently terminated sheep's spine, and a surgeon, would provide a way of collecting those Faradaic measurements and additional post-termination samples.

Parameters for the interface model in sheep have been fitted. It appears as though buffered saline, of any concentration, is not an ideal representation of a live sheep's spinal cavity. The next research question is to determine if it is possible to create a solution that better matches those impedance characteristics.

Chapter 9
Creating Phantom Spinal Cavities

Utilising the measurement methods used previously, a liquid that better replicates a biological impedance is sought. This work is of benefit to engineers of medical implant devices. Having the ability to formulate a solution that mimics the electrical conditions inside a living mammal reduces the resources required to test electronic implants.

It was previously mentioned that the developers of spinal cord stimulator implants used solutions of PBS having a one-tenth concentration as a test fluid for their implants. This solution was the best substitute for an actual live spine that these engineers had. Solutions of the 0.1 X PBS are held in drums within the electronics laboratories for use whenever quick tests needed to be carried out. Electrodes were submerged into these drums in order to recreate the electrical conditions inside a human spine. This was not the only way to simulate the impedance conditions inside a person. As presented in section 8.2, anaesthetised sheep are also used. A sheep's spine is smaller than a human's but is a good approximation in terms of geometry. However, the resources involved with conducting a live sheep trial are high, such as use of a hospital operating theatre, surgeon veterinarian, and medical equipment. Engineers have no way of knowing how well those baths of saline represented a sheep's spine. It was shown in section 8.2 that the match between the two was weak. With that knowledge, and the measurement techniques developed thus far, research into creating a solution that better matches sheep spine is carried out. Strengthening that match would reduce the number of surgical operations the test engineers might need to conduct, saving resources and reducing time.

Ingredients

To determine how certain additives affect the impedance of the interface, a range of ingredients are mixed and measured. Various mixtures were created

using a heuristic approach until trends emerged which eliminated many of the ingredients. The following consumer grade ingredients were used for mixture testing and creation:

- Cellulose
- Citric acid
- Cornflour
- Gelatine
- Glycerol
- Isopropyl alcohol - 99.9% pure
- Methylated spirits
- Potassium chloride - agricultural grade
- Sodium bicarbonate
- Sodium carbonate
- Sodium chloride - non-iodised

Those ingredients were chosen as they include alcohols, sugars, salts, acids, and inert fillers. The filling agents are expected to reduce the capacitive nature of the CPE by adding non participant species to the electrolyte. As the CPE's behaviour relies on being able to reorient, repel, and attract species in the liquid phase, those ingredients should reduce the overall effect. These ingredients are also easily obtainable in large quantities. Measurement data from many trialled mixtures can be found in appendix F. They have not been included here for brevity.

Measurement

The focus when creating the solution is to match the CPE and series resistance of the sheep measurements to the newly-created solution. The measurement setup used to measure the CPE response and series resistance is mostly the same as was used for characterising PBS and the sheep's spinal cavity. This configuration differs only in the electrodes used on the array, necessary due to electrode two having its connection broken, and is shown in fig. 9.1. A 10 kΩ resistor was used to measure the current driven between electrodes eight and

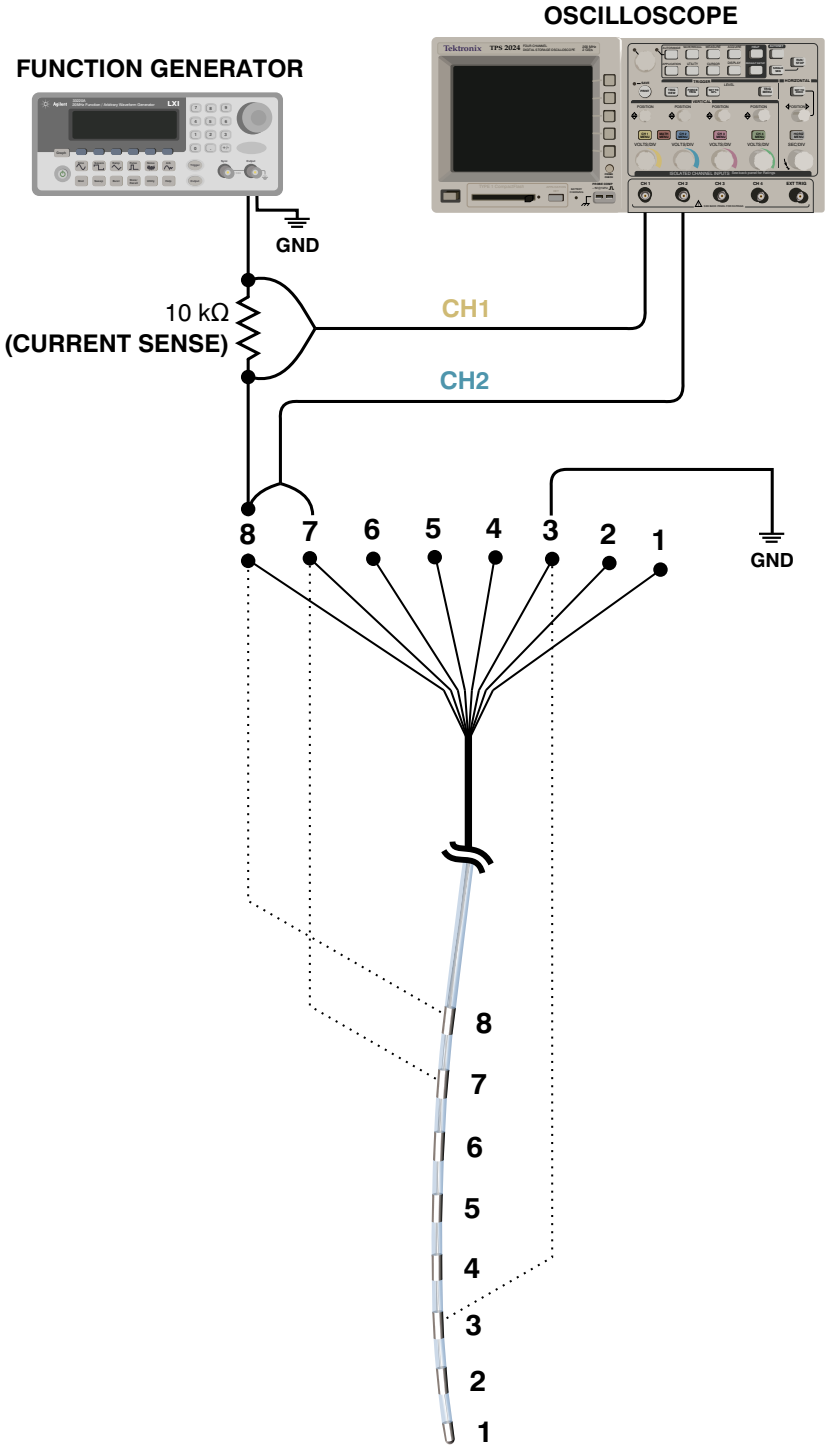


Figure 9.1: Diagram showing the measurement configuration used to measure the CPE response and resistivity of mixed solutions

three. It had a measured resistance of 9.990 kΩ, as measured with a Fluke digital multimeter.

Each measurement run begins at the upper end of the frequency spectrum and proceeds towards the low frequency endpoint. Starting with the higher frequencies offers a chance to confirm correct measurement set-up early in the measurement since they take less time to complete. A sample at the lowest frequency can take over a minute to acquire, where the higher frequencies take less than one second. The same frequencies were chosen that were used to measure the CPE's response in sheep's spinal cavity. Using those same frequencies makes comparison between the sheep data and the impedance of mixed solutions much easier. Measurements were fully automated via a Linux based computer running Python scripts. These scripts controlled the output settings of the waveform generator and acquired the resulting waveforms from the oscilloscope. The scripts had the ability to set the horizontal and vertical scales on the oscilloscope channels in order to ensure appropriate settings were used. The measurement procedure followed by the script is shown as a simplified flowchart in fig. 9.2. The programme steps through each of the required frequencies, making sure the target voltage is developed across electrodes seven and eight, before calculating the interface impedance.

The target voltage across the interface is 20 mV. This voltage was previously determined as a safe stimulus voltage in that it does not trigger Faradaic reactions at the electrode's surface and was used in for PBS measurements. Because the impedance of the interface changes with frequency it is necessary to alter the output amplitude to keep the voltage across electrodes seven and eight consistent.

Results & Discussion

Not surprisingly, the salt had the greatest influence on the response per gram added. The other non-inert ingredients had much the same effect but required larger quantities to achieve the same effect. This suggests that the primary contribution those ingredients offered was increasing the overall conductivity. The salt was capable of bringing the broad-band response in line with each of the PBS traces, which were themselves salt based solutions. The real question was whether any of the ingredients would be capable of moving the CPE's slope independently of the solution's conductivity. A solution of 0.25X PBS roughly matches the bulk conductivity of sheep spine, however the sheep's spine offered a much higher impedance at low frequencies – where the CPE dominates. What was needed was a way to increase the impedance offered

```
frequencies = [
 40000,
 16000,
 6700,
 2700,
 1100,
 450,
 190,
 76,
 31,
 13,
 5.1,
 2.1,
 0.86,
 0.35]
```

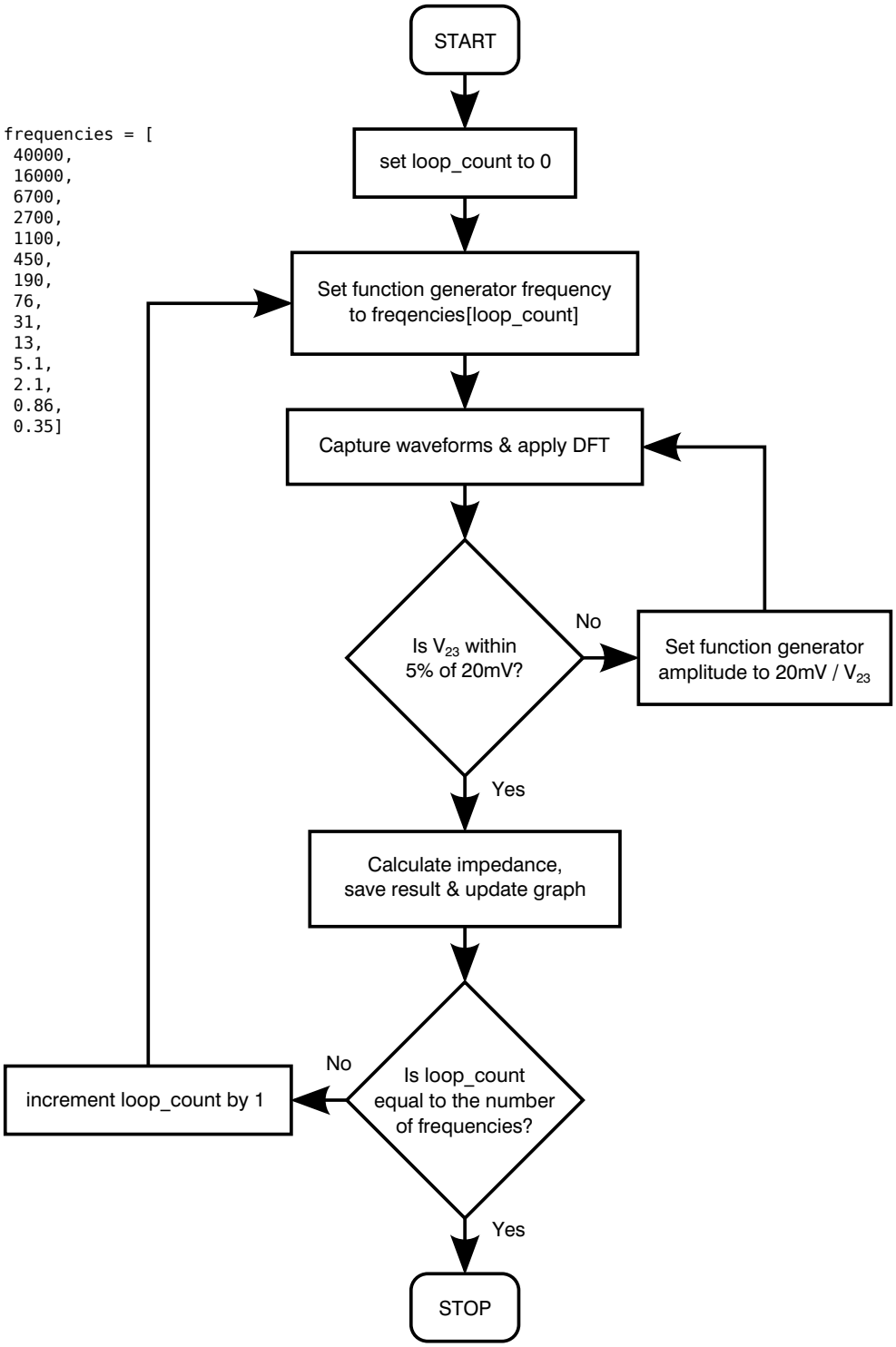


Figure 9.2: Diagram showing the execution of the measurement script

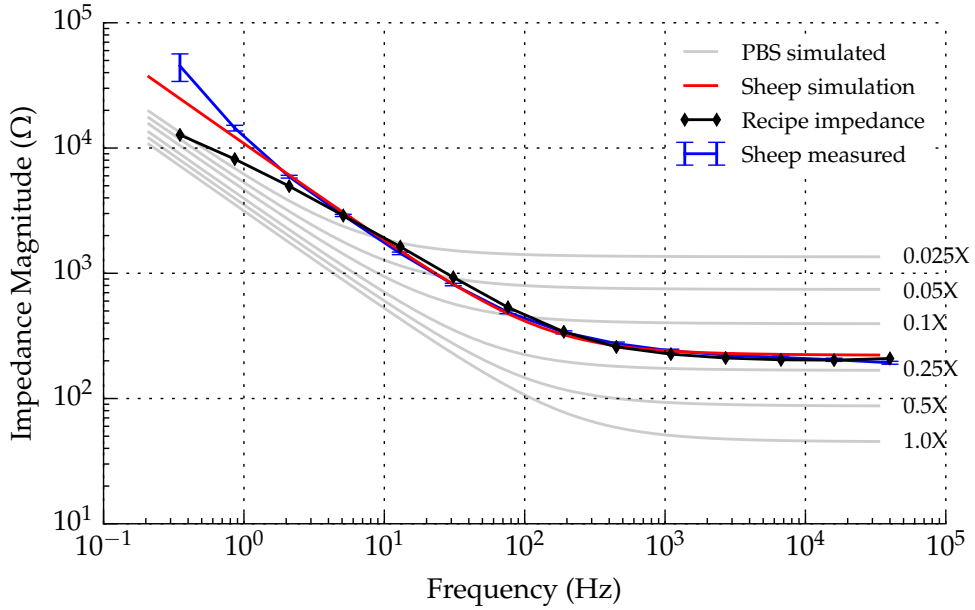


Figure 9.3: Graph showing impedance magnitude versus frequency (log-log) for 250 g cornflour mixed with 175 ml distilled water and 1.9 g table-salt.

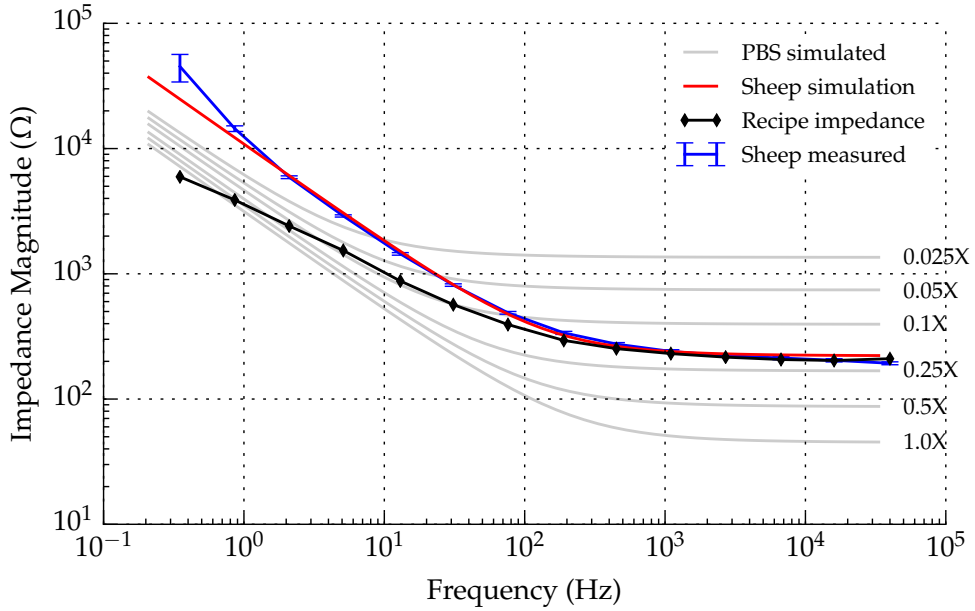


Figure 9.4: Graph showing impedance magnitude versus frequency (log-log) for 250 g cornflour mixed with 180 ml distilled water and 1.9 g table-salt.

by the CPE without affecting the bulk conductivity of the solution.

One popular and fun use of cornflour is creating a non-Newtonian fluid by mixing it with water. It behaves like a classical fluid until pressure is applied to it, then it temporarily solidifies around the point of pressure. There is a critical point when creating the mixture when adding additional water quickly turns it from a dry mix to wet volume, i.e. the consistency becomes highly sensitive to relatively small quantities water being added. Figure 9.3 shows the measured CPE response of a mixture of 250 g cornflour, 1.9 g salt, and 175 ml of water, at the point when the mixture transitions from having a dry base to a wet one. This particular solution manages to match the impedance response to sheep better than any saline solution. What is interesting is what adding another 5 ml of water did to the impedance response, shown in fig. 9.4. Adding cornflour to the saline solution made negligible difference to the response until close to the point when the water becomes saturated with cornflour, corresponding to the critical mixing point mentioned earlier. Comparing fig. 9.3 to fig. 9.4, the series resistance has stayed the same but the CPE slope has changed. No other ingredients tried changed the slope of the CPE's response and it appears as though the critical point of mixing the cornflour and water is having a large impact on that change.

The drop in impedance at low frequencies may be a result of a lack of buffering agent in the saline solution. This drop is also seen in measurements of straight unbuffered saline solutions, indicating that it is not the addition of cornflour that is causing the drop. Adding a buffering agent to the cornflour and saline mixture would most likely remove that impedance drop. It may also be possible to create an increase in impedance at low frequency by increasing the capacity of the buffering agent.

Finally, figs. 9.5 and 9.6 show both the magnitude and phase response of the closest match, as determined by a least squares error method. Again, the lack of buffering agent in these mixtures is expected to be responsible for the low frequency deviation from the measured response in sheep's spine. The ingredients that were used to create the final solution are summarised in table 9.1. These results are promising and no doubt will be of use not only to implant designers but to anyone trying to recreate the electrical impedance of biological fluids.

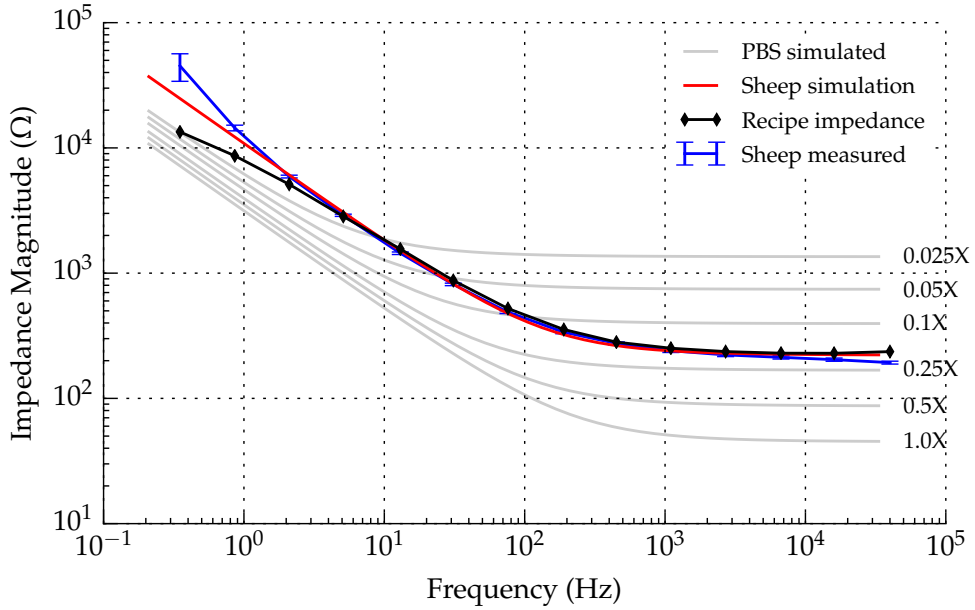


Figure 9.5: Graph of measured impedance magnitude versus frequency (log-log) for 190 g cornflour mixed with 190 ml distilled water and 0.858 g table-salt.

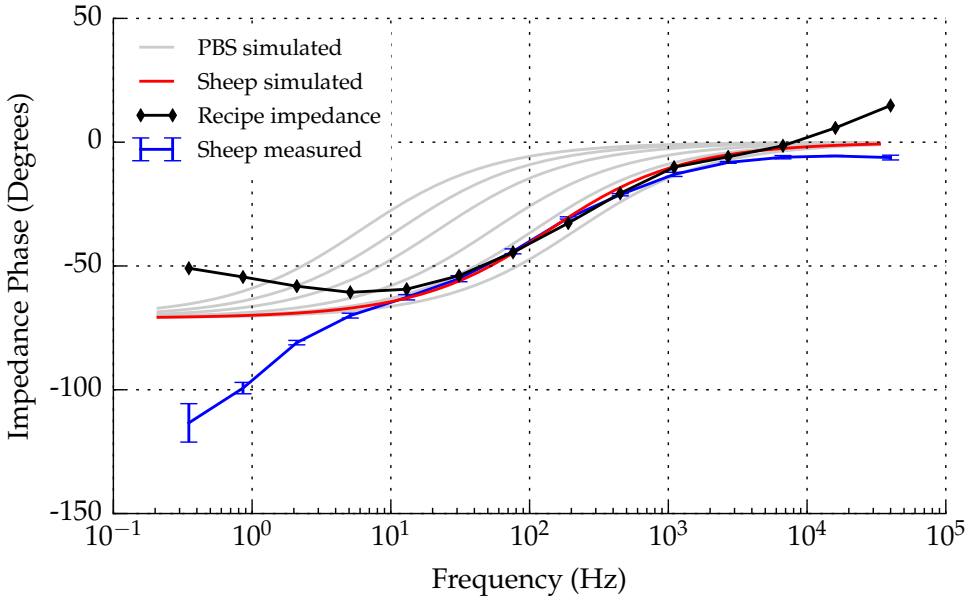


Figure 9.6: Graph of impedance phase versus frequency (log-log) for 190 g cornflour mixed with 190 ml distilled water and 0.858 g table-salt.

Ingredient	Quantity	Unit
Cornflour	190	g
Distilled water	190	ml
Salt	858	mg

Table 9.1: Ingredients used to create the mixture that matches both the CPE's response and the interface series resistance.

Chapter 10
Summary

An impedance model of a pair of electrodes in a solution of phosphate buffered saline has been taken and extended to cover a wider range of salinities. Relationships between that model's parameter values and saline concentration were determined and represented as equations. They showed that when varying the salinity, the impedance of the constant phase element scaled much slower than the series resistance. With those parameters extended to a range of salinities, the model proves to be a useful reference point for comparison to other solutions. Measurements of electrical current associated with Faradaic reactions showed that cyclic voltammetry measurements are poorly suited to capture Faradaic current. After moving to step based measurements, the collected data was consistent and matched the expected response of exponential growth. They revealed the dependence of capacitance on the current density at the electrode as proposed by Schwan [41]. These measurements also revealed that at low voltages, below 900 mV across a pair of electrodes, saline concentration has no measurable effect on the reaction rate. That observation should be especially important to implant engineers since they must avoid any Faradaic reactions in-vivo. Knowing that an electrolyte's salinity is not linked to the magnitude of those reactions at their onset simplifies that constraint. Those measurements also suggest that the interface is setting the ionic concentration at the surface, in a localised volume surrounding the electrode, in constant voltage situations. That volume of ions is likely to be the double layer itself, which is consumed by Faradaic reactions and replenished by ion migration from the electrolyte bulk. Once the layer has been consumed, the rate reaction rate of Faradaic reactions is determined by the saline concentration of the bulk solution. This hypothesis fits the data presented fig. 8.15.

With the model fitted to solutions of phosphate buffered saline, a comparison is made to a living mammal. In-vivo measurements in live sheep spine show that no single concentration of phosphate buffered saline matches the

impedance of both the CPE and series resistance. If a single concentration of buffered saline had to be used to match both the CPE and series resistance then the one-tenth concentration solution is a good trade-off, but is a poor match to both. The measurements also show that the spinal cavity provides a significant amount of reactance, which is not present in saline solutions. Modelling that reactance could be done by fitting inductive and/or capacitive elements to the resistive mesh of the model. That step was not taken due to the complexity of the geometry inside the spinal canal and variability expected between epidural insertions.

Seeing that the match between saline and a sheep's spinal cavity were poor I then created a mixture that improved the match. Tests with a variety of ingredients showed that a mix of salt, cornflour and water improved the match dramatically. The impedance presented by the CPE moved with ratio of cornflour to water, whereas the series resistance was determined by the salt. It appears as though the cornflour, when in high enough concentrations, reduced the capacitive effect by displacing active species in the liquid with inert filler.

Appendix A
Charged Drips for Energy Harvesting

In 1867 William Thomson (Lord Kelvin) described an apparatus that could generate electrostatic charge using drops of water [77]. Figure A.1 shows original artwork of the device from that paper. It works by inducing charge onto drops of water before they detach from the source of the drips. This device was the starting point of investigation into the use of charged water drops to generate electrical power.

Generating Charge

Figure A.2 shows charge generating mechanism of Lord Kelvin's electrostatic generator. This mechanism is comprised of three main components:

1. a jet of water which breaks into droplets,
2. an inducting ring surrounding the area where the jet breaks up,
3. a receiver where the charged droplets are collected.

A diagram showing a variation of that design with both sides present is shown as fig. A.3. The polarity of each receiver and induction ring is represented by its colour. Equal and opposite charge is accumulated in the receivers below the nozzles. The receivers are electrically isolated from each other, but are connected to the induction rings of the opposite side. The induction rings push charge away from the drop as it forms on the positive side, and pulls charge onto drops forming on the negative side. Gravity then pulls the drops down into the receivers below. Because the receiver and the drip both have the same polarity electric field, they repel each other. The drop, because of gravity and the height it falls from, is doing work as it falls into the receiver as the integral of the electrostatic force over the distance it travels. The result of that work is an increase in static charge held by the receiver, measured as voltage.

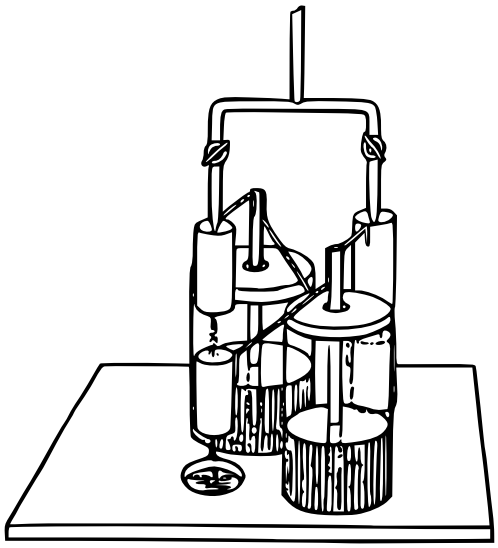


Figure A.1: Drawing of Lord Kelvin's electrostatic generator [1].

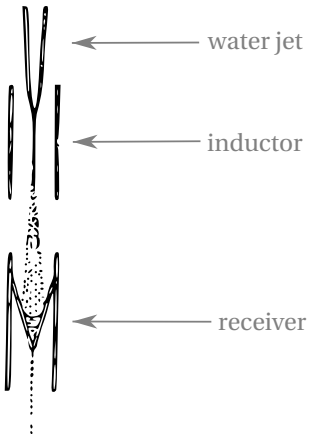


Figure A.2: Drawing of the charging mechanism for Lord Kelvin's electrostatic generator

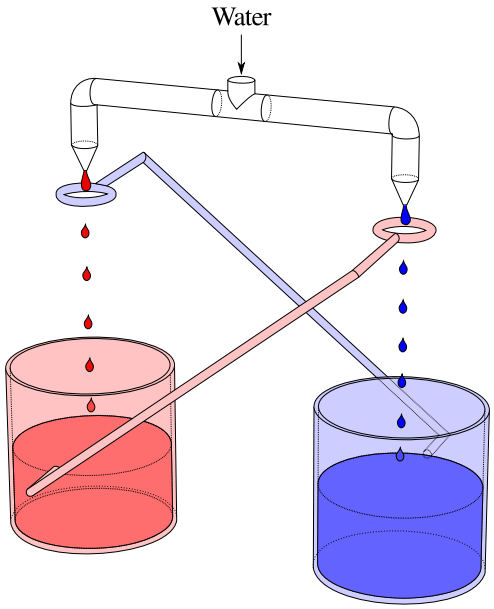


Figure A.3: Simplified diagram of Lord Kelvin's water dropper configuration

Optimising output

Summer research student Jonathon McMullen assembled a test-rig to recreate the experiment. Once constructed, another summer research student, Wayne Crump, and myself took measurements. Then we sought to optimise the design in the hopes that is may be suitable as an energy harvester.

Drop volume and frequency

The first optimisation question was "is it better to have many small or fewer but larger drops?". A simplified experiment was made with the help of Wayne Crump. We aimed to remove as many variables from the experiment that was previously performed by Jonathan McMullen. By doing so we hoped to isolate the effect of varying drop size, induction voltage, and flow rate, had on output power.

Experimental setup

A photo of the measurement setup is shown in fig. A.4. A simplified diagram of that same setup is shown in fig. A.5. Drips are formed from a syringe needle which then fall through the induction ring before hitting the tin foil. A close-up of photo of the needle and inductor is shown in fig. A.6. The drips frequency is determined by a microphone paced under the foil and the flow of water is set by the syringe pump (shown in fig. A.7). The volume of each

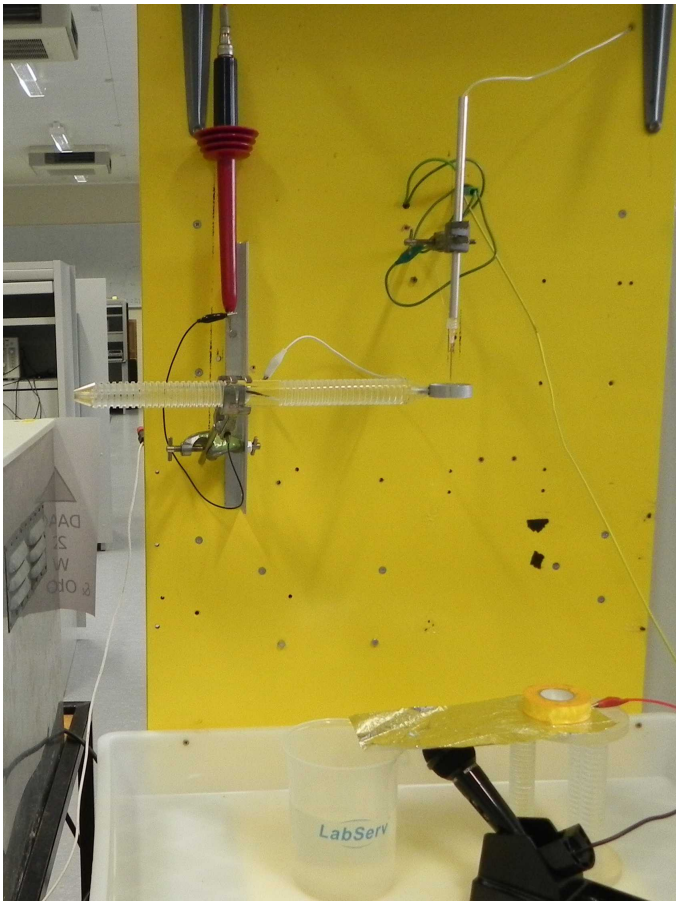


Figure A.4: Photo of experimental setup for charge on drip measurements.

drip is calculated by dividing the flow rate by the drip frequency. The charge on each drop is determined by dividing the average current through the multimeter by the drip frequency. Measured current was in the nano Ampere range so direct current measurement with a multimeter was not possible. Instead, the multimeter was set to measure voltage and the internal resistance of the meter itself was used as the current sense resistor. The multimeter had an internal resistance of 10 MΩ.

Results

Figure A.11 shows the effect of increasing the flow on the output current; a relatively linear response. Figure A.8 shows the effect of drip volume on the bound charge per drop. This curve resembles the surface area of a sphere against the volume of a sphere. This was expected as excess charge will distribute itself over the outer surface of the drop, and therefore be proportional to its surface area. If the same data is plotted in terms of charge per volume versus volume of a drip, as is shown in Figure A.9, it is evident

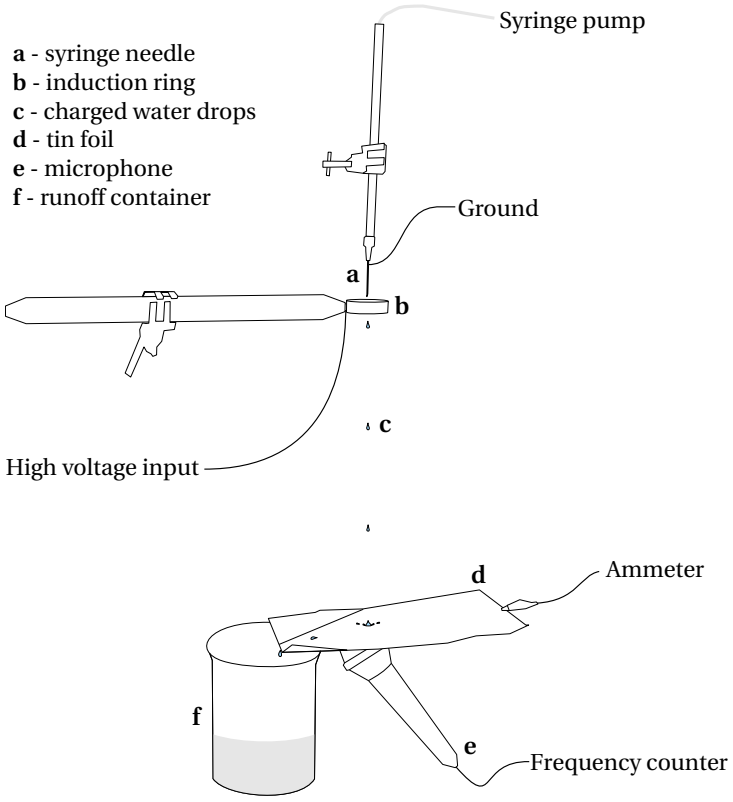


Figure A.5: Diagram of experimental setup for charge on drip experiments.

that smaller drop sizes equate to a higher charge per volume of water ratio. Figure A.10 shows the results of changing the induction voltage on the average charge carried per drop. The results show the charge induced on a drop is proportional to the induction voltage. Variation in the measurement data is due to variations in room temperature.

Conclusion & discussion

Increasing the output of the generator is possible by:

- reducing the drop size for a given volumetric flow rate,
- increasing the volumetric flow rate,
- increasing the voltage on the induction ring.

Increasing the volumetric flow through a nozzle causes drops to turn into a stream above a flow threshold. If that stream does not break into drops before connecting with the receiving vessel then charge held in the vessel can travel up the stream; short-circuiting the device. If the stream does reliably break into droplets then the induction ring should be positioned as close to the

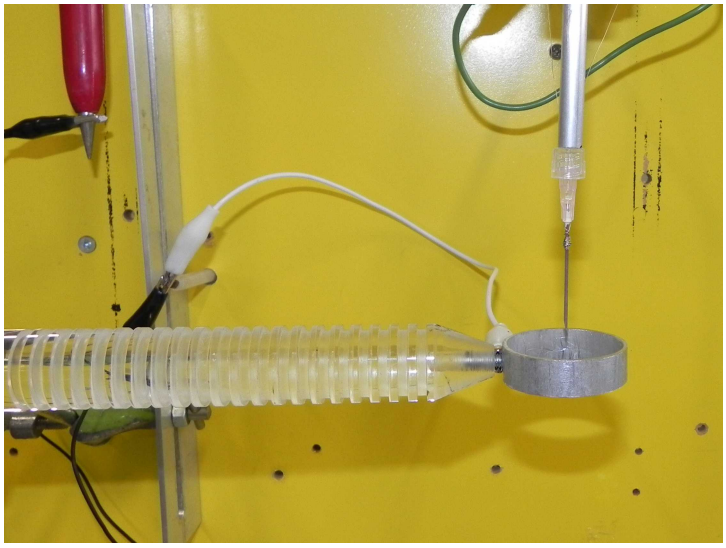


Figure A.6: Photo of the dropper and high voltage inductor.

transition point as possible. Reducing drop size and increasing the induction voltage both increase the charge to mass ratio of individual drops. As this ratio increases, the movement of drops becomes increasingly dominated by the electrostatic force between the receiver and the drop. The force repels the drop from the receiver meaning that the drops start to bend away from the receiver as they fall. Increasing the charge-to-mass ratio too far cause drops to escape the receiver. It is possible with a sufficiently large charge-to-mass ratio for the water droplets path to bend back upward to the induction ring, i.e., the water "falls upwards" back to the high voltage ring. A nozzle capable of producing consistent drop volumes allows for the greatest efficiency since the charge per drop can be maximised with the least reduction of drop catchment.

Scale

Lord Kelvin's original design of electrostatic generator is too big to fit inside a water meter. Because of its size the voltage differential required to create the target electrostatic field strength is large; approximately 4 kV. A reduction in physical size would lower this voltage while keeping the electric field gradient the same. Notice in fig. A.6 how revolved cuts have been made in the arm supporting the induction ring to increase electrical isolation. Lower voltages would mean less electrical isolation is necessary keep the device from self-discharging. Reducing the scale of the device would allow it to generate a maximum output at voltages that are easier to process using conventional electronics.

Figure A.7: Photo of the syringe pump used to produce drops and control flow rate.

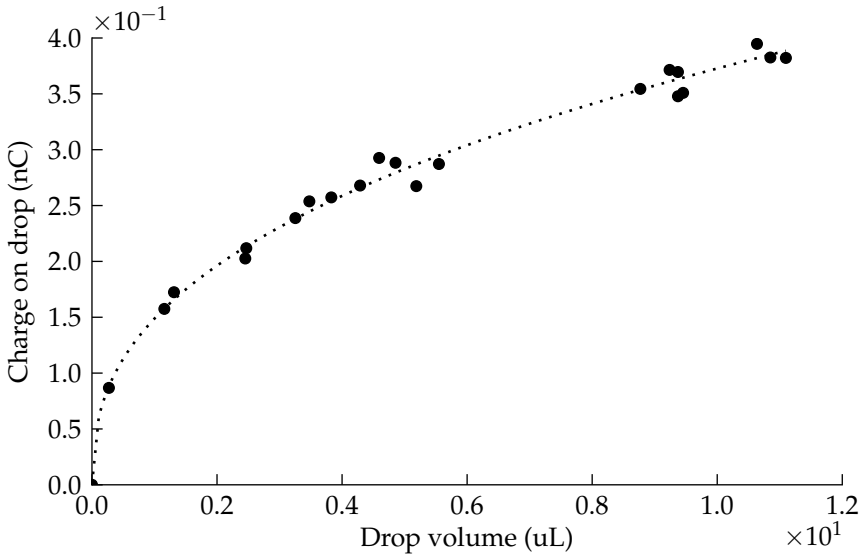


Figure A.8: Charge on drip versus drip volume for a fixed induction voltage of 2.5 kV.

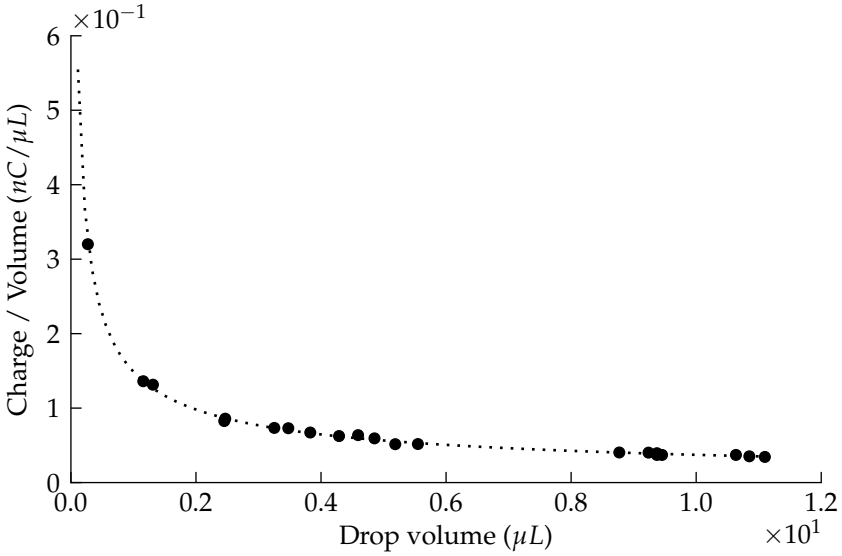


Figure A.9: Charge per volume versus drop volume for a fixed induction voltage of 2.5 kV.

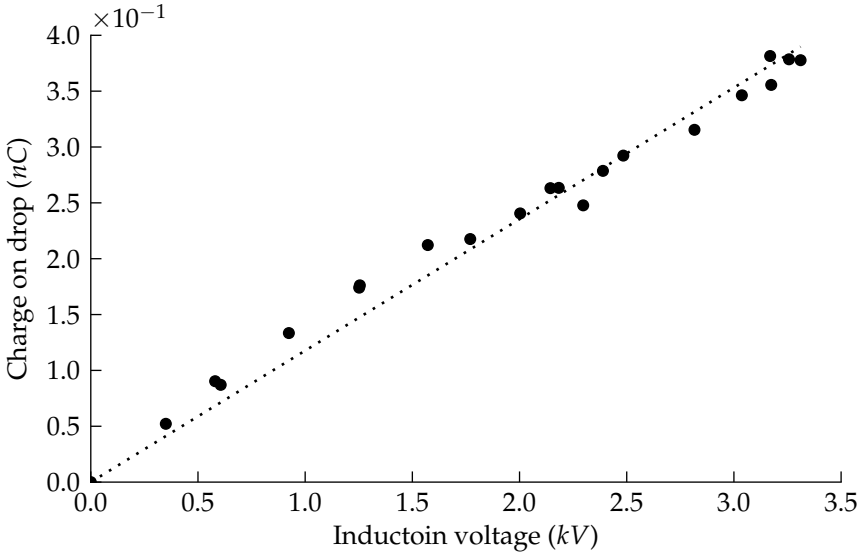


Figure A.10: Graph showing charge on a drop versus induction voltage for a fixed drop volume.

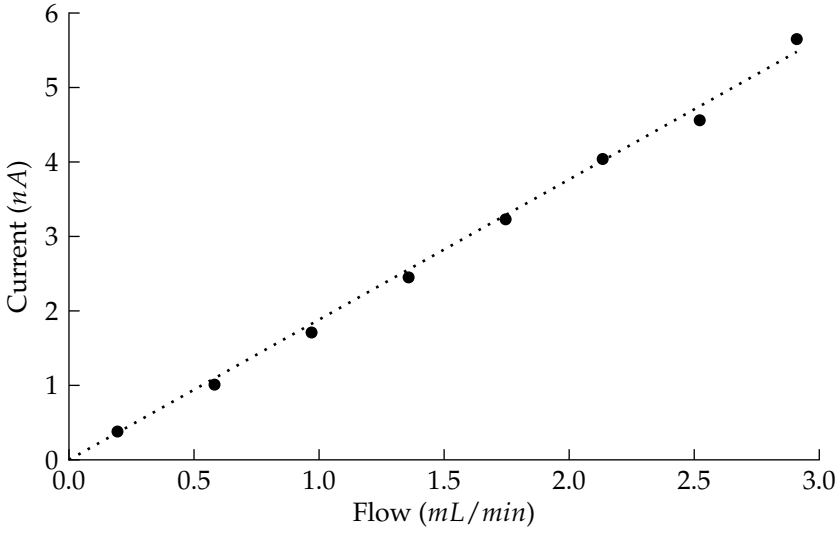


Figure A.11: Graph showing current versus flow rate for an induction voltage of 3.8 kV and drop volume of 3.1 µl.

Appendix B
Streaming Cell Voltage Measurements

For compactness, the pressure/voltage graphs for individual streaming cell channel heights were omitted from chapter 3. They are presented here in order of increasing internal channel widths. The graph showing all results combined is presented earlier as fig. 3.10.

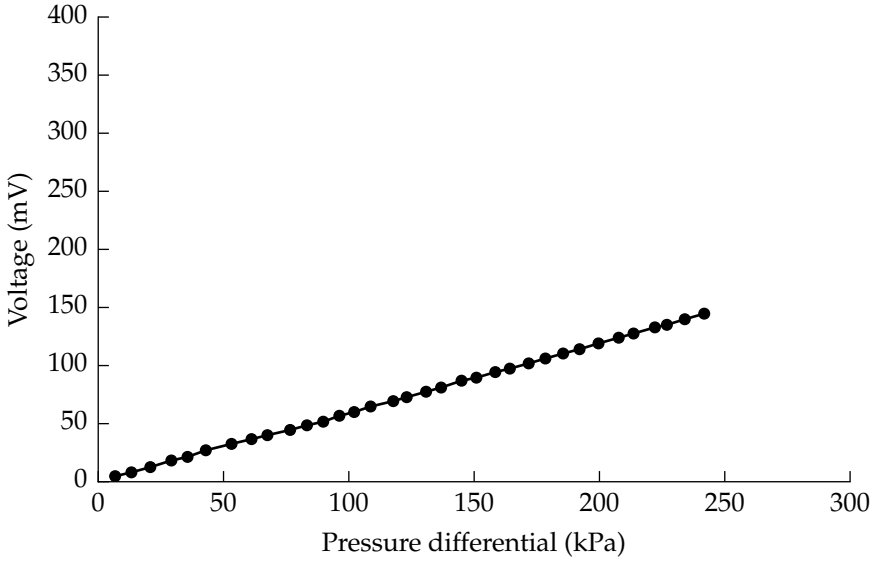


Figure B.1: Graph showing the voltage output with applied pressure differential across a 26 µm glass micro-channel (38 mV offset added)

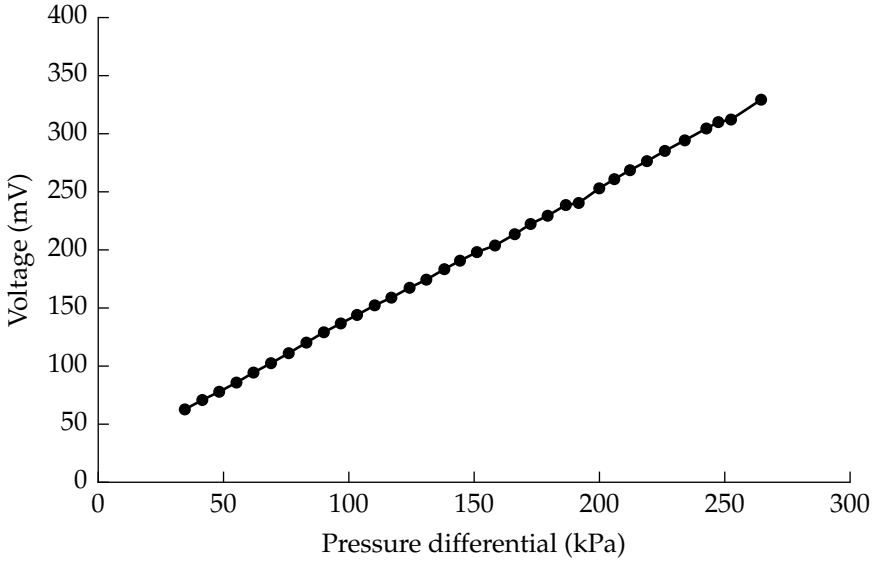


Figure B.2: Graph showing the voltage output with applied pressure differential across a 52 µm glass micro-channel (23 mV offset added)

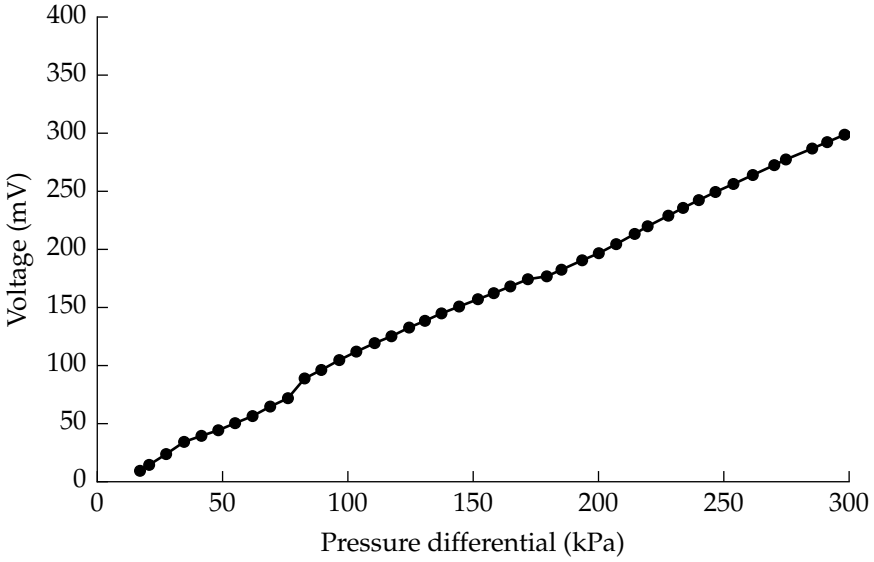


Figure B.3: Graph showing the voltage output with applied pressure differential across a 56 µm glass micro-channel (405 mV offset added)

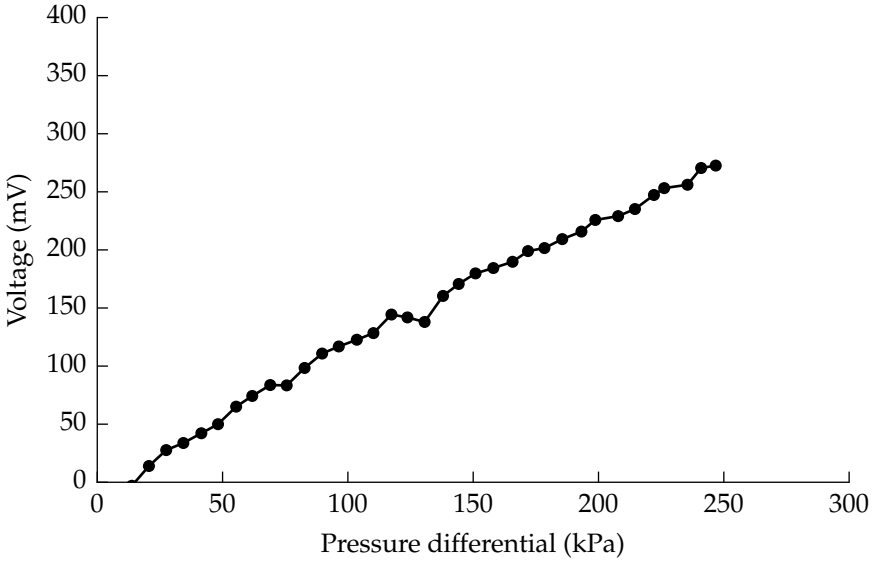


Figure B.4: Graph showing the voltage output with applied pressure differential across a 71 µm glass micro-channel (44 mV offset added)

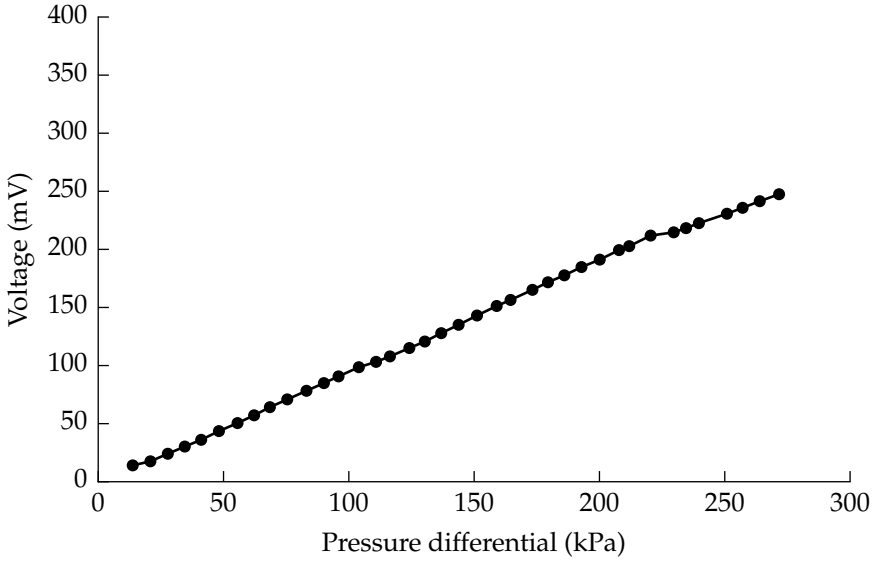


Figure B.5: Graph showing the voltage output with applied pressure differential across a 75 µm glass micro-channel (20 mV offset added)

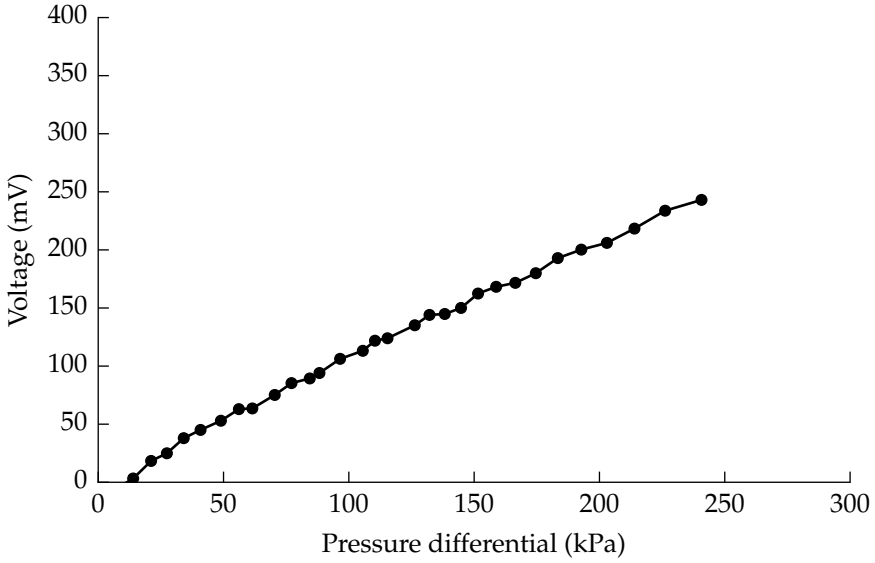


Figure B.6: Graph showing the voltage output with applied pressure differential across a 106 µm glass micro-channel (56 mV offset added)

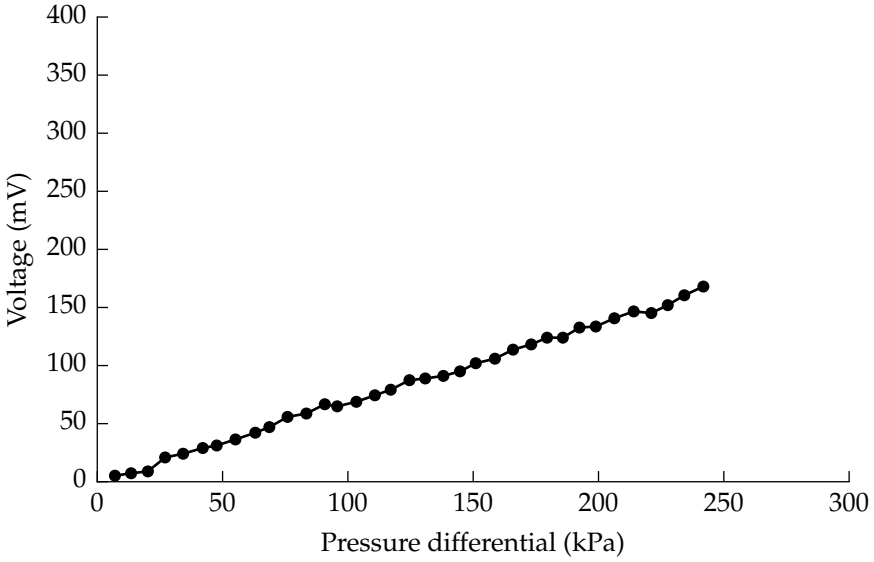


Figure B.7: Graph showing the voltage output with applied pressure differential across a 125 µm glass micro-channel (5 mV offset added)

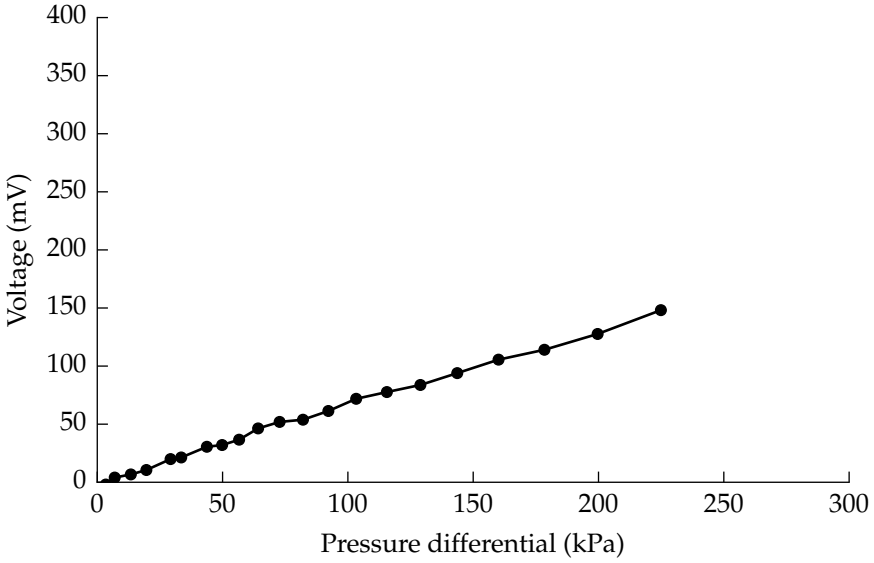


Figure B.8: Graph showing the voltage output with applied pressure differential across a 161 µm glass micro-channel (23 mV offset added)

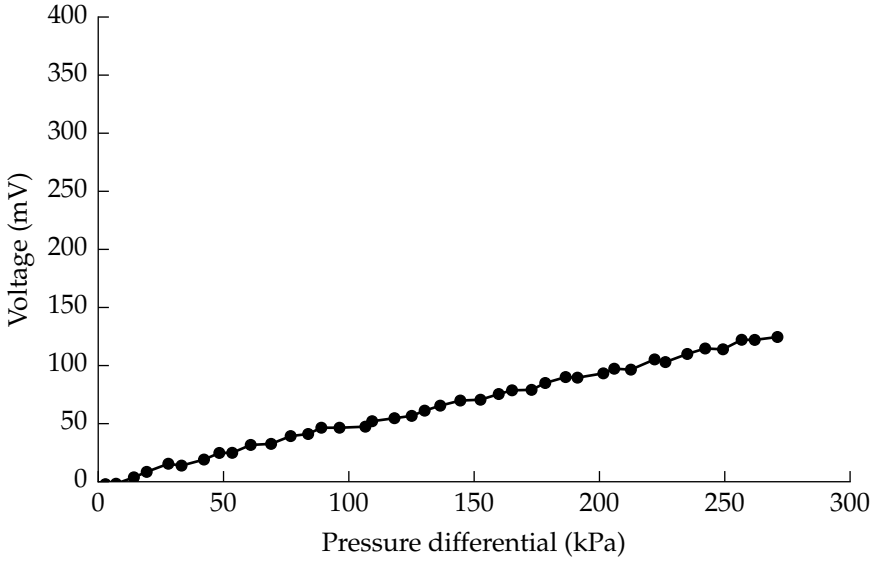


Figure B.9: Graph showing the voltage output with applied pressure differential across a 178 µm glass micro-channel (13 mV offset added)

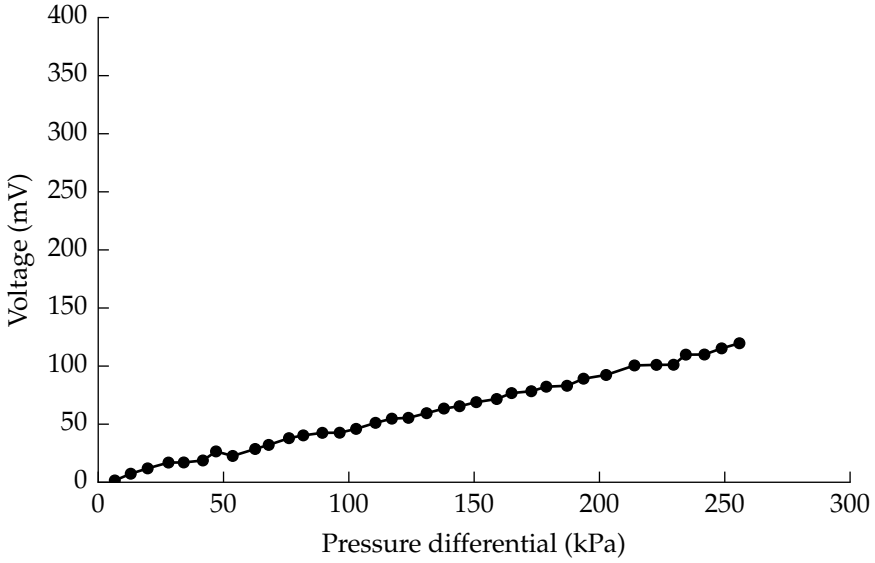


Figure B.10: Graph showing the voltage output with applied pressure differential across a 245 µm glass micro-channel (27 mV offset added)

Appendix C
Microprocessor Energy Measurements

Measurement scripts

Energy consumption measurements were made using an Agilent E5270B 8-Slot Precision Measurement Mainframe, a Tektronix MSO 4054 Mixed Signal Oscilloscope and a desktop PC. Operation of the E5270 was done via GLIB interface using a USB connection and Agilent IO Libraries Version 16.0.1458.0. From here the machine was interfaced using custom Python scripts (appended) and PyVisa (available from http://pyvisa.sourceforge.net/pyvisa/).

Measurement of chip energy consumption was carried out using the following measurement script. This script is written in Python and was executed using PyLab from the Enthought Python bundle.

Measurement Data

It should be noted that the Freescale M9S08QG8 and the Microchip PIC16F1827 were unable to boot reliably into a low power state at 1.8V. To prevent this from happening the chips were booted at 2.0V and lowered to 1.8V to prevent the chips entering a state where current consumption was in the hundreds of microamps range.

Sleep Mode

The following tables list the unprocessed current measurements for a sweep of Vdd from 1.8V to 5.5V. Sweeps have been restricted to the input voltage ranges as specified in each chips datasheet.

Vdd	Tiny13V	Tiny25V	12F675	16F1827	M9S08QG8
1.80	3.41E-08	9.27E-08	N/A	9.914E-07	3.82E-04
1.90	4.04E-08	9.30E-08	N/A	1.008E-06	3.86E-04
2.00	4.73E-08	9.33E-08	4.497E-10	1.023E-06	4.88E-07
2.10	5.47E-08	9.36E-08	4.677E-10	1.038E-06	4.89E-07
2.20	6.26E-08	9.39E-08	4.797E-10	1.054E-06	4.91E-07
2.30	7.10E-08	9.42E-08	4.907E-10	1.068E-06	4.92E-07
2.40	7.99E-08	9.45E-08	5.057E-10	1.087E-06	4.94E-07
2.50	8.92E-08	9.47E-08	5.210E-10	1.102E-06	4.96E-07
2.60	9.90E-08	9.50E-08	5.347E-10	1.119E-06	4.99E-07
2.70	1.09E-07	9.53E-08	5.487E-10	1.137E-06	5.02E-07
2.80	1.20E-07	9.56E-08	5.610E-10	1.166E-06	5.05E-07
2.90	1.31E-07	9.59E-08	5.760E-10	1.195E-06	5.10E-07
3.00	1.42E-07	9.63E-08	5.903E-10	1.234E-06	5.15E-07
3.10	1.54E-07	9.66E-08	6.057E-10	1.290E-06	5.21E-07
3.20	1.67E-07	9.70E-08	6.217E-10	1.300E-06	5.28E-07
3.30	1.79E-07	9.74E-08	6.397E-10	1.301E-06	5.37E-07
3.40	1.92E-07	9.78E-08	6.590E-10	1.302E-06	5.47E-07
3.50	2.06E-07	9.82E-08	6.763E-10	1.301E-06	5.60E-07
3.60	2.20E-07	9.87E-08	6.970E-10	1.304E-06	5.75E-07
3.70	2.35E-07	9.93E-08	7.200E-10	1.307E-06	N/A
3.80	2.49E-07	1.00E-07	7.437E-10	1.309E-06	N/A
3.90	2.65E-07	1.01E-07	7.703E-10	1.311E-06	N/A
4.00	2.80E-07	1.02E-07	7.960E-10	1.312E-06	N/A
4.10	2.97E-07	1.03E-07	8.273E-10	1.317E-06	N/A
4.20	3.13E-07	1.04E-07	8.690E-10	1.319E-06	N/A
4.30	3.31E-07	1.05E-07	9.033E-10	1.330E-06	N/A
4.40	3.48E-07	1.07E-07	9.487E-10	1.334E-06	N/A
4.50	3.67E-07	1.09E-07	9.977E-10	1.342E-06	N/A
4.60	3.86E-07	1.11E-07	1.056E-09	1.351E-06	N/A
4.70	4.05E-07	1.13E-07	1.117E-09	1.360E-06	N/A
4.80	4.25E-07	1.16E-07	1.197E-09	1.373E-06	N/A
4.90	4.46E-07	1.19E-07	1.296E-09	1.385E-06	N/A
5.00	4.67E-07	1.23E-07	1.403E-09	1.401E-06	N/A
5.10	4.90E-07	1.27E-07	1.513E-09	1.422E-06	N/A
5.20	5.13E-07	1.32E-07	1.661E-09	1.445E-06	N/A
5.30	5.37E-07	1.38E-07	1.811E-09	1.469E-06	N/A
5.40	5.62E-07	1.45E-07	1.975E-09	1.497E-06	N/A
5.50	5.88E-07	1.53E-07	2.164E-09	1.529E-06	N/A

Table C.1: Raw sleep measurements (Vdd 1.8V – 5.5V)

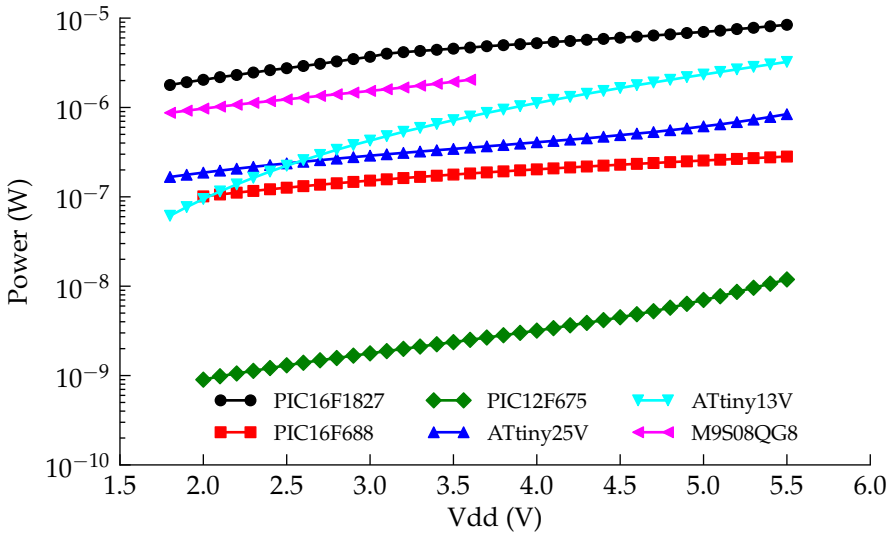


Figure C.1: Graph of power consumption of each microprocessor while in sleep mode versus supply voltage.

Clocking

ATtiny13V

ATtiny25V

Microchip PIC16F1827

Microprocessor Test Code

ATMEL ATtiny13V and ATtiny25

Code was written in AVRStudio 4.18 (build 684) and compiled using WinAVR (AVR-GCC compiler for windows available from http://winavr.sourceforge.net/. Chip programming was done using an Atmel AVR STK500 demonstration board with serial interface.

Sleep

Programming fuses were all disabled (Watchdog, brown-out detect, clock divider) and clock selection was set to 'Int. RC Osc 128kHz; Start-up time; 14 CK + 0ms'

```
1 #include <avr/io.h>
2
3 int main(void)
4 {
5     //POWER REDUCTION TIPS:
```

Vdd	9.6 MHz	4.8 MHz	1.2 MHz	600 kHz	128 kHz	16 kHz
1.80	N/A	N/A	3.900E-04	2.148E-04	8.098E-05	5.393E-05
1.90	N/A	N/A	4.121E-04	2.258E-04	8.387E-05	5.499E-05
2.00	N/A	N/A	4.242E-04	2.329E-04	8.708E-05	5.608E-05
2.10	N/A	N/A	4.429E-04	2.421E-04	8.897E-05	5.721E-05
2.20	N/A	N/A	4.634E-04	2.523E-04	9.126E-05	5.839E-05
2.30	N/A	N/A	4.859E-04	2.632E-04	9.415E-05	5.964E-05
2.40	N/A	N/A	5.088E-04	2.748E-04	9.738E-05	6.099E-05
2.50	N/A	N/A	5.323E-04	2.865E-04	1.006E-04	6.214E-05
2.60	N/A	N/A	5.557E-04	2.984E-04	1.038E-04	6.396E-05
2.70	2.759E-03	1.664E-03	5.799E-04	3.108E-04	1.071E-04	6.560E-05
2.80	2.879E-03	1.742E-03	6.065E-04	3.235E-04	1.102E-04	6.743E-05
2.90	3.001E-03	1.834E-03	6.384E-04	3.366E-04	1.136E-04	6.926E-05
3.00	3.116E-03	1.945E-03	6.705E-04	3.540E-04	1.174E-04	7.100E-05
3.10	3.231E-03	2.022E-03	6.876E-04	3.582E-04	1.124E-04	6.809E-05
3.20	3.358E-03	2.109E-03	7.144E-04	3.710E-04	1.150E-04	6.878E-05
3.30	3.486E-03	2.199E-03	7.421E-04	3.843E-04	1.175E-04	6.996E-05
3.40	3.619E-03	2.295E-03	7.700E-04	3.979E-04	1.200E-04	7.115E-05
3.50	3.757E-03	2.394E-03	7.976E-04	4.114E-04	1.226E-04	7.235E-05
3.60	3.905E-03	2.491E-03	8.269E-04	4.251E-04	1.250E-04	7.354E-05

Table C.2: Atmel ATtiny13V clocking current (Vdd 1.8V – 3.6V).

Vdd	9.6 MHz	4.8 MHz	1.2 MHz	600 kHz	128 kHz	16 kHz
3.70	4.046E-03	2.580E-03	8.555E-04	4.392E-04	1.275E-04	7.471E-05
3.80	4.185E-03	2.661E-03	8.854E-04	4.535E-04	1.298E-04	7.706E-05
3.90	4.322E-03	2.740E-03	9.156E-04	4.685E-04	1.321E-04	7.701E-05
4.00	4.464E-03	2.822E-03	9.467E-04	4.834E-04	1.343E-04	7.814E-05
4.10	4.606E-03	2.900E-03	9.780E-04	4.990E-04	1.365E-04	7.929E-05
4.20	4.753E-03	2.932E-03	1.010E-03	5.152E-04	1.386E-04	8.045E-05
4.30	4.901E-03	2.991E-03	1.043E-03	5.313E-04	1.406E-04	8.163E-05
4.40	5.061E-03	3.001E-03	1.077E-03	5.491E-04	1.425E-04	8.283E-05
4.50	5.219E-03	2.987E-03	1.110E-03	5.655E-04	1.444E-04	8.406E-05
4.60	5.389E-03	3.073E-03	1.144E-03	5.833E-04	1.462E-04	8.531E-05
4.70	5.547E-03	3.162E-03	1.178E-03	6.002E-04	1.480E-04	8.661E-05
4.80	5.710E-03	3.257E-03	1.213E-03	6.216E-04	1.498E-04	8.795E-05
4.90	5.872E-03	3.348E-03	1.248E-03	6.395E-04	1.502E-04	8.932E-05
5.00	6.046E-03	3.443E-03	1.285E-03	6.567E-04	1.484E-04	9.072E-05
5.10	6.228E-03	3.548E-03	1.320E-03	6.746E-04	1.470E-04	9.213E-05
5.20	6.391E-03	3.647E-03	1.357E-03	6.940E-04	1.465E-04	9.346E-05
5.30	6.575E-03	3.753E-03	1.394E-03	7.134E-04	1.469E-04	9.482E-05
5.40	6.753E-03	3.857E-03	1.434E-03	7.331E-04	1.485E-04	9.640E-05
5.50	6.955E-03	3.970E-03	1.471E-03	7.526E-04	1.506E-04	9.818E-05

Table C.3: Current consumed during processing for the Atmel ATtiny13V (Vdd 3.7V – 5.5V).

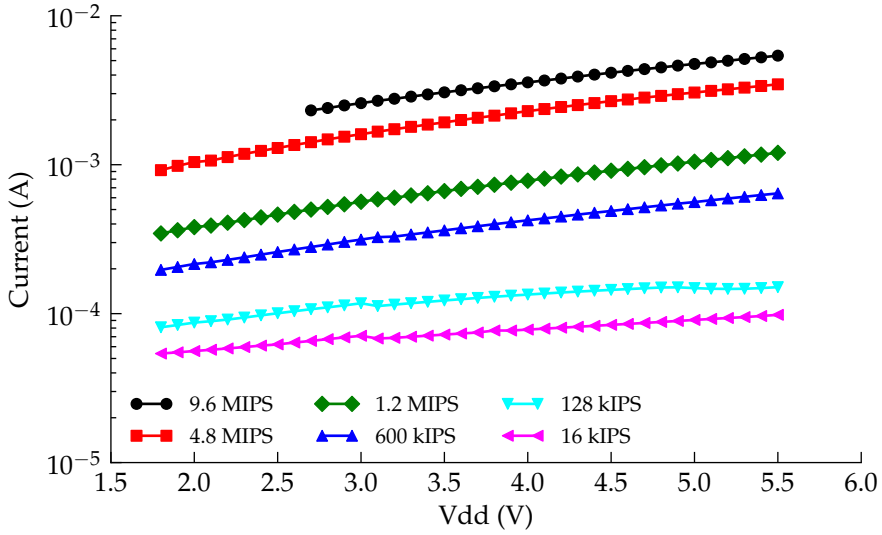


Figure C.2: Graph of current consumption of the Atmel ATtiny13V while clocking versus supply voltage.

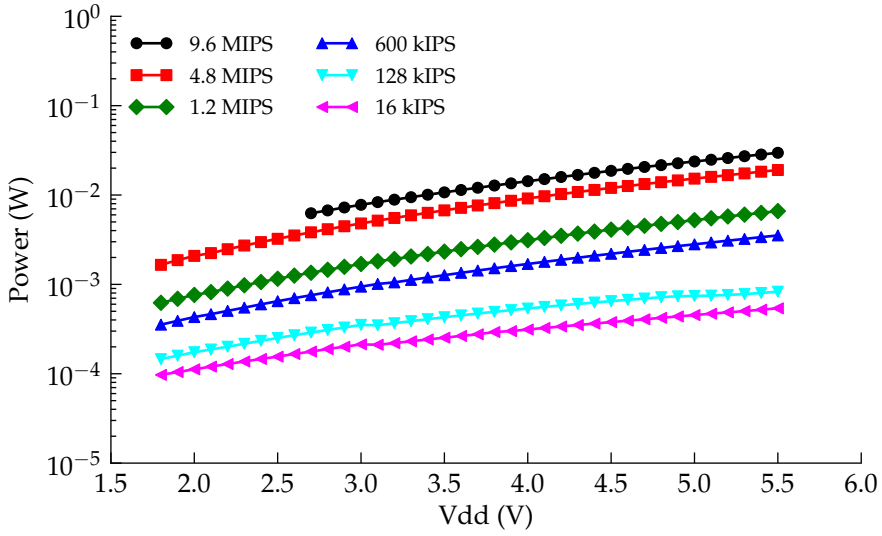


Figure C.3: Graph of power consumption of the Atmel ATtiny13V while clocking versus supply voltage.

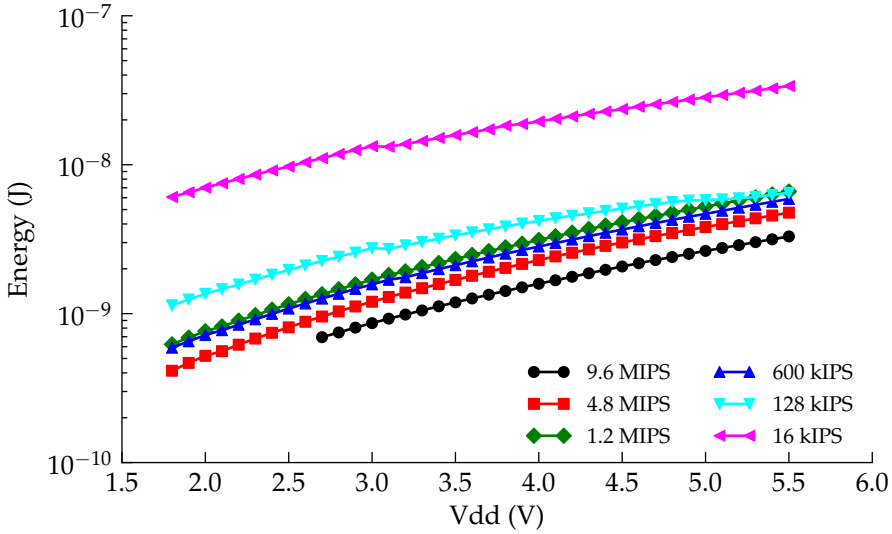


Figure C.4: Graph of energy consumed per instruction for the Atmel ATtiny13V versus supply voltage.

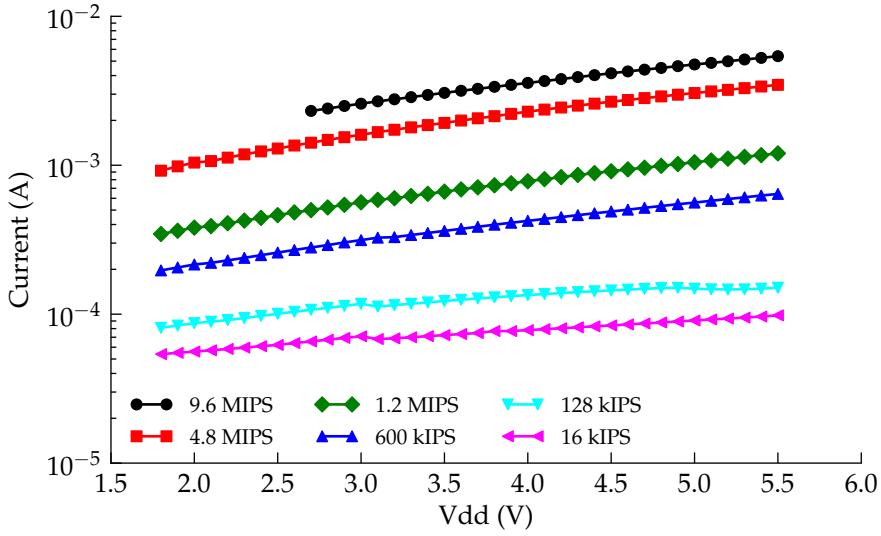


Figure C.5: Graph of current consumption of the Atmel ATtiny25V while clocking versus supply voltage

Vdd	16 MHz	8 MHz	6.4 MHz	2 MHz	1 MHz	800 kHz	128 kHz	16 kHz
1.80	N/A	N/A	N/A	1.148E-03	4.106E-04	1.146E-03	7.622E-05	1.666E-04
1.90	N/A	N/A	N/A	1.205E-03	4.319E-04	1.209E-03	7.858E-05	1.737E-04
2.00	N/A	N/A	N/A	1.261E-03	4.531E-04	1.274E-03	8.121E-05	1.813E-04
2.10	N/A	N/A	N/A	1.320E-03	4.754E-04	1.339E-03	8.464E-05	1.903E-04
2.20	N/A	N/A	N/A	1.378E-03	4.984E-04	1.404E-03	8.981E-05	1.994E-04
2.30	N/A	N/A	N/A	1.434E-03	5.195E-04	1.471E-03	9.486E-05	2.066E-04
2.40	N/A	N/A	N/A	1.491E-03	5.413E-04	1.537E-03	9.642E-05	2.126E-04
2.50	N/A	N/A	N/A	1.540E-03	5.671E-04	1.604E-03	9.903E-05	2.193E-04
2.60	N/A	N/A	N/A	1.597E-03	5.912E-04	1.671E-03	1.008E-04	2.257E-04
2.70	6.283E-03	3.124E-03	1.765E-03	1.651E-03	6.163E-04	1.740E-03	1.030E-04	2.320E-04
2.80	6.551E-03	3.256E-03	1.842E-03	1.706E-03	6.425E-04	1.811E-03	1.053E-04	2.378E-04
2.90	6.826E-03	3.392E-03	1.916E-03	1.766E-03	6.683E-04	1.883E-03	1.072E-04	2.414E-04
3.00	7.097E-03	3.537E-03	1.995E-03	1.832E-03	6.963E-04	1.959E-03	1.094E-04	2.434E-04
3.10	7.380E-03	3.682E-03	2.081E-03	1.906E-03	7.284E-04	2.050E-03	1.118E-04	2.440E-04
3.20	7.695E-03	3.825E-03	2.172E-03	1.973E-03	7.623E-04	2.131E-03	1.143E-04	2.469E-04
3.30	7.996E-03	3.975E-03	2.253E-03	2.036E-03	7.925E-04	2.210E-03	1.170E-04	2.521E-04
3.40	8.288E-03	4.122E-03	2.340E-03	2.101E-03	8.216E-04	2.286E-03	1.190E-04	2.587E-04
3.50	8.605E-03	4.271E-03	2.425E-03	2.165E-03	8.516E-04	2.371E-03	1.208E-04	2.657E-04
3.60	8.915E-03	4.421E-03	2.510E-03	2.230E-03	8.824E-04	2.452E-03	1.222E-04	2.732E-04

Table C.4: Atmel ATtiny25V clocking current (Vdd 3.7V – 5.5V)

Vdd	16 MHz	8 MHz	6.4 MHz	2 MHz	1 MHz	800 kHz	128 kHz	16 kHz
3.70	9.232E-03	4.582E-03	2.599E-03	2.298E-03	9.122E-04	2.535E-03	1.235E-04	2.805E-04
3.80	9.550E-03	4.754E-03	2.687E-03	2.365E-03	9.436E-04	2.622E-03	1.246E-04	2.867E-04
3.90	9.908E-03	4.921E-03	2.776E-03	2.436E-03	9.746E-04	2.706E-03	1.258E-04	2.950E-04
4.00	1.025E-02	5.077E-03	2.864E-03	2.506E-03	1.006E-03	2.795E-03	1.269E-04	2.951E-04
4.10	1.057E-02	5.245E-03	2.962E-03	2.578E-03	1.040E-03	2.882E-03	1.280E-04	3.015E-04
4.20	1.092E-02	5.422E-03	3.054E-03	2.652E-03	1.074E-03	2.970E-03	1.278E-04	3.085E-04
4.30	1.128E-02	5.607E-03	3.153E-03	2.724E-03	1.108E-03	3.062E-03	1.279E-04	3.159E-04
4.40	1.161E-02	5.783E-03	3.250E-03	2.795E-03	1.143E-03	3.155E-03	1.270E-04	3.235E-04
4.50	1.198E-02	5.958E-03	3.348E-03	2.873E-03	1.177E-03	3.248E-03	1.266E-04	3.313E-04
4.60	1.235E-02	6.139E-03	3.445E-03	2.949E-03	1.210E-03	3.340E-03	1.254E-04	3.390E-04
4.70	1.272E-02	6.312E-03	3.549E-03	3.027E-03	1.248E-03	3.430E-03	1.240E-04	3.468E-04
4.80	1.311E-02	6.501E-03	3.653E-03	3.105E-03	1.284E-03	3.528E-03	1.230E-04	3.547E-04
4.90	1.349E-02	6.689E-03	3.757E-03	3.190E-03	1.319E-03	3.633E-03	1.229E-04	3.625E-04
5.00	1.387E-02	6.898E-03	3.860E-03	3.270E-03	1.358E-03	3.732E-03	1.237E-04	3.703E-04
5.10	1.427E-02	7.099E-03	3.965E-03	3.356E-03	1.396E-03	3.832E-03	1.248E-04	3.781E-04
5.20	1.464E-02	7.302E-03	4.075E-03	3.437E-03	1.432E-03	3.931E-03	1.262E-04	3.857E-04
5.30	1.506E-02	7.502E-03	4.181E-03	3.527E-03	1.471E-03	4.033E-03	1.277E-04	3.930E-04
5.40	1.547E-02	7.713E-03	4.297E-03	3.616E-03	1.512E-03	4.142E-03	1.293E-04	3.999E-04
5.50	1.588E-02	7.908E-03	4.415E-03	3.705E-03	1.552E-03	4.245E-03	1.311E-04	4.063E-04

Table C.5: Atmel ATtiny25V clocking current (VDD 3.7V – 5.5V)

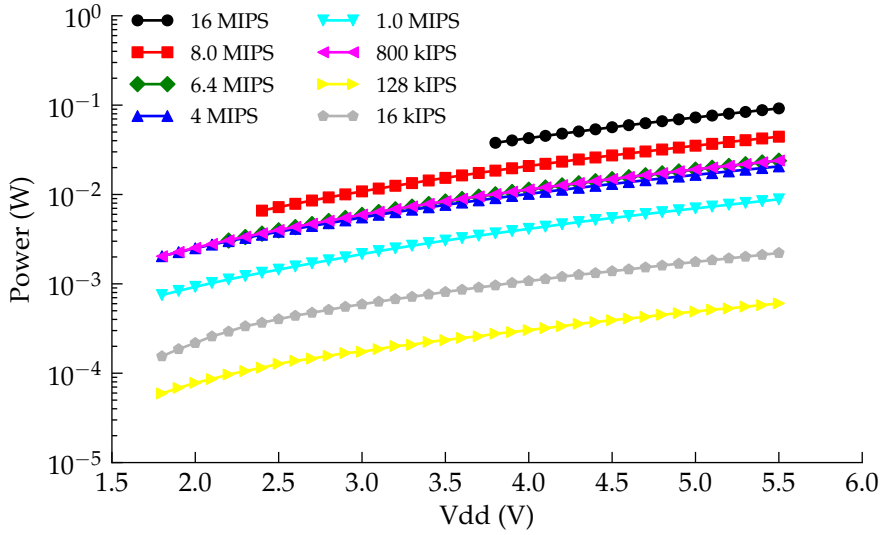


Figure C.6: Graph of power consumption of the Atmel ATtiny25V while clocking versus supply voltage.

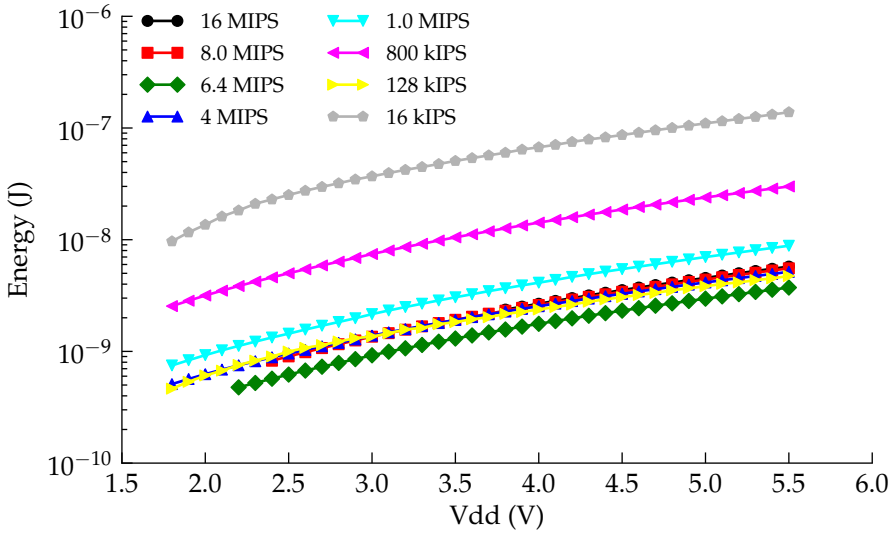


Figure C.7: Graph of energy consumed per instruction for the Atmel ATtiny25V versus supply voltage.

Vdd	32 MHz	16 MHz	8 MHz	4 MHz	2 MHz	1 MHz	500 kHz	31 kHz
1.80	N/A	1.250E-03	8.043E-04	5.163E-04	3.912E-04	3.290E-04	1.327E-04	4.679E-06
1.90	N/A	1.322E-03	8.483E-04	5.391E-04	4.066E-04	3.405E-04	1.367E-04	4.892E-06
2.00	N/A	1.395E-03	8.930E-04	5.619E-04	4.220E-04	3.520E-04	1.402E-04	5.063E-06
2.10	N/A	1.468E-03	9.382E-04	5.848E-04	4.375E-04	3.637E-04	1.436E-04	5.307E-06
2.20	N/A	1.542E-03	9.841E-04	6.079E-04	4.528E-04	3.754E-04	1.471E-04	5.546E-06
2.30	N/A	1.620E-03	1.031E-03	6.310E-04	4.683E-04	3.870E-04	1.503E-04	5.796E-06
2.40	N/A	1.698E-03	1.078E-03	6.544E-04	4.838E-04	3.987E-04	1.535E-04	5.996E-06
2.50	2.711E-03	1.773E-03	1.126E-03	6.778E-04	4.996E-04	4.105E-04	1.567E-04	6.282E-06
2.60	2.826E-03	1.848E-03	1.172E-03	7.008E-04	5.150E-04	4.223E-04	1.599E-04	6.536E-06
2.70	2.939E-03	1.919E-03	1.214E-03	7.232E-04	5.301E-04	4.337E-04	1.631E-04	6.734E-06
2.80	3.050E-03	1.990E-03	1.255E-03	7.456E-04	5.450E-04	4.449E-04	1.662E-04	6.912E-06
2.90	3.163E-03	2.062E-03	1.297E-03	7.685E-04	5.605E-04	4.565E-04	1.695E-04	7.109E-06
3.00	3.277E-03	2.134E-03	1.338E-03	7.916E-04	5.758E-04	4.683E-04	1.728E-04	7.349E-06
3.10	3.391E-03	2.207E-03	1.380E-03	8.152E-04	5.917E-04	4.802E-04	1.763E-04	7.534E-06
3.20	3.507E-03	2.281E-03	1.422E-03	8.388E-04	6.074E-04	4.919E-04	1.797E-04	7.601E-06
3.30	3.623E-03	2.355E-03	1.465E-03	8.626E-04	6.235E-04	5.039E-04	1.829E-04	7.604E-06
3.40	3.445E-03	2.263E-03	1.438E-03	8.666E-04	6.386E-04	5.273E-04	1.940E-04	7.638E-06
3.50	3.445E-03	2.263E-03	1.439E-03	8.663E-04	6.390E-04	5.276E-04	1.942E-04	7.582E-06
3.60	3.447E-03	2.264E-03	1.439E-03	8.663E-04	6.389E-04	5.275E-04	1.943E-04	7.616E-06

Table C.6: Microchip PIC16F1827 clocking current (Vdd 0V – 3.6V)

Vdd	32 MHz	16 MHz	8 MHz	4 MHz	2 MHz	1 MHz	500 kHz	31 kHz
3.70	3.449E-03	2.265E-03	1.439E-03	8.663E-04	6.392E-04	5.277E-04	1.945E-04	7.617E-06
3.80	3.449E-03	2.265E-03	1.440E-03	8.665E-04	6.393E-04	5.279E-04	1.947E-04	7.649E-06
3.90	3.450E-03	2.265E-03	1.440E-03	8.667E-04	6.397E-04	5.283E-04	1.950E-04	7.603E-06
4.00	3.451E-03	2.267E-03	1.441E-03	8.669E-04	6.396E-04	5.283E-04	1.952E-04	7.634E-06
4.10	3.452E-03	2.267E-03	1.441E-03	8.672E-04	6.401E-04	5.286E-04	1.955E-04	7.633E-06
4.20	3.453E-03	2.268E-03	1.442E-03	8.673E-04	6.401E-04	5.286E-04	1.957E-04	7.667E-06
4.30	3.454E-03	2.269E-03	1.442E-03	8.675E-04	6.404E-04	5.290E-04	1.960E-04	7.620E-06
4.40	3.454E-03	2.269E-03	1.443E-03	8.679E-04	6.405E-04	5.293E-04	1.961E-04	7.660E-06
4.50	3.456E-03	2.270E-03	1.443E-03	8.681E-04	6.410E-04	5.294E-04	1.964E-04	7.663E-06
4.60	3.457E-03	2.271E-03	1.444E-03	8.684E-04	6.412E-04	5.297E-04	1.966E-04	7.704E-06
4.70	3.458E-03	2.271E-03	1.444E-03	8.687E-04	6.416E-04	5.300E-04	1.969E-04	7.664E-06
4.80	3.458E-03	2.272E-03	1.445E-03	8.690E-04	6.417E-04	5.303E-04	1.972E-04	7.710E-06
4.90	3.460E-03	2.273E-03	1.446E-03	8.692E-04	6.421E-04	5.307E-04	1.976E-04	7.721E-06
5.00	3.461E-03	2.274E-03	1.446E-03	8.695E-04	6.424E-04	5.310E-04	1.979E-04	7.757E-06
5.10	3.463E-03	2.275E-03	1.447E-03	8.697E-04	6.428E-04	5.314E-04	1.983E-04	7.733E-06
5.20	3.464E-03	2.276E-03	1.448E-03	8.699E-04	6.433E-04	5.319E-04	1.986E-04	7.762E-06
5.30	3.465E-03	2.277E-03	1.448E-03	8.703E-04	6.437E-04	5.326E-04	1.991E-04	7.797E-06
5.40	3.467E-03	2.278E-03	1.449E-03	8.708E-04	6.444E-04	5.332E-04	1.996E-04	7.777E-06
5.50	3.468E-03	2.279E-03	1.450E-03	8.711E-04	6.449E-04	5.336E-04	2.002E-04	7.833E-06

Table C.7: Microchip PIC16F1827 clocking current (Vdd 3.7V – 5.5V)

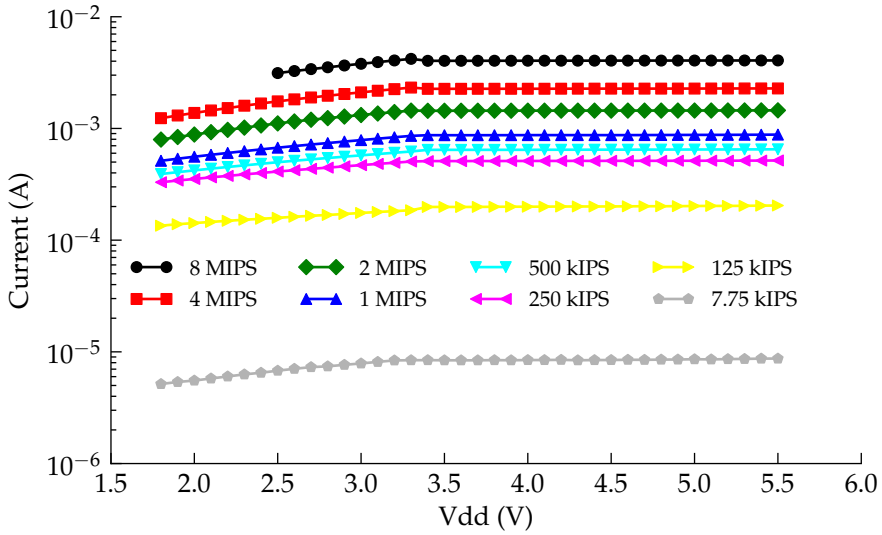


Figure C.8: Current consumption of the Microchip PIC16F1827 while clocking

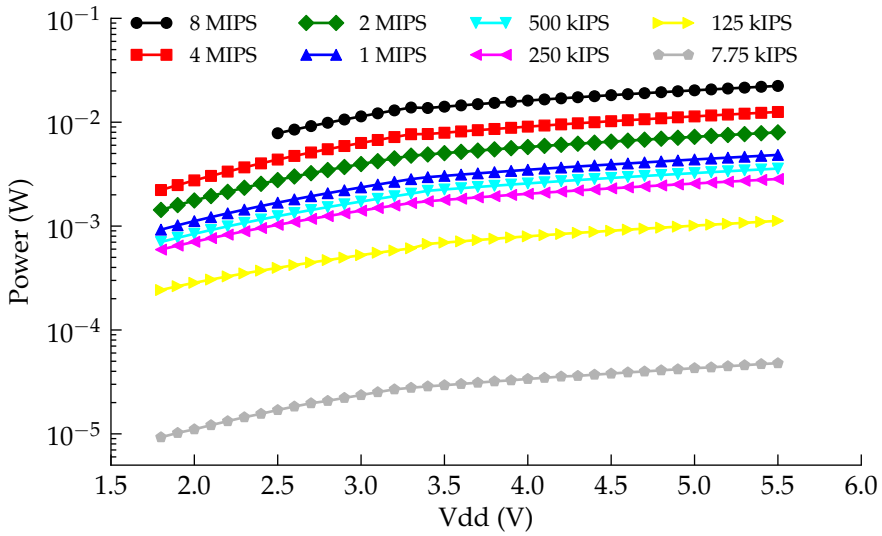


Figure C.9: Graph of power consumption of the Microchip PIC16F1827 while clocking versus supply voltage.

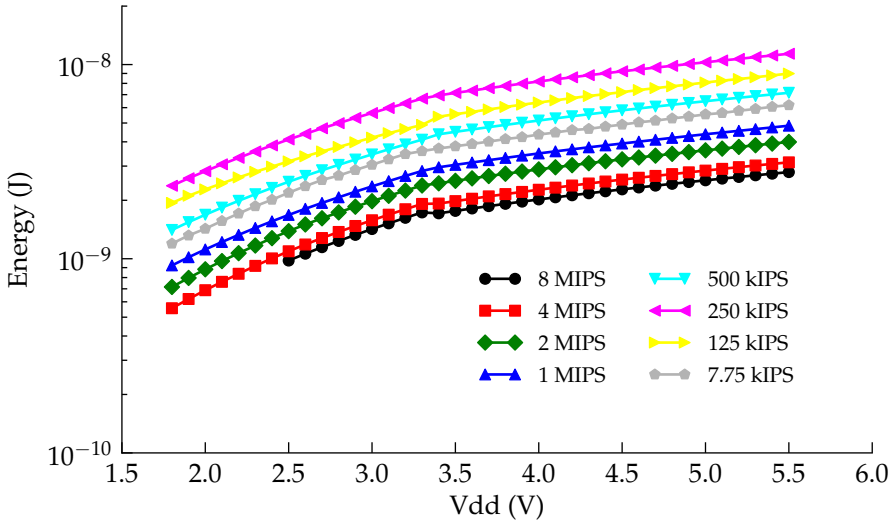


Figure C.10: Graph of energy consumed per instruction for the Microchip PIC16F1827 versus supply voltage.

```
6   //  * Disable DWEN fuse
7   //  * Disable BODLEVEL fuse
8
9   //Disable interrups
10    SREG = 0x00;
11
12  //Set PortB as outputs
13    DDRB = 0b00000000;
14
15  //Disable analog comparitor
16  ACSR = 0x80;
17
18  //Disable digital input
19  DIDR0 = 0xFF;
20
21  //Disable ADC before sleep
22  ADCSRA = 0x00;
23
24  //Set GPIOs as high
25    PORTB = 0xFF;
26
27  //Sequence to disable brown-
28  //out detect while sleeping
29    MCUCR = 0b00110000;
30
31    while(1)
32  {
33    asm("sleep");
```

```
34    }
35 }
```

Listing C.1: ATtiny25V Sleep Procedure

Clocking

```
1  #include <avr/io.h>
2
3  int main(void)
4  {
5      unsigned char rand;
6      SREG = 0x00;
7    //Set PortB as outputs
8      DDRB = 0b11111111;
9    //Set pins high
10     PORTB = 0xFF;
11     for(;;)
12     {
13         rand++;
14         asm("nop");
15     }
16
17 }
```

Listing C.2: ATtiny25V Clocking Procedure

Microchip 12F675

Code was written in MPLAB v8.6 in C and compiled using HI-TECH C v9.60. Chip programming was done using PICkit 2 Programmer software v2.61. [1]

Sleep

Microchip 16F1827

Code written in MPLAB v8.6 in C and compiled using HI-TECH C v 9.60. Chip programming was done using PICkit 2 Programmer software v2.61, however in order to program the 16F1827 with the PICkit 2 programmer a patch was applied to the device file list. This patch was retrieved from http://www.uploadarchief.net/files/download/pk2patch16x.zip on the 12th May 2011.

[1] Downloading to the 12F675s with MPLAB led to corruption of the internal oscillators, causing them fail.

Sleep

The specified sleep current of 30nA was not achievable. In an attempt to reach the specified current, the program was written in assembler. The assembler version of the code was used in the measurement data. The assembler version gave a lower sleep current than the C version, probably not as a result of the compiler but as due to a more thorough initialisation routine.

```
1  #include <htc.h>
2
3  __CONFIG(FOSC_INTOSC & WDTE_OFF & MCLRE_OFF & PWRTE_OFF
4    & BOREN_OFF & FCMEN_OFF & IESO_OFF & CLKOUTEN_OFF
5    & CP_OFF & CPD_OFF & LVP_ON & BORV_19 & STVREN_ON
6    & PLLEN_OFF & WRT_OFF);
7
8
9  void main(void)
10 {
11   //Set system clock to Internal osc block
12     SCS0 = 0;
13     SCS1 = 1;
14
15   //Set internal osc freq = 31kHz
16     IRCF3 = 0;
17     IRCF2 = 0;
18     IRCF1 = 0;
19     IRCF0 = 0;
20
21   //Disable interrupts
22     GIE = 0;
23
24     //Set all pins high (Tied to VDD via 10k)
25     PORTA = 0xFF;
26     PORTB = 0xFF;
27
28   //Put to sleep
29     SLEEP();
30     while(1)
31     {
32
33     }
34 }
```

Listing C.3: 16F1827 Sleep Procedure - HI-TECH C version

```
1  LIST    P=PIC16F1827
2  #include <P16F1827.INC>
3
```

```
4   __CONFIG _CONFIG1 , _FOSC_INTOSC & _WDTE_OFF & _PWRTE_OFF &
        _MCLRE_OFF & _CP_OFF & _CPD_OFF & _BOREN_OFF & _CLKOUTEN_OFF &
        _IESO_OFF & _FCMEN_OFF
5   __CONFIG _CONFIG2 , _WRT_OFF & _PLLEN_OFF & _STVREN_OFF & _BORV_19
        & _LVP_ON
6
7
8 ORG 0x0000 ; Specifies where to place the following code (which in
      this case is at the beginning of memory space )
9
10 START
11   ;Disable interrupts
12   BANKSEL INTCON
13   CLRF INTCON
14   ;Disable watchdog
15   BANKSEL WDTCON
16   CLRF WDTCON
17   ;Disable capacitive sensing
18   BANKSEL CPSCON0
19   CLRF CPSCON0
20   ;Disable modulation control
21   BANKSEL MDCON
22   CLRF MDCON
23   ;Disable perhiperal interrupts
24   BANKSEL PIE1
25   CLRF PIE2
26   ;Disable timer 1
27   BANKSEL T1CON
28   CLRF T1CON
29   ;Disable DAC
30   BANKSEL DACCON0
31   CLRF DACCON0
32   ;Disable ADC
33   BANKSEL ADCON0
34   CLRF ADCON0
35   BANKSEL ADCON1
36   CLRF ADCON1
37   ;Disable timers
38   BANKSEL T2CON
39   CLRF T2CON
40   BANKSEL T4CON
41   CLRF T4CON
42   BANKSEL T6CON
43   CLRF T6CON
44   ;Init PortA
45   BANKSEL PORTA ;
46   CLRF PORTA ;Init PORTA
47   BANKSEL LATA ;Data Latch
```

```
48    CLRF LATA ;
49    COMF LATA ;
50    BANKSEL ANSELA ;
51    CLRF ANSELA ;digital I/O
52    BANKSEL TRISA ;
53    CLRF TRISA ;SET AS OUTPUT
54    ;Init PortB
55    BANKSEL PORTB ;
56    CLRF PORTB ;Init PORTB
57    BANKSEL LATB
58    CLRF LATB ;
59    COMF LATB ;
60    BANKSEL ANSELB
61    CLRF ANSELB ;Make RB<7:0> digital
62    BANKSEL TRISB ;
63    ;and RB<3:0> as outputs
64    CLRF TRISB ;
65
66  LOOP ; Label this position (forms the start of a loop)
67    SLEEP
68    GOTO LOOP
69  END
```

Listing C.4: 16F1827 Sleep Procedure - MPASM assembler version

Clocking

```
1  #include <htc.h>
2
3  __CONFIG(FOSC_INTOSC & WDTE_OFF & MCLRE_OFF & PWRTE_OFF
4    & BOREN_OFF & FCMEN_OFF & IESO_OFF & CLKOUTEN_OFF
5    & CP_OFF & CPD_OFF & LVP_ON & BORV_19 & STVREN_ON
6    & PLLEN_ON & WRT_OFF);
7
8  unsigned int count;
9
10 void main(void)
11 {
12   //Set system clock to Internal osc block
13     SCS0 = 0;
14     SCS1 = 0;
15
16   //Set internal osc freq = 31kHz
17     IRCF3 = 1;
18     IRCF2 = 1;
19     IRCF1 = 1;
20     IRCF0 = 0;
```

```
21
22   //Disable interrupts
23     GIE = 0;
24
25     //Disable serial ports
26     SSP1CON1 = 0x00;
27     SSP2CON1 = 0x00;
28
29     ADCON0 = 0x00;
30
31     //Disable modulation
32     MDSRC = 0b10000000;
33     MDCARH = 0b10000000;
34     MDCARL = 0b10000000;
35
36     //Add pins as outputs
37     TRISA = 0x00;
38     ANSELA = 0x00;
39     TRISB = 0x00;
40     ANSELB = 0x00;
41
42     //Set all pins high (Tied to VDD via 10k)
43     PORTA = 0xFF;
44     PORTB = 0xFF;
45
46   //Put to sleep
47     while(1)
48     {
49       count++;
50     }
51 }
```

Listing C.5: PIC16F1827 Clocking Procedure

Freescale M9S08QG8

Code was written using Freescale's bundled IDE, CodeWarrior v5.90, and downloaded using a supplied USB demo board (DEMO9S08QG8E).

Sleep

```
1 #include <hidef.h> /* for EnableInterrupts macro */
2 #include "derivative.h" /* include peripheral declarations */
3
4 void main(void) {
5
6
```

```
7   SOPT1 = 0b00100000;
8   //         |||    ||
9   //         |||    |\- RESET Pin Enable (0)
10  //         |||    \-- Background Debug Mode Pin Enable (1)
11  //         ||\------- Stop Mode Enable (0)
12  //         |\-------- COP Watchdog Timeout (1)
13  //         \--------- COP Watchdog Enable (1)
14
15  //Enable low power bit
16  ICSC2 = 0b01001000;
17
18  //Disable Low Voltage Detect
19  SPMSC1 = 0x00;
20
21  //Enable power down control
22  //Disable partial power down
23  SPMSC2 = 0x02;
24
25  for(;;)
26  {
27    //Enter sleep mode
28    _Stop;
29  }
30 }
```

Listing C.6: M9S08QG8 Sleep Procedure

Appendix D
Electrolyte Impedance Measurement

This section contains supplementary data to the measurements made in chapter 9. It details some of the testing done to ensure that the measurement setup was providing expected results, and test the electrodes for defects. It also contains measurements not presented in the thesis body. They have been included purely for interests sake.

Due to the impedance drop at low frequencies that was encountered when measuring the various solutions presented in chapter 9, some testing of the measurement setup was done. Figure D.1 shows the configuration used to test the electrode. The electrodes were included to determine whether they themselves were contributing to the impedance magnitude drop at low frequency.

Figures D.2 and D.3 show that the measurement setup *is* capable of measuring a target 10 kΩ resistance placed between electrode one and two. These measurements include the electrode array in the loop so any effect from the internal wiring will be evident here.

Figures D.4 and D.5 show the effect of submerging the electrode array in liquid. The resistors from the previous test are still in place, and therefore are also submerged. The resistance has dropped slightly, as expected, but no impedance deviation at low frequency is evident. This rules out the possibility of a wet electrode array having an effect on the measurement results due to leakage.

Figures D.6 and D.7 shows the effect of adding salt to the solution. That the characteristic impedance drop at low frequencies is evident. This result adds weight to the theory that it is ions in the solution that is responsible for the impedance magnitude drop at low frequency.

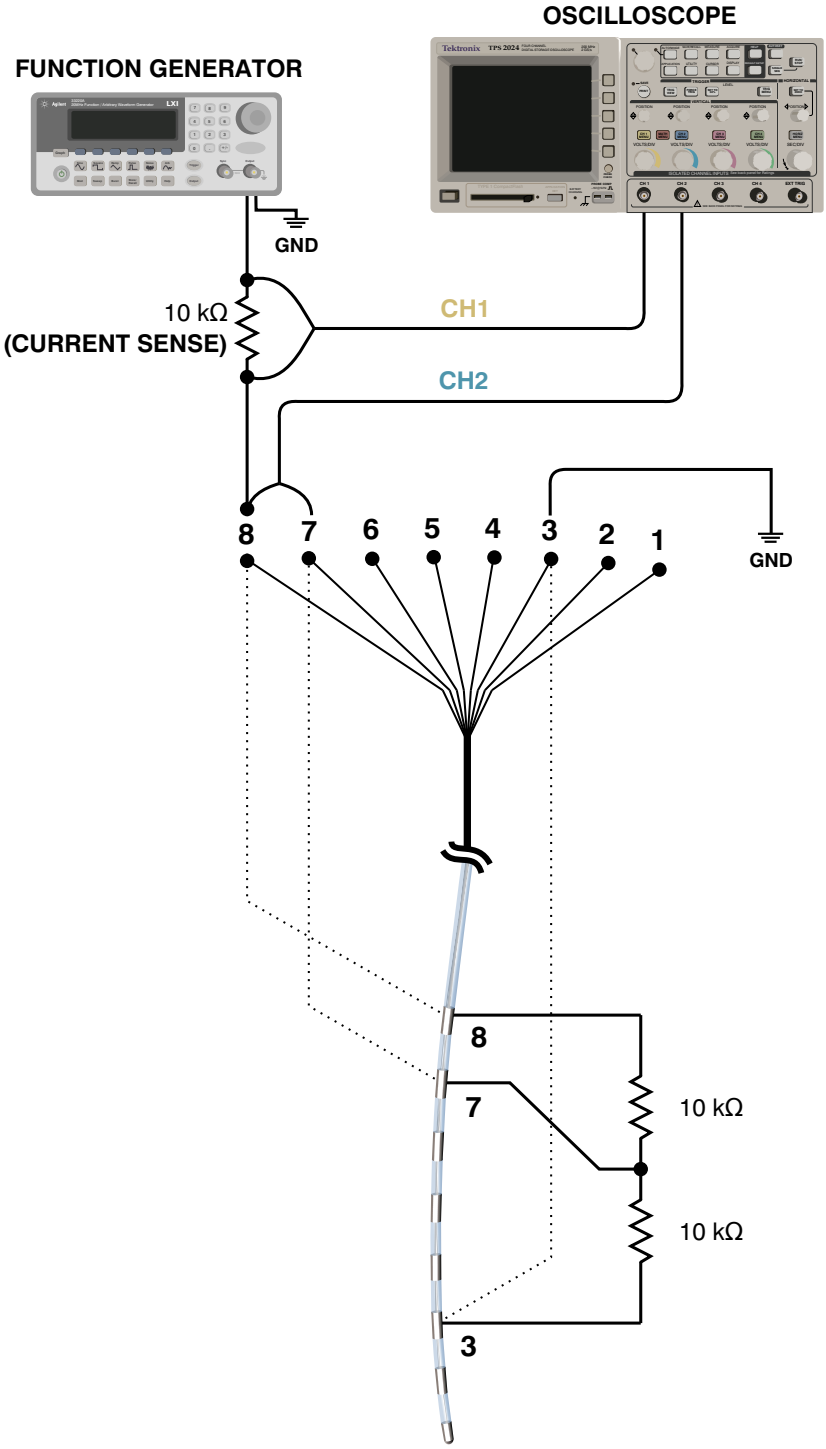


Figure D.1: Diagram showing the instruments and electrode configuration used to test the measurement setup.

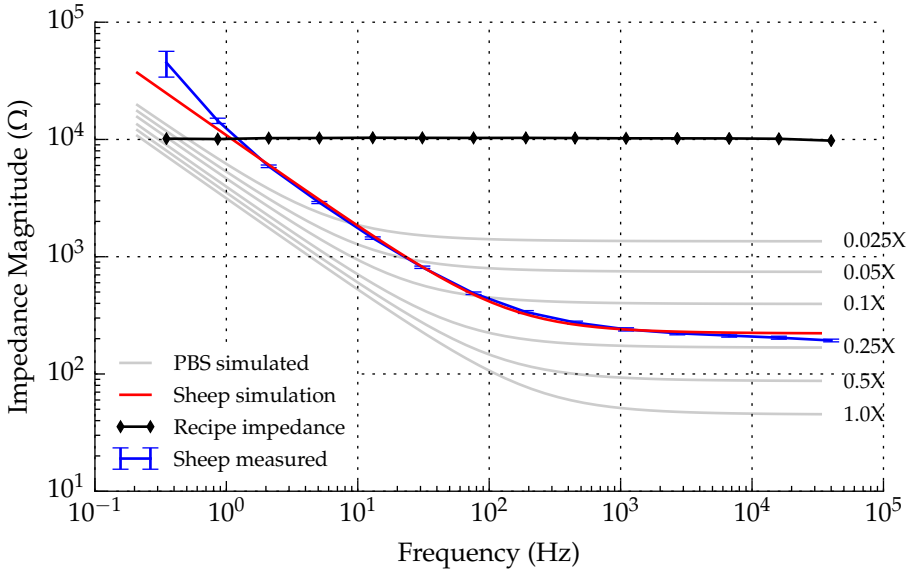


Figure D.2: Graph of impedance magnitude versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five.

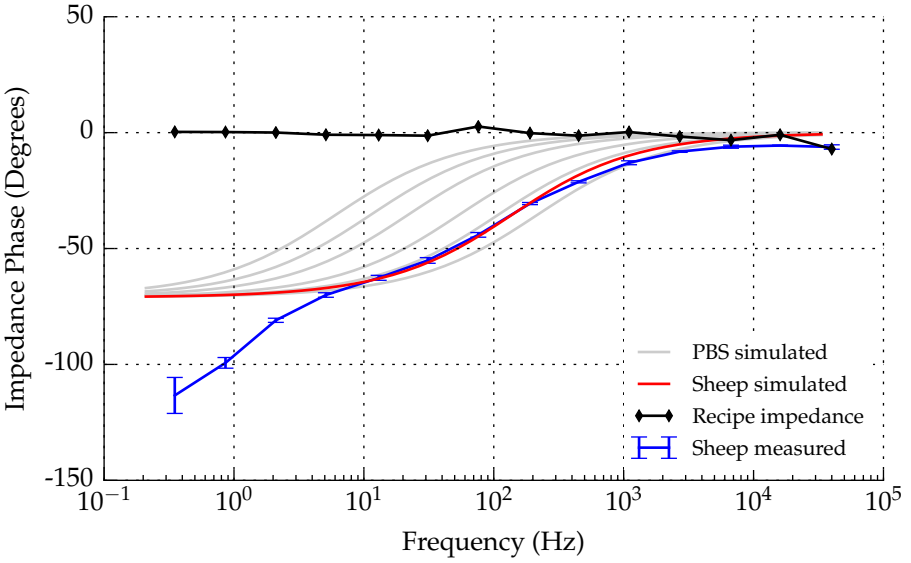


Figure D.3: Graph of impedance phase versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five.

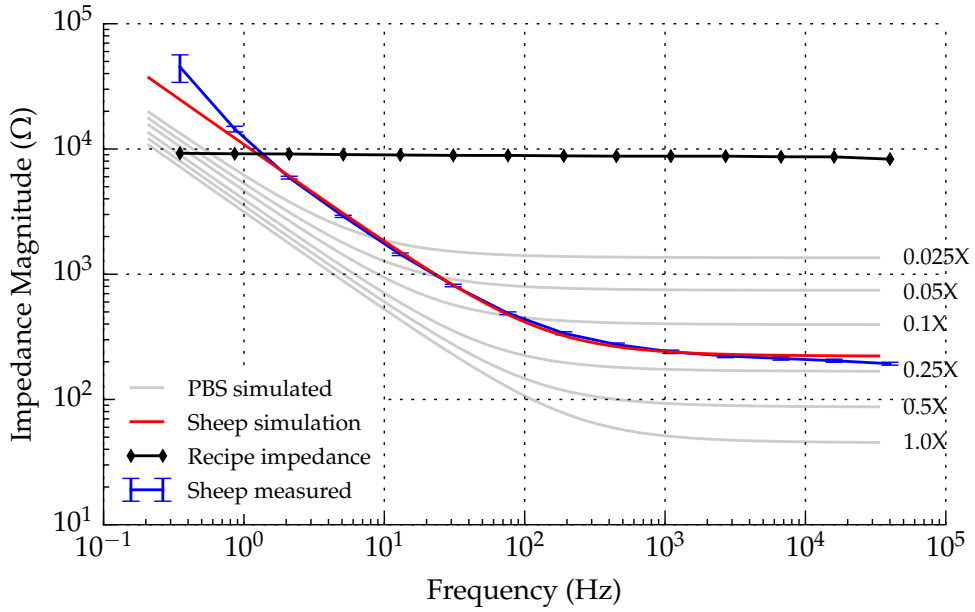


Figure D.4: Graph of impedance magnitude versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five. The electrodes and resistors are submerged in distilled water.

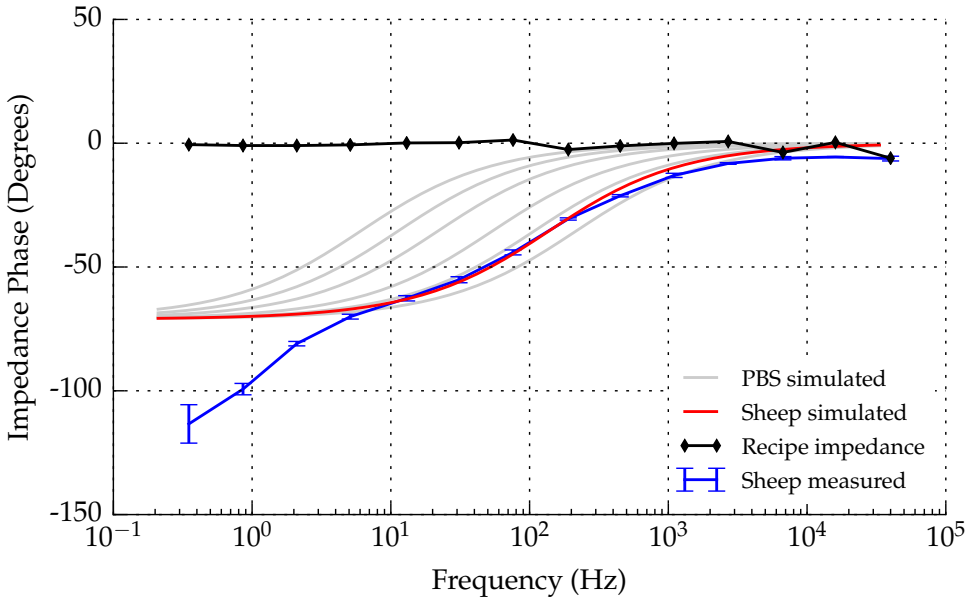


Figure D.5: Graph of impedance phase versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five. The electrodes and resistors are submerged in distilled water.

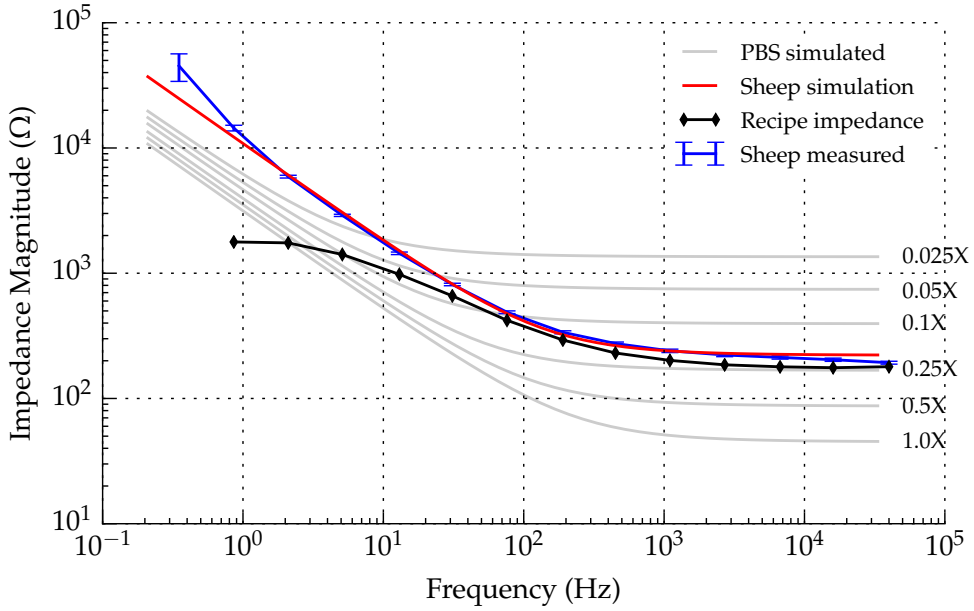


Figure D.6: Graph of impedance magnitude versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five. The electrodes and resistors are submerged in saline.

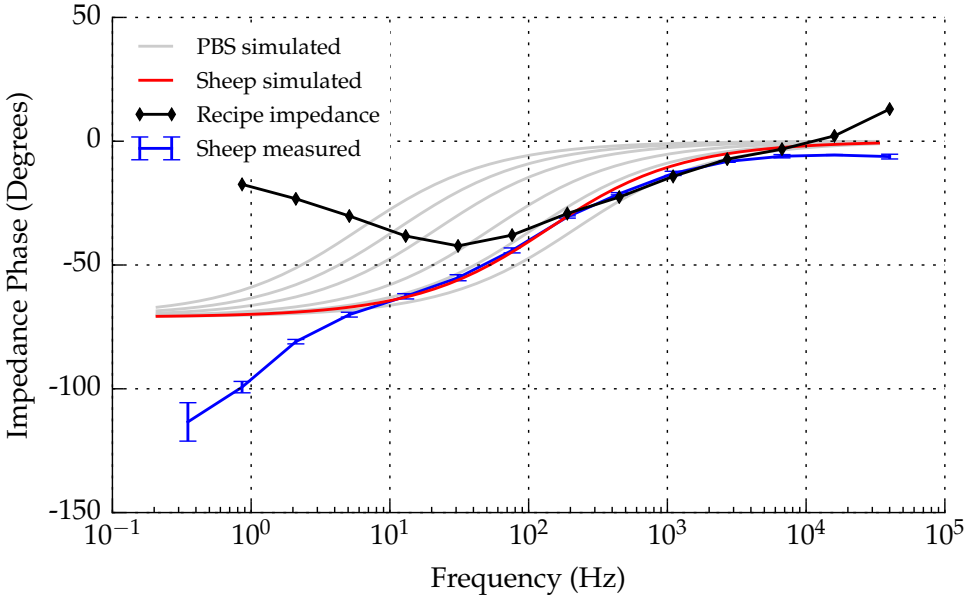


Figure D.7: Graph of impedance phase versus frequency (log-log) for a 10 kΩ resistor placed between electrodes one and two, and another 10 kΩ resistor placed between electrodes two and five. The electrodes and resistors are submerged in saline.

Appendix E

Interface Simulation Scripts

Source code for interface modelling used here can be found on GitHub at:
http://github.com/MarkHedleyJones
Parts of the code used to generate and run a simulation using ngSPICE are shown below.

```
# Custom libraries
import lib_functions
import lib_simulate

# Math helpers
import numpy as np
import mpmath

def simulate_model(conc=None,
                   measurements=None,
                   params=None,
                   res=0.1):
    """Runs the model and returns the results as a numpy array -
        pass the solution concentration."""
    #=========================================================================
    # Set basic simulation parameters
    #=========================================================================
    fmin = 20e-3  # Set frequency bandwidth for CPE building
    fmax = 1e5


    cond = lib_simulate.pbs_conductivity(conc)
    electrodes = 8

    #=========================================================================
    # Build spice circuit simulation components
    #=========================================================================
    Rr, Rv = lib_simulate.ladderResistorValues(cond)

    ladderSubckt, ladderName = lib_simulate.generate_ladder(electrodes=electrodes,
                                                            margin=3,
                                                            depth=5,
                                                            Rr_electrode=Rr,
                                                            Rv_commence=Rv)

    if params is None:
        params = lib_simulate.get_defaultParams()
        print("No params passed - DEFAULTS WILL BE USED!")

    fracpoleSubckt, fracpoleName = lib_simulate.generate_fracpoleSubcircuit(conc,
                                                                            fmin,
                                                                            fmax)

    #=========================================================================
    # Define the measurement points
    #=========================================================================
```

```
    if measurements is None:
        measurements = [('v_mag', 'VR(input,el6)'),
                        ('v_phi', 'VI(input,el6)'),
                        ('current', 'i(V1)')]

    #========================================================================
    # Build the spice file
    #========================================================================
    spice_ckt = "circuit\n"
    spice_ckt += ladderSubckt
    spice_ckt += fracpoleSubckt
    spice_ckt += ".SUBCKT interface a b\n"
    spice_ckt += "X1 a mid fracpole\n"
    spice_ckt += "Rs mid b " + str(params['rs']) + "\n"
    spice_ckt += ".ENDS interface\n"
    spice_ckt += "\n"
    spice_ckt += "**************************************\n"
    spice_ckt += "*          Circuit description       *\n"
    spice_ckt += "**************************************\n"
    spice_ckt += "XLadder el1 el2 el3 el4 el5 el6 el7 el8 resistorLadder\n"
    spice_ckt += "\n* Interface models from (Water to Electrode)\n"
    spice_ckt += "XInterface2 em2 el2 interface\n"
    spice_ckt += "XInterface7 em7 el7 interface\n"
    spice_ckt += "\n* Connections to power\n"
    spice_ckt += "R_IN input em7 0\n"
    spice_ckt += "R_OUT 0 em2 0\n"
    spice_ckt += "\n"
    spice_ckt += "\n* Power supply\n"
    spice_ckt += "V1 input 0 0 AC 1\n"
    spice_ckt += "\n"
    spice_ckt += "**************************************\n"
    spice_ckt += "*           Simulation options       *\n"
    spice_ckt += "**************************************\n"
    spice_ckt += ".control\n"
    spice_ckt += "set appendwrite\n"
    spice_ckt += "AC DEC 10 0.05 10000\n"
    spice_ckt += "wrdata output_spice "
    for measurement in measurements:
        spice_ckt += measurement[1] + ' '
    spice_ckt += "\n"
    spice_ckt += ".endc\n"
    spice_ckt += ".END"
    #========================================================================
    # Simulate the circuit and return the results
    #========================================================================
    simulation = lib_simulate.simulate_spice(spice_ckt)


    voltages = map(lambda (r, i): mpmath.mpc(r, i), zip(simulation[1], simulation[3]))
    currents = map(lambda (r, i): mpmath.mpc(r, i), zip(simulation[5], simulation[6]))
    data = np.array(zip(simulation[0],
                        voltages,
                        currents),
                        dtype={'names':('frequency',
                                        'voltage',
                                        'current'),
                               'formats':('f', 'complex', 'complex')})
    return data


# Simulate a solution of 0.5X PBS
concentration = 0.5
result = simulate_model(concentration)

# Print the results
for frequency in result['frequency']:
    print(result['frequency'], result['voltage'], result['current'])
```

Listing E.1: Example Python script to generate and run a simulation of the interface

```
# Import required libraries
import numpy as np
```

```
3  import math
4  import mpmath
5  import subprocess
6  import threading
7  import os
8
9  # Optimised parameter repository
10 optimise = {}
11 optimise['cpe'] = {'slope':-0.79052566,
12                   'mag':{'a':3284,
13                          'b':-0.158}}
14 optimise['rs'] = {'a': 13.38,
15                  'b':-0.8397}
16
17 command = None
18
19 def ladderResistorValues(conductivity):
20     """
21     Given a solution conductivity return suitable Rr Rv values
22     (radial resistance (electrode) and vertical resistor commencing value,
23     respecitvely)
24     """
25     Rr_b = 0.407  # Determined from optimisation
26     Rv_b = 3.71  # Determined from optimisation
27     Rr = Rr_b / conductivity
28     Rv = Rv_b / conductivity
29     return (Rr, Rv)
30
31
32 def rs(conc, a=None, b=None):
33     """
34     Given a concentration of PBS returns the value of Rs (The series resistance
35     in the model)
36     """
37     if a is None:
38         a = optimise['rs']['a']
39     if b is None:
40         b = optimise['rs']['b']
41     return a * math.pow(conc, b)
42
43
44 def combine_subcircuitParallel(subCktA_name, subCktB_name, newName, rs=0.0):
45     """
46     Combines the provided two port subcircuits in parallel for easy
47     use in a spice file.
48     """
49     out = ""
50     out += "*************************************\n"
51     out += "*    Combine " + subCktA_name + " and " + subCktB_name + " in\n"
52     out += "*    parallel to provide " + newName + "\n"
53     out += "*************************************\n"
54     out += ".SUBCKT " + newName + " a b\n"
55     out += "XC1 a mid " + subCktA_name + "\n"
56     out += "XC2 a mid " + subCktB_name + "\n"
57     out += "R1 mid b " + str(rs) + "\n"
58     out += ".ENDS " + newName + "\n"
59     return out
60
61
62 def generate_faradaicSubcircuit(conc,
63                                  params={},
64                                  i0=None,
65                                  n=None,
66                                  cm=None,
67                                  rm=None,
68                                  memristor=True):
69     """
70     Generates the ngspice compatible subcircuit named faradaic that simulates
71     the faradic component in the interface. This includes the diode and
72     memristor branches
73     """
74
75     # Load defaults
```

```
76      i0 = 3.5e-7
77      n = -0.025 * float(conc) + 0.164
78      cm = 2.316e-04 + 1.224e-04 * math.exp(-conc / 6.832e-01)
79      rm = 10000000.0
80
81      if 'i0' in params:
82          i0 = params['i0']
83      if 'n' in params:
84          n = params['n']
85      if 'cm' in params:
86          cm = params['cm']
87      if 'rm' in params:
88          rm = params['rm']
89
90      if memristor:
91          out = [".param i0=" + str(i0),
92                 ".param cm=" + str(cm),
93                 ".param rm=" + str(rm),
94                 ".param n=" + str(n),
95                 ".subckt faradaic n1 n2",
96                 "Bdm1 n1 n2 I=i0*exp(v(n1,n2)/n)",
97                 "Bdm2 n2 n1 I=i0*exp(v(n2,n1)/n)",
98                 "Bdm1cpy 0 mset I=i0*exp(v(n1,n2)/n)",
99                 "Bdm2cpy mset 0 I=i0*exp(v(n2,n1)/n)",
100                "C_M mset 0 cm",
101                "R_M mset 0 rm",
102                ".ends"]
103     else:
104         out = [".param i0=" + str(i0),
105                ".param cm=" + str(cm),
106                ".param rm=" + str(rm),
107                ".param n=" + str(n),
108                ".subckt faradaic n1 n2",
109                "Bdm1 n1 n2 I=i0*(1-v(mset))*exp(v(n1,n2)/n)",
110                "Bdm2 n2 n1 I=i0*(1+v(mset))*exp(v(n2,n1)/n)",
111                "Bdm1cpy 0 mset I=i0*(1-v(mset))*exp(v(n1,n2)/n)",
112                "Bdm2cpy mset 0 I=i0*(1+v(mset))*exp(v(n2,n1)/n)",
113                "C_M mset 0 cm",
114                "R_M mset 0 rm",
115                ".ends"]
116
117     tmp = ""
118     tmp += "**************************************\n"
119     tmp += "*        Faradaic branch start       *\n"
120     tmp += "**************************************\n"
121     return (tmp + "\n".join(out) + "\n\n", 'faradaic')
122
123
124 def param_cpe_slope(conc):
125     """
126     Return the magnitude and slope of the CPE.
127     """
128     slope = optimise['cpe']['slope']
129     mag = optimise['cpe']['mag']['a'] * math.pow(conc,
130                                                  optimise['cpe']['mag']['b'])
131     return (mag, slope)
132
133 def get_cpeParams(freq_min, freq_max, m, perDecade=3):
134     """
135     Calculates the parameters required to create a sufficiently accurate
136     Constant Phase Element (CPE) from the given parameters.
137
138     Returns (pts, k, y_theta) where:
139         pts:     an array of frequencies at which to place RC elements
140         k:       a parameter that controls multiplicity
141         y_theta: another parameter, not sure of its exact definition
142     """
143     # Extend the range so no funny stuff happens at the endpoints
144     freq_min /= 1000
145     freq_max *= 1000
146
147     # Calculate the number of elements to place in this range
148     numPts = (math.log10(freq_max) - math.log10(freq_min)) * perDecade
```

```
149
150     # Calculate the frequency scaling factor
151     k_f = math.exp((math.log(freq_max) - math.log(freq_min)) / numPts)
152
153     # Generate the x positions for cpe elements
154     pts = []
155     for i in range(int(numPts) + 1):
156         pts.append(freq_min * math.pow(k_f, i))
157
158     # Determine k - the multiplicity factor
159     k = math.pow(k_f, 1 / m)
160
161     # k gets used here to create the y_theta variables
162     # which are passed to generate_fracpoleSubcircuit and used
163     # to choose the value of capacitance in each RC branch.
164     y_theta = ((math.pi / (m * math.log(k))) *
165                mpmath.sec(0.5 * math.pi * (1 - (2 / m))))
166
167     return (pts, y_theta)
168
169
170 def generate_fracpoleSubcircuit(conc, fmin, fmax, m=1.34):
171     """
172     Generates the ngspice compatible subcircuit named fracpole ready for
173     inclusion into a spice file
174     """
175     slope_a, slope_b = param_cpe_slope(conc)
176     pts, y_theta = get_cpeParams(fmin, fmax, m)
177
178     out = ""
179     out += "**************************************\n"
180     out += "*           Fracpole/CPE start         *\n"
181     out += "**************************************\n"
182
183     fracpoleElements = []
184
185     for point in pts:
186         omega = 2 * math.pi * point
187         R = slope_a * math.pow(point, slope_b)
188         C = math.pow((R / (y_theta * R)), m) / (omega * R)
189         fracpoleElements.append({'frequency': point, 'R': R, 'C': C})
190
191     out += ".SUBCKT fracpole a b\n"
192     for num, facpoleElement in enumerate(fracpoleElements):
193         out += ("R" + str(num) + " a " + str(num + 1)
194                             + " " + str(facpoleElement['R']) + "\n")
195         out += ("C" + str(num) + " " + str(num + 1)
196                             + " b " + str(facpoleElement['C']) + "\n")
197     out += ".ENDS fracpole\n"
198     out += "\n"
199
200     return (out, 'fracpole')
201
202
203 def pbs_conductivity(conc):
204     """
205     Converts a concentration of PBS into a conductivity according to a
206     least squares fit of the solutions used - fit code follows...
207     """
208     #========================================================================
209     # Linear relationship
210     #========================================================================
211     m = 1.67296736e-02  # Determined from optimisation
212     c = 8.54665149e-05  # Determined from optimisation
213     return m * conc + c
214
215
216 def pbs_resistivity(conc):
217     return 1 / pbs_conductivity(conc)
218
219
220 def generate_ladder(electrodes, margin, depth, Rr_electrode, Rv_commence):
221     """
```

```
222     Generates a resistor ladder circuit for insertion into a spice file.
223     Parameters:
224         electrodes:   Integer number of electrodes to create the ladder around.
225         margin:       Integer number of dummy rows at each end of ladder (helps
226                       to prevent end effects).
227         depth:        How deep (column-wise) to generate the ladder.
228         Rr_electrode: Radial resistance at electrode.
229         Rv_commence:  Initial value of the vertical resistor.
230     """
231     # Derived from the value of Rr_insulator
232     Rr_insulator = Rr_electrode * (3 / 4.0)
233
234     # Populate the latitude resistor value array
235     Rv = []
236     for i in range(depth):
237         Rv.append(float(Rv_commence) / pow(4, i))
238
239     # Keep track of which nodes correspond to which electrodes
240     nodes = {}
241
242     # Generate the resistor ladder
243     out = ""
244     out += "****************************************\n"
245     out += "*          Resistor ladder Start       *\n"
246     out += "****************************************\n"
247     out += ".SUBCKT resistorLadder"
248     for electrode in range(electrodes):
249         out += " e" + str((electrode + 1))
250     out += "\n"
251
252     # Figure out which nodes correspond to electrodes
253     for row in range(((electrodes + electrodes - 1) * 2 - 1) + 4 * margin):
254         for col in range(depth):
255             if col == 0 and row % 2 == 0:
256                 actRow = row / 2
257                 if actRow < (margin + (electrodes * 2) - 1):
258                     segment = (actRow - margin)
259                     if segment % 2 == 0 and segment >= 0:
260                         nodes[col + (int(row / 2) * 5) + 1] = (int(segment / 2)
261                                                                + 1)
262
263     # Step over each component adding as necessary
264     for row in range(((electrodes + electrodes - 1) * 2 - 1) + 4 * margin):
265         for col in range(depth):
266
267             fromNode = col + (int(row / 2) * 5) + 1
268
269             if (row % 2) == 0:
270                 if (row / 2 >= margin and
271                     row / 2 < (margin + (electrodes * 2) - 1) and
272                     (row / 2 - margin) % 2 != 0):
273                     value = Rr_insulator
274                 else:
275                     value = Rr_electrode
276
277                 component = "RRAD_" + str(row + 1) + "_" + str(col + 1)
278                 if col == (len(Rv) - 1):
279                     toNode = 1000
280                 else:
281                     toNode = col + (int(row / 2) * 5) + 2
282             else:
283                 value = Rv[col]
284                 toNode = col + (int(row / 2 + 1) * 5) + 1
285                 component = "RVERT_" + str(row + 1) + "_" + str(col + 1)
286
287             if fromNode in nodes:
288                 fromNode = 'e' + str(nodes[fromNode])
289
290             if toNode in nodes:
291                 toNode = 'e' + str(nodes[toNode])
292             out += (str(component) + ' ' +
293                     str(fromNode) + ' ' +
294                     str(toNode) + ' ' +
```

```
295                         str(value) + "\n")
296
297     out += ".ENDS resistorLadder\n"
298     out += "\n"
299     return (out, 'resistorLadder')
300
301
302 class Command(object):
303     """
304     Responsible for running a subprocess but with the ability to terminate
305     that process if and when if executes for a pre-determined amount of time.
306     """
307     def __init__(self, cmd):
308         self.cmd = cmd
309         self.process = None
310
311     def run(self, timeout):
312         def target():
313             with open(os.devnull, 'w') as dnull:
314                 self.process = subprocess.Popen(self.cmd,
315                                                 stdout=dnull,
316                                                 stderr=subprocess.STDOUT,
317                                                 shell=True)
318             self.process.communicate()
319
320         thread = threading.Thread(target=target)
321         thread.start()
322
323         thread.join(timeout)
324         if thread.is_alive():
325             print 'Terminating process'
326             self.process.terminate()
327             thread.join()
328             return False
329         else:
330             return True
331
332
333 def simulate_spice(circuit,
334                    filename=None,
335                    outputName=None,
336                    debug=False,
337                    cleanup=False,
338                    timeout=None):
339     """
340     Takes a string representing a spice circuit, saves it to a file,
341     runs it through ngspice and returns the results as an array
342     """
343     global command
344
345     # Sort out some defaults
346     if filename is None:
347         filename = 'spicemodel'
348     if outputName is None:
349         outputName = 'output_spice'
350
351     outputName += '.data'
352     filename += '_netlist.spice'
353
354     filename = '../tmp/' + filename
355
356     try:
357         # Clean up temp files
358         os.remove(filename)
359         os.remove(outputName)
360     except OSError:
361         pass
362
363     with open(filename, 'w') as f:
364         f.write(circuit)
365
366     # Run the simulation
367     if debug == False:
```

```
        # This will use the timeout provided
        if command is None:
            command = Command("ngspice -bp " + filename)
        if command.run(timeout) == False:
            print "A errant simulation terminated"
            return None
    else:
        subprocess.call(["ngspice", "-bp", filename])

    # Fetch the results and place into an array
    data = None
    with open(outputName, 'r') as f:

        lines = f.readlines()
        for i, line in enumerate(lines):
            parts = [float(part) for part in line.split()]

            if data is None:
                data = np.zeros((len(parts), len(lines)))

            for j, part in enumerate(parts):
                data[j][i] = part

    # Clean up so the working directory doesnt get clogged up
    if cleanup:
        try:
            os.remove(filename)
            os.remove(outputName)
            print("Cleaned up temporary files")
        except OSError:
            pass

    return data


def get_defaultParams():
    return {'cm': 0.0002,  # Memristor responce capacitance
            'rm': 1500000,  # Memristor memory fade resistance
            'rs': 400  # Series resistance
            }
```

Listing E.2: Python library for SPICE model generation of the interface model

```
electrodeModel
**************************************
*         Faradaic branch start      *
**************************************
.SUBCKT faradaic n1 n2
.PARAM Vt=0.025875
.PARAM i0=2.75674884748e-12
.PARAM n=1.35992195766
.PARAM nVt=n*Vt
Bdm1 n1 n2 I=i0*exp(v(n1,n2)/nVt)
Bdm2 n2 n1 I=i0*exp(v(n2,n1)/nVt)
R_b n1 n2 1e10
.ENDS faradaic
**************************************
*          Resistor ladder start     *
**************************************
.SUBCKT ladder e1 e2 e3 e4 e5 e6 e7 e8
RRAD_1_1 1 2 24.2043775561
RRAD_1_2 2 3 24.2043775561
RRAD_1_3 3 4 24.2043775561
RRAD_1_4 4 5 24.2043775561
RRAD_1_5 5 1000 24.2043775561
RVERT_2_1 1 6 220.634498116
RVERT_2_2 2 7 55.158624529
RVERT_2_3 3 8 13.7896561322
RVERT_2_4 4 9 3.44741403306
RVERT_2_5 5 10 0.861853508265
RRAD_3_1 6 7 24.2043775561
RRAD_3_2 7 8 24.2043775561
```

```
RRAD_3_3  8  9  24.2043775561
RRAD_3_4  9  10  24.2043775561
RRAD_3_5  10  1000  24.2043775561
RVERT_4_1  6  11  220.634498116
RVERT_4_2  7  12  55.158624529
RVERT_4_3  8  13  13.7896561322
RVERT_4_4  9  14  3.44741403306
RVERT_4_5  10  15  0.861853508265
RRAD_5_1  11  12  24.2043775561
RRAD_5_2  12  13  24.2043775561
RRAD_5_3  13  14  24.2043775561
RRAD_5_4  14  15  24.2043775561
RRAD_5_5  15  1000  24.2043775561
RVERT_6_1  11  e1  220.634498116
RVERT_6_2  12  17  55.158624529
RVERT_6_3  13  18  13.7896561322
RVERT_6_4  14  19  3.44741403306
RVERT_6_5  15  20  0.861853508265
RRAD_7_1  e1  17  24.2043775561
RRAD_7_2  17  18  24.2043775561
RRAD_7_3  18  19  24.2043775561
RRAD_7_4  19  20  24.2043775561
RRAD_7_5  20  1000  24.2043775561
RVERT_8_1  e1  21  220.634498116
RVERT_8_2  17  22  55.158624529
RVERT_8_3  18  23  13.7896561322
RVERT_8_4  19  24  3.44741403306
RVERT_8_5  20  25  0.861853508265
RRAD_9_1  21  22  18.1532831671
RRAD_9_2  22  23  18.1532831671
RRAD_9_3  23  24  18.1532831671
RRAD_9_4  24  25  18.1532831671
RRAD_9_5  25  1000  18.1532831671
RVERT_10_1  21  e2  220.634498116
RVERT_10_2  22  27  55.158624529
RVERT_10_3  23  28  13.7896561322
RVERT_10_4  24  29  3.44741403306
RVERT_10_5  25  30  0.861853508265
RRAD_11_1  e2  27  24.2043775561
RRAD_11_2  27  28  24.2043775561
RRAD_11_3  28  29  24.2043775561
RRAD_11_4  29  30  24.2043775561
RRAD_11_5  30  1000  24.2043775561
RVERT_12_1  e2  31  220.634498116
RVERT_12_2  27  32  55.158624529
RVERT_12_3  28  33  13.7896561322
RVERT_12_4  29  34  3.44741403306
RVERT_12_5  30  35  0.861853508265
RRAD_13_1  31  32  18.1532831671
RRAD_13_2  32  33  18.1532831671
RRAD_13_3  33  34  18.1532831671
RRAD_13_4  34  35  18.1532831671
RRAD_13_5  35  1000  18.1532831671
RVERT_14_1  31  e3  220.634498116
RVERT_14_2  32  37  55.158624529
RVERT_14_3  33  38  13.7896561322
RVERT_14_4  34  39  3.44741403306
RVERT_14_5  35  40  0.861853508265
RRAD_15_1  e3  37  24.2043775561
RRAD_15_2  37  38  24.2043775561
RRAD_15_3  38  39  24.2043775561
RRAD_15_4  39  40  24.2043775561
RRAD_15_5  40  1000  24.2043775561
RVERT_16_1  e3  41  220.634498116
RVERT_16_2  37  42  55.158624529
RVERT_16_3  38  43  13.7896561322
RVERT_16_4  39  44  3.44741403306
RVERT_16_5  40  45  0.861853508265
RRAD_17_1  41  42  18.1532831671
RRAD_17_2  42  43  18.1532831671
RRAD_17_3  43  44  18.1532831671
RRAD_17_4  44  45  18.1532831671
RRAD_17_5  45  1000  18.1532831671
```

```
RVERT_18_1 41 e4 220.634498116
RVERT_18_2 42 47 55.158624529
RVERT_18_3 43 48 13.7896561322
RVERT_18_4 44 49 3.44741403306
RVERT_18_5 45 50 0.861853508265
RRAD_19_1 e4 47 24.2043775561
RRAD_19_2 47 48 24.2043775561
RRAD_19_3 48 49 24.2043775561
RRAD_19_4 49 50 24.2043775561
RRAD_19_5 50 1000 24.2043775561
RVERT_20_1 e4 51 220.634498116
RVERT_20_2 47 52 55.158624529
RVERT_20_3 48 53 13.7896561322
RVERT_20_4 49 54 3.44741403306
RVERT_20_5 50 55 0.861853508265
RRAD_21_1 51 52 18.1532831671
RRAD_21_2 52 53 18.1532831671
RRAD_21_3 53 54 18.1532831671
RRAD_21_4 54 55 18.1532831671
RRAD_21_5 55 1000 18.1532831671
RVERT_22_1 51 e5 220.634498116
RVERT_22_2 52 57 55.158624529
RVERT_22_3 53 58 13.7896561322
RVERT_22_4 54 59 3.44741403306
RVERT_22_5 55 60 0.861853508265
RRAD_23_1 e5 57 24.2043775561
RRAD_23_2 57 58 24.2043775561
RRAD_23_3 58 59 24.2043775561
RRAD_23_4 59 60 24.2043775561
RRAD_23_5 60 1000 24.2043775561
RVERT_24_1 e5 61 220.634498116
RVERT_24_2 57 62 55.158624529
RVERT_24_3 58 63 13.7896561322
RVERT_24_4 59 64 3.44741403306
RVERT_24_5 60 65 0.861853508265
RRAD_25_1 61 62 18.1532831671
RRAD_25_2 62 63 18.1532831671
RRAD_25_3 63 64 18.1532831671
RRAD_25_4 64 65 18.1532831671
RRAD_25_5 65 1000 18.1532831671
RVERT_26_1 61 e6 220.634498116
RVERT_26_2 62 67 55.158624529
RVERT_26_3 63 68 13.7896561322
RVERT_26_4 64 69 3.44741403306
RVERT_26_5 65 70 0.861853508265
RRAD_27_1 e6 67 24.2043775561
RRAD_27_2 67 68 24.2043775561
RRAD_27_3 68 69 24.2043775561
RRAD_27_4 69 70 24.2043775561
RRAD_27_5 70 1000 24.2043775561
RVERT_28_1 e6 71 220.634498116
RVERT_28_2 67 72 55.158624529
RVERT_28_3 68 73 13.7896561322
RVERT_28_4 69 74 3.44741403306
RVERT_28_5 70 75 0.861853508265
RRAD_29_1 71 72 18.1532831671
RRAD_29_2 72 73 18.1532831671
RRAD_29_3 73 74 18.1532831671
RRAD_29_4 74 75 18.1532831671
RRAD_29_5 75 1000 18.1532831671
RVERT_30_1 71 e7 220.634498116
RVERT_30_2 72 77 55.158624529
RVERT_30_3 73 78 13.7896561322
RVERT_30_4 74 79 3.44741403306
RVERT_30_5 75 80 0.861853508265
RRAD_31_1 e7 77 24.2043775561
RRAD_31_2 77 78 24.2043775561
RRAD_31_3 78 79 24.2043775561
RRAD_31_4 79 80 24.2043775561
RRAD_31_5 80 1000 24.2043775561
RVERT_32_1 e7 81 220.634498116
RVERT_32_2 77 82 55.158624529
RVERT_32_3 78 83 13.7896561322
```

```
RVERT_32_4 79 84 3.44741403306
RVERT_32_5 80 85 0.861853508265
RRAD_33_1 81 82 18.1532831671
RRAD_33_2 82 83 18.1532831671
RRAD_33_3 83 84 18.1532831671
RRAD_33_4 84 85 18.1532831671
RRAD_33_5 85 1000 18.1532831671
RVERT_34_1 81 e8 220.634498116
RVERT_34_2 82 87 55.158624529
RVERT_34_3 83 88 13.7896561322
RVERT_34_4 84 89 3.44741403306
RVERT_34_5 85 90 0.861853508265
RRAD_35_1 e8 87 24.2043775561
RRAD_35_2 87 88 24.2043775561
RRAD_35_3 88 89 24.2043775561
RRAD_35_4 89 90 24.2043775561
RRAD_35_5 90 1000 24.2043775561
RVERT_36_1 e8 91 220.634498116
RVERT_36_2 87 92 55.158624529
RVERT_36_3 88 93 13.7896561322
RVERT_36_4 89 94 3.44741403306
RVERT_36_5 90 95 0.861853508265
RRAD_37_1 91 92 24.2043775561
RRAD_37_2 92 93 24.2043775561
RRAD_37_3 93 94 24.2043775561
RRAD_37_4 94 95 24.2043775561
RRAD_37_5 95 1000 24.2043775561
RVERT_38_1 91 96 220.634498116
RVERT_38_2 92 97 55.158624529
RVERT_38_3 93 98 13.7896561322
RVERT_38_4 94 99 3.44741403306
RVERT_38_5 95 100 0.861853508265
RRAD_39_1 96 97 24.2043775561
RRAD_39_2 97 98 24.2043775561
RRAD_39_3 98 99 24.2043775561
RRAD_39_4 99 100 24.2043775561
RRAD_39_5 100 1000 24.2043775561
RVERT_40_1 96 101 220.634498116
RVERT_40_2 97 102 55.158624529
RVERT_40_3 98 103 13.7896561322
RVERT_40_4 99 104 3.44741403306
RVERT_40_5 100 105 0.861853508265
RRAD_41_1 101 102 24.2043775561
RRAD_41_2 102 103 24.2043775561
RRAD_41_3 103 104 24.2043775561
RRAD_41_4 104 105 24.2043775561
RRAD_41_5 105 1000 24.2043775561
.ENDS ladder
****************************************
*            Fracpole/CPE start        *
****************************************
.SUBCKT displacement a b
R0 a 1 42769491660.0
C0 1 b 0.000359405061629
R1 a 2 23314469349.4
C1 2 b 0.00030602676341
R2 a 3 12709163937.9
C2 3 b 0.000260576129615
R3 a 4 6928008764.81
C3 4 b 0.000221875755469
R4 a 5 3776590315.45
C4 5 b 0.000188923102579
R5 a 6 2058691738.85
C5 6 b 0.000160864528044
R6 a 7 1122232310.52
C6 7 b 0.000136973170722
R7 a 8 611750333.967
C7 8 b 0.000116630121791
R8 a 9 333476827.926
C8 9 b 9.93083918356e-05
R9 a 10 181784608.179
C9 10 b 8.45592591134e-05
R10 a 11 99094272.8358
```

```
249 C10 11 b 7.20006453598e-05
250 R11 a 12 54018186.7277
251 C11 12 b 6.13072179982e-05
252 R12 a 13 29446348.5512
253 C12 13 b 5.22019623559e-05
254 R13 a 14 16051768.775
255 C13 14 b 4.44490055623e-05
256 R14 a 15 8750126.70443
257 C14 15 b 3.78475062299e-05
258 R15 a 16 4769861.71536
259 C15 16 b 3.22264516315e-05
260 R16 a 17 2600143.0096
261 C16 17 b 2.74402275925e-05
262 R17 a 18 1417387.77219
263 C17 18 b 2.33648463361e-05
264 R18 a 19 772645.231177
265 C18 19 b 1.98947345633e-05
266 R19 a 20 421183.719075
267 C19 20 b 1.69399985624e-05
268 R20 a 21 229595.314972
269 C20 21 b 1.44240955003e-05
270 R21 a 22 125156.805142
271 C21 22 b 1.22818505702e-05
272 R22 a 23 68225.3724348
273 C22 23 b 1.04577686292e-05
274 R23 a 24 37190.9576838
275 C23 24 b 8.90459658967e-06
276 R24 a 25 20273.5035967
277 C24 25 b 7.5820993212e-06
278 R25 a 26 11051.4752424
279 C25 26 b 6.45601735437e-06
280 R26 a 27 6024.37089629
281 C26 27 b 5.49717938452e-06
282 R27 a 28 3284.0
283 C27 28 b 4.68074658522e-06
284 R28 a 29 1790.17132007
285 C28 29 b 3.98556915512e-06
286 R29 a 30 975.856685504
287 C29 30 b 3.39363842947e-06
288 R30 a 31 531.958176275
289 C30 31 b 2.8896203633e-06
290 R31 a 32 289.980593984
291 C31 32 b 2.46045830089e-06
292 R32 a 33 158.073977688
293 C32 33 b 2.09503474134e-06
294 R33 a 34 86.169153869
295 C33 34 b 1.78388333826e-06
296 R34 a 35 46.972456739
297 C34 35 b 1.51894367273e-06
298 R35 a 36 25.6055861411
299 C35 36 b 1.29335244713e-06
300 R36 a 37 13.9580955979
301 C36 37 b 1.10126569045e-06
302 R37 a 38 7.60882534167
303 C37 38 b 9.37707369436e-07
304 R38 a 39 4.14771647566
305 C38 39 b 7.98440483821e-07
306 R39 a 40 2.2609997194
307 C39 40 b 6.79857306217e-07
308 R40 a 41 1.23251426686
309 C40 41 b 5.78885923475e-07
310 R41 a 42 0.671867141321
311 C41 42 b 4.92910658358e-07
312 R42 a 43 0.366247651428
313 C42 43 b 4.19704310074e-07
314 R43 a 44 0.199648611946
315 C43 44 b 3.57370458332e-07
316 R44 a 45 0.108832283556
317 C44 45 b 3.0429433633e-07
318 R45 a 46 0.0593265629473
319 C45 46 b 2.59100999995e-07
320 R46 a 47 0.0323400461347
321 C46 47 b 2.20619709877e-07
```

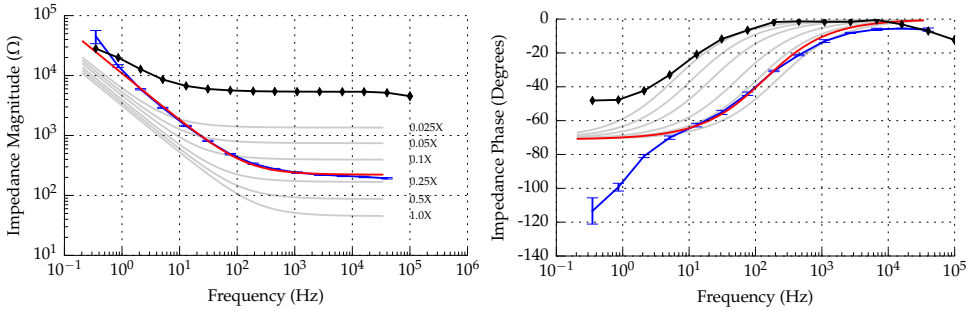
```
R47 a 48 0.0176291787698
C47 48 b 1.87853602985e-07
R48 a 49 0.00961000311513
C48 49 b 1.59953868919e-07
R49 a 50 0.00523859682171
C49 50 b 1.36197761318e-07
R50 a 51 0.00285565949684
C50 51 b 1.15969875023e-07
R51 a 52 0.00155667470497
C51 52 b 9.8746203923e-08
R52 a 53 0.000848573206917
C52 53 b 8.40805665025e-08
R53 a 54 0.000462573513399
C53 54 b 7.15930474542e-08
R54 a 55 0.000252157684869
C54 55 b 6.09601559193e-08
R55 a b 10000000000.0
.ENDS displacement
**************************************
*          Combine subcircuits       *
**************************************
.SUBCKT interface a b
X_1 a n1 faradaic
X_2 a n1 displacement
R3 n1 b 13.38
.ENDS interface
**************************************
*          Circuit description       *
**************************************
X_ladder w1 w2 w3 w4 w5 w6 w7 w8 ladder
X_interface1 e1 w1 interface
X_interface2 e2 w2 interface
X_interface3 e3 w3 interface
X_interface4 e4 w4 interface
X_interface5 e5 w5 interface
X_interface6 e6 w6 interface
X_interface7 e7 w7 interface
X_interface8 e8 w8 interface
R_in 1 e7 0
R_out 0 e2 0
V1 1 0 DC 0 PWL(66 0.5 67 0.55 131 0.55 132 0.6 196 0.6 197 0.65 261 0.65 262 0.7 326 0.7 327 0.75 391
      0.75 392 0.8 456 0.8 457 0.85 521 0.85 522 0.9 586 0.9 587 0.95 651 0.95 652 1.0 )
**************************************
*           Simulation options       *
**************************************
.control
TRAN 1 651 0
wrdata model_data v(e7,e2) i(V1) * -1.00
.endc
.END
```

Listing E.3: Full spice model for 1.0X PBS generated from fitted parameters

Appendix F

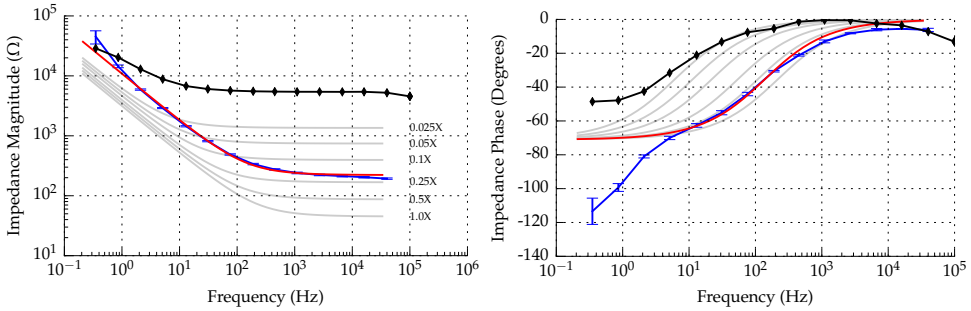
Other Biological Solution Attempts

The graphs presented in this appendix are other solutions that were tried before the cornflour and salt solution was found. In each case, the black trace represents the measured response of the solution tried (with ingredients listed in the caption). The blue trace shows the measured response in sheep spine, with the red trace being a simulated fit to that measured data. This trace was used as a reference point for how well the measured mixture matched the situation in sheep spine. The light grey traces show the various concentrations of phosphate buffered saline measured in chapter 8. The measurement setup used here is the same as was used in chapter 9. The mix of gelatine used in these measurements (1.0X) contains 10 g of gelatine dissolved in 500 ml of distilled water. These graphs are outputs from the measurement scripts used to measure the various solutions tried. Those measurements generated 186 graphs in total, corresponding to 93 separate measurements, however many of those have not been included here. Of the graphs omitted, many were calibration runs, where others were so close to previous measurements there is no visible change.



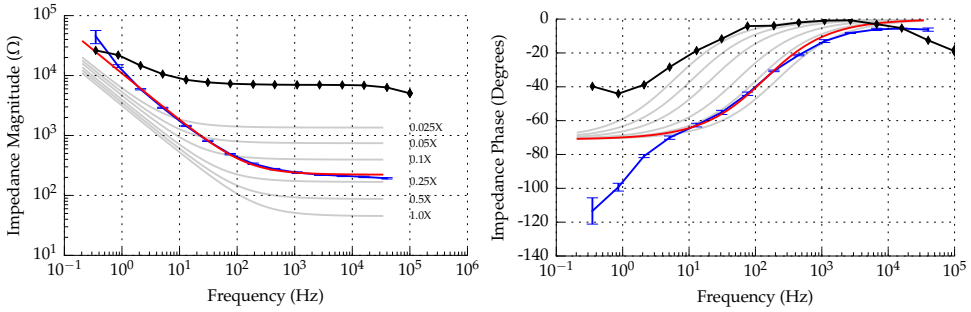
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.1: Tap water



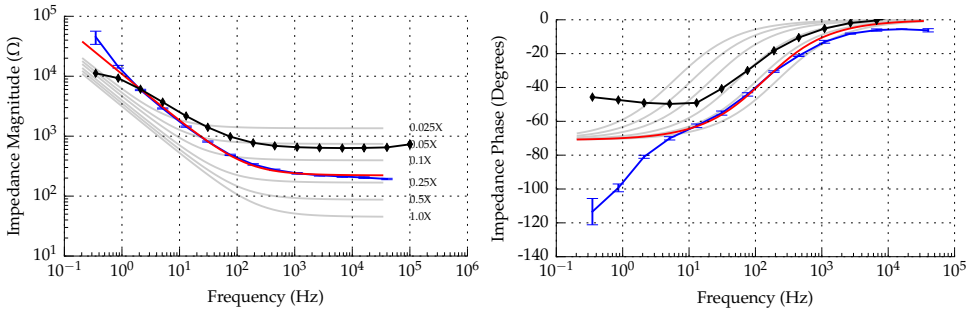
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.2: 1000 ml of tap water and 2.5 ml of glycerol



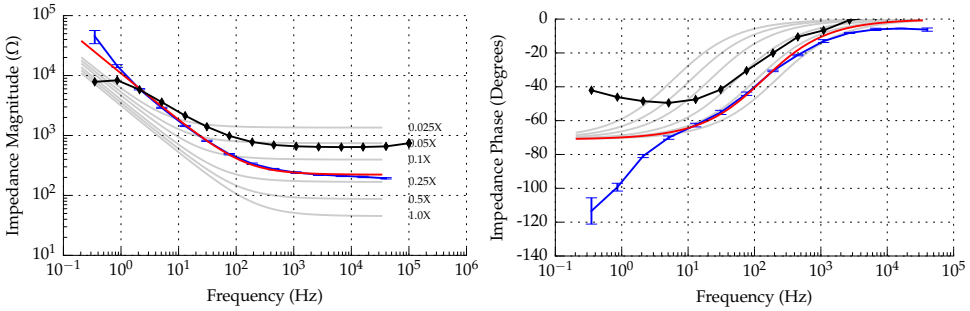
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.3: 1000 ml of tap water and 2.5 ml of methylated spirits



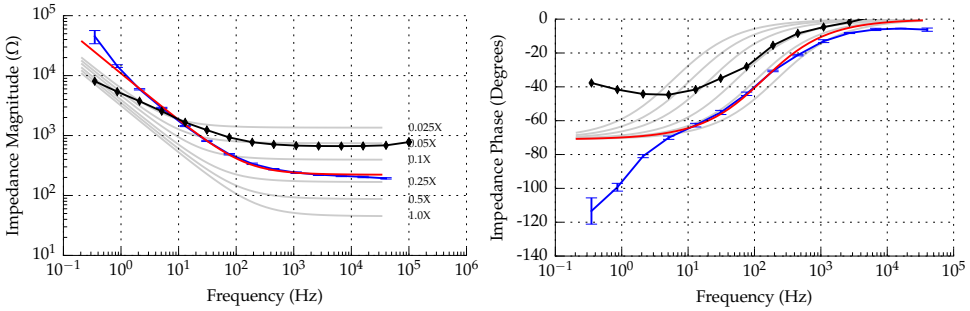
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.4: 1000 ml of tap water and 1 g sodium bicarbonate



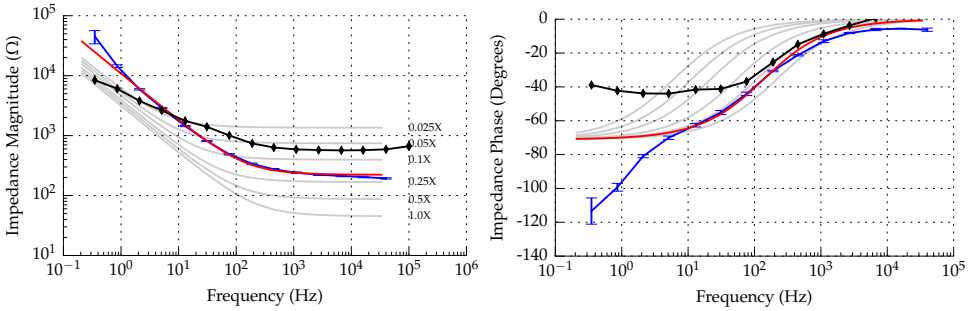
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.5: 1000 ml of tap water, 1 g sodium bicarbonate, and 10 ml of glycerol



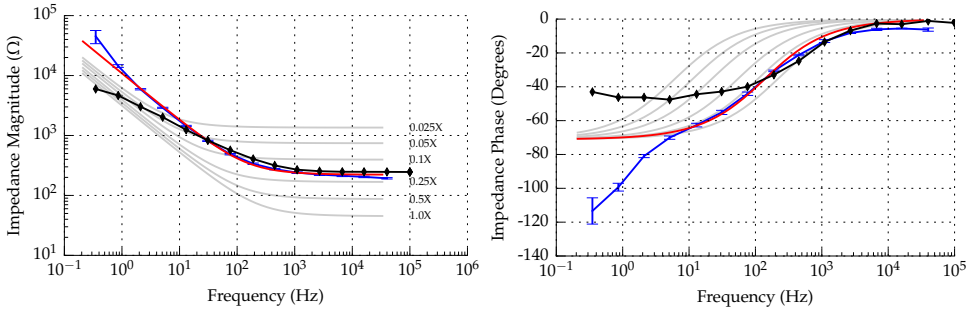
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.6: 1000 ml of tap water, 1 g sodium bicarbonate, and 30 ml of glycerol



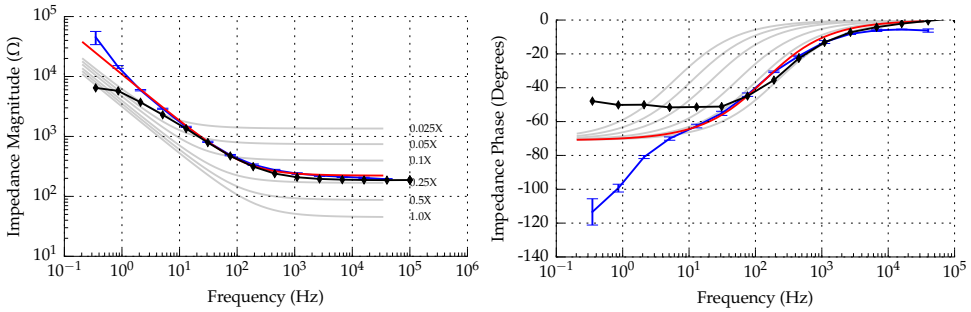
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.7: 1000 ml of tap water, 1 g sodium bicarbonate, 30 ml of glycerol, and 0.1 g NaCl



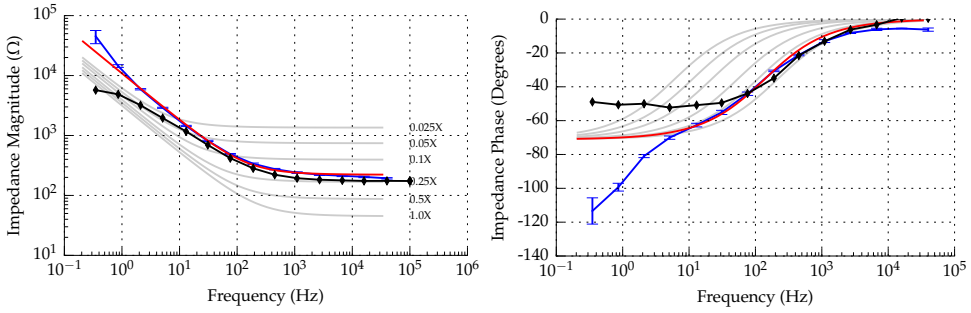
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.8: 1000 ml of tap water, 1 g sodium bicarbonate, 30 ml of glycerol, and 1.0 g NaCl



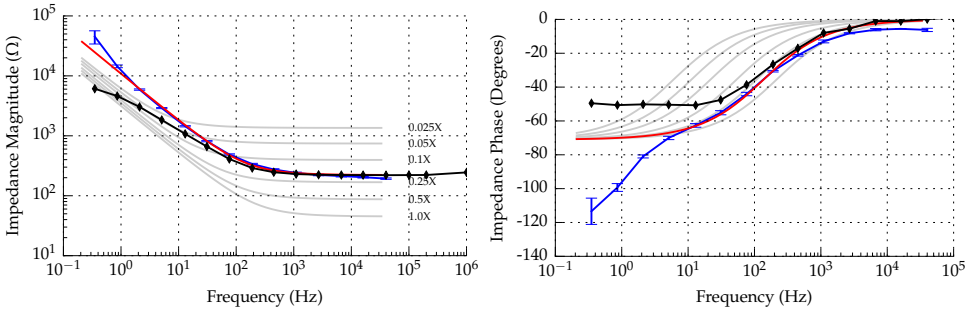
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.9: 600 ml of tap water and 1.0 g NaCl



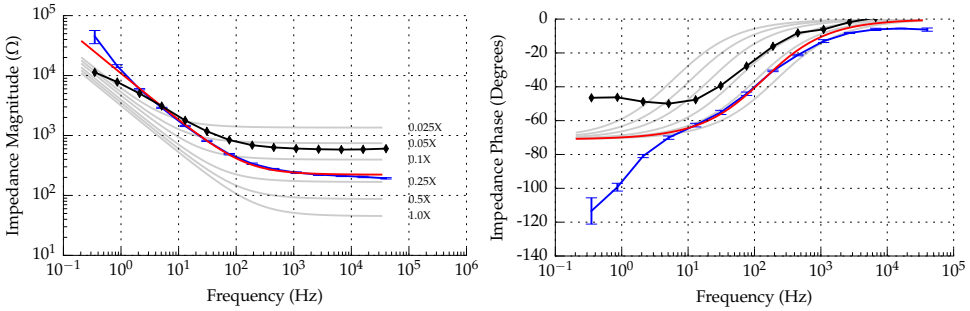
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.10: 600 ml of tap water, 1.0 g NaCl, and 2.5 ml sugar soap



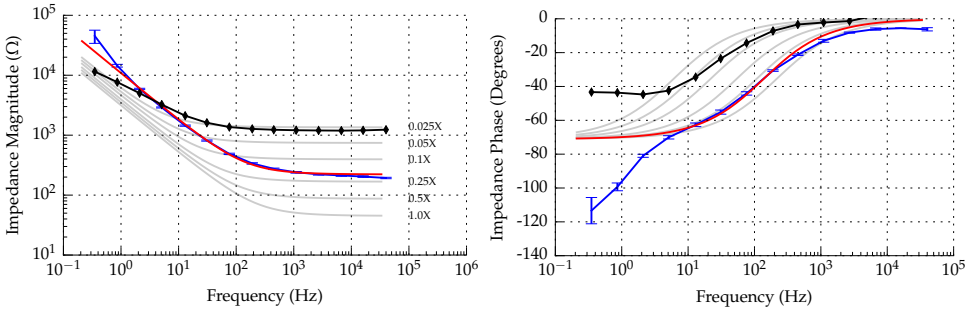
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.11: 600 ml of distilled water and 1.0 g NaCl



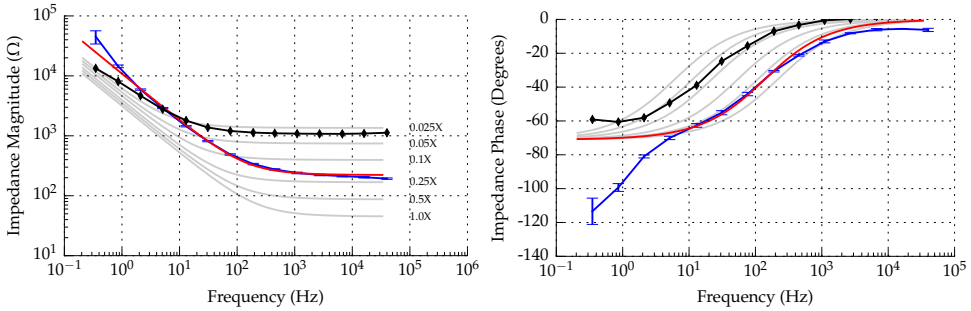
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.12: A 1.0X mixture of gelatine



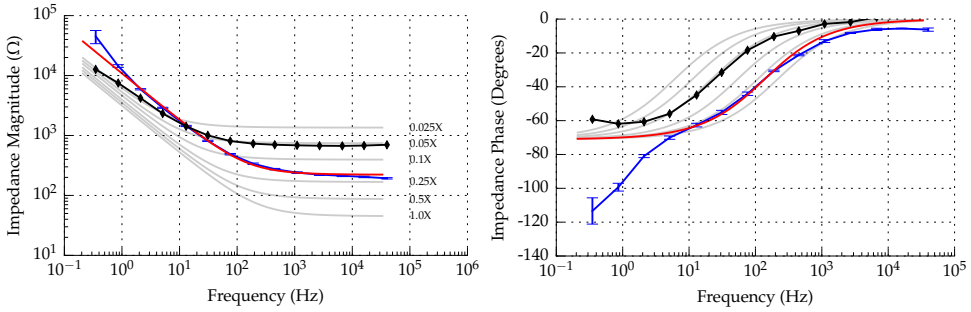
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.13: A 0.5X mixture of gelatine



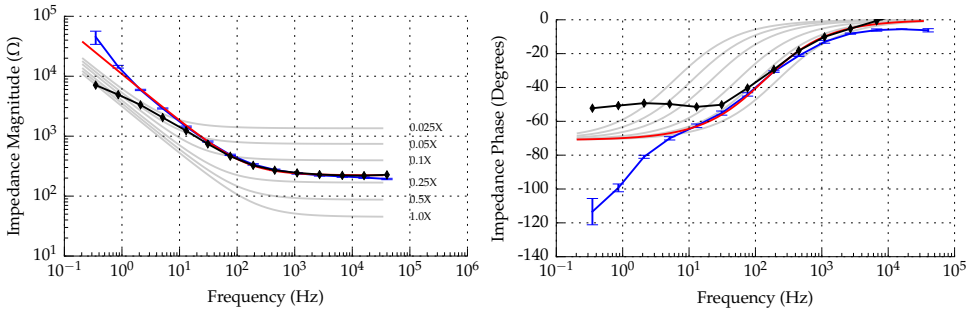
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.14: A 1.0X mixture of gelatine having a temperature of 1° Celcius



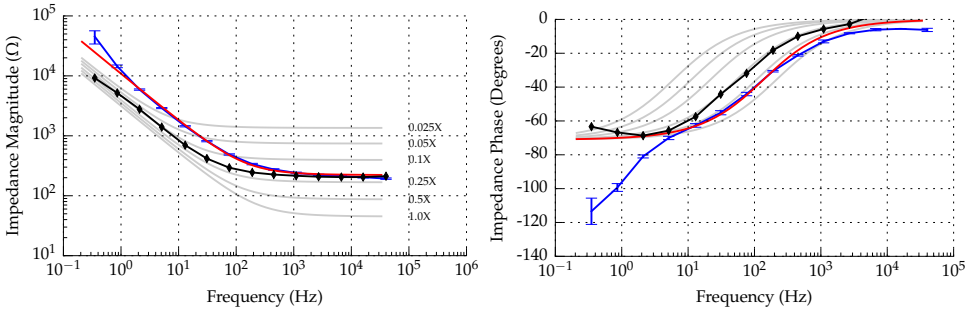
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.15: A 1.0X mixture of gelatine having a temperature of 20° Celcius



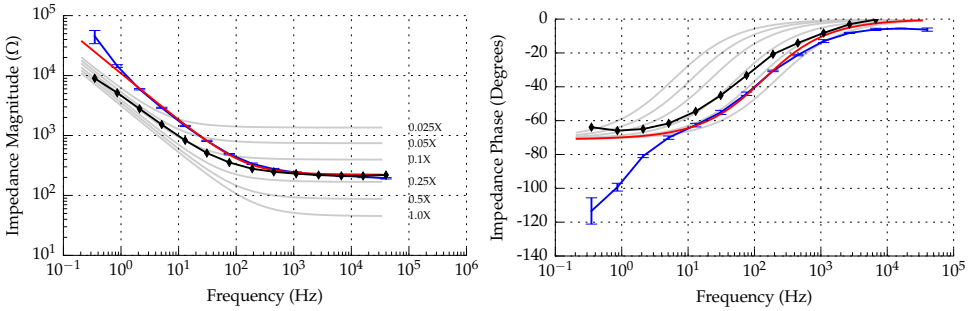
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.16: A 0.5X mixture of gelatine and 0.581 g NaCl



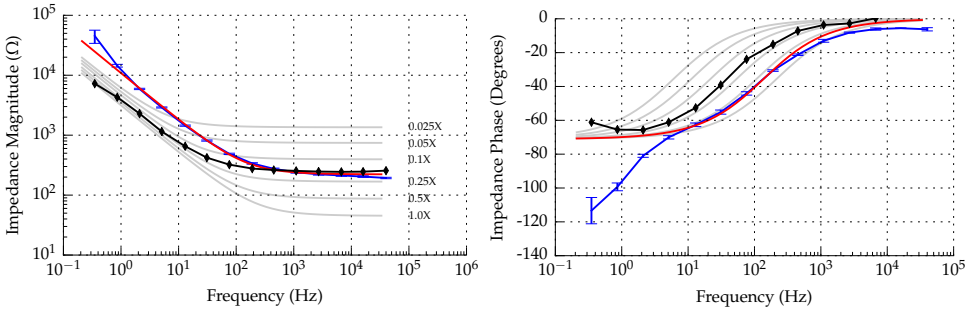
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.17: 600 ml of distilled water, 1.0 g NaCL, and 0.175 g citric acid



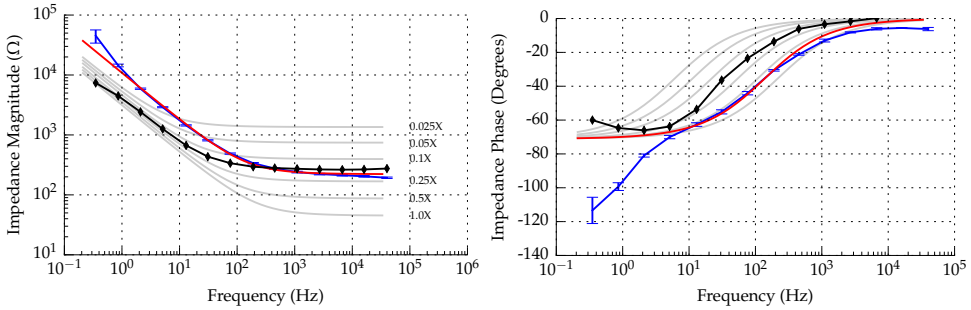
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.18: 600 ml of distilled water and 1.01 g sodium carbonate



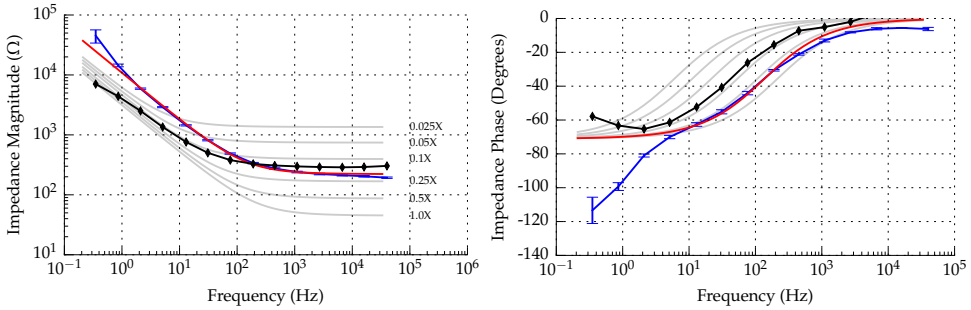
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.19: 600 ml of distilled water, 1.01 g sodium carbonate, and 1.00 g citric acid



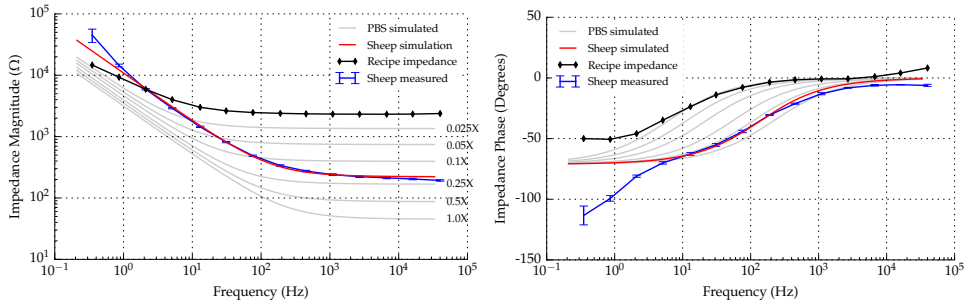
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.20: 600 ml of distilled water, 1.01 g sodium carbonate, and 2.00 g citric acid



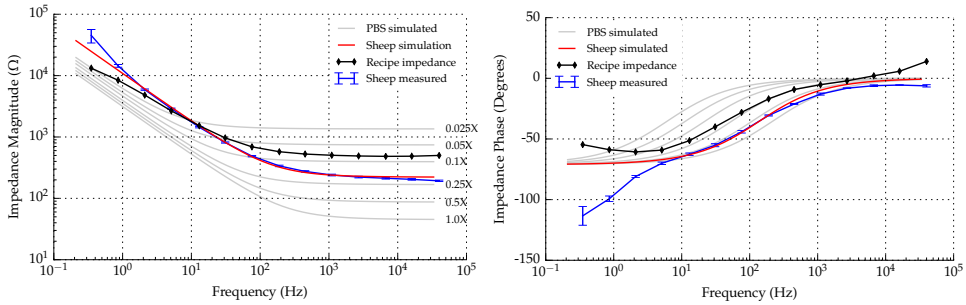
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.21: 600 ml of distilled water, 1.01 g sodium carbonate, and 4.00 g citric acid



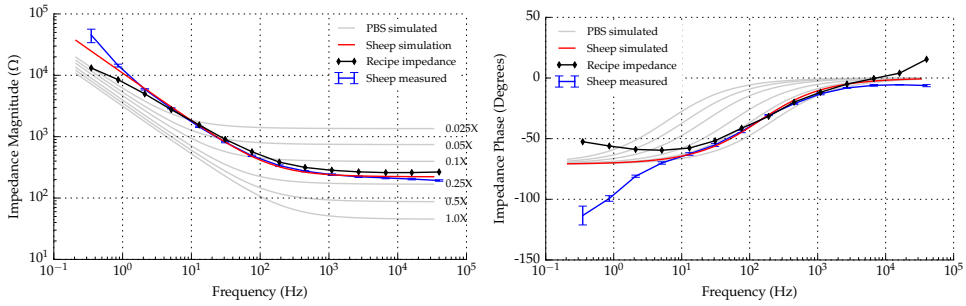
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.22: 190 ml of distilled water and 190 g cornflour



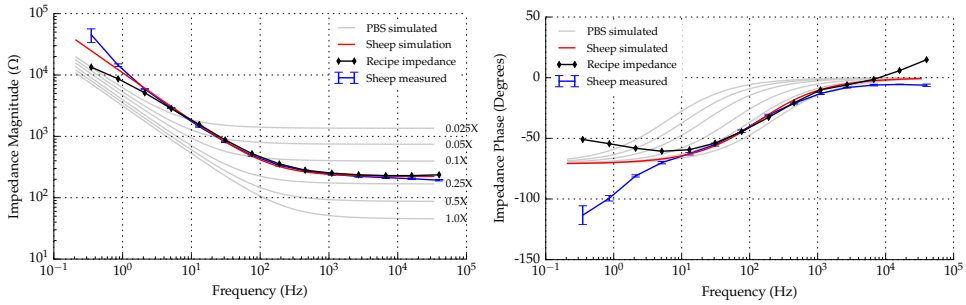
(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.23: 190 ml of distilled water, 190 g cornflour, and 0.358 g NaCl



(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.24: 190 ml of distilled water, 190 g cornflour, and 0.758 g NaCl



(a) Magnitude vs. frequency **(b)** Phase vs. frequency

Figure F.25: 190 ml of distilled water, 190 g cornflour, and 0.858 g NaCl

Bibliography

[1] W. Thomson. On a Self-Acting Apparatus for Multiplying and Maintaining Electric Charges, with Applications to Illustrate the Voltaic Theory. *Proceedings of the Royal Society of London*, 16:67–72, 1867. 1.1, A.1

[2] Peter Bruesch. The electric double layer at a metal electrode in pure water. *Journal of Applied Physics*, 95(5):2846, 2004. 2.1.1

[3] KW Horch and GS Dhillon. *Neuroprosthetics: theory and practice*. Volume 2 edition, 2004. 2.1.2

[4] L a Geddes. Historical evolution of circuit models for the electrode-electrolyte interface. *Annals of biomedical engineering*, 25(1):1–14, 1997. 2.1.2, 2.3.1

[5] G B Salieb-Beugelaar, K D Dorfman, A van den Berg, and J C T Eijkel. Electrophoretic separation of DNA in gels and nanostructures. *Lab on a chip*, 9(17):2508–23, sep 2009. 2.1.2

[6] Allen J. Bard and Larry R. Faulkner. *Electrochemical methods: Fundamentals and Applications*. JOHN WILEY & SONS, INC., 1980. 2.1.2, 2.1.2

[7] David Leonard Chapman. LI. A contribution to the theory of electrocapillarity. *Philosophical Magazine Series 6*, 25:475–481, 1913. 2.1.2

[8] Otto Stern. Theory of the electrical double layer. *Electrochemistry*, 30:508–516, 1924. 2.1.2

[9] W Olthuis, B Schippers, J Eijkel, and A Vandenberg. Energy from streaming current and potential. *Sensors and Actuators B: Chemical*, 111-112(April):385–389, 2005. 2.1.2, 2.2.1, 3.1.1, 3.1.3

[10] Zhijun Jiang and Derek Stein. Electrofluidic gating of a chemically reactive surface. *Langmuir : the ACS journal of surfaces and colloids*, 26(11):8161–73, jun 2010. 2.1.3

[11] Jacob N. Israelachvili. *Intermolecular and Surface Forces*. Academic Press, 2011. 2.1.3

[12] Brian J Kirby and Ernest F Hasselbrink. Zeta potential of microfluidic substrates: 1. Theory, experimental techniques, and effects on separations. *Electrophoresis*, 25(2):187–202, jan 2004. 2.2

[13] JF Osterle. Electrokinetic energy conversion. *Journal of Applied Mechanics*, pages 161–164, 1964. 2.11, 2.2.1

[14] D Burgreen and F. R. Nakache. Electrokinetic flow in ultrafine capillary slits1. *The Journal of Physical Chemistry*, (1):1084–1091, 1964. 2.2.1

[15] D. Burgreen and F. R. Nakache. Efficiency of Pumping and Power Generation in Ultrafine Electrokinetic Systems. *Journal of Applied Mechanics*, pages 675–679, 1965. 2.2.1

[16] CL Rice and R Whitehead. Electrokinetic flow in a narrow cylindrical capillary. *The Journal of Physical Chemistry*, 69(11), 1965. 2.2.1

[17] J Yang, F Lu, LW Kostiuk, and DY Kwok. Electrokinetic microchannel battery by means of electrokinetic and microfluidic phenomena. *Journal of Micromechanics . . .* , 13:963–970, 2003. 2.2.1, 2.2.1

[18] Chih-Chang Chang and Ruey-Jen Yang. Electrokinetic energy conversion in micrometer-length nanofluidic channels. *Microfluidics and Nanofluidics*, 9(2-3):225–241, dec 2009. 2.2.1

[19] Hirofumi Daiguji, Peidong Yang, Andrew J. Szeri, and Arun Majumdar. Electrochemomechanical Energy Conversion in Nanofluidic Channels. *Nano Letters*, 4(12):2315–2321, dec 2004. 2.2.1, 2

[20] Frank van der Heyden, Derek Stein, and Cees Dekker. Streaming Currents in a Single Nanofluidic Channel. *Physical Review Letters*, 95(11):9–12, sep 2005. 2.2.1

[21] Frank H. J. van der Heyden, Douwe Jan Bonthuis, Derek Stein, Christine Meyer, and Cees Dekker. Electrokinetic energy conversion efficiency in nanofluidic channels. *Nano Letters*, 6(10):2232–7, oct 2006. 2.2.1, 2.2.1, 3.3, 6

[22] Hirofumi Daiguji, Yukiko Oka, Takuma Adachi, and Katsuhiro Shirono. Theoretical study on the efficiency of nanofluidic batteries. *Electrochemistry Communications*, 8(11):1796–1800, nov 2006. 2.2.1

[23] Frank H. J. van der Heyden, Douwe Jan Bonthuis, Derek Stein, Christine Meyer, and Cees Dekker. Power generation by pressure-driven transport of ions in nanofluidic channels. *Nano Letters*, 7(4):1022–1025, 2007. 2.2.1, 2.2.1

[24] Christian Davidson and Xiangchun Xuan. Effects of Stern layer conductance on electrokinetic energy conversion in nanofluidic channels. *Electrophoresis*, 29(5):1125–30, mar 2008. 2.2.1

[25] Sumita Pennathur, Jan C T Eijkel, and Albert van den Berg. Energy conversion in microsystems: is there a role for micro/nanofluidics? *Lab on a chip*, 7(10):1234–7, oct 2007. 2.2.1

[26] Christian Davidson and Xiangchun Xuan. Electrokinetic energy conversion in slip nanochannels. *Journal of Power Sources*, 179(1):297–300, apr 2008. 2.2.1

[27] Yongqiang Ren and Derek Stein. Slip-enhanced electrokinetic energy conversion in nanofluidic channels. *Nanotechnology*, 19(19):195707, may 2008. 2.2.1

[28] Jan C T Eijkel. Liquid slip in micro-and nanofluidics: recent research and its possible implications. *Lab on a Chip*, pages 299–301, 2007. 2.2.1

[29] Jun Yang, Fuzhi Lu, Larry W Kostiuk, and Daniel Y Kwok. Electrokinetic Power Generation via Streaming Potentials in Microchannels: A Mobile-Ion-Drain Method to Increase Streaming Potentials. In Dr. Wael Badawy and Dr. Walied Moussa, editors, *International Conference on MEMS, NANO and Smart Systems*, number 780, pages 675–679, Banff, Alberta - Canada, 2004. IEEE. 2.2.1

[30] Ming-Chang Lu, Srinath Satyanarayana, Rohit Karnik, Arun Majumdar, and Chi-Chuan Wang. A mechanical-electrokinetic battery using a nano-porous membrane. *Journal of Micromechanics and Microengineering*, 16(4):667–675, apr 2006. 2.2.1

[31] Yanbo Xie, Xinwei Wang, Jianming Xue, Ke Jin, Long Chen, and Yugang Wang. Electric energy generation in single track-etched nanopores. *Applied Physics Letters*, 93(16):163116, 2008. 2.2.1

[32] Allen J. Bard, Hector D. Abruna, Chris E. Chidsey, Larry R. Faulkner, Stephen W. Feldberg, Kingo Itaya, Marcin Majda, Owen Melroy, and Royce W. Murray. The electrode/electrolyte interface - a status report. *The Journal of Physical Chemistry*, 97(28):7147–7173, 1993. 2.16

[33] E. Warburg. Über das Verhalten sogenannter unpolarisirbarer Elektroden gegen Wechselstrom. *Annalen der Physik und Chemie*, 67:493–499, 1899. 2.3.1

[34] Hugo Fricke. XXXIII. The theory of electrolytic polarization. *Philosophical Magazine Series 7*, 14:310–318, 1932. 2.3.1

[35] C. C. Murdock and E. E. Zimmerman. Polarization Impedance at Low Frequencies. *Physics*, 7:211, 1936. 2.3.1

[36] J. E. B. Randles. Kinetics of rapid electrode reactions, 1947. 2.3.1

[37] I. Epelboin, M. Keddam, and J. C. Lestrade. Faradaic impedances and intermediates in electrochemical reactions, 1973. 2.3.1

[38] Margaretha Sluyters-Rehbach and Jan H Sluyters. Sine wave methods in the study of electrode processes. *Electroanalytical Chemistry*, 4:1–128, 1970. 2.3.1

[39] Leslie Alexander Geddes and Lee Edward Baker. *Principles of Applied Biomedical Instrumentation*. JOHN WILEY & SONS, INC., 1975. 2.3.1

[40] B. Onaral and H. P. Schwan. Linear and nonlinear properties of platinum electrode polarisation. Part 1: frequency dependence at very low frequencies. *Medical & Biological Engineering & Computing*, 20(May):299–306, 1982. 2.3.1

[41] H P Schwan. Electrode polarization impedance and measurements in biological materials. *Annals of the New York Academy of Sciences*, 148:191–209, 1968. 2.3.1, 8.1.3, 10

[42] Jonathan Scott and Peter Single. Compact nonlinear model of an implantable electrode array for spinal cord stimulation (SCS). *IEEE Transactions on Biomedical Circuits and Systems*, 8:382–390, 2014. 2.3.1, 7.1.1, 7.3.1, 8.1

[43] R Morrison. RC Constant-Argument Driving-Point Admittances. *IRE Transactions on Circuit Theory*, 6(3):310–317, 1959. 2.3.1, 7.7, 7.1.3

[44] Camelia Gabriel. Compilation of the Dielectric Properties of Body Tissues at RF and Microwave Frequencies. *Environmental Health*, Report No.(June):21, jun 1996. 2.3.1

[45] Chun Yang, D Li, and JH Masliyah. Modeling forced liquid convection in rectangular microchannels with electrokinetic effects. *International Journal of Heat and Mass Transfer*, pages 3118–3138, 1998. 3.1.1

[46] AA Kornyshev. Double-layer in ionic liquids: paradigm change? *The Journal of Physical Chemistry B*, 2007. 3.1.1

[47] Gu Yongan and Dongqing Li. The ζ-Potential of Glass Surface in Contact with Aqueous Solutions. *Journal of Colloid and Interface Science*, 226(2):328–339, jun 2000. 3.1.1, 3.1.1, 3.2.1, 3.2.2, 3.3, 3.14, 3.5, 3.5, 3.15, 3.16

[48] I K Varga and R S Seymour. Observation of electrical effects with cavitating liquid flow. *Journal of Physics D: Applied Physics*, 19:2293–2300, 1986. 3.2.1, 3.2

[49] G Mala. Flow characteristics of water through a microchannel between two parallel plates with electrokinetic effects. *International Journal of Heat and Fluid Flow*, 18(5):489–496, oct 1997. 3.3

[50] Peter J. Scales, Franz Grieser, Thomas W. Healy, Lee R. White, and Derek Y. C. Chan. Electrokinetics of the silica-solution interface: a flat plate streaming potential study. *Langmuir*, 8(3):965–974, mar 1992. 3.3

[51] Honeywell. Installation Instructions for the 26PC SMT Series Pressure Sensor, 2003. 3.8

[52] Python-ivi. python-vxi11, 2014. 3.3.1

[53] Nancy Chang. Smart Gas And Water Meter Trends: Impacts on Meter Designs. *Metering International*, (4):38—-39, 2012. 4, 4.1

[54] "Water New Zealand". Metering Overview, 2011. 4.1

[55] Tracy C. Britton, Rodney a. Stewart, and Kelvin R. O'Halloran. Smart metering: enabler for rapid and effective post meter leakage identification and water loss management. *Journal of Cleaner Production*, 54:166–176, sep 2013. 4.1

[56] B Meters. Hydrolink Remote Reading Systems, 2014. 4.1, 4.2

[57] Matthias Heinrich. Water Use in Auckland Households: Auckland Water Use Study (AWUS). Technical Report October 2008, BRANZ Ltd and WaterCare Services Ltd, 2008. 4.1, 4.3

[58] Matthias Heinrich. Water End Use and Efficiency Project (WEEP): Final Report. Technical Report 159, BRANZ Ltd, 2007. 4.3

[59] Elster. KENT V100T PSM Product Specifications, 2008. 4.5

[60] Watercare New Zealand. Private communication, 2014. 4.3

[61] Microchip Technology Inc. PIC16(L)F1826/27 Data Sheet, 2011. 5.1.2.1

[62] Freescale Semiconductor Inc. MC9S08QG8/4 Data Sheet, 2009. 5.1.2.1

[63] Atmel Corporation. ATtiny13. Technical Datasheet, 2010. 5.1.2.1

[64] Atmel Corporation. ATtiny25/25V/45/45V/85/85V, 2010. 5.1.2.1

[65] Microchip Technology Inc. PIC12F629/675. Technical Datasheet, 2003. 5.1.2.1

[66] Mark Donald Hill, Norman Paul Jouppi, and Gurindar Sohi. *Readings in Computer Architecture*. 2000. 5.1.2.1

[67] Wikipedia. Linear feedback shift register, 2015. 5.1.2.3

[68] Malcolm Souness - WEL Networks. Private communication, 2012. 5.2

[69] Li Li, Mehmet C Vuran, and Ian F Akyildiz. Characteristics of underground channel for wireless underground sensor networks. *The Sixth Annual Mediterranean Ad Hoc Networking Workshop*, 6:12–15, 2007. 5.2

[70] Mohammad Arsalan, Ahmad Umair, and Varun Kumar Verma. Dash7 : Performance. *Journal of Electronics and Communication Engineering (IOSR-JECE)*, 2(5):8–11, 2012. 5.2

[71] Paul Ben Ishai, Mark S Talary, Andreas Caduff, Evgeniya Levy, and Yuri Feldman. Electrode polarization in dielectric measurements: a review. *Measurement Science and Technology*, 24(OCTOBER):102001, 2013. II

[72] Jonathan Scott, Senior Member, and Peter Single. Compact Nonlinear Model of an Implantable Electrode Array for Spinal Cord Stimulation (SCS). *JOURNAL OF BIOMEDICAL ENGINEERING*, pages 1–8, 2013. 7.1, 7.1.4

[73] Daniel R Merrill, Marom Bikson, and John G R Jefferys. Electrical stimulation of excitable tissue: design of efficacious and safe protocols. *Journal of neuroscience methods*, 141(2):171–98, feb 2005. 7.1.4

[74] ET McAdams, A Lackermeier, J.A. McLaughlin, D. Macken, and J. Jossinet. The linear and non-linear electrical properties of the electrode-electrolyte interface. *Biosensors and Bioelectronics*, 10(1-2):67–74, jan 1995. 7.1.4

[75] Shahar Kvatinsky, Keren Talisveyberg, Dmitry Fliter, Avinoam Kolodny, Uri C. Weiser, and Eby G. Friedman. Models of memristors for SPICE simulations. In *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, pages 1–5. IEEE, nov 2012. 7.1.5

[76] R. Stanley Williams. How we found the missing Memristor. *IEEE Spectrum*, 45(december):28–35, 2008. 7.1.5

[77] Sir William Thomson. Sir W. Thomson on an Apparatus. *Philosphical Magazine and Journal of Science*, 34:391–396, 1867. A