DESIGN CONSIDERATIONS FOR THE CHARACTERIZATION OF CAPACITORS

by

MICHAEL L. DELIBERO

Submitted in partial fulfillment of the requirements for
the degree of Master of Science

Thesis Advisor: Dr. Merat

Department of Electrical Engineering
& Computer Science

CASE WESTERN RESERVE UNIVERSITY

August, 2015

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the master of science of

Michael L. DeLibero

candidate for the Master of Science degree*.

*We also certify that written approval has been obtained for any proprietary material contained herein.

Dedication

Dedication text

Table of Contents

List of Figures

List of Tables

Preface

Unless otherwise stated, all portions of this master's thesis are the sole and original work of the author, Michael L. DeLibero. Section: 2.1 History of Capacitors is largely a compilation of other research and is cited as such. Section: 5 Regression Analysis and Modeling is an application of Levy's [30] and Sanathanan's [44] work and is cited as such. The transimpedance amplifier seen in Appendix: A Schematic is largely based on previous work done with Steven Ehret which was published in his master's thesis [10].

Acknowledgments

I would like to thank my thesis advisor, Dr. Merat, for his support and instruction in helping me complete this work. I would also like to thank Dr. Welsch and ARPA-E, who's leadership and funding under ARPA-E contract DE–AR000016 made this work possible. Furthermore, I would like to thank Steven Ehret, who's initial work on this contract enabled my own.

Acronyms

ADC Analog to Digital Converter. 22, 25, 26

DA Dielectric Absorption. 16, 17

DAC Digital to Analog Controller. 20

DF Dissipation Factor. 14, 15

DUT Device Under Test. 18, 19, 21, 22, 24, 25, 27, 42

ESL Equivalent Series Inductance. 12, 14, 17

ESR Equivalent Series Resistance. 7, 12–14, 17

LSE Least Squares Estimate. 29–31

MCU Microcontroller Unit. 26

MLCC Multi-Layer Ceramic Capacitor. 3, 4

Q Quality Factor. 15

RF Radio Frequency. 25

USB Universal Serial Bus. 26

Abstract

Various industry needs, such as electric vehicles and defibrillators, are requiring capacitors with a high energy and power density that can operate in the presence of a DC bias in the 100s of volts. Researchers have been developing technologies such as electrochemical capacitors for lower voltages and Titanium based capacitors for the higher voltage applications. A critical step in this development is the characterization of the capacitors operating with a high DC bias. This thesis outlines a practical method to characterize a capacitor of up to tens of microfards with a DC bias up to 500 volts. A circuit has been designed to characterize a capacitor through discharge curves and a swept frequency analysis of the impedance. It has a frequency range of 100Hz $\rightarrow$ 100kHz, a DC bias range of 0$\rightarrow$500V, and an AC signal amplitude range of 0$\rightarrow$5V. An iterative regression analysis has been developed to fit measured data to a multi-term capacitor model. Simulated data has been generated using Muratas 34-term model for a ceramic capacitor and fit to a six-term model which incorporates dielectric absorption, leakage and resistance. The accuracy of the fit to the data is largely determined by the number of parameters in the capacitor model and typically has the greatest relative error near resonance. The six-term model fit the simulated data within 2Ω for the magnitude of the impedance and within 2° for the phase of the impedance. The relative error, outside of a single peak in the phase plot, stays well below 5%.

1 Introduction

This thesis presents a method to automate capacitor characterization in order to determine the physical differences between large numbers of research grade capacitors. It focuses on measuring capacitor characteristics through swept frequency analysis and discharge curve measurements in the presence of up to a $500V$ DC bias. The method then generates values for the physical parameters of the capacitors by fitting a model to the empirical data by applying a regression analysis.

2 Background

2.1 History of Capacitors

This section will chronolog the history of capacitors. It will link various introductions in the technology to advances in industry, and it will map the driving forces behind capacitor development.

Capacitors have their origin in the invention of the Leyden jar by Peter van Musschenbroek of Leiden University in 1745 [23]. Before this point, scientists were able to generate static electricity through electrostatic machines, but had a very limited ability to store this electrical energy [20]. The most common design for the Leyden jar was to use a glass jar with metal foil lining the inside and out. The inner foil was typically charged with an electrostatic generator, while the outer foil was connected to ground. The charge would stay on the metal foil until a short or small resistance was connected between them. Charge could be stored this way, allowing scientists and showmen, to use greater amounts of current than they could generate at any one moment. Since the Leyden jar, many different types of capacitors have risen and fallen in prominence in the market. This section covers the historical introduction of some of the major types.

In 1876 Fitzgerald introduced wax impregnated paper dielectric capacitors with foil electrodes [9, ch. 11][29]. They were typically used for power supply filtering in radios. By the early 1920s, they existed as tubes encapsulated in plain, Bakelite cardboard, with bitman sealing the ends and waxed paper as a dielectric [9, ch 3]. Paper capacitors were upgraded to impregnated paper capacitors, which used paper that was soaked in mineral oil. They were interleaved with metal foil and then rolled to make the capacitor [41, ch. 8.2.1.1]. During WWII, paper capacitors were upgraded with metal-cased tubes with a rubber end [41, ch. 8.1]. These metalized paper capacitors were constructed similarly with the improvement that one side of the paper dielectric was sprayed with metal [13].

Karol (Charles) Pollak discovered the principle of the electrolytic capacitor in 1886 while he was researching the anodization of metals, and received a patent for the borax-solution aluminum electrolytic capacitor in 1897. In 1936 Cornell-Dubilier opened a factory to produce aluminum electrolytic capacitors. After the start of WWII, increased funding and effort applied to the cause of electrolytic capacitors and techniques such as "etching and pre-anodizing" greatly increased their reliability [17][58].

In 1957 GE extended the science of electrolytic capacitors into what is now known as electrochemical capacitors. This technology, also known as electric double layer capacitors, was not commercialized until NEC licensed Standard Oil's patent in 1978 [34]. This "supercapacitor" formed the basis of today's electrochemical capacitors. They are characterized by a large capacitance in the range of kilo farads, but with a working voltage of only 2.7V [34]. Some of today's typical applications for EC capacitors are in battery replacement, electric vehicles, and regenerative breaking.

M. Bauer of Germany invented the mica capacitor in 1874. The original mica capacitor was a "clamped" style capacitor, which was used through the 1920s [59] and then replaced by silver mica capacitors [29]. Mica's inherent inertness and

reliability allowed for extreme reliability and efficiency in a packaged capacitor [50]. Mica capacitors were heavily used in the radio industry due to their superb stability at RF frequencies and their physical robustness [40]. Capacitors created with mica allowed for a comparatively smaller product [9, f. 37-41] with the ability to survive shock from weapons better than its glass counterpart. Consequently, mica capacitors began to be produced in large quantities during WWI. In light of mica supply chain problems and the emergence of ceramic capacitors during WWII, mica capacitors fell from prominence to a niche market [5, Ch 3, Sec II].

The first glass ceramic dielectric capacitor was the Leyden jar. While this early capacitor was used mainly for scientific experiments, commercial glass capacitors came later. Glass tubular capacitors, known as Moscicki tubes, appeared in 1904 and were used in Marconi's experiments in wireless transmission. They continued to be used in wireless communication until about WWI [9, p. 102].

Scientists in Germany created the first steatite ceramic capacitors in 1920 [5, Ch 3 Sec II][16]. Also known as talc, this ceramic capacitor variant was able to closely match the temperature coefficient of mica [15]. Rutile was later introduced into ceramic capacitor technology. On its own, it produced a dielectric constant of 10 times that of steatite, but the two were typically blended to get a better temperature coefficient.

A ceramic composition with barium titanate ($BaTiO_3$) was first discovered in 1941. Barium titanate was quickly found to be able to exhibit a dielectric constant over 1000; an order of magnitude greater than the best capacitor at that time (rutile - $Ti0_2$). It was not until 1947, that barium titanate appeared in its first commercial devices, phonograph pickups [22][13][5, Ch 3 Sec III]. Barium titanate is still used today in certain types of Multi-Layer Ceramic Capacitors (MLCCs).

Thanks to the semiconductor industry, MLCCs grew to dominate the capacitor market and by 1979, AVX alone reached $95 million in MLCC sales [3]. MLCCs are

divided up into three classes. Class 1 type MLCCs are known for their extremely good temperature characteristics. COG/NPO types can have 0-30ppm/^{o}C. These capacitors are typically made by combining TiO_2 with additives in order to adjust its temperature characteristics [37]. Additionally, class 1 MLCCs have comparatively poor volumetric efficiency, and will tend to come in larger packages than class 2 MLCCs of the same capacitance value. Class 2 types typically have worse temperature coefficients than class 1 types, but they have a much higher volumetric efficiency. They are constructed with a ferroelectric base material, typically barium titanate [37]. Class 2 MLCCs are the most common type of ceramics used. Class 3 types have very high capacitance, but a working voltage of several volts [13][5, Ch 3 Sec VI][12]. They were originally developed as a potential replacement for liquid electrolytics, but they have largely been replaced by Class 2 MLCCs [57]. The advances in Class 2 MLCCs in maximum capacitance shrunk the gap between these two classes until the difference was no longer viable [57].

Bell labs invented the first solid tantalum capacitor in 1956. They created it in conjuncture with, and for, transistors [9, f. 56-64]. Tantalum capacitors typically have better characteristics than aluminum electrolytics, but have a lower maximum capacitance and working voltage [29]. Sprague patented the first commercially viable solid tantalum capacitor in 1960. It offered an increased capacitance per unit volume and greater reliability [42]. In the 1970s, Sprague released the first surface mount tantalum capacitor [47]. One of the historical problems with tantalum capacitors has been a limited and volatile supply of tantalum in the world market. As a result of a price spike around 1980, the industry reacted by creating finer grain tantalum powders. This allowed a unit to be made with less overall tantalum, reducing price and package size [19, ch 3.1]. Over time, this volatility resulted in better tantalum capacitors, but decreased their popularity and led to a desire for a suitable replacement.

One of the possible alternatives to tantalum capacitors is a titanium based capacitor. Some of the instrumental work in titanium alloys was done in 1939 by Fast [11] and more recently in 1995 by Kobayashi [27][7]. Early work on titanium based capacitors resulted prohibitively high leakage currents [26], but research done by Welsch [54][55] with titanate (TiO_2) shows a promising a solution to this issue. This work also indicates that a Ti capacitor can be made with a much higher energy and power density when compared with tantalum [10].

2.2 Ti Capacitors

Even though the work in this thesis can characterize wide range of components, its primary focus is on titanium capacitors. As stated in the previous section, titanium capacitors have many advantages over their primary competitor, tantalum capacitors. They have fewer sourcing issues, as titanium is both cheaper and can be mined in conflict free zones. Also, the higher power and energy density, along with a high working voltage, make titanium capacitors an attractive option. These characteristics allow for a more efficient product, that is both cheaper and lighter than tantalum capacitors. The main reason why titanium capacitors do not have the market share of tantalum capacitors is that they have been plagued with high leakage currents. The initial work by Ehret[10] on the ARPA-E DE–AR000016 contract focused on measuring this quantity. The circuitry designed in this thesis (Section: 4) expands upon that design to also measure additional characteristics.

One of the main potential applications for titanium capacitors is the DC-link capacitor for electric vehicles [56]. Its purpose is to provide an energy source and storage mechanism that has a higher power density than the high energy density batteries in the vehicle [51]. They allow for a high surge of power when the vehicle is accelerating, and also allow the vehicle to recapture some of the expended energy through regenerative breaking.

Another potential use of titanium capacitors is in modern defibrillators. These devices require capacitors that can deliver a quick burst of energy while operating at no less than several hundred volts [45][31]. One of the main requirements of the capacitor in defibrillators is extremely high reliability. The titanium capacitors proposed by Welsch [54] provide an attractive option here because of their self-healing characteristic. Small defects in the dielectric are automatically healed, allowing the capacitor to continue to function while minimizing failure events.

3 Capacitor Parameters

This section will begin by describing some of the practical uses of capacitors and then transition into the various parameters that are used to describe capacitors. Section: 5 will show a method which allows these parameters to be extrapolated from empirical data.

3.1 Practical Capacitor Uses

The most basic reason for wanting to use capacitors is that they have the ability to store charge; the ability to store electrical energy. Capacitors have the ability to store and release electrical energy quickly, in order to be able to react to the needs of the circuit. This section will describe some of the most common uses for capacitors.

3.1.1 Bypassing

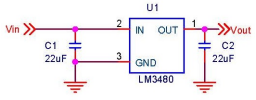


Figure 1: Power Supply Bypassing Circuit

One of the most common uses of capacitors is in power supply bypassing.

Capacitors are nearly always attached from a power rail to ground on a supply or IC. They provide a reservoir of charge that limits inductive voltage spikes, such as when a digital circuit switches, and limits voltage dips, caused by things such as a current surge when a processor boots. Both LDOs (Figure: 1) and switchers use bypass capacitors on their input and output voltage rails. Input capacitors are divided into two main categories, ripple reduction and bulk. Ripple reduction capacitors need a low Equivalent Series Resistance (ESR) (Section: 3.6) and are meant to decrease the magnitude of any AC signals that ride on top of the input DC voltage. Bulk capacitors are meant to deliver surge currents. Output capacitors have a similar purpose as input capacitors. The main difference is in the case of switching power supplies. In that application, the output capacitor is a major component in the feedback loop. It contributes to both the transient and stability properties of the switcher.

3.1.2 Analog Filtering

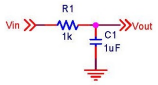


Figure 2: Analog Filtering Circuit

Another use for capacitors is in analog filtering. The low-pass filter in Figure: 2 attenuates frequencies above a cutoff point, set by the values of the resistor and capacitor. Low pass filters are needed in many applications, such as anti-aliasing, clock filtering, and integration.

3.1.3 DC Blocking

Designers often take advantage of a capacitor's characteristic of passing AC current while blocking DC current. As in Figure: 3, a capacitor can be used to block a DC

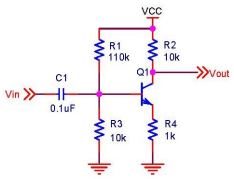


Figure 3: DC Blocking Capacitor

[38][ch 2.2.7 fig 2.35 pg 88]

offset before an amplifier.

3.1.4 Oscillators

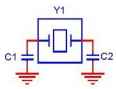


Figure 4: Oscillator Circuit

Stable capacitors of a very specific value are required to make a parallel resonant oscillator function properly (Figure: 4). These oscillators provide the clock base for most modern digital circuits using microcontroller.

3.2 Capacitance

There is a distinct difference between a capacitor and capacitance. While a capacitor's dominant characteristic is capacitance, it cannot be modeled entirely as such in most practical applications. There are also various inductive and resistive components to a capacitor that are important in various circumstances.

$$C = \frac{Q}{V} \tag{1}$$

Capacitance is the ability to store electrical charge. Equation: (1) shows that capacitance is stored charge that is spread throughout a volume. A device that can

store a lot of charge in a small space has a large capacitance. The basic equation for a commercial capacitor is seen in Equation: (2).

$$C = \frac{\epsilon_0 A}{d} \tag{2}$$

When using a capacitor in a single-pole low-pass filter, the cutoff frequency can be determined by Equation: (3). The circuit designer chooses a value for C and R in order to meet the cutoff frequency restraint.

$$f = \frac{1}{2\pi RC} \tag{3}$$

Varying the capacitance used in the filter moves the cutoff frequency and consequently produces a different response in the filter. The effect of this can be seen in Figure: 5.

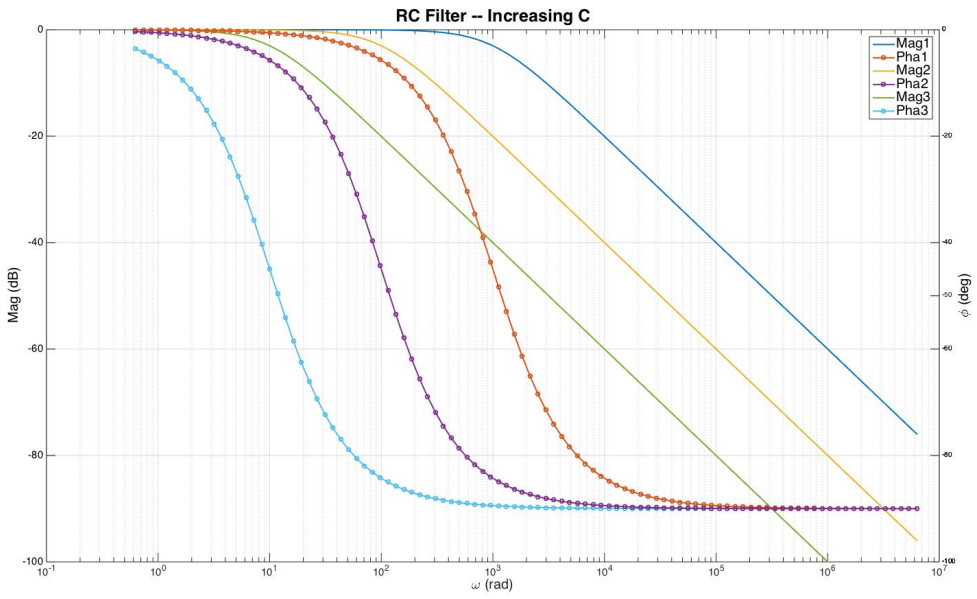


Figure 5: Low Pass Filter – Varying C

3.3 Impedance

The impedance of a capacitor is the "AC resistance" of the device. It determines the AC current that will flow when an AC voltage is applied to the capacitor via Ohm's law (Equation: (4)). Real capacitors have a complicated impedance equation, but ideal capacitors have a much simpler representation, as seen in Equation: (5).

$$\vec{V} = \vec{I}\vec{Z} \tag{4}$$

$$\vec{Z} = \frac{1}{j\omega C} \tag{5}$$

$$Z = |\vec{Z}| = \frac{1}{\omega C} \tag{6}$$

In most AC applications designers are interested in the magnitude of the impedance. Real capacitors have a more complicated impedance, but an ideal capacitor's magnitude equation can be simplified down to Equation (6). When capacitors are used in bypassing power supplies, the goal is to have a low impedance for common or expected noise frequencies. Using a large valued capacitor to bypass a wide range of frequencies does not work in practical situations due to parasitics in a real capacitor. The parasitic elements, described in the following sections, cause various undesirable effects, such as the impedance of a capacitor increasing after a certain frequency. This results in a more complicated impedance plot than the ideal version shown in Figure: 6.

3.4 Phase

The phase of a combination of resistive and reactive components can be written as in Equation: (7).

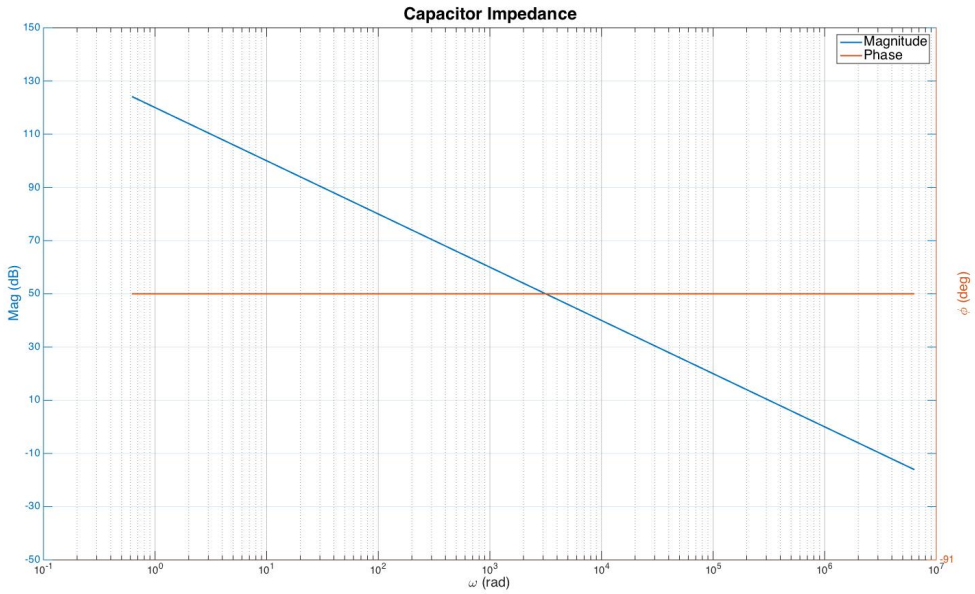


Figure 6: Capacitor Magnitude Over Frequency

$$\phi = tan^{-1}[\frac{X_c}{R_c}] \tag{7}$$

For an ideal capacitor, having no resistance and only capacitance, the phase angle can be simplified to:

$$\phi = -i = -90^0 \tag{8}$$

The practical implication of this is seen in the phase response of a low pass filter (Figure: 5). The capacitor introduces a phase lag relative to the input signal's frequency. If you would compare the input and output signals in time, the output's peak would lag behind the input's by the phase amount predicted in the phase response.

3.5 ESL

The Equivalent Series Inductance (ESL) of a capacitor is a lumped estimate of all of the inductive components of a capacitor. It is typically modeled as an inductor in series with the bulk capacitance (See Figure 7).

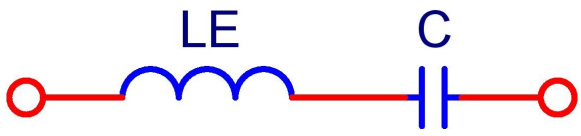


Figure 7: ESL Capacitor Model

Adding ESL to the capacitive model creates a new impedance equation (Equation: (9)). Note that for L<<C, this equation simplifies to Equation: (5) for low frequencies. In other words, the ideal impedance equation can reasonably be used for low frequencies.

$$\vec{Z_c} = j\frac{\omega^2 LC - 1}{\omega C} \tag{9}$$

Figure: 8 shows a graphical representation of a capacitor's magnitude and phase, once ESL is considered. This plot shows that after the resonance point, the impedance of the inductor (which increases with frequency) begins to dominate. This makes the capacitor ineffective as a bypass element at frequencies higher than its resonance point. Typically, this frequency point and the capacitor's value have an inverse relationship. This is why power supplies and other chips are bypassed by a range of different valued capacitors.

3.6 ESR

The ESR is the practical result of the fact that the materials used to create a capacitor have resistance. In simple cases, this can be approximated by a resistance in series with the main capacitor (See Figure: 9).

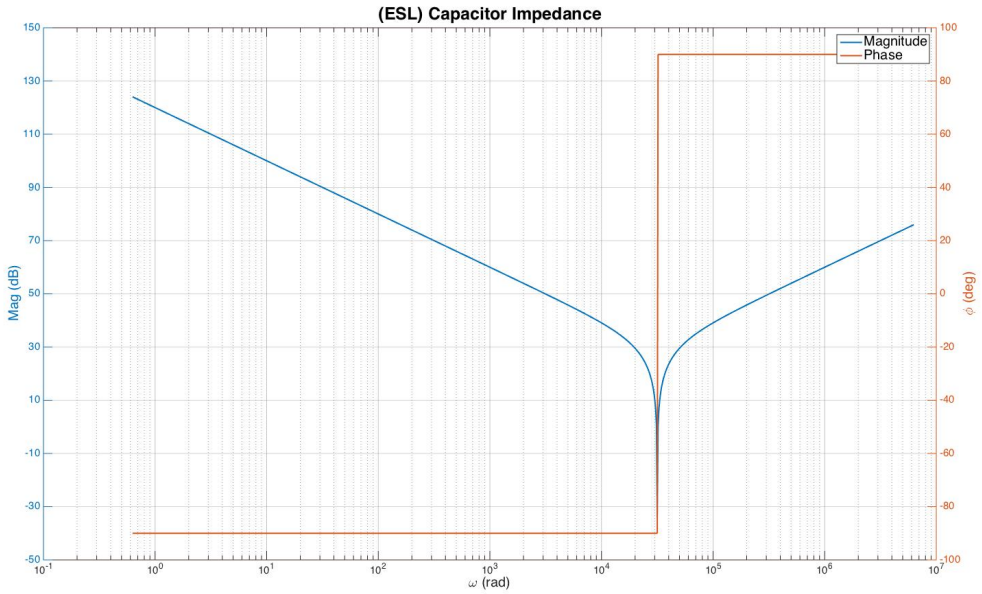


Figure 8: Capacitor Impedance with ESL

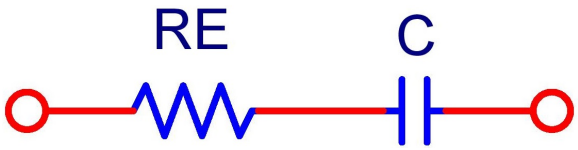


Figure 9: ESR Capacitor Model

ESR becomes important when thinking about DCDC switch mode power supplies. The output ripple voltage of the converter will cause a ripple current to pass through the ESR and dissipate heat as per Equation: (10). It is important to choose a low ESR capacitor in order to reduce failures.

$$P_E = I_{C,RMS}^2 * R_E \text{ [32]} \tag{10}$$

Another important thing to note about ESR is that even though it is shown as a resistance in simple models, it is not constant across all frequencies. It is a simplification of the resistive and capacitive elements in a capacitor that are dominated by resistance. That said, it is still sufficient for a basic understanding

of a capacitor's impedance (Equation (11)).

$$\vec{Z_c} = \frac{1 + j\omega R_E C + (j\omega)^2 L_E C}{j\omega C} \tag{11}$$

3.7 Resonance Frequency

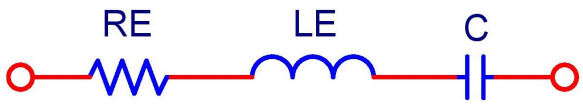


Figure 10: RLC Capacitor Model

Once C, ESL, and ESR are included into the capacitor model (Figure: 10), a parameter known as the self-resonant frequency becomes evident. Equation: (11) shows that when $Z_{ESL} == Z_C$, the capacitor is at its resonance point. At this frequency, the capacitor's impedance is determined solely by the ESR. This frequency can be calculated by Equation: (12).

$$f_r = \frac{1}{2\pi\sqrt{LC}} \tag{12}$$

3.8 Dissipation Factor

The Dissipation Factor (DF), otherwise known as the loss-tangent, is a measure of the energy stored to the energy dissipated per cycle. It is a measurement of the efficiency of the capacitor. The DF can be quantified through Equation: (13).

$$D = \frac{R_E}{X_C} \tag{13}$$

The loss tangent can be seen in Figure: 11. The greater the angle, the more efficient the capacitor will be.

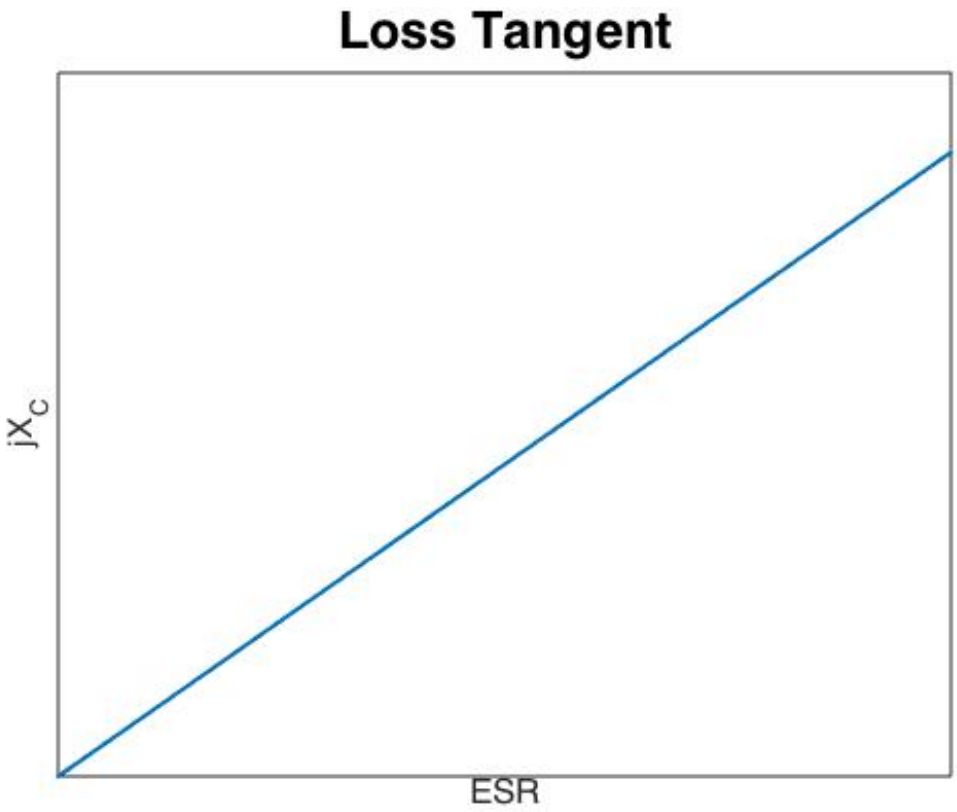


Figure 11: Loss Tangent

3.9 Quality Factor

$$Q = \frac{1}{D} \tag{14}$$

The Quality Factor (Q) of a capacitor is found by taking the reciprocal of the Dissipation Factor, Equation: (14). It is defined as the ratio of the energy stored to the energy dissipated per cycle.

3.10 Leakage Resistance

Every capacitor will have some DC leakage resistance associated with it (Figure:12). This resistance affects the capacitor's ability to store charge. A capacitor with a high leakage resistance has a low self-discharge rate. This characteristic is especially

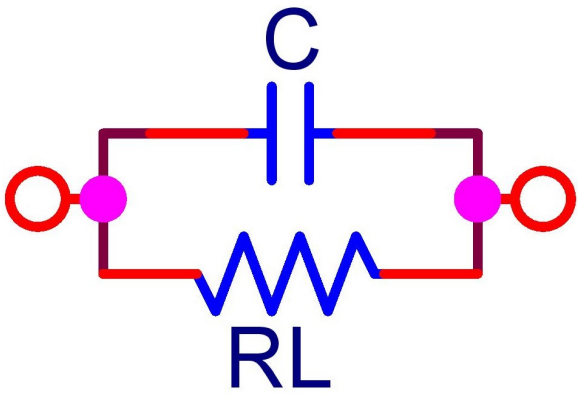


Figure 12: Capacitor Leakage Model

important in sample and hold circuits.

3.11 Dielectric Absorption

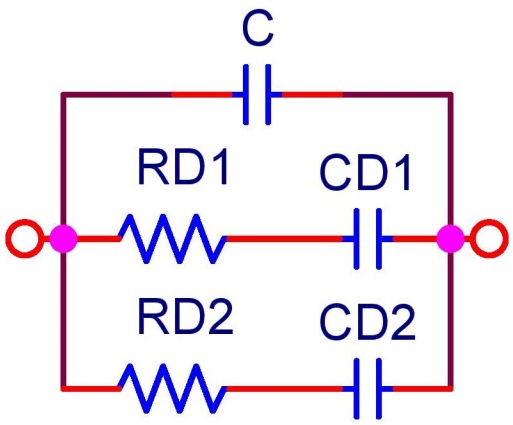


Figure 13: Dielectric Absorption

Dielectric Absorption (DA) in a capacitor is a characteristic which describes the unit's ability to "regenerate" a voltage after being shorted to ground for a brief time.

As seen in Figure: 13, a capacitor can be modeled with multiple RC elements in parallel with the bulk capacitance. When the main capacitor is shorted to ground for a short time and then released, the other capacitors are not guaranteed to have been fully discharged. After several minutes, they can recharge the main capacitance

to a significant portion of its original charge. This is why large valued electrolytic capacitors get shipped with a resistor across their terminals.

3.12 Six Term Model

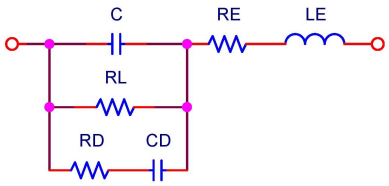


Figure 14: 6 Term Model

The model shown in Figure: 14 combines bulk capacitance ESR, ESL, leakage, and DA. It describes a capacitor in simple manner that allows for an understanding of its physical characteristics. It is used in the regression analysis in Section: 5 to determine the characteristics of unknown capacitors.

3.13 Murata Model

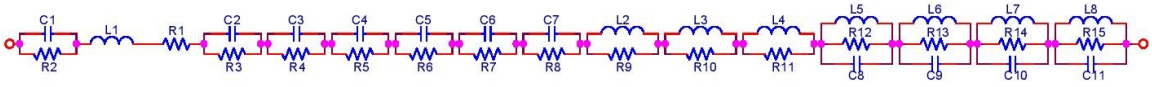


Figure 15: Murata GRM31MR71H105KA88 Model [46]

The model shown in Figure: 15 is a functional representation of a ceramic capacitor. It contains a single element for the bulk capacitance (C1), leakage (R2), and ESL (L1). R2, along with the rest of the components, represents an accurate model of the ESR. While this precisely describes how a particular capacitor changes over frequency, the large number of parameters does not lend to an intuitive understanding of the physical characteristics of the capacitor. This model will instead be used to provide an ideal capacitor frequency response to evaluate the regression analysis in Section: 5.

3.14 Electrochemical Capacitor Model

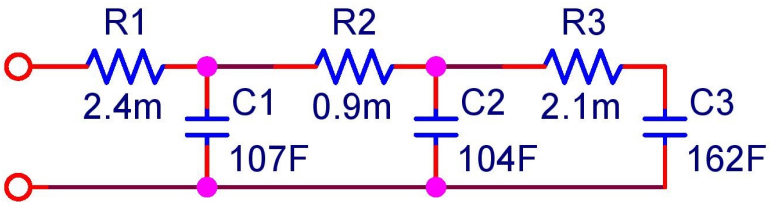


Figure 16: Electrochemical Capacitor [34][Fig: 8]

The model in Figure: 16, shown by Miller [34], describes the functionality of an electrochemical capacitor. It has a very high capacitance and several time constants on the order of 10s to 100s of milliseconds. Dependent upon future work, it may be an accurate representation for Titanium based capacitors which have energy and power densities on the same order of magnitude as electrochemical capacitors.

4 Measurement Circuitry

The proposed circuit in this section will perform the measurements described in the Introduction and is based in part on the measurement of anodized titanium capacitors done by Ehret [10]. The circuit will measure the respones of capacitors to various test inputs in order to generate data which can then be fit to an appropriate capacitor model (Section: 5). This method will allow for an analysis of the effect of the DC bias on the output model.

This circuity provides for two main testing modes. In the first mode, a large (up to 500VDC) bias voltage is coupled onto an AC signal via an optocoupler. This voltage is then applied to the Device Under Test (DUT) and the resulting current flow is measured through a transimpedance amplifier. The signal is then filtered and its magnitude and phase are measured and digitized. The user has control of the DC bias, AC amplitude, and frequency. These control parameters can be swept to obtain

a characterization of the DUT.

The second test mode provides a means to measure the discharge characteristics of the device. A DC bias is used to charge up the DUT and then disconnected. The circuity can either connect a discharge capacitor to the DUT or leave it open in order to measure the discharge current.

4.1 Power Supplies

The power supply circuitry shows the complete power generation scheme used for the circuit board. 24VDC is fused and then fed through an EMC compliance filter [28]. The initial fuse value is set to roughly twice the current rating of the switching power supplies. Once a practical current draw value is obtained, this value can be decreased. In order to ensure that the three switching regulators meet their rated regulation specifications, output shunt resistors are used to set a minimum load of 10%. All of the power supplies were chosen such that their power ratings were at least twice the maximum expected load. The power supply generating $\pm 15V_{ISO}$ is slightly different than the rest, as its output ground is attached to the DC bias voltage instead of signal ground. This purpose of this seen in the optocoupler circuitry described in Section: 4.3.

4.2 DC Bias

The DC bias control circuitry controls a Stanford Research Supply SRS-PS350 power supply to generate the bias voltage for the circuit. Some of the models of this supply provide a serial input which can be used to set their output voltage. This scenario is preferred and the communications for it are handled through the RS232 transceiver found on Schematic Page: 3. Other models of this supply only provide an analog input option to control their output voltage. The specifications for this method are

shown in Table: 1. The control signal is generated by using the microcontroller to set the Digital to Analog Controller (DAC) output over a SPI bus. It is then buffered and clamped to under 1.5V. Since the SRS-PS350 has a gain of 500, this clamps its output to 750VDC. The circuit is only designed to operate at up to 500VDC, but this value is still within the specified ratings of the components in the signal path.

Input Scale	0 to +10V for 0 to 5kV
Input Impedance	1 MΩ
Accuracy	±0.2% of full scale
Update Rate	15 Hz
Output Slew Rate	<0.3s for 0 to full scale (full load)

Table 1: SRS-PS350 Analog Control Characteristics [49][48]

The DAC has a 2.5V internal reference with a 14 bit resolution, giving the control signal an ideal resolution of $150\mu V$ and the SRS-PS350 an ideal output resolution of $76mV$. The update rate of this device is not important to this application because the DAC will be set infrequently and only when the system is not actively collecting data.

If different characteristics are needed, the SRS-PS350 can be exchanged for an alternate power supply, but the control logic may not be supported.

4.3 Optocoupler

The optocoupler circuitry is based upon IXYS's application note AN-107 [21], and its purpose is to inject an AC signal ontop of a high DC bias. The optocoupler is configured in what is known as photovoltaic mode [21][Sec: 3.3], with both photoreceivers acting as coupled current sources.

In this mode, the front end op-amp's feedback path causes the first photoreceiver to draw current through the first resistor such that both of its terminals stay at ground

potential. The phototransmitter's resistor sets a full scale current of 20mA, which is well within the absolute maximum ratings of the device. The second phototransmitter pulls current through the back end op-amp's feedback resistor with a voltage gain of 2.5.

The output of the optocoupler is referenced to the DC bias voltage from the SRS-PS350 instead of circuit ground. The limiting isolation specification comes from the $\pm 15V$ supply at 1.5kV, far below the 500V limit imposed by the system. The input sine wave is fed into the magnitude and phase measurement circuitry seen on Schematic Page: 7. The optically coupled sine wave is fed into the DUT as seen on Schematic Page 5.

4.4 Charging Circuitry

The DUT charging circuitry is meant to only be used to prepare the DUT for a test. The current is purposely limited in order to minimize the load on the SRS-PS350. The charging operation functions by enabling the high-current measurement and charging relays, and then slowly ramping the DC bias voltage until the DUT is fully charged.

4.5 Discharging Circuitry

The discharging circuitry provides a means to determine the bulk capacitance value of the DUT from the RC time constant. Once the DUT is charged with the operation described in Section: 4.4, the charging relay will open and the high-current discharge relay will close. The transimpedance amplifier described in Section: 4.6 will measure the current through the DUT as it decays.

There are three switched discharge stages that allow for the decay rate to be regulated. Each stage allows for a range of voltages and capacitances needed to stay within safety and operational constraints. As seen in Figure: 17, there are constraints

on both the voltage and capacitance. The maximum voltage is constrained by either the power rating on the lowest valued resistor or the overall safety limit of 500VDC. The minimum capacitance value is constrained by setting a 100 Analog to Digital Converter (ADC) samples per time constant with a sampling rate of 250KSPS.

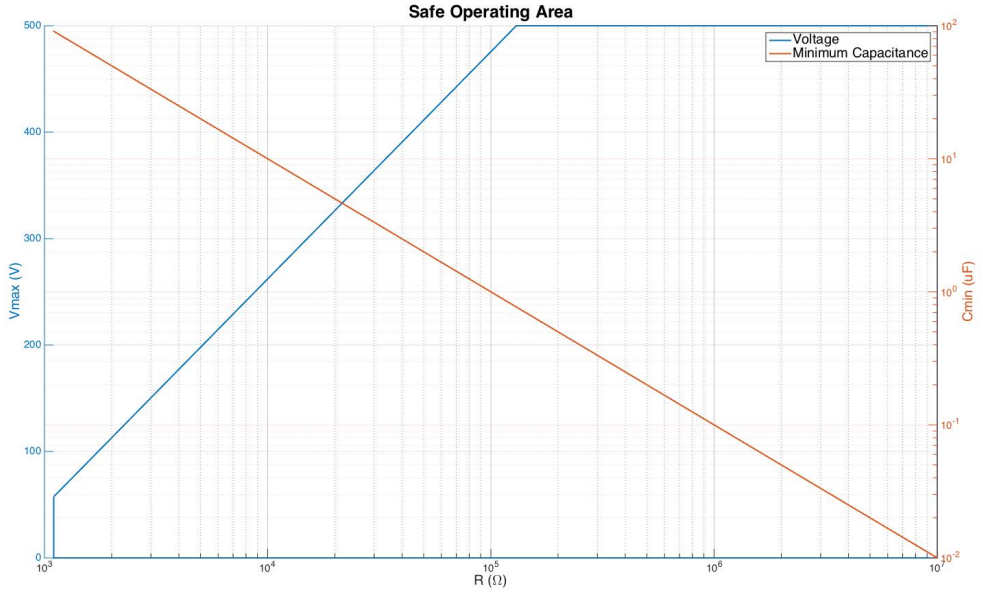


Figure 17: Operating Area

4.6 Current Measurement

In light of the high DC bias voltage present in the system, the current measurement circuitry utilizes a low-side current measurement technique. This allows the measurement electronics to operate at relative low (5VDC) voltages while measuring current from the DUT with a DC voltage in the 100s of volts.

The current measurement circuit utilizes a transimpedance amplifier designed from the reference circuit in [10]. It utilizes three switched, feedback stages to measure nine decades of current (See Table: 2). The circuit operates by creating a virtual ground at the negative terminal of the DUT, and then directing the current through

R_f Value	Range	Current
5	Hi	1A - 1mA
5k	Med	1mA - 1uA
5M	Low	1uA - 1nA

Table 2: Current Measurement Ranges

one of its feedback paths. The resultant voltage in Equation: (15) is then filtered and sent to the next stage for further signal conditioning.

$$Vo = I * R_f \tag{15}$$

The AD817 op-amp has the precision and bandwidth needed to operate in a transimpedance amplifier configuration, but does not have the required drive strength for the high current range. This limitation is overcome by inserting a BJT based current booster in the feedback path. This preserves the benefits of the op-amp, while providing the ability to drive a high-current signal.

4.7 Filtering

The filtering circuitry implements two filtering paths which each consist of a 6-pole, low-pass, Sallen-Key filter to condition the signal.

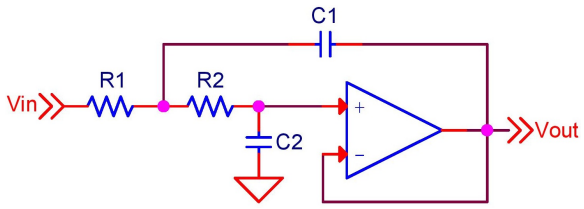


Figure 18: Low-Pass Sallen-Key Filter [43]

$$f_c = \frac{1}{2\pi R_2 C_2 \sqrt{\frac{R_1 C_1}{R_2 C_2}}} \text{ [38][}ch\text{ 6.3.2.}D\text{ }equ\text{ 6.5 }pg\text{ 410]} \tag{16}$$

$$Q = \frac{\sqrt{\frac{R_1 C_1}{R_2 C_2}}}{1 + \frac{R_1}{C_1}} \text{ [38][}ch\text{ 6.3.2.}D\text{ }equ\text{ 6.6 }pg\text{ 410]} \tag{17}$$

A generic, single-stage, low-pass, Sallen-Key filter is shown in Figure: 18. The filter is basically a 2-pole, passive, RC filter with the ground of C_1 tied to V_{out} through the op-amp's active feedback path. This implementation is a simplified version of the Sallen-Key filter with the gain (listed as K in most explanations) equal to 1. The transition frequency and quality factor can be set according to Equations: (16) and (17) respectively by choosing values for the passives.

Filter	**Filter Order**	**Gain**	f_c	**Stopband Attenuation**
Discharging	6	1	25kHz	-65dB
AC Measurement	6	1	100kHz	-100dB

Table 3: Sallen-Key Filter Specifications

The filters for this circuit were designed with TI's WEBENCH designer [53]. Table: 3 shows the specifications for each filter. The AC current measurement branch has a cutoff frequency of 100kHz. With the overall measurement frequency range capped at 10kHz, this provides sufficient anti-aliasing for the measurement. The discharge measurement filter has a cutoff of 25kHz. With a $1k\Omega$ resistor and a 1uF DUT having a time constant of $10\mu s$, this provides a sufficient margin for 100 points per time constant requirement while still filtering out higher frequency components.

4.8 Magnitude

The magnitude circuitry measures both the magnitude of the input sine wave and the magnitude of the resultant current waveform through the DUT. The design for

this circuit is based on Analog Device's application note for the AD8277 [35].

The function of this circuit is to perform a full-wave rectification operation on the input signal and then pass the output to be filtered and then sampled. The first difference amplifier is configured as a voltage follower. Since it is only powered from a single rail, it passes the positive half of the waveform and shunts the negative half to ground. The second difference amplifier operates in different modes, based upon the output of the first differential amplifier. During the negative going portions of the waveform, its positive terminal is held at ground, and it operates as a unity gain, inverting amplifier. This rectifies the input signal to the output. During the positive going portions of the waveform, positive input terminal is held at the value of the input terminal. This forces the second difference amplifier to act as a voltage follower. The resultant waveform is a full-wave rectified version of the input signal. It is then bypassed with an output capacitor and fed into the ADC to calculate the value of the rms voltage.

4.9 Phase

The phase circuitry measures both the phase of the input sine wave and the phase of the resultant current waveform through the DUT. It uses two, redundant phase measurement techniques.

The first technique utilizes the AD8302, a Radio Frequency (RF) phase detector, in a low frequency configuration as per AN-691 [39]. The capacitors on the IC inputs tune the front end high pass filter cutoff to roughly 20Hz. The phase measurement is accomplished internally by a magnitude independent phase multiplier configuration. The output consists of a 10mV per degree signal that is low passed at 2Hz to prevent aliasing. The filtered signal is then fed to the ADC for digitization.

The second technique is added for a low frequency comparison against the RF phase detector. The comparator acts as a sine to square wave converter. The default

configuration relies on the comparator's internal hysteresis, but provisions for wider thresholds are available if needed. The outputs of the two converters are fed into timer inputs on the microcontroller. The phase is then calculated by the Microcontroller Unit (MCU) by measuring the period of the signals and the time difference between their rising edges.

4.10 ADC

The ADC circuitry makes use of the AD7656, a 6 channel, 250ksps, 16bit, SAR ADC. It sends the converted values over three parallel SPI outputs to the MCU for further processing and analysis.

4.11 Communications

The communications circuitry is used to communicate to off-board processors.

4.11.1 USB

The Universal Serial Bus (USB) section is used to communicate with a PC for data logging and post processing. This circuitry centers around a FTDI FT232H serial to USB chip. It allows seamless USB communication to a PC's COM port with the MCU only using a single UART port. This significantly lowers the complexity of the communication bus, as the MCU is not responsible for running the USB stack.

4.11.2 RS-232

The board also has two, bidirectional RS-232 ports. They are able to be used for general purposes, but will most often be used for communicating with a sine-wave generator and the DC Bias supply.

4.12 Circuity Capabilities

Under most operating conditions, the circuit described in Section: 4 can measure the DUT with a theoretical error of less than 1%. This section will describe the operating conditons which lead to a larger measurement error. The OPA365 operational amplifier in Section: 4.6 has an input offset voltage of $200\mu V$. This causes about a 4% error in the impedance measurement at the low end of each current measurement range (Table: 2). This error drops to less than 1% by the time the current increases by half of an order of magnitude. The OPA365 also has an input offset current of $10pA$. This results in a maximum error of 1% when measuring a current of $1nA$. These error sources combine for a worst case erorr of 5%.

The filtering circuitry described in Section: 4.7 introduces an additional error. In this implementation, the OPA365's input offset voltage voltage is multiplied by 3 to introduce a $600\mu V$ offset error. This equates to a 12% error at the low end of each current range and a 1.2% error at an order of magnitude above that. The main affect of this error is to reduce the maximum current that can be measured in each range. The circuit can be calibrated in software to remove the offset error from the final result so that it does not appreciably affect the measurement for most of the range.

The AD8392 circuit described in Section: 4.8 introduces a frequency dependent error (if left uncompensated). An impedance analysis $10\mu F$ capacitor with a $100mV$ excitation voltage results in a maximum magnitude error of 10% at 500Hz. This is due to the fact that the AD8392 has an error that increases with the input gain. 500Hz is the first frequency in this scenario that requires the high gain stage of the current measurement. The phase error for the AD8392 is always less than 1% when the capacitor is not near the resonance point.

4.13 Comparison with Agilent 4294A

There are various instrumentation units on the market that provide a similar function as the circuit described in this section. For instance, the Agilent 4294A impedance analyzer has the ability to measure a capacitor's impedance at a greater frequency range and with a basic accuracy of 0.08% [2]. Also, with the 16065A attachement, it has the ability to measure these parameters at up to a 200V bias [1], as opposed to the 500V capability in this circuit. The other main disadvantage of the Agilent meter is that it does not provide the ability to measure discharge curves. As stated in Section: 5.3, this characteristic is important in determining the capacitor's ability to deliver power.

5 Regression Analysis and Modeling

Once the circuitry described in Section: 4 collects the data for a capacitor, an analysis is needed to relate that information to the physical characteristics described in Section: 3. Various models, such as in Figures: 14 & 16, can be used to represent a capacitor. They differ based upon the type of capacitor, the input parameters of interest, and the desired accuracy. This section demonstrates a regression analysis technique which fits a capacitor model to the impedance data shown in Figure: 19. The accuracy of these fits will be evaluated to determine the model's effectiveness. See Appendix: B for the code used to generate the images shown in this section.

5.1 Regression Analysis

5.1.1 Basic LSE

At its core, regression analysis is an optimization problem whose purpose is to fit the equation of a line to a data set. A commonly used regression analysis technique is

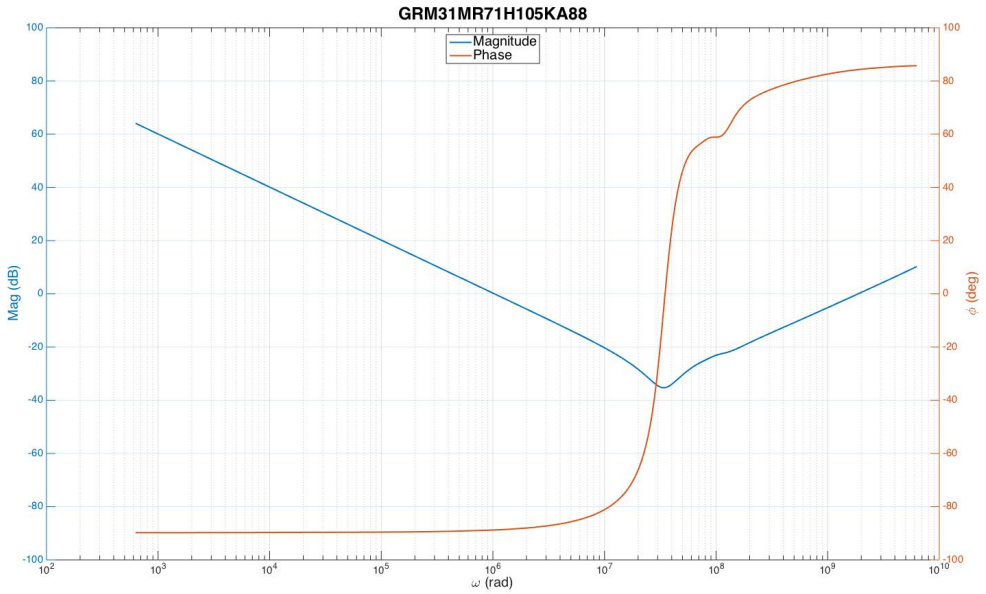


Figure 19: GRM31MR71H105KA88 Capacitor Data [46]

called the Least Squares Estimate (LSE). It attempts to find a model which minimizes the squared error between an empirical set of data and itself.

The first step in applying an LSE is to choose the form of the equation that best represents the data. The equation of a line, Equation: (18), is chosen when only a simple linear fit is needed. Then the squared error equation is generated, as in Equations: (19) & (20).

$$y = a_0 + a_1 x \tag{18}$$

$$E^2 = \sum_{i=1}^{n} (y_i - y)^2 \tag{19}$$

$$E^2 = \sum_{i=1}^{n} (y_i - (a_0 + a_1 x_i))^2 \tag{20}$$

In order to minimize the squared error over the data set, one needs to take the

partial derivate of Equation: (20) with respect to each of the unknown parameters, the coefficients, separately. While a_0 and a_1 will be constants in the final equation, they are treated as variables here until they are known. Conversely, all x_i values are treated as constants. This results in Equations: (21) & (22).

$$\frac{\partial E^2}{\partial a_0} = 0 = \sum_{i=1}^{n}(-2y_i + 2a_0 + 2a_1x_i) \tag{21}$$

$$\frac{\partial E^2}{\partial a_1} = 0 = \sum_{i=1}^{n}(-2y_ix_i + 2a_0x_i + 2a_1x_i^2) \tag{22}$$

Up to this point most LSE methods analyze follow the same basic path. The rest of the steps will depend upon the complexity of the model and solution techniques. The following steps in the basic LSE use transformations and substitutions to solve for the unknown variables. In this case, Equation: (23) is used to remove the summation terms from the equation.

$$\sum_{i=1}^{n} y_i = \bar{y}n \tag{23}$$

This results in Equations: (24) & (25) with solutions shown in Equations: (26) & (27). The empirical data is then used to find the values of a_0 and a_1. At this point, the best fit line can be used to estimate new points on the plot or to compare against other data sets (Figure: 20).

$$0 = \bar{y} - (a_0 + 2a_1\bar{x}) \tag{24}$$

$$0 = \bar{xy} - (a_0\bar{x} + 2a_1\bar{x^2}) \tag{25}$$

$$a_0 = \bar{y} - a_1\bar{x} \tag{26}$$

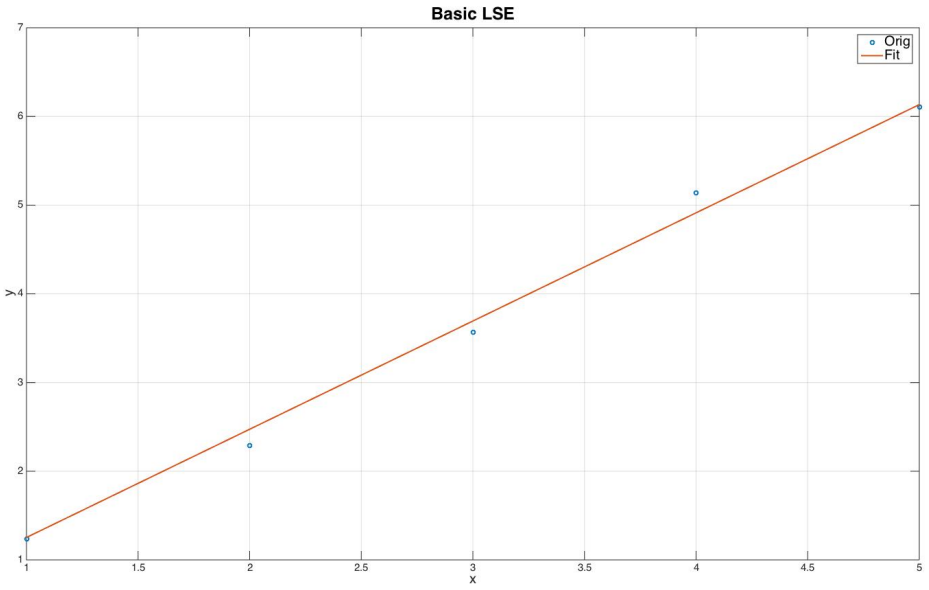


Figure 20: Basic LSE

$$a_1 = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x^2} - \bar{x}^2} \tag{27}$$

5.1.2 Levy's Technique - Complex Curve Fitting

While the basic LSE technique is sufficient for many circumstances, it is not directly applicable in situations where one needs to fit a model to a complex line, such as a transfer function. Levy [30] shows an extension of the simple LSE example that is valid for a generic polynomial transfer function. This method is important because it not only allows for a complex-valued transfer function, but it also prevents the necessity of rederiving the system of equations for each new model.

$$G(s) = \frac{A_0 + A_1 s + A_2 s^2 + ... + A_n s^n}{B_0 + B_1 s + B_2 s^2 + ... + B_m s^m} \; [30][Eq.\ 3] \tag{28}$$

Using Equation: (28) as the generic model, Levy shows that you can use Equations: (29), (30), (31), & (32) to simplify the series of partial derivatives into a

single matrix multiplication equation shown in (33), (34), (35), & (36).

$$\lambda_h = \sum_{k=0}^{m} \omega_k^h \text{ [30][}Eq.\text{ 15]} \tag{29}$$

$$S_h = \sum_{k=0}^{m} \omega_k^h R_k \text{ [30][}Eq.\text{ 16]} \tag{30}$$

$$T_h = \sum_{k=0}^{m} \omega_k^h I_k \text{ [30][}Eq.\text{ 17]} \tag{31}$$

$$U_h = \sum_{k=0}^{m} \omega_k^h (R_k^2 + I_k^2) \text{ [30][}Eq.\text{ 18]} \tag{32}$$

$$MN = C \text{ [30][}Eq.\text{ 20]} \tag{33}$$

$$M = \begin{bmatrix} \lambda_0 & 0 & -\lambda_2 & 0 & \lambda_4 & \cdots & T_1 & S_2 & -T_3 & -S_4 & T_5 & \cdots \\ 0 & \lambda_2 & 0 & -\lambda_4 & 0 & \cdots & -S_2 & T_3 & S_4 & -T_5 & -S_6 & \cdots \\ \lambda_2 & 0 & -\lambda_4 & 0 & \lambda_6 & \cdots & T_3 & S_4 & -T_5 & -S_6 & T_7 & \cdots \\ 0 & \lambda_4 & 0 & -\lambda_6 & 0 & \cdots & -S_4 & T_5 & S_6 & -T_7 & -S_8 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \\ T_1 & -S_2 & -T_3 & S_4 & T_5 & \cdots & U_2 & 0 & -U_4 & 0 & U_6 & \cdots \\ S_2 & T_3 & -S_4 & -T_5 & S_6 & \cdots & 0 & U_4 & 0 & -U_6 & 0 & \cdots \\ T_3 & -S_4 & -T_5 & S_6 & T_7 & \cdots & U_4 & 0 & -U_6 & 0 & U_8 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \text{ [30][}Eq.\text{ 21}a\text{]} \tag{34}$$

$$N = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ \vdots \\ B_1 \\ B_2 \end{bmatrix} \text{ [30][}Eq.\text{ 21}b\text{]} \tag{35}$$

$$C = \begin{bmatrix} S_0 \\ T_1 \\ S_2 \\ T_3 \\ \vdots \\ 0 \\ U_2 \\ 0 \\ \vdots \end{bmatrix} \quad [30][Eq.\ 21c] \qquad (36)$$

Levy's technique works well for applications where there is a small dynamic frequency range and a small number of coefficients, but there a several problems with it. The first problem is that for models with a wide bandwidth, the solution to Equation: (33), involves an ill-conditioned matrix. This means that the solution to the system of equations will experience precision errors. The second problem is that this technique is heavily biased by high frequency data. As can be seen in Figure: 21, applying Levy's technique with a 3^{rd} order model provides a unusable fit to the empirical data.

5.1.3 Weighted LSE

One improvement that can be made upon Levy's method is to iterate with a frequency dependent weighting function until the error term is minimized [44]. By multiplying Levy's error function by the weighting term in Equation: (37), we get Equation: (38), which can be used to obtain a new system of equations. The L subscript stands for the current iteration, while $L-1$ stands for the previous iteration.

$$W_{kL} = \frac{1}{|Q(jw_k)_{L-1}|^2} \ [44] \qquad (37)$$

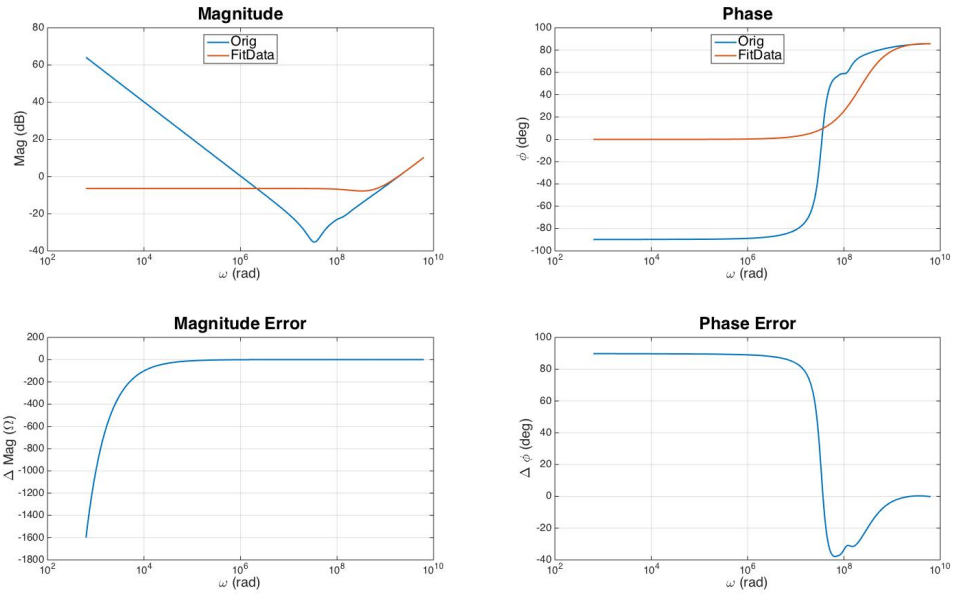


Figure 21: Levy's Technique

$$E = \sum_{k=1}^{n} |\epsilon_k^{'}|^2 W_{kL} \ [44][Eq.\ 7] \tag{38}$$

Equations (33), (34), (35), & (36) are the same, with Equations (29), (30), (31), & (32) being replaced with Equations: (39), (40), (41), & (42).

$$\lambda_h = \sum_{k=0}^{m} \omega_k^h W_{kL} \ [44][Eq.\ 9] \tag{39}$$

$$S_h = \sum_{k=0}^{m} \omega_k^h R_k W_{kL} \ [44][Eq.\ 10] \tag{40}$$

$$T_h = \sum_{k=0}^{m} \omega_k^h I_k W_{kL} \ [44][Eq.\ 11] \tag{41}$$

$$U_h = \sum_{k=0}^{m} \omega_k^h (R_k^2 + I_k^2) W_{kL} \ [44][Eq.\ 12] \tag{42}$$

Dependent upon the initial conditions and the model order, this particular method is not guaranteed to converge. Figure: 22 was generated with an initial condition of $Q(jw_k) == 1$ and a model order of 7. It shows that, for this data set, the squared error of the magnitude and phase do not converge after a particular number of iterations. Furthermore, they do not reach their minimums at the same iteration. In order to select the desired iteration, the magnitude and phase squared plots are normalized as in Equation: (43). The index of the minimum of Figure: 22 is selected as the best fit.

$$n = min(\frac{E_{Mag}}{max(E_{Mag})} + \frac{E_{pha}}{max(E_{Pha})}) \tag{43}$$

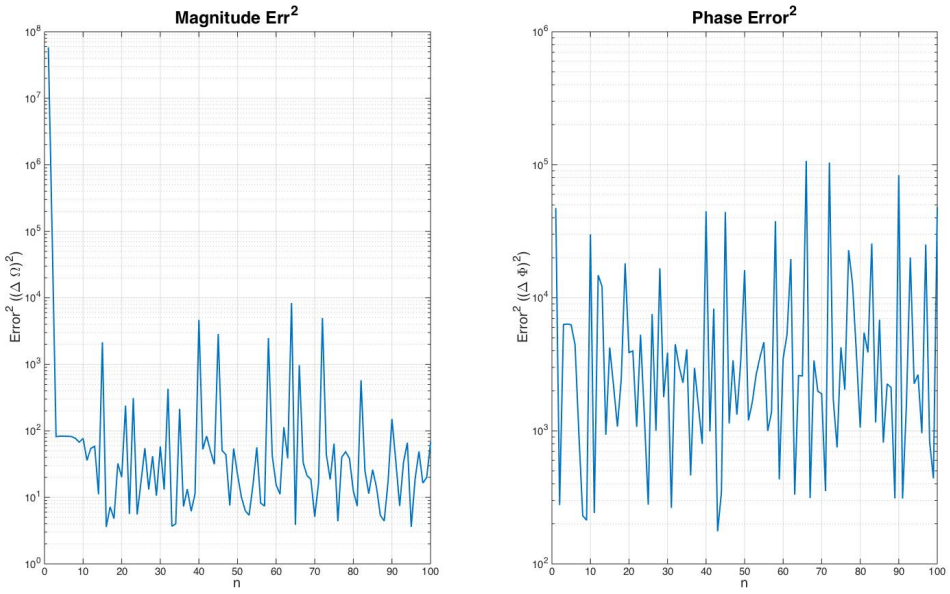


Figure 22: LSE + Iteration – Magnitude and Phase Error

Figure: 24 shows that this method can result in a much improved result over the Levy's original method, as seen in Figure: 21.

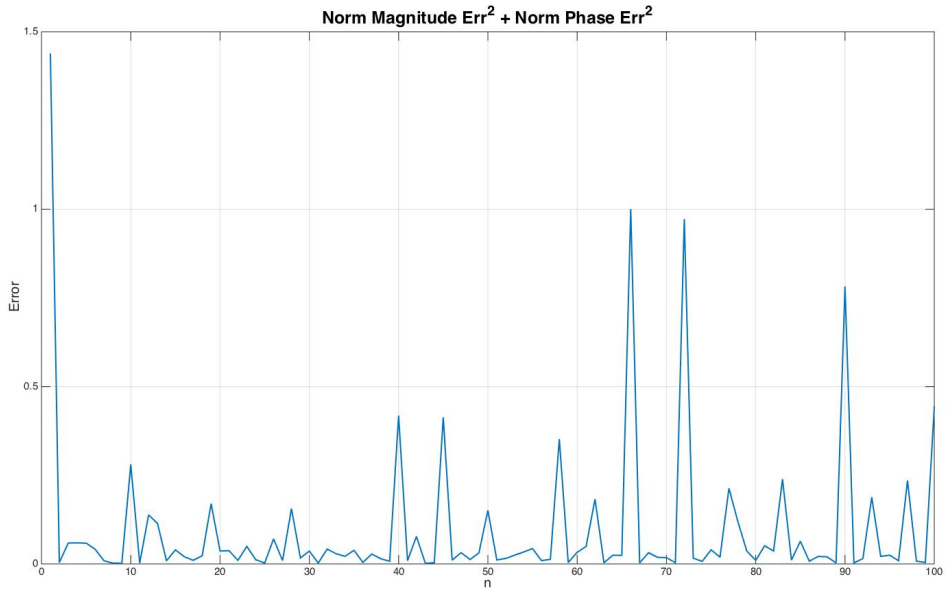


Figure 23: LSE + Iteration – Combined Error

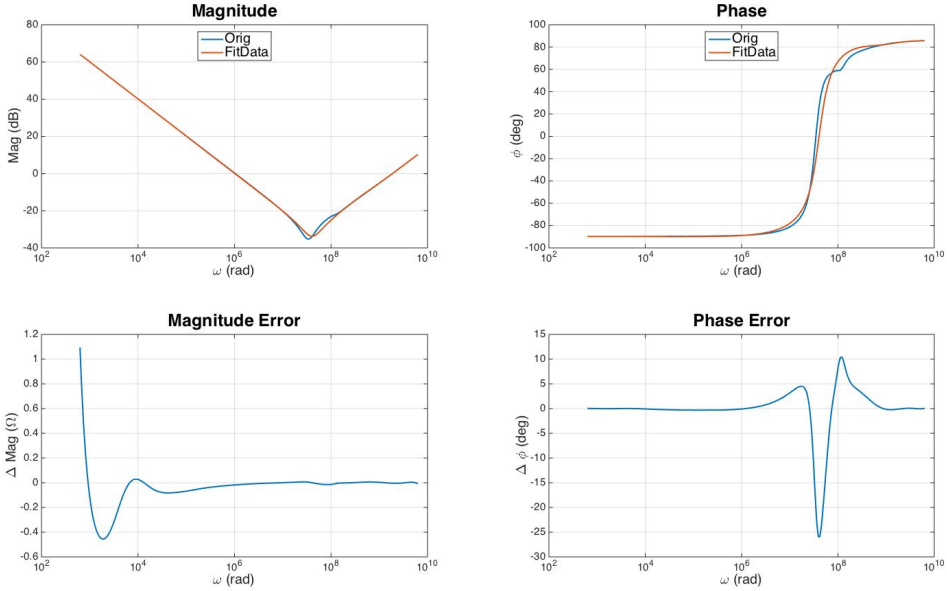


Figure 24: LSE + Iteration

5.2 Modeling

This section will investigate a model which includes the most common parameters of merit. It will show how to fit the model to a data set with Levy's method described in Section: 5.1, and will describe its effectiveness and limitations in doing so. The data in Figure: 19 used to fit against was generated from the model shown in Figure: 15. Since the model described in this section is of a smaller order than the model used to generate the data, there will be a loss of precision. This section will explore the error seen with the abbreviated model.

5.2.1 Six Term Model

The model shown in Figure: 14 shows several of the most important parameters for a capacitor. It's impedance, shown in Equation: (44), and generalized in Equation: (45) can be used as the basis for a regression analysis.

$$\begin{aligned}\bar{Z}(s) = &\frac{(R_E + R_L) + (L_E + C_D R_D R_E + C_D R_D R_L + C R_E R_L + C_D R_E R_L)s}{1 + (C_D R_D + C R_L + C_D R_L)s + C C_D R_D R_L s^2} \\ &+ \frac{(C_D L_E R_D + C L_E R_L + C_D L_E R_L + C C_D R_D R_E R_L)s^2 + C C_D L_E R_D R_L s^3}{1 + (C_D R_D + C R_L + C_D R_L)s + C C_D R_D R_L s^2}\end{aligned} \tag{44}$$

$$\bar{Z}(s) = \frac{a_0 + a_1 s + a_2 s^2 + a_3 s^3}{b_0 + b_1 s + b_2 s^2} \tag{45}$$

For this model, Equations: (34), (35), & (36) simplify down to Equations: (46), (47), & (48).

$$M = \begin{bmatrix} \lambda_0 & 0 & -\lambda_2 & 0 & T_1 & S_2 \\ 0 & \lambda_2 & 0 & -\lambda_4 & -S_2 & T_3 \\ \lambda_2 & 0 & -\lambda_4 & 0 & T_3 & S_4 \\ 0 & \lambda_4 & 0 & -\lambda_6 & -S_4 & T_5 \\ T_1 & -S_2 & -T_3 & S_4 & U_2 & 0 \\ S_2 & T_3 & -S_4 & -T_5 & 0 & U_4 \end{bmatrix} \tag{46}$$

$$N = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ B_1 \\ B_2 \end{bmatrix} \quad (47) \qquad C = \begin{bmatrix} S_0 \\ T_1 \\ S_2 \\ T_3 \\ 0 \\ U_2 \end{bmatrix} \quad (48)$$

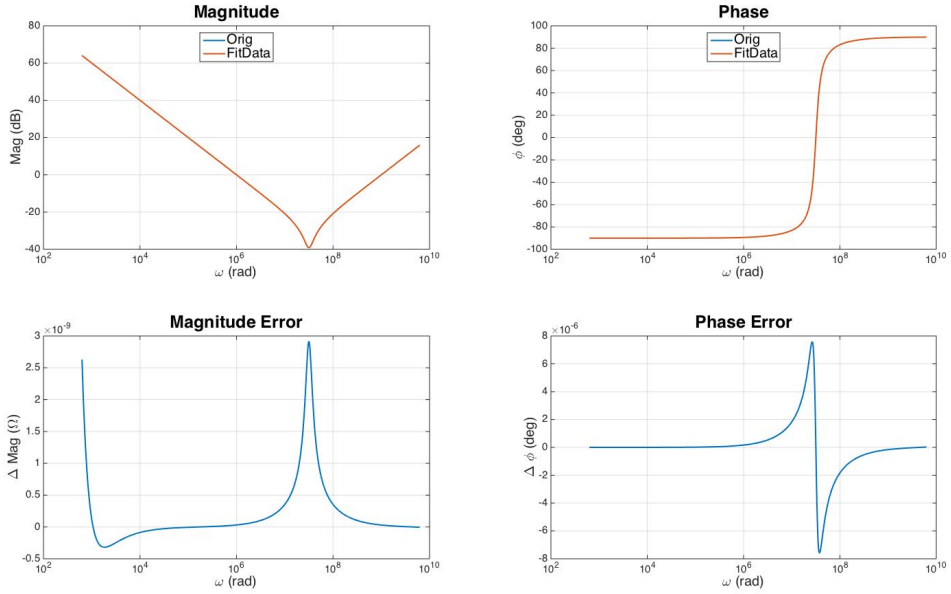


Figure 25: 6 Term Model Validation

Figure: 25 shows the results from simplifying the model in Figure: 15 to the six term model and then applying this regression analysis to that model. Figure: 26 shows that the relative error in fitting this model to itself is negligible.

Using this model, the regression analysis method described in Section: 5.1.3 generates the output seen in Figure: 27. The fit tracks the original data well, except at low frequencies and near resonance.

Substituting the polynomial coefficients seen in Equation: (56) into Equations:

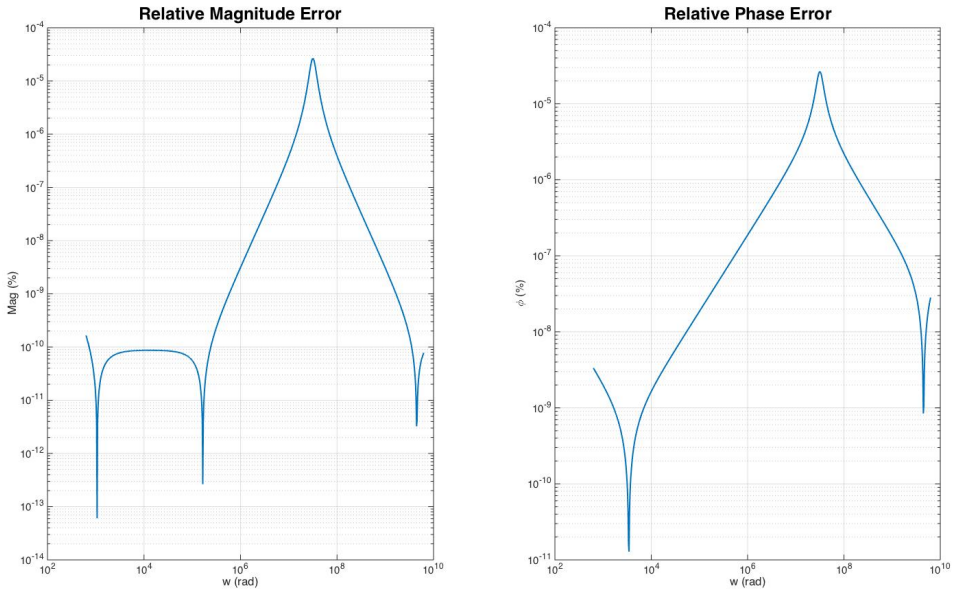


Figure 26: 6 Term Model Validation: Relative Erorr

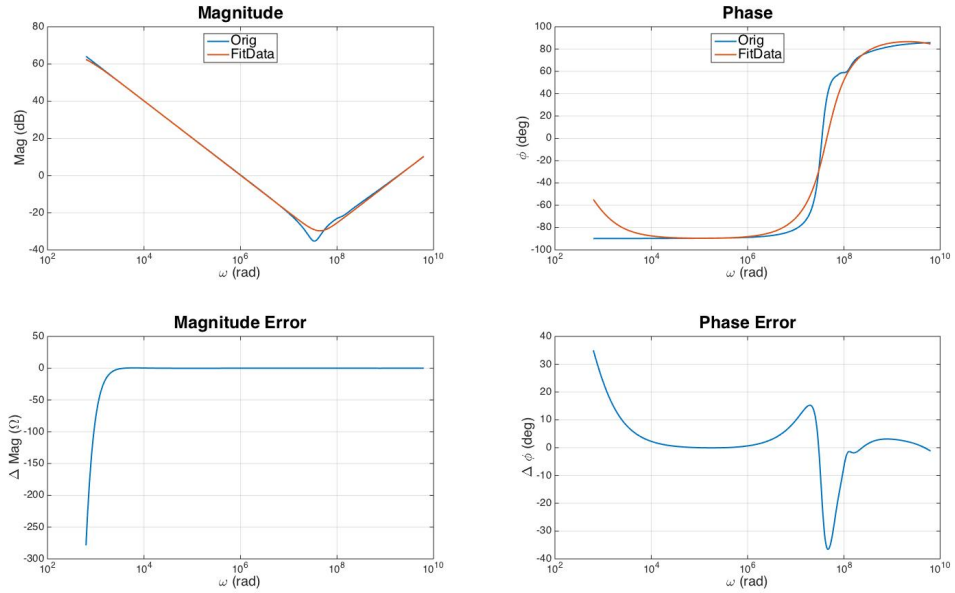


Figure 27: 6 Term Model: Bad Initilization

(49)-(55), returns the model parameter values as seen in Equation: (57).

$$a_0 = R_E + R_L \tag{49}$$

$$a_1 = L_E + C_D R_D R_E + C_D R_D R_L + C R_E R_L + C_D R_E R_L \tag{50}$$

$$a_2 = C_D L_E R_D + C L_E R_L + C_D L_E R_L + C C_D R_D R_E R_L \tag{51}$$

$$a_3 = C C_D L_E R_D R_L \tag{52}$$

$$b_0 = 1 \tag{53}$$

$$b_1 = C_D R_D + C R_L + C_D R_L \tag{54}$$

$$b_2 = C C_D R_D R_L \tag{55}$$

Even though the polynomial coefficients gave an acceptable fit to the data, it resulted in unacceptable circuit parameters. Since C and R_D are negative, this model is not physically realizable.

$$\begin{aligned} a_0 &= 5.9991E^{+03} \\ a_1 &= 1.7934E^{-04} \\ a_2 &= 3.3158E^{-12} \\ a_3 &= 6.8295E^{-22} \\ b_0 &= 1.0000 \\ b_1 &= 5.9057E^{-03} \\ b_2 &= 1.4067E^{-12} \end{aligned} \tag{56}$$

$$\begin{aligned} C &= -8.2563E^{-10} \\ R_E &= 3.1886E^{-01} \\ L_E &= 4.8551E^{-10} \\ R_L &= 4.8551E^{-10} \\ C_D &= 9.8536E^{-07} \\ R_D &= -2.8824E^{-01} \end{aligned} \tag{57}$$

This problem stems from the author's [44] suggestion that the initial guess for $Q(jw_k) == 1$. While an initial guess is required to calculate the weighting function during the first iteration, this particular choice causes the problem seen in Equations: (58) & (59) for this application. They show that this initial condition does not start

with rational solution for the circuit parameters.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 1 \\ a_2 &= 1 \\ a_3 &= 1 \\ b_0 &= 1 \\ b_1 &= 0 \\ b_2 &= 0 \end{aligned} \tag{58}$$

$$\begin{aligned} C &= INF \\ R_E &= INF \\ L_E &= INF \\ R_L &= IND \\ C_D &= IND \\ R_D &= IND \end{aligned} \tag{59}$$

Starting instead with the rational set of circuit parameters seen in Equation: (61), the starting coefficients are as seen in Equation: (60).

$$\begin{aligned} a_0 &= 2 \\ a_1 &= 5 \\ a_2 &= 4 \\ a_3 &= 1 \\ b_0 &= 1 \\ b_1 &= 3 \\ b_2 &= 1 \end{aligned} \tag{60}$$

$$\begin{aligned} C &= 1 \\ R_E &= 1 \\ L_E &= 1 \\ R_L &= 1 \\ C_D &= 1 \\ R_D &= 1 \end{aligned} \tag{61}$$

Figure: 28 shows a good fit across the frequency spectrum for both magnitude and phase. The model does deviate at resonance, but that is not surprising, as the

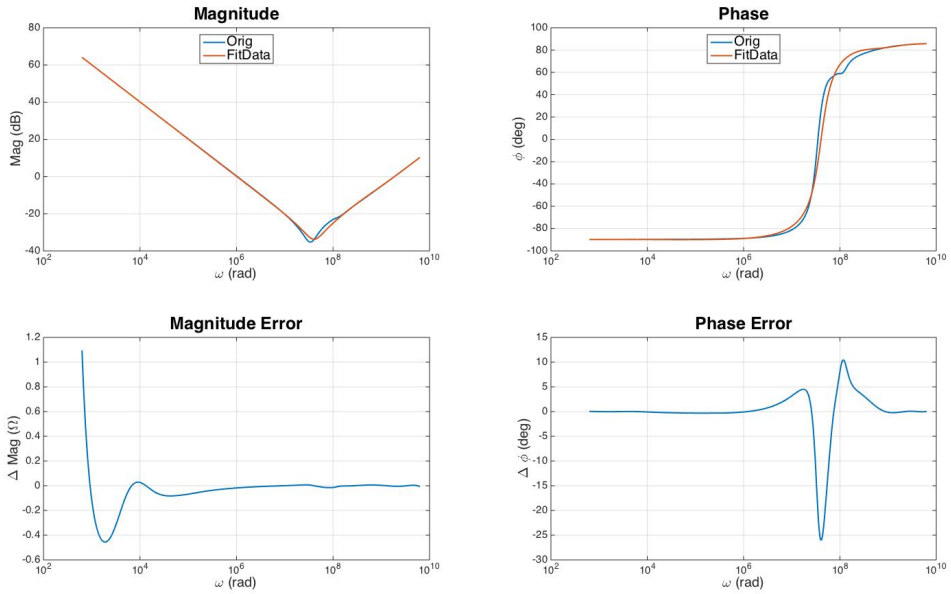


Figure 28: 6 Term Model: Good Initilization

number of parameters in the model is fairly low when compared with the model used to generate the data. As can be seen in Figure: 29, the relative error is the worst around resonance. It is at that point where there is simultaneously the lowest impedance and the sharpest feature in the plot.

The 6 term model described in this section provides a reasonable starting point for the capacitor characterization. As mentioned in the Conclusion and Future Work sections, this model has enough accuracy to characterize new capacitors. Empirical testing is needed in order to determine if the model needs to be adjusted to better characterize a specific DUT of interest.

5.3 Modeling Discharge Waveforms

Electric vehicles often rely on their DC-Link capacitors for acceleration. The capacitor's behavior in theses circumstances can be predicted by fittings its discharge curves to a model. Miller's model, seen in Figure: 16, can be fit to a discharge

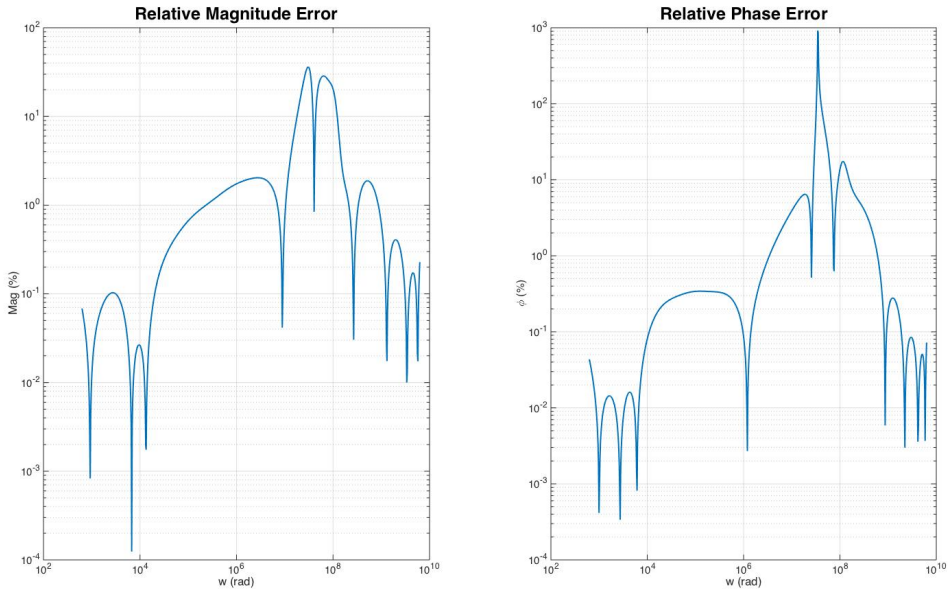


Figure 29: 6 Term Model: Relative Error

waveform to show how an electrochemical capacitor delivers power. This task can be accomplished by using MATLAB's "optimset" and "lsqnonlin" functions in the method developed by Hokanson [18]. The discharge current equation corresponding to Figure: 16 is omitted for brevity, but it can be generated by running the Mathematica code shown in the Discharge Equations Appendix.

6 Conclusion

In conclusion, the circuitry devolved in this thesis, along with the regression analysis developed, shows a promising ability to accurately model capacitors. The method shown provides the ability to model a capacitor over a $0 \rightarrow 500$VDC bias and a 100Hz $\rightarrow$ 100kHz range. The regression analysis provides a means to compare new capacitors with unknown characteristics. The accuracy of the model is mostly determined by the number of parameters and typically has the greatest relative error near resonance.

Outside of resonance and low frequencies, the accuracy of the fit stays within 2Ω and 2°. The relative error, outside of a single peak in the phase plot, stays well below 5%.

7 Future Work

This section will describe the future work that needs to be accomplished in order to complete and further the stated goals of this thesis. It will focus on the practical circuitry implementation and other aspects needed to make it a viable tool.

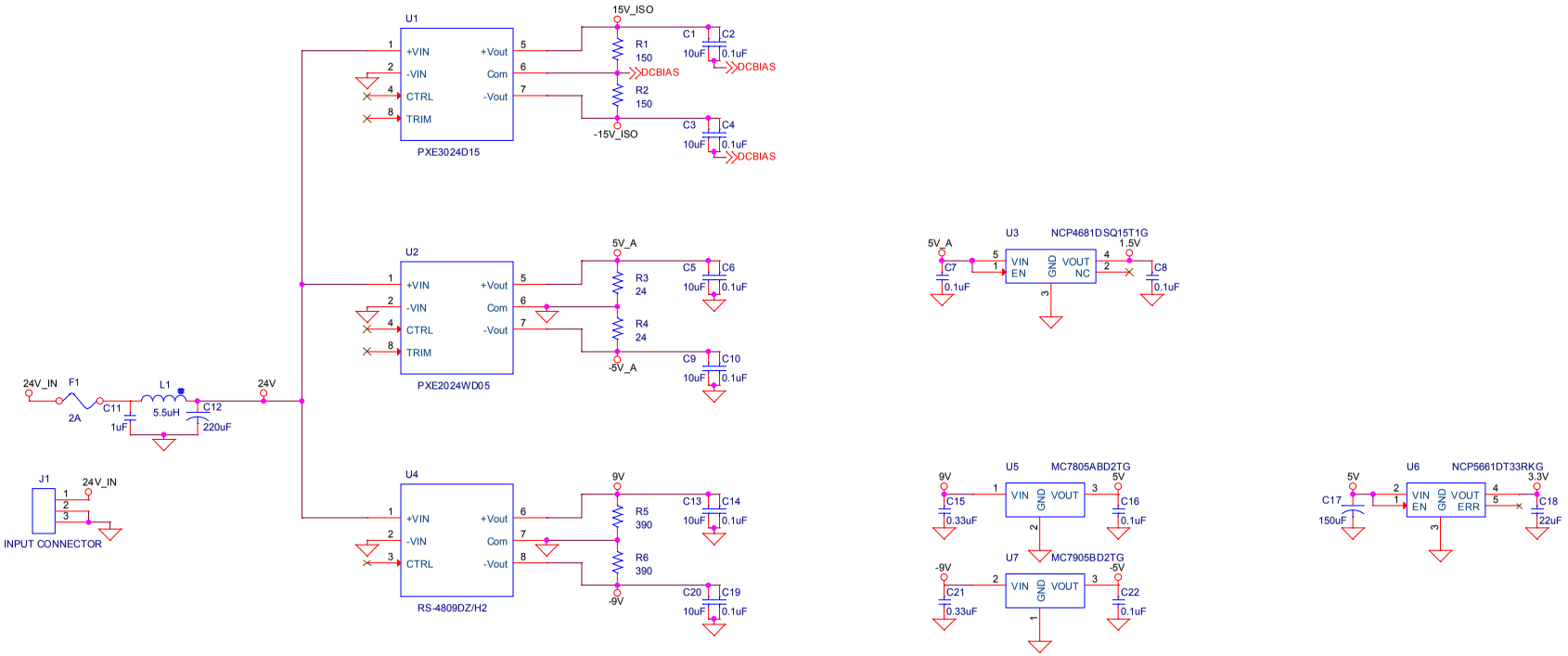
- Build the circuit and validate against the stated capabilities and accuracy.
- Increase the available frequency range for the measurement circuitry.
- Add additional fail-safe protection to the circuit.
- Update the regression technique for better accuracy (See Section: C).
- Evaluate the accuracy of the six term model described in Section: 5 with empirical data.

A Schematic

This section shows the schematic described in Section: 4.

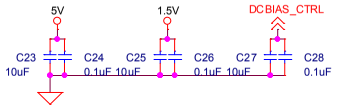
TABLE OF CONTENTS

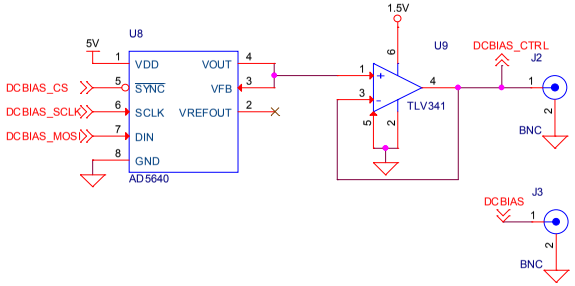
Title: TABLE OF CONTENTS		
Size: B	Document Number: DELIBERO_THESIS	Rev: 1
Date: Thursday, June 25, 2015	Sheet 1 of 11	



POWER SUPPLIES

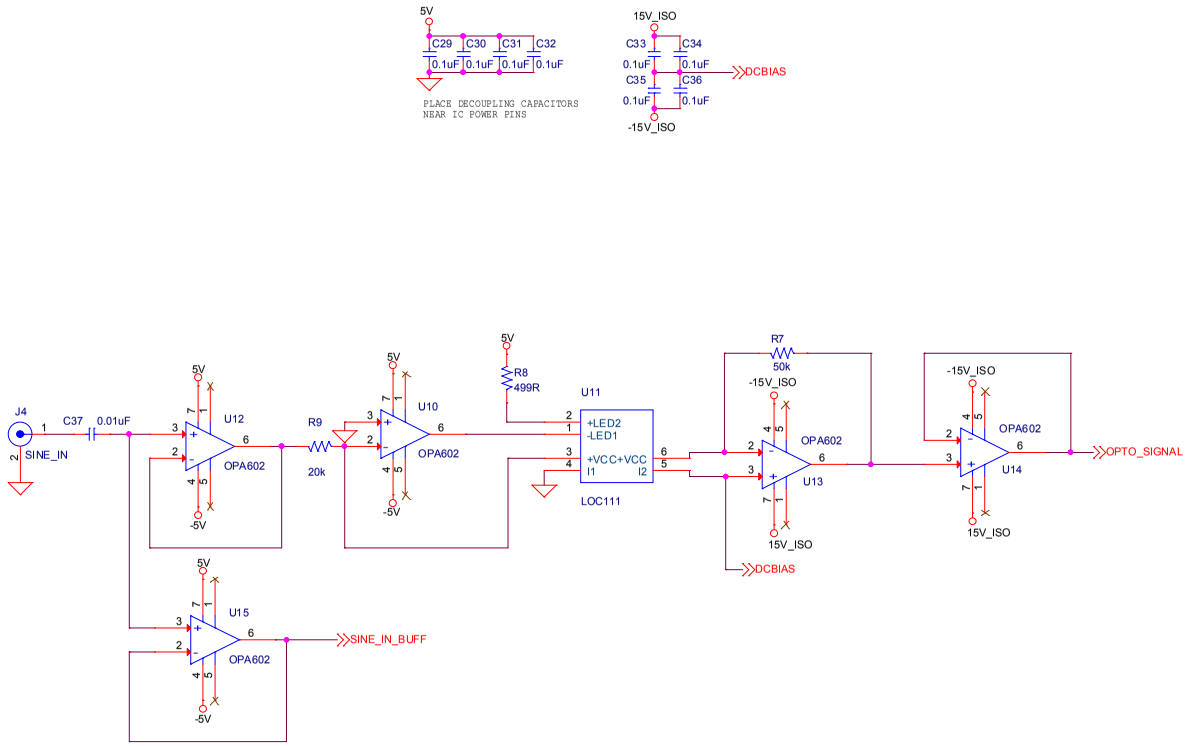
Title	POWER SUPPLIES	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Wednesday, June 24, 2015	Sheet 2 of 11	





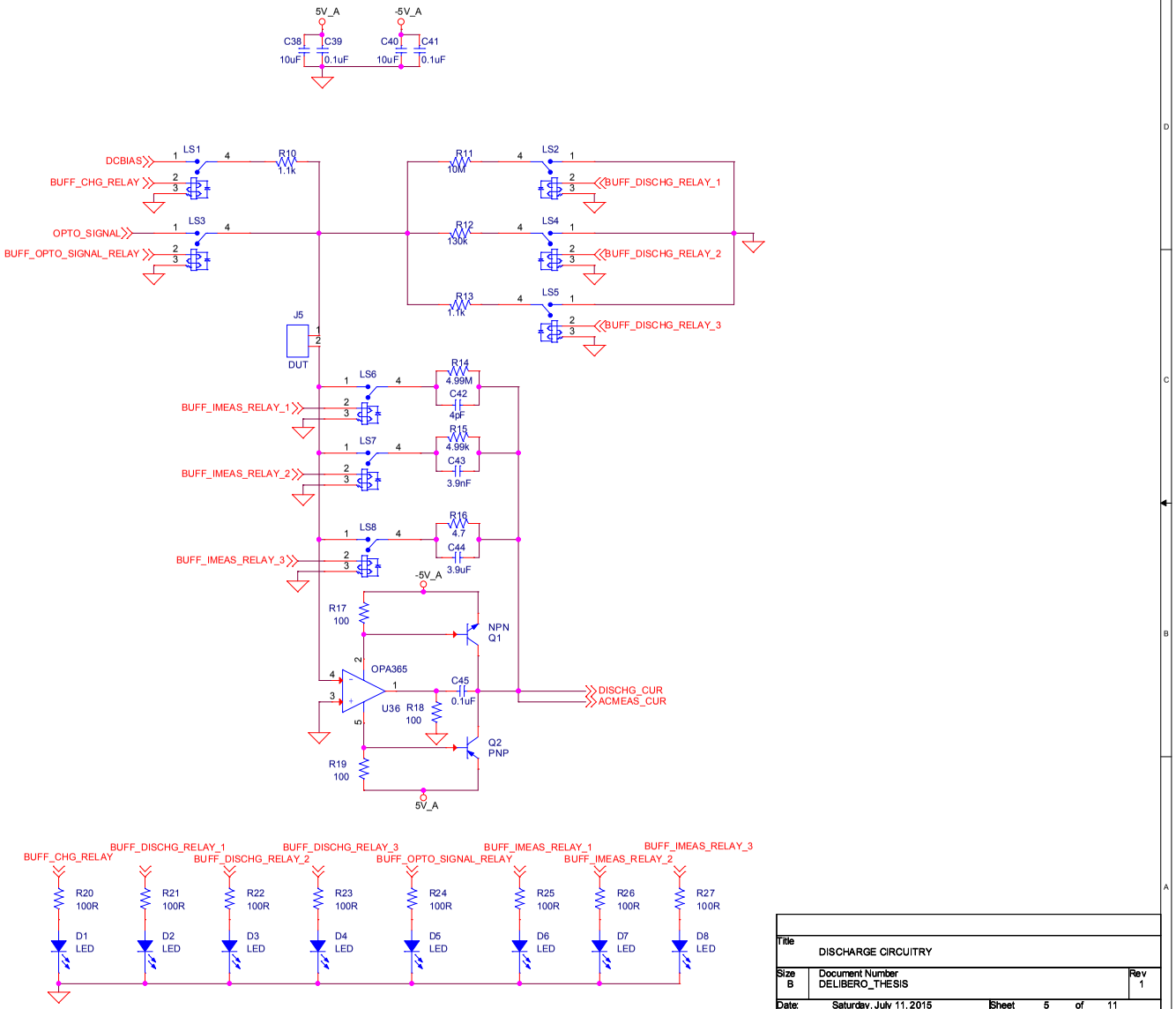
DC BIAS CONTROL

Title	DC BIAS CONTROL	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Thursday, June 25, 2015	Sheet 3 of 11	



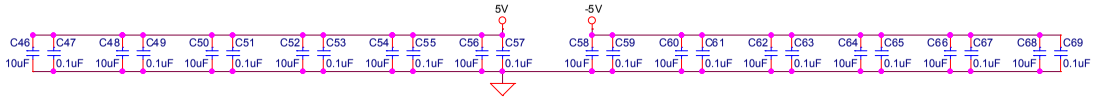
OPTOCOUPLER

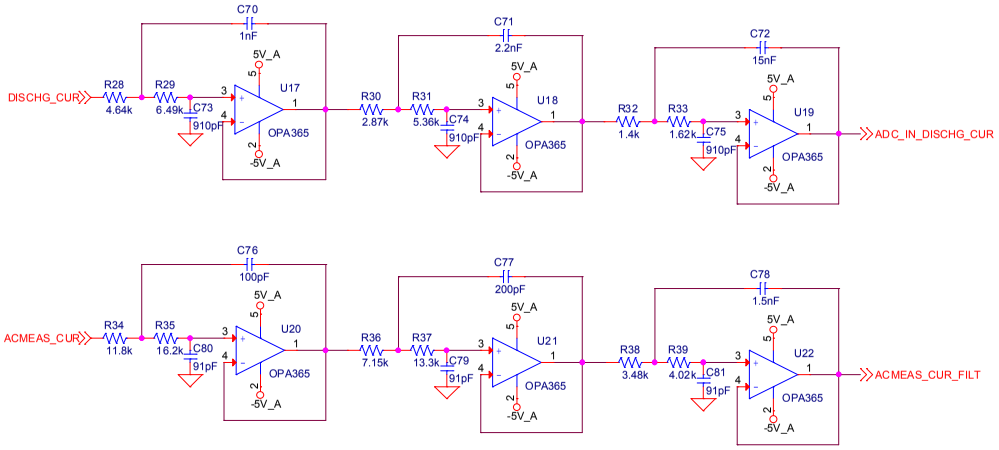
Title	OPTOCOUPLER	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Thursday, June 25, 2015	Sheet 4 of 11	



DISCHARGING CIRCUITRY

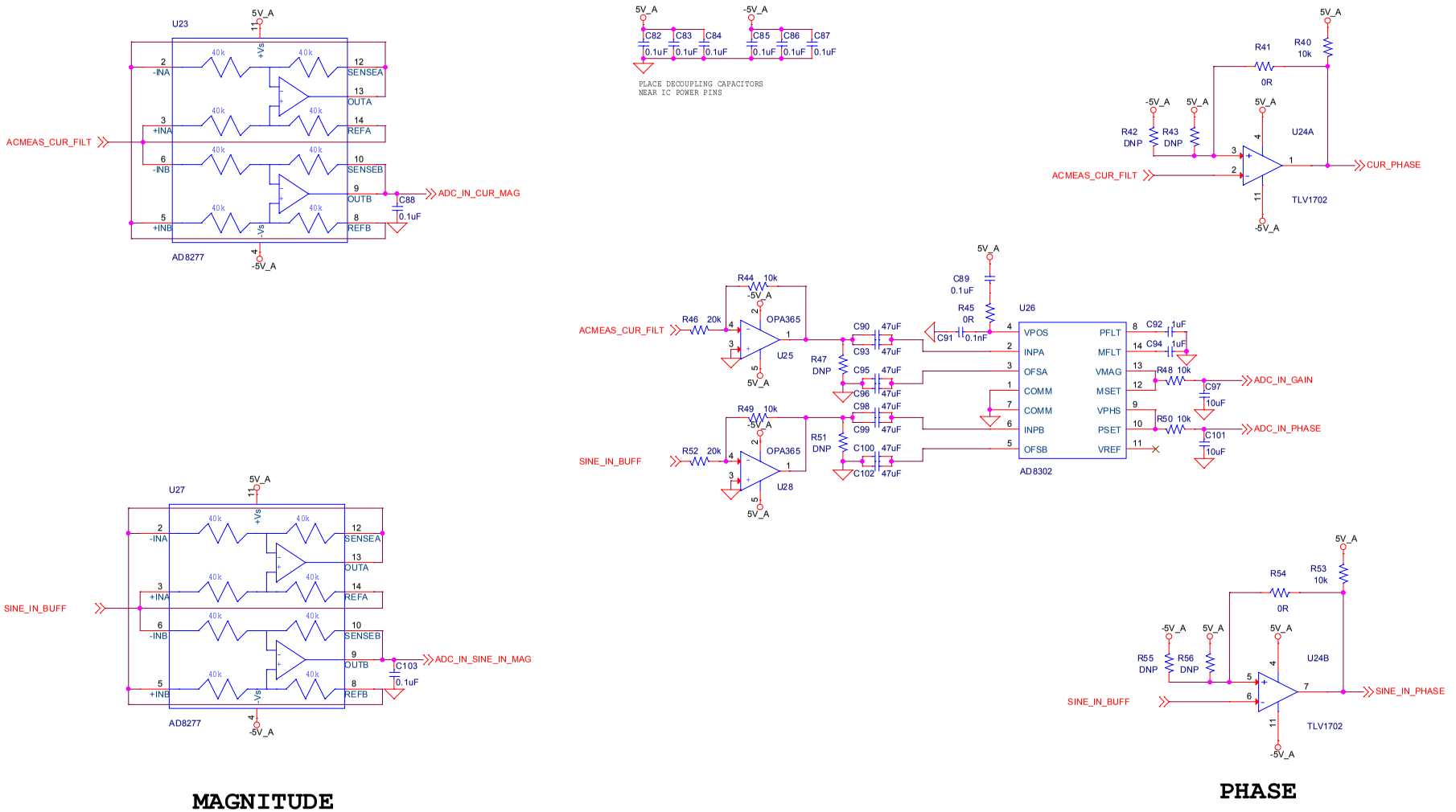
Title	DISCHARGE CIRCUITRY	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Saturday, July 11, 2015	Sheet 5 of 11	





FILTERING

Title: FILTERING		
Size: B	Document Number: DELIBERO_THESIS	Rev: 1
Date: Saturday, July 11, 2015	Sheet 6 of 11	

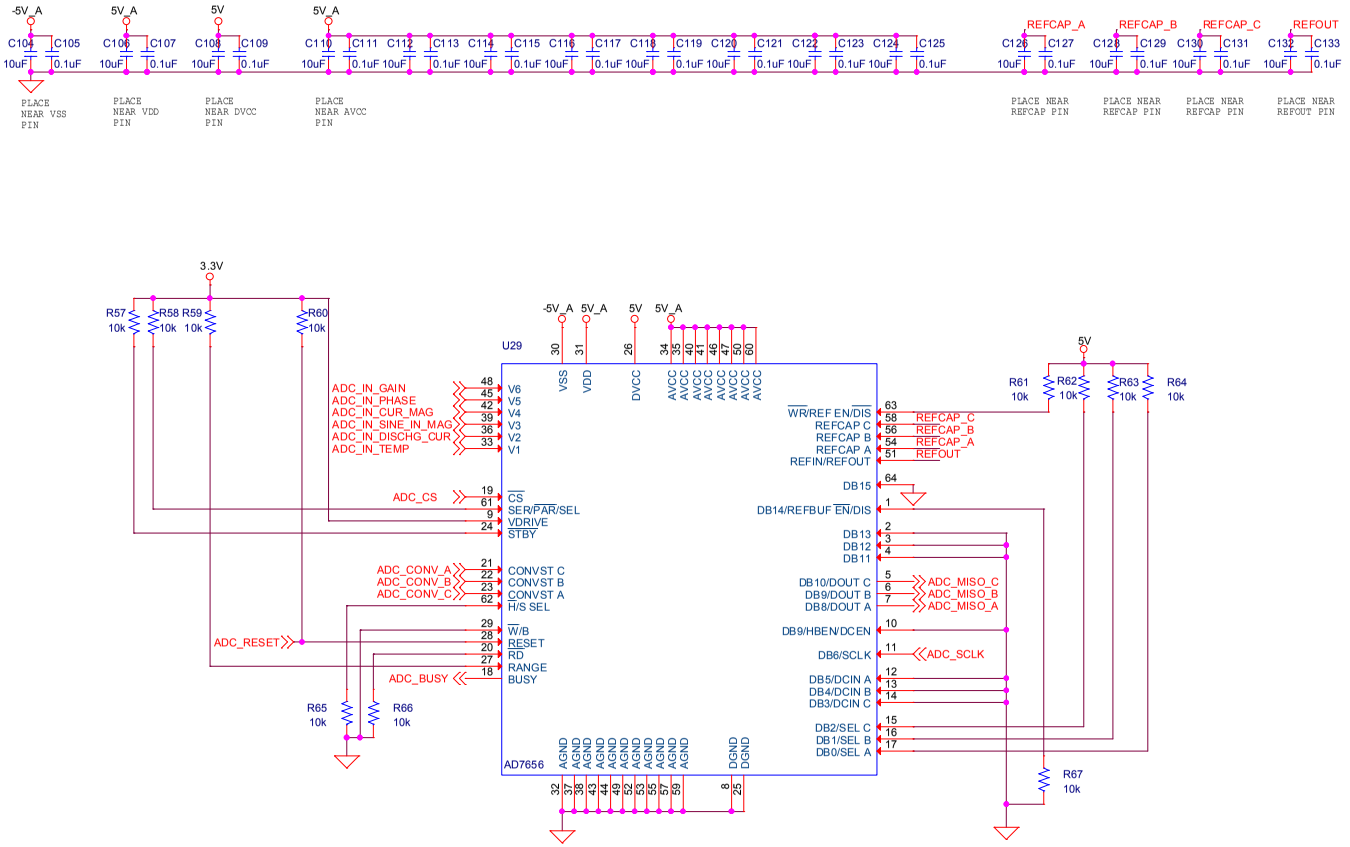


MAGNITUDE

PHASE

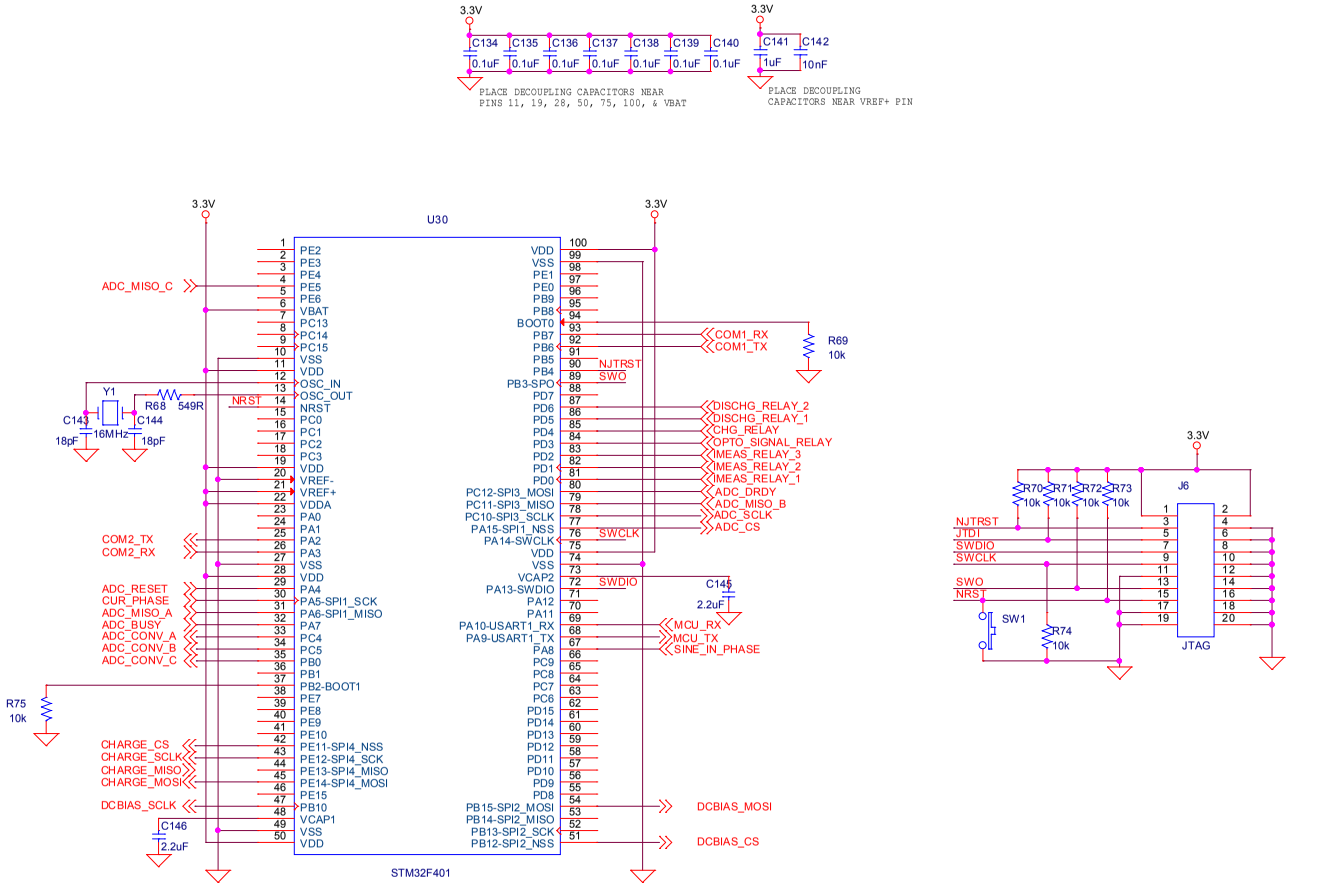
IMPEDANCE / PHASE

Title	IMPEDANCE / PHASE	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Thursday, June 25, 2015	Sheet 7 of 11	



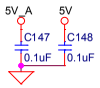
ADC

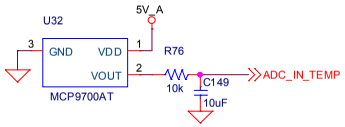
Title: ADC		
Size: B	Document Number: DELIBERO_THESIS	Rev: 1
Date: Thursday, June 25, 2015	Sheet 8 of 11	



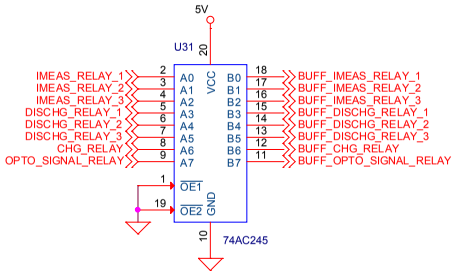
MCU I/O

Title	MCU I/O	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Thursday, June 25, 2015	Sheet 9 of 11	





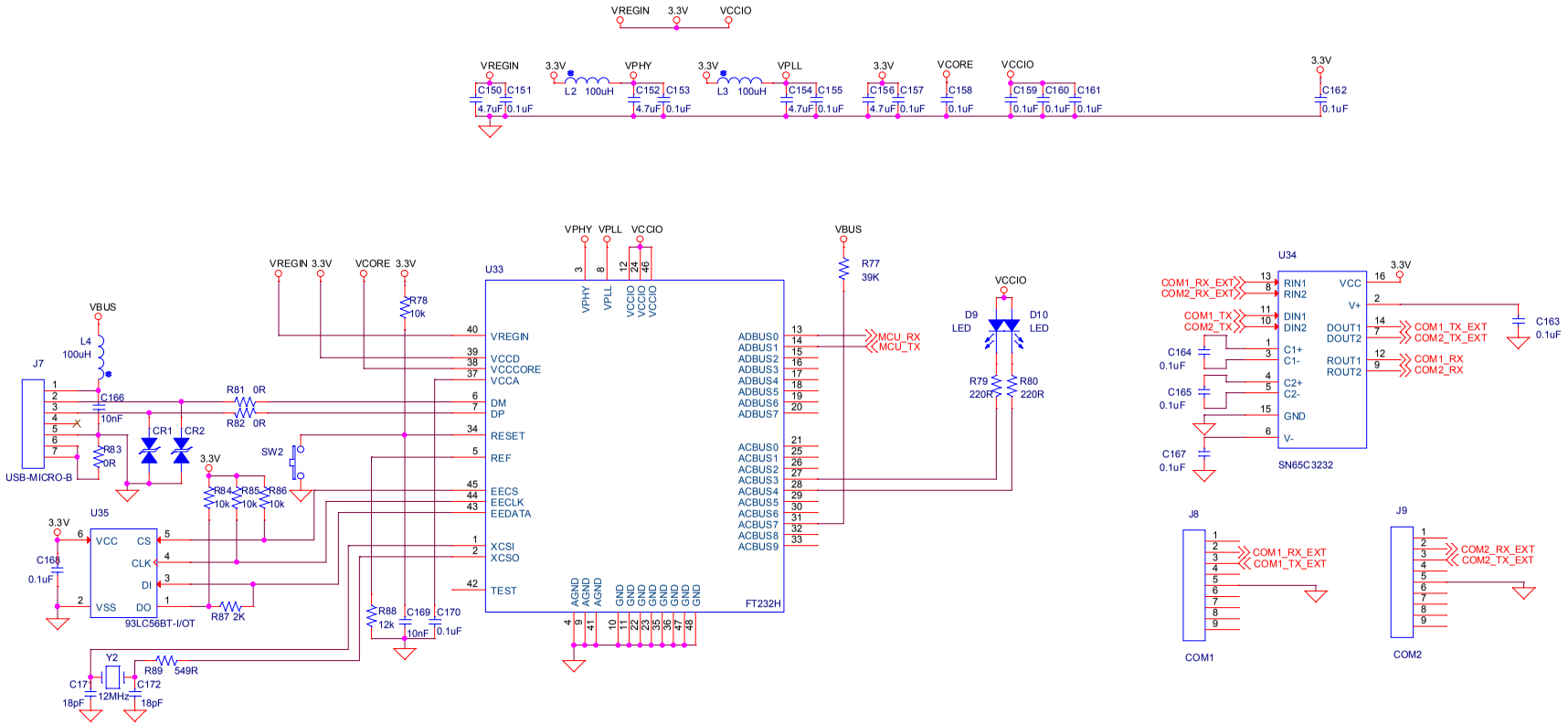
TEMPERATURE



BUFFER

MISCELLANEOUS

Title: MISCELLANEOUS		
Size: B	Document Number: DELIBERO_THESIS	Rev: 1
Date: Thursday, June 25, 2015	Sheet 10 of 11	



COMMUNICATIONS

Title	USB COMMUNICATIONS	
Size B	Document Number DELIBERO_THESIS	Rev 1
Date: Thursday, June 25, 2015	Sheet 11 of 11	

B Generating Modeling Images

This appendix will list all of the matlab code and supporting files needed to generate the images seen in Section: 5.

B.1 Example Data: Figure: 19

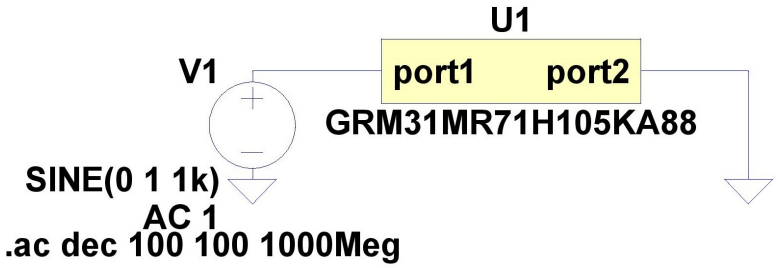


Figure 30: LTSpice Schematic for Capacitor Model

This section will describe how to obtain and generate the example data used for the regression fitting. This method uses LTSpice to generate impedance vs frequency data from one of Murata's capacitor models. First, go to Murata's online SimSurfing tool [46] and select the "Monolithic Ceramic Capacitors" button. Download a SPICE *.mod file (Appendix: B.1.1) for the capacitor of interest, by selecting it from the list and clicking the "netlist" button. Open the *.mod file in LTSpice, right click on the part name in the line starting with ".SUBCKT," and select "Create Symbol." Create an LTSpice schematic similar to Figure: 30 and plot $\frac{V(n001)}{-I(V1)}$. The negative sign is important because LTSpice defines current as coming out of a node. If the negative sign is omitted, the phase will be offset by 180^o, and the regression analysis will solve for negative capacitance! With the plot window selected, select "$File \rightarrow Export \rightarrow Cartesian \rightarrow OK$" with the impedance plot selected as the waveform. Open the resultant *.txt file, delete the first line, and change all tabs to commas ("$<:> \%s/ < ctrl + v >< TAB > /,/g$" , "$\%s/\hat{}I/,/g$" in vim). Make sure to save the resultant file in "/scripts/data."

B.1.1 Capacitor Subcircuit Model

```
1  *---------------------------------------------------------------------
2  * SPICE Model generated by Murata Manufacturing Co., Ltd.
3  * Copyright(C) Murata Manufacturing Co., Ltd.
4  * Description :3216/X7R/1uF/50V
5  * Murata P/N :GRM31MR71H105KA88
6  * Property : C = 1[uF]
7  *---------------------------------------------------------------------
8  * Applicable Conditions:
9  *   Frequency Range = 100Hz-6000000000Hz
10 *   Temperature = 25 degC
11 *   DC Bias Voltage = 0V
12 *   Small Signal Operation
13 *---------------------------------------------------------------------
14 .SUBCKT GRM31MR71H105KA88 port1 port2
15 C1 port1 11 1.00e-6
16 L2 11 12 2.82e-11
17 R3 12 13 6.65e-3
18 C4 13 14 1.53e-4
19 R4 13 14 22.8
20 C5 14 15 2.02e-4
21 R5 14 15 1.85
22 C6 15 16 1.51e-4
23 R6 15 16 3.53e-1
24 C7 16 17 1.60e-4
25 R7 16 17 6.68e-2
26 C8 17 18 9.85e-5
27 R8 17 18 1.80e-2
28 C9 18 19 1.20e-4
29 R9 18 19 2.94e-3
```

```
L10 19 20 3.24e-11
R10 19 20 1.02e-1
L11 20 21 1.94e-10
R11 20 21 3.05e-2
L12 21 22 4.09e-11
R12 21 22 2.85e-2
C13 22 23 5.18e-6
L13 22 23 3.93e-11
R13 22 23 1.00e-2
C14 23 24 9.49e-7
L14 23 24 8.93e-11
R14 23 24 1.68e-2
C15 24 25 1.33e-11
L15 24 25 9.25e-11
R15 24 25 6.24
C16 25 port2 1.03e-12
L16 25 port2 3.80e-10
R16 25 port2 171
R100 port1 11 5.00e+8
.ENDS GRM31MR71H105KA88
```

B.1.2 Plot ExCapData Script

The functions used to get and plot the data can be found in Appendix: B.4.

```
% plot_ExCapData.m
% This script outputs a plot of the example data taken from ...
    Murata's Sim Surfing online tool.

% Clear environment
clearvars;
close all;
```

```
format shorte;

filename = './data/GRM31MR71H105KA88.txt';
[w, cData, rData, iData] = getData(filename);
myPlotType   = plotType.cData;
plotFit(myPlotType, cData, cData, w); % ...
    figures/regression/levyIter.jpg
```

B.2 Basic LSE Image: Figure: 20

```
% run_basicLse.m
% This script runs the basic LSE for a line

% Clear environment
clearvars;
close all;
format shorte;

%% Generate data
n = 5;
err = rand(1,n)*1.3;
x = linspace(1,n,n);
b = rand(1)*ones(1,n);
y = x + err + b;

%% LSE
avg_y   = sum(y);
avg_x   = sum(x);
avg_xy  = sum(y .* x);
avg_xsq = sum(x .^ 2);
```

```
21
22 a_1 = (avg_xy - avg_x *avg_y) / (avg_xsq - avg_x^2)
23 a_0 = avg_y - a_1 * avg_x
24
25 y_lse = a_0*ones(1,length(x)) + a_1 .* x;
26
27 %% Plot
28 plotFit(plotType.DIFF_PLOT,y,y_lse,x);
```

B.3 Levy's Method: Figures: 21, 22, 23, & 24

```
1  % scripts/regression/run_levy_iter.m
2  % This script runs the example in Sanathanan's paper showing how
3  % to improve Levy's method by an iterative method.
4
5  % Clear environment
6  clearvars;
7  close all;
8  format shorte;
9
10 % Add funs to path
11 addpath('../utilityFuns');
12 addpath('../');
13
14 NumDeg = 7;
15 DenDeg = 7;
16 iterations = 100;
17 filename = '../data/GRM31MR71H105KA88.txt';
18
19 [w, cData, rData, iData] = getData(filename);
```

```
20 initDen = getInitGuess(w,modelTypes.FULL_MODEL);
21 [G, numCoeffs, denCoeffs, E, minIndex] = ...
22 regression_levy_iter(cData, w, iterations, NumDeg, DenDeg, initDen);
23
24 %% Error minimization
25 Emag = E(:,1);
26 Epha = E(:,2);
27 EmagNorm = Emag ./ max(Emag);
28 EphaNorm = Epha ./ max(Epha);
29 E2 = EmagNorm + EphaNorm;
30 n = find(E2==min(E2),1);
31
32 %% Relative Error
33 abs_magErr = abs(G(minIndex,:)) - abs(cData);
34 abs_phaErr = phase(G(minIndex,:)) - phase(cData);
35
36 rel_magErr = abs(abs_magErr ./ abs(cData))*100;
37 rel_phaErr = abs(abs_phaErr ./ phase(cData))*100;
38
39 %% Plot
40 plotcDiff = plotType.cVectorsDiff;
41 plotErrs  = plotType.twoErrors;
42 plotErr   = plotType.oneError;
43 plotRelErr= plotType.MULTPLOT;
44
45 plotFit(plotcDiff , cData, G(n,:), w); % ...
      figures/regression/levyIter.jpg
46 plotFit(plotErrs  , Emag , Epha); % ...
      figures/regression/levyIter_Err1.jpg
47 plotFit(plotErr   , E2         ); % ...
      figures/regression/levyIter_Err2.jpg
```

```
48 [h_fig, h_leg, h_title, ax, p] = plotFit(plotRelErr, rel_magErr, ...
       rel_phaErr,w);
49 h_title(1).String = 'Relative Magnitude Error';
50 h_title(2).String = 'Relative Phase Error';
51 xlabel(ax(1),'w (rad)');
52 xlabel(ax(2),'w (rad)');
53 ylabel(ax(1,1), 'Mag (%)');
54 ylabel(ax(1,2), '\phi (%)');
```

```
1  % scripts/getInitGuess.m
2  % This function returns the appropriate initial guess
3  % for Sanathanan's extension to Levy's method.
4  function [Den] = getInitGuess(w,modelType)
5      if modelType == modelTypes.FULL_MODEL
6          C  = 1;
7          RE = 1;
8          LE = 1;
9          RL = 1;
10         CD = 1;
11         RD = 1;
12
13         b1 = CD*RD+C*RL+CD*RL;
14         b2 = C*CD*RD*RL;
15         [¬,¬,Den] = calcDataFromCoeffs([1;1;1],[1,b1,b2],w);
16
17     else % Set denominator to all ones
18         Den = ones(length(w));
19     end % if modelType == modelTypes.FULL_MODEL
20 end % function [Den] = getInitGuess(w)
```

```
% scripts/regression_levy_iter.m
% This script runs the calculations for iterating Levy's method
function [G, numCoeffs, denCoeffs, E, minIndex] = ...
    regression_levy_iter(cData, w, iterations, numbNumCoeffs, ...
    numbDenCoeffs, initDen)
    % G(s) = \frac{a0 + a1*s + a2*s^2 + ... + an*s^n}
    %             {b0 + b1*s + b2*s^2 + ... + bm*s^m}

    % Solve
    numCoeffs = ones(iterations,numbNumCoeffs);
    denCoeffs = ones(iterations,numbDenCoeffs);
    W         = ones(length(w)); % Weighting function

    % Set initial guess of the denomiator. This also will cause it
    % to calc Levy's original method if the iteraiton == 1.
    Den = initDen;
    G = zeros(iterations,length(w));

    for iter = 1:iterations
        W = 1 ./ abs(Den).^2;
        [numCoeffs(iter,:), denCoeffs(iter,:)] = ...
            calcCoeffs(cData, w, W, numbNumCoeffs, numbDenCoeffs);
        [G(iter,1:length(w)),~,Den] = ...
            calcDataFromCoeffs(numCoeffs(iter,:),denCoeffs(iter,:),w);
        E(iter,:) = sumError(cData,G(iter,1:length(w)));
        iter
    end

    Emag = E(:,1);
    Epha = E(:,2);
    EmagNorm = Emag ./ max(Emag);
    EphaNorm = Epha ./ max(Epha);
```

```
29      E2 = EmagNorm + EphaNorm;
30      minIndex = find(E2==min(E2),1);
31  end % function [G,numCoeffs,denCoeffs] = regression_levy_iter...
32
33  % This calculates the polynomial coeffs from the polynomial series
34  function [numCoeffs, denCoeffs] = calcCoeffs(cData, w, W, ...
        numbNumCoeffs, numbDenCoeffs)
35
36      % Prepare variables
37      Mrows = numbNumCoeffs + numbDenCoeffs - 1;
38      Mcols = Mrows;
39
40      M = ones(Mrows,Mcols);
41      N = ones(Mrows,1);
42      C = ones(Mrows,1);
43
44      sign  = 0;
45      power = 0;
46
47      %% Calculate
48      % Populate M
49      for row = 1:Mrows
50          for col = 1:Mcols
51              % Find out which section of M you are in:
52              if ((row ≤ numbNumCoeffs) && (col ≤ numbNumCoeffs))
53                  %Upper Left Quadrant -- lambdas
54                  xrow = row;
55                  xcol = col;
56
57                  if (xor(mod(xrow,2) == 0, mod(xcol,2) == 0) == 1)
58                  % If both are odd or both are even
59                      M(xrow,xcol) = 0;
```

```
60
61                  else
62                      power = xrow + xcol - 2;
63
64                      if ((mod(xcol,3) == 0) && (mod(xrow,2) == 1))
65                          sign = -1;
66                      elseif ((mod(xcol,4) == 0) && (mod(xrow,2) == 0))
67                          sign = -1;
68                      else
69                          sign =  1;
70                      end
71
72                      M(xrow,xcol) = sign * lambda(W,w,power);
73                  end % else
74
75
76              elseif (row ≤ numbNumCoeffs)
77                  % Upper Right quadrant -- S and T
78                  xrow = row;
79                  xcol = col - numbNumCoeffs;
80                  power = xrow + xcol - 1;
81
82                  if      (mod(xcol,4) == 1) && (mod(xrow,2) == 1)
83                      sign =  1;
84                  elseif (mod(xcol,4) == 1) && (mod(xrow,2) == 0)
85                      sign = -1;
86                  elseif (mod(xcol,4) == 2)
87                      sign =  1;
88                  elseif (mod(xcol,4) == 3) && (mod(xrow,2) == 1)
89                      sign = -1;
90                  elseif (mod(xcol,4) == 3) && (mod(xrow,2) == 0)
91                      sign = 1;
```

```
92                  elseif (mod(xcol,4) == 4)
93                      sign = -1;
94                  end
95
96                  if  (mod(power,2) == 1)
97                      M(row,col) = sign * T(W,w,imag(cData),power);
98                  else
99                      M(row,col) = sign * S(W,w,real(cData),power);
100                 end
101
102
103             elseif ((row > numbNumCoeffs) && (col ≤ numbNumCoeffs))
104                 % Lower Left -- S and T
105                 xrow = row - numbNumCoeffs;
106                 xcol = col;
107                 power = xrow + col - 1;
108
109                 if     (mod(xcol,4) == 1)
110                     sign =  1;
111                 elseif (mod(xcol,4) == 2) && (mod(xrow,2) == 1)
112                     sign = -1;
113                 elseif (mod(xcol,4) == 2) && (mod(xrow,2) == 0)
114                     sign =  1;
115                 elseif (mod(xcol,4) == 3)
116                     sign = -1;
117                 elseif (mod(xrow,2) == 1)
118                     sign =  1;
119                 elseif (mod(xrow,2) == 0)
120                     sign = -1;
121                 end
122
123                 if (mod(power,2) == 0)
```

```
124                     M(row,col) = sign * S(W,w,real(cData),power);
125                 else
126                     M(row,col) = sign * T(W,w,imag(cData),power);
127                 end
128
129             else
130                 % Lower Right -- U
131                 xrow = row - numbNumCoeffs;
132                 xcol = col - numbNumCoeffs;
133
134                 % If both are odd or both are even
135                 if (xor(mod(xrow,2) == 0, mod(xcol,2) == 0) == 1)
136                     M(row,col) = 0;
137                 else
138                     power = xrow + xcol;
139
140                     if      (mod(xcol,4) == 1)
141                         sign = 1;
142                     elseif (mod(xcol,4) == 2)
143                         sign = 1;
144                     else
145                         sign = -1;
146                     end
147
148                     M(row,col) = sign * ...
                           U(W,w,real(cData),imag(cData),power);
149                 end % else -- check for zeros
150             end % else -- searching through quadrants
151         end % for row = 1:Mrows
152     end % for col = 1:Mcols
153
154     % Populate C
```

```
155     for (row = 1:length(C))
156         if (row ≤ numbNumCoeffs)
157             xrow = row;
158             power = xrow - 1;
159             if (mod(power,2) == 0)
160                 C(xrow) = S(W,w,real(cData),power);
161             else
162                 C(xrow) = T(W,w,imag(cData),power);
163             end
164         else
165             xrow = row - numbNumCoeffs;
166             power = xrow;
167             if (mod(xrow,2) == 1)
168                 C(row) = 0;
169             else
170                 C(row) = U(W,w,real(cData),imag(cData),power);
171             end
172         end
173     end
174
175     % Solve For N
176     N = inv(M) * C;
177
178     numCoeffs = N(1:numbNumCoeffs);
179     denCoeffs = [1;N(numbNumCoeffs+1:length(N))]; % Add 1 as coeff_0
180
181     %Sub Functions
182     function [Lh] = lambda(W,wk,h)
183         Lh = 0;
184         for k = 1:1:length(wk)
185             Lh = Lh + wk(k)^h*W(k);
186         end
```

```
187     end
188
189     function [S] = S(W,wk,Rk,h)
190         S = 0;
191         for k = 1:1:length(wk)
192             S = S + wk(k)^h * Rk(k)*W(k);
193         end
194     end
195
196     function [T] = T(W,wk,Ik,h)
197         T = 0;
198         for k = 1:1:length(wk)
199             T = T + wk(k)^h * Ik(k)*W(k);
200         end
201     end
202
203     function [U] = U(W,wk,Rk,Ik,h)
204         U = 0;
205         for k = 1:1:length(wk)
206             U = U + wk(k)^h * (Rk(k)^2 + Ik(k)^2)*W(k);
207         end
208     end
209 end % function [b,a] = calcCoeffs(w_norm, T, cData)
```

B.4 Utility Functions

This appendix holds common "utility" functions used by many of the MATLAB scripts. You are required to manually save each plot after calling the "plotfit.m."

```
1 % Getdata.m
2 % This function parses data from an LTSpice file into a
```

```
% format that can be used for regression analysis.
function [w, cData, rData, iData] = getData(filename)
    data  = csvread(filename);
    w     = (data(:,1) * 2 * pi)';
    rData = data(:,2)';
    iData = data(:,3)';
    cData = rData + 1i * iData;
end
```

```
% plotFit.m
% Outputs different plots for the regression analysis.
function [h_fig, h_leg, h_title, ax, p] = plotFit(pType, Data1, ...
    Data2, x)
%% Common Prep
titleSize     = 25;
legendSize    = 20;
axisTitleSize = 20;
axisSize      = 15;

h_fig = figure;

%% Plot specific request
if pType == plotType.DIFF_PLOT;
    ax = subplot(1,1,1);
    p(1) = plot(x,Data1,'o'); hold on;
    p(2) = plot(x,Data2);
    h_title = title('Basic LSE','FontSize',titleSize);
    h_leg = legend('Orig','Fit');
    set(h_leg,'FontSize',legendSize);
    xlabel(ax, 'x', 'FontSize', axisTitleSize);
    ylabel(ax, 'y', 'FontSize', axisTitleSize);
```

```
22
23 elseif pType == plotType.cData
24     [ax,p(1),p(2)] = plotyy(x,mag2db(abs(Data1)),x, ...
25     rad2deg(phase(Data1)),'semilogx','semilogx');
26
27     h_title = title('GRM31MR71H105KA88','FontSize',titleSize);
28     h_leg = legend('Magnitude','Phase','Location','north');
29     set(h_leg,'FontSize',legendSize);
30     xlabel(ax(1), '\omega (rad)'  , 'FontSize', axisTitleSize);
31     ylabel(ax(1), 'Mag (dB)'      , 'FontSize', axisTitleSize);
32     ylabel(ax(2), '\phi (deg)'    , 'FontSize', axisTitleSize);
33     min_y1 = min(get(ax(1),'ytick'));
34     max_y1 = max(get(ax(1),'ytick'));
35     min_y2 = min(get(ax(2),'ytick'));
36     max_y2 = max(get(ax(2),'ytick'));
37     set(ax(1),'ytick',min_y1:20:max_y1);
38     set(ax(2),'ytick',min_y2:20:max_y2);
39
40 elseif pType == plotType.MULTCDATA
41     colors = [0.000,0.447,0.741 ;
42     0.850,0.325,0.098 ;
43     0.929,0.694,0.125 ;
44     0.494,0.184,0.556 ;
45     0.466,0.674,0.188 ;
46     0.301,0.745,0.933];
47
48     [ax(1,:),p(1),p(2)] = plotyy(x,mag2db(abs(Data1(1,:))),x,...
49     rad2deg(phase(Data1(1,:))),'semilogx','semilogx'); hold on;
50     [ax(2,:),p(3),p(4)] = plotyy(x,mag2db(abs(Data1(2,:))),x,...
51     rad2deg(phase(Data1(2,:))),'semilogx','semilogx'); hold on;
52     [ax(3,:),p(5),p(6)] = plotyy(x,mag2db(abs(Data1(3,:))),x,...
53     rad2deg(phase(Data1(3,:))),'semilogx','semilogx');
```

```
54
55      h_title = title('RC Filter -- Increasing ...
            C','FontSize',titleSize);
56      h_leg = legend(p,'Mag1','Pha1','Mag2','Pha2',...
57      'Mag3','Pha3','Location','northeast');
58      set(h_leg,'FontSize',legendSize);
59      xlabel(ax(1,1), '\omega (rad)'  , 'FontSize', axisTitleSize);
60      ylabel(ax(1,1), 'Mag (dB)'      , 'FontSize', axisTitleSize);
61      ylabel(ax(1,2), '\phi (deg)'    , 'FontSize', axisTitleSize);
62
63      for ind = 1:6
64          p(ind).Color = colors(ind,:);
65          ax(ind).YColor = [0,0,0]; % Black
66      end
67
68      p(2).Marker = 'o';
69      p(4).Marker = 'o';
70      p(6).Marker = 'o';
71
72  elseif pType == plotType.MULTDATA
73      ax = subplot(1,1,1);
74
75      for ind = 1:size(Data1,1)
76          p(ind) = plot(x,Data1(ind,:)); hold on;
77      end
78
79      h_title = title('','FontSize',titleSize);
80      h_leg = legend();
81      set(h_leg,'FontSize',legendSize);
82      xlabel(ax(1,1), 'x'  , 'FontSize', axisTitleSize);
83      ylabel(ax(1,1), 'y'  , 'FontSize', axisTitleSize);
84
```

```
85 elseif pType == plotType.cVectorsDiff
86     rows = 2;
87     cols = 2;
88
89     ax(1) = subplot(rows,cols,1);
90     p(1)  = semilogx(x,mag2db(abs(Data1))); hold on;
91     p(2)  = semilogx(x,mag2db(abs(Data2)));
92     h_title(1) = title('Magnitude','FontSize',titleSize);
93     h_leg = legend('Orig','FitData','Location','north');
94     set(h_leg,'FontSize',legendSize);
95     xlabel('\omega (rad)'  ,'FontSize',axisTitleSize);
96     ylabel('Mag (dB)','FontSize',axisTitleSize);
97     min_y1 = min(get(ax(1),'ytick'));
98     max_y1 = max(get(ax(1),'ytick'));
99     set(ax(1),'ytick',min_y1:20:max_y1);
100
101    ax(2) = subplot(rows,cols,2);
102    p(3)  = semilogx(x,rad2deg(phase(Data1))); hold on;
103    p(4)  = semilogx(x,rad2deg(phase(Data2)));
104    h_title(2) = title('Phase','FontSize',titleSize);
105    h_leg = legend('Orig','FitData','Location','north');
106    set(h_leg,'FontSize',legendSize);
107    xlabel('\omega (rad)','FontSize',axisTitleSize);
108    ylabel('\phi (deg)','FontSize',axisTitleSize);
109    min_y2 = min(get(ax(2),'ytick'));
110    max_y2 = max(get(ax(2),'ytick'));
111    set(ax(2),'ytick',min_y2:20:max_y2);
112
113    ax(3) = subplot(rows,cols,3);
114    p(5)  = semilogx(x,abs(Data2)-abs(Data1));
115    h_title(3) = title('Magnitude Error','FontSize',titleSize);
116    xlabel('\omega (rad)','FontSize',axisTitleSize);
```

```
117     ylabel('\Delta Mag (\Omega)','FontSize',axisTitleSize);
118
119     ax(4) = subplot(rows,cols,4);
120     p(6)  = semilogx(x,rad2deg(phase(Data2))-rad2deg(phase(Data1)));
121     h_title(4) = title('Phase Error','FontSize',titleSize);
122     xlabel('\omega (rad)','FontSize',axisTitleSize);
123     ylabel('\Delta \phi (deg)','FontSize',axisTitleSize);
124
125 elseif pType == plotType.oneError
126     rows = 1;
127     cols = 1;
128     ax = subplot(rows,cols,1);
129     p  = plot(Data1);
130     h_title = title('Norm Magnitude Err^2 + Norm Phase ...
            Err^2','FontSize',titleSize);
131     h_leg = legend();
132     xlabel('n','FontSize',axisTitleSize);
133     ylabel('Error','FontSize',axisTitleSize);
134
135 elseif pType == plotType.twoErrors
136     rows = 1;
137     cols = 2;
138
139     ax(1) = subplot(rows,cols,1);
140     p (1) = semilogy(Data1);
141     h_title(1) = title('Magnitude Err^2','FontSize',titleSize);
142     xlabel('n','FontSize',axisTitleSize);
143     ylabel('Error^2 ((\Delta \Omega)^2)','FontSize',axisTitleSize);
144     h_leg = legend();
145
146     ax(2) = subplot(rows,cols,2);
147     p (2) = semilogy(Data2);
```

```
148     h_title(2) = title('Phase Error^2','FontSize',titleSize);
149     xlabel('n','FontSize',axisTitleSize);
150     ylabel('Error^2 ((\Delta \Phi)^2)','FontSize',axisTitleSize);
151
152 elseif pType == plotType.MULTPLOT
153     rows = 1;
154     cols = 2;
155
156     ax(1) = subplot(rows,cols,1);
157     p (1) = loglog(x,Data1);
158     h_title(1) = title('Magnitude Err^2','FontSize',titleSize);
159     xlabel('n','FontSize',axisTitleSize);
160     ylabel('Error^2 ((\Delta \Omega)^2)','FontSize',axisTitleSize);
161     h_leg = legend();
162
163     ax(2) = subplot(rows,cols,2);
164     p (2) = loglog(x,Data2);
165     h_title(2) = title('Phase Error^2','FontSize',titleSize);
166     xlabel('n','FontSize',axisTitleSize);
167     ylabel('Error^2 ((\Delta \Phi)^2)','FontSize',axisTitleSize);
168
169 elseif pType == plotType.OPAREA
170     [ax,p(1),p(2)] = ...
          plotyy(x,Data1,x(3:5),Data2,'semilogx','loglog');
171
172     h_title = title('Safe Operating Area','FontSize',titleSize);
173     h_leg = legend('Voltage','Minimum ...
          Capacitance','Location','northeast');
174     set(h_leg,'FontSize',legendSize);
175     xlabel(ax(1), 'R (\Omega)', 'FontSize', axisTitleSize);
176     ylabel(ax(1), 'Vmax (V)'  , 'FontSize', axisTitleSize);
177     ylabel(ax(2), 'Cmin (uF)' , 'FontSize', axisTitleSize);
```

```
178
179      set(ax(1),'ytick',0:100:500);
180 %     set(ax(2),'YLim',[10^-9 10^-3]);
181      ax(2).YTickMode='Auto';
182 end % if pType == plotType.cVectorsDiff
183
184 %% Common plotting options
185 for ind = 1:length(ax)
186          grid(ax(ind),'on');
187          set(ax(ind),'FontSize',axisSize);
188      end % for ind = 1:length(ax)
189
190      for ind = 1:length(p)
191          p(ind).LineWidth = 2;
192      end % for ind = 1:length(p)
193 end % function plotFit()
```

```
1 % plotType.m
2 % Holds the enum for the various plot types.
3 classdef plotType
4      enumeration
5          DIFF_PLOT, MULTCDATA, MULTDATA, MULTPLOT, OPAREA, cData, ...
               cVectorsDiff, oneError, twoErrors
6      end
7 end
```

C Additional Regression Techniques

This section will cover the theory behind a computational method that should improve the results obtained in Section: 5. As previously stated, the main problem with

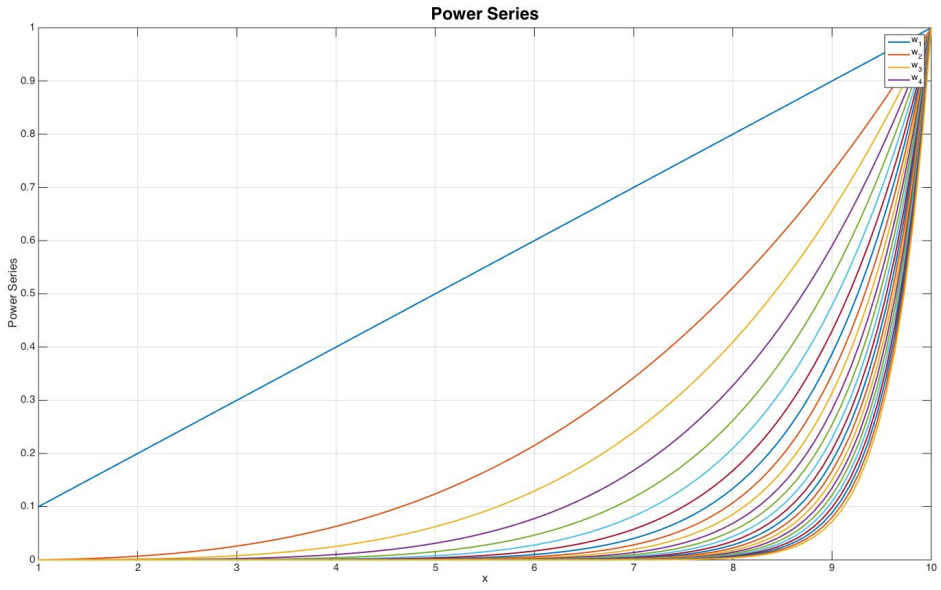


Figure 31: Power Series [4][Fig: 1]

applying Levy's approach to capacitor modeling is that it is ill-suited for wide bandwidth applications. As stated by Beyene, "this is because the ordinary power series $\omega^0, \omega^1, \omega^2, \omega^3, ...$ have a large dynamic range, and they become almost parallel at higher orders. As show in Figure: 31, for higher orders, the shapes of the power series become very similar over most of the normalized frequency range. [4]"

$$T_{n+1} = 2xT_n - T_{n-1} \tag{62}$$

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x
\end{aligned}
\tag{63}
$$

The proposed methodology uses Chebyshev polynomials of the first kind to circumvent this problem. As seen in Equation: (62), they are recursively defined with $T_0 = 1$ and $T_1 = x$. This results in polynomials that are orthogonal over a normalized frequency range, as seen in Figure: 32. Gao[14] shows that the frequency terms in an LSE can be replaced with Chebyshev polynomials. The coefficients found with this method are not the same as the coefficients in the frequency space. One needs to use Clenshaw's Recurrence Formula [14][Eq. 11] in order to get the desired coefficients. This method should produce a result that is more accurate due to its avoidance of the ill-conditioned matrix while solving the system of equations.

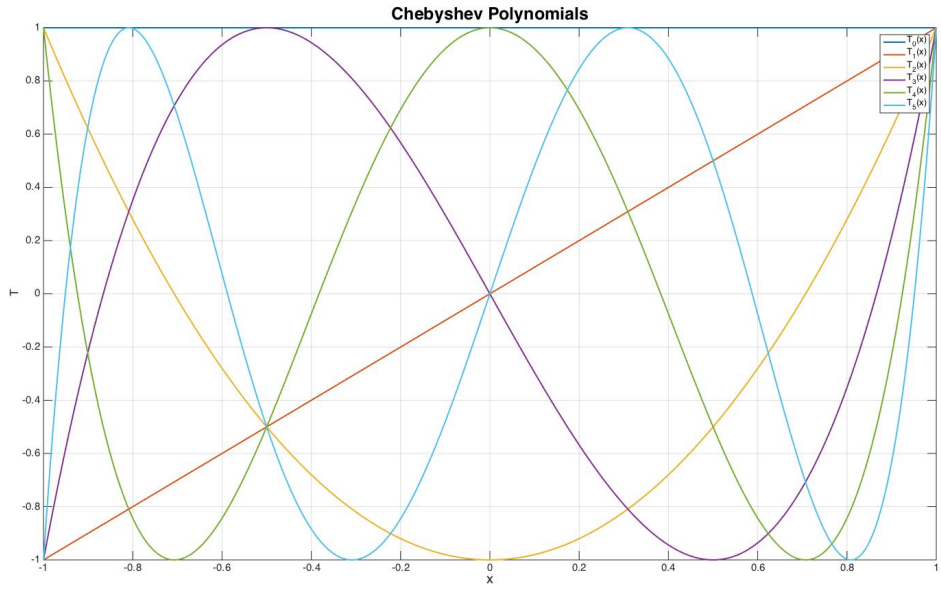


Figure 32: Chebyshev Polynomials [60]

D Discharge Equations

This section lists the discharge equations from Miller's electrochemical capacitor model [34] seen in Figure: 16.

```
(* 1 Branch *)
ClearAll["Global`*"]
v1 = DSolve[{v'[t] == -1 / (c1 * (r1 + rd)) * v[t], v[0] == 1}, v[t], t];
id = rd / (rd + r1) * v1;
rd = 1100;
r1 = .001;
c1 = 100;
id;
```

```
(* 2 Branches *)
ClearAll["Global`*"]
sol = DSolve[{c1 * v1'[t] + c2 * v2'[t] == v1[t] / (rd + r1), v2'[t] * c2 ==
      (v2[t] - v1[t]) / r2 + v1[t] (r1 + rd), v1[0] == 1, v2[0] == 1}, {v1[t], v2[t]}, t];
rd = 1100;
vc1 = sol[[All, 1]];
ic1 = rd (rd + r1) * vc1;

r1 = .001;
r2 = .001;
c1 = 100;
c2 = 100;

vc1;
ic1;
```

```
(* 3 Branches *)
ClearAll["Global`*"]
sol =
  DSolve[{c1 * v1'[t] + c2 * v2'[t] + c3 * v3'[t] == v1[t] / (rd + r1), (v3[t] - v2[t]) / r3 ==
      c3 * v3'[t], c2 * v2'[t] == (v3[t] - v2[t]) / r3 - (v2[t] - v1[t]) / r2,
    v3[0] == 1, v2[0] == 1, v1[0] == 1}, {v1[t], v2[t], v3[t]}, t];
rd = 1100;
vc1 = sol[[All, 1]];
ic1 = rd (rd + r1) * vc1;

r1 = .001;
r2 = .001;
r3 = .001;
c1 = 100;
c2 = 100;
c3 = 100;

vc1;
ic1;
```

References

[1] 16065a 200 vdc external voltage bias fixture. http://www.keysight.com/en/pd-1000000484%3Aepsg%3Apro-pn-16065A/200-vdc-external-voltage-bias-fixture?cc=US&lc=eng, 2015.

[2] 4294a precision impedance analyzer, 40 hz to 110 mhz. http://www.keysight.com/en/pd-1000000858%3Aepsg%3Apro-pn-4294A/precision-impedance-analyzer-40-hz-to-110-mhz?cc=US&lc=eng, 2015.

[3] Avx corporation history. http://www.fundinguniverse.com/company-histories/avx-corporation-history/.

[4] W.T. Beyene. Improving time-domain measurements with a network analyzer using a robust rational interpolation technique. *Microwave Theory and Techniques, IEEE Transactions on*, 49(3):500–508, Mar 2001.

[5] Relva C. Buchanan, editor. *Ceramic Materials for Electronics*. Marcel Dekker, 3 edition, June 2004.

[6] What are mica capacitors? http://www.capacitorguide.com/mica-capacitor/, 2014.

[7] PETER CORWIN. Synthesis and characterization of titanium zirconium based alloys for capacitor use. Master's thesis, Case Western Reserve University, 2013.

[8] Research Department. Expansion of functions in terms of chebyshev polynomials. Report 10, BBC, 1969.

[9] G W A Dummer. *Electronics Inventions and Discoveries*. Institute of Physics Publishing, 1997.

[10] Steven Ehret. *Instrumentation For Anodization and In-Situ Testing of Titanium Alloys for Capacitor Anodes*. Masters thesis, Case Western Reserve University, January 2012.

[11] JD Fast. The transition point diagram of the zirconium-titanium system. *Recueil des Travaux Chimiques des Pays-Bas*, 58(11):973–983, 1939.

[12] Richard Fore. Understanding temperature coefficients of ceramics, January 2005.

[13] S. Fujishima. The history of ceramic filters. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 47(1):1–7, Jan 2000.

[14] Rong Gao, Y.S. Mekonnen, W.T. Beyene, and J.E. Schutt-Aine. Black-box modeling of passive systems by rational function approximation. *Advanced Packaging, IEEE Transactions on*, 28(2):209–215, May 2005.

[15] James M. Gleason. Steatite for high frequency insuation. In *Journal of the British Institution of Radio Engineers*. Institute of Radio Engineers, 1945.

[16] J.M. Herbert. *Ceramic Dielectrics in Capacitors*. Gordon and Breach Scientific Publishers, 1985.

[17] J. Ho, T.R. Jow, and S. Boggs. Historical introduction to capacitor technology. *Electrical Insulation Magazine, IEEE*, 26(1):20–25, January 2010.

[18] Jeffrey Hokanson. Exponential fitting code. `https://github.com/jeffrey-hokanson/exponential_fitting_code`, November 2013.

[19] Brian Holman. *The Electrical Characterization of Tantalum Capacitors as MIS Devices*. ProQuest LLC, 2008.

[20] Capacitors. `http://www.ieeeghn.org/wiki/index.php/Capacitors`.

[21] IXYS. *LOC Series Linear Optocouplers Application Note: AN-107*, September 2013.

[22] B. Jaffe, W.R Cook Jr., and H. Jaffe. *Piezoelectric Ceramics*. Adacemic Press Ince (London) Ltd, 1971.

[23] Jill Jonnes. *Empires of Light*. Random House, 2003.

[24] Thomas R. Cuthbert Jr. *Circuit Design Using Personal Computers*. Wiley and Sons, 1983.

[25] James Karki. Analysis of the sallen-key architecture. Online, September 2002.

[26] Jun-Wan Ki. *Titanium sponge on titanium substrate for titanium electrolytic capacitor anodes*. PhD thesis, Case Western Reserve University, 2005.

[27] Equo Kobayashi, Shigeru Matsumoto, Takayuki Yoneyama, Hitoshi Hamanaka, et al. Mechanical properties of the binary titanium-zirconium alloys and their potential for biomedical materials. *Journal of biomedical materials research*, 29(8):943–950, 1995.

[28] Lambda. *PXE Series (Dual Output) DC-DC Converters*, July 2005.

[29] Learning about electronics. `http://www.learningaboutelectronics.com/Articles/Types-of-capacitors`.

[30] E. C. Levy. Complex-curve fitting. *Automatic Control, IRE Transactions on*, AC-4(1):37–44, 1959.

[31] Byron Gilman Mark W. Kroll, Karl Kroll. Idiot-proofing the defribrillator, 2008.

[32] Maxim Integrated. *An Efficiency Primer for Switch-Mode, DCDC Converter Power Supplies*, December 2008.

[33] Francis Merat Michael DeLibero, Steven Ehret. Instrumentation for the anodization and characterization of titanium electrodes for electrolytic capacitors. In *Energytech, 2012 IEEE*. IEEE, May 2012.

[34] J.R. Miller. Introduction to electrochemical capacitor technology. *Electrical Insulation Magazine, IEEE*, 26(4):40–47, July 2010.

[35] Reem Malik Moshe Gerstenhaber. More value from your absolute value circuit. *Back Burner*, 44(04), April 2010.

[36] Roland Neumayer, A. Stelzer, F. Haslinger, and R. Weigel. On the synthesis of equivalent-circuit models for multiports characterized by frequency-dependent parameters. *Microwave Theory and Techniques, IEEE Transactions on*, 50(12):2789–2796, Dec 2002.

[37] Ming-Jen Pan and Clive A. Randall. A brief introduction to ceramic capacitors. *Electrical Insulation Magazine, IEEE*, 26(3):44–50, May 2010.

[38] Winfield Hill Paul Horowitz. *The Art of Electronics*. Cambridge University Press, 3 edition, 2015.

[39] Matthew Pilotte. Operation of rf detector products at low frequency: An-691, 2005.

[40] Ian Poole. Silver mica capacitor. `http://www.radio-electronics.com/info/data/capacitor/silver-mica-capacitor.php`.

[41] S. Pooranchandra, B. Sasikala, and Afzal Khan. *Introduction to Electrical , Electronics and Communication Engineering*. FireWall Media, 2005.

[42] Krik K. Reed. Characterization of tantalum polymer capacitors, 2005.

[43] R.P. Sallen and E.L. Key. A practical method of designing rc active filters. *Circuit Theory, IRE Transactions on*, 2(1):74–85, March 1955.

[44] C.K. Sanathanan and J. Koerner. Transfer function synthesis as a ratio of two complex polynomials. *Automatic Control, IEEE Transactions on*, 8(1):56–58, Jan 1963.

[45] RJ Shephard. The design of a cardiac defibrillator. *British heart journal*, 23(1):7, 1961.

[46] Sim surfing. `http://ds.murata.co.jp/software/simsurfing/en-us/index.html?intcid5=com_xxx_xxx_cmn_nv_xxx`.

[47] Sprague 50 year timeline. `http://www.vishay.com/landingpage/50year/sprague.html`.

[48] High voltage power supplies. `http://www.thinksrs.com/downloads/PDFs/Catalog/PS300c.pdf`.

[49] Series ps300 high voltage power supplies. `http://www.thinksrs.com/downloads/PDFs/Manuals/PS300m.pdf`.

[50] Electronic components - dipped mica capacitors. `http://www.tedss.com/LearnMore/Mica-Capacitors`.

[51] Phatiphat Thounthong, Stéphane Raël, and Bernard Davat. Control strategy of fuel cell/supercapacitors hybrid power sources for electric vehicle. *Journal of Power Sources*, 158(1):806–814, 2006.

[52] Virgilio Valdivia, D Lopez del Moral, M Sanz, A Barrado, Antonio Lázaro, and F Tonicello. Simple modelling method of tantalum capacitors. *Electronics letters*, 47(1):22–23, 2011.

[53] Webench filter designer. http://www.ti.com/lsds/ti/analog/webench/webench-filters.page.

[54] GE Welsch. Doped titanium for forming capacitors with high energy density. In *The 22nd Advanced Aerospace Materials and Processes (AeroMat) Conference and Exposition*. ASM, May 2011.

[55] Gerhard Welsch and Donald McGervey. Directionally-grown capacitor anodes, May 1 2001. US Patent 6,226,173.

[56] H. Wen, W. Xiao, and Xuhui Wen. Comparative evaluation of dc-link capacitors for electric vehicle application. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, pages 1472–1477, May 2012.

[57] Ceramic capacitor. http://en.wikipedia.org/wiki/Ceramic_capacitor#Class_3_ceramic_capacitors, November 2014.

[58] Electrolytic capacitor. http://en.wikipedia.org/wiki/Electrolytic_capacitor.

[59] Silver mica capacitor. http://en.wikipedia.org/wiki/Silver_mica_capacitor, January 2014.

[60] Chebyshev polynomials of the first kind. http://mathworld.wolfram.com/ChebyshevPolynomialoftheFirstKind.html, 2015.