Developing Steady-State Thermoreflectance and 3ω Methods For Thermally Characterizing Microwave Atom Chips

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science with Honors in
Physics from the College of William and Mary in Virginia,

by

Paul T. Kelemen

Advisor: Prof. S. Aubin

Prof. S. Mordijck

Prof. D. Maliniak

Williamsburg, Virginia
May 19 2026

Contents

Acknowledgments

I dedicate this thesis to my brother, Cole, who walks through the valley and will rest upon the mountain. My deepest appreciation and thanks to Prof. Seth Aubin, whose outstanding and always genial support and mentorship were invaluable to my project and development as a scientist. Thank you to Trevor Tingle and Brian Hurley for their deposition and characterization work, Kisa Avrutina for their computational guidance, Dr. Adam Vernon for their assistance calibrating equipment, and Russell Kamback for their advice and friendship. And thank you to my mother and father, to whom I owe every success. May the ball keep rolling.

List of Figures

List of Tables

Abstract

Physicists use devices called microwave atom chips to spatially localize or "trap" atoms for cold atom experiments. These chips can require up to fifty Watts of microwave power to pass through its copper traces. In such high-power applications, trace overheating can damage or destroy the chip [1, 2]. To mitigate this risk, microwave atom chips consist of a 50 μm thick-film layer of aluminum oxynitride (ALON), which dissipates heat away from the chip's traces. These thick-films are called microwave heatsinks, and their performance is determined by AlON's thermal conductivity, which is not well-known [2]. We investigate two methods for thermally characterizing thick-film AlON and construct two experimental setups to perform non-intrusive thermometry measurements using a steady-state thermoreflectance (SSTR) and 3ω method [3, 4, 5]. Using the SSTR method, we determine the thermoreflectance coefficient of a protected aluminum mirror desposited onto fused silica to be $\beta = (3.38 \pm 0.62) \cdot 10^{-5}$ K^{-1}. Interestingly, this value does not agree with the widely accepted literature value for aluminum's thermoreflectance coefficient of $\beta \approx 1.25 \cdot 10^{-4}$ K^{-1} [3, 6]. For the 3ω method, we present our observations of the 3ω effect in a through-hole 100 Ω resistor over an AC ($\frac{\omega}{2\pi} = 10$ Hz) current range from 0.05 Amps to 0.105 Amps and determine maximum oscillatory temperature changes of $\Delta T_{AC} = 5.0K \pm 1.3K$ (Figure 4.6) and $\Delta T_{AC} = 7.4K \pm 1.8K$ (Figure 4.7). While the SSTR experimental setup we develop poses some challenges for future development, our 3ω setup shows great promise for thermally characterizing thick-film AlON. Work is presently underway applying our 3ω experimental setup to thick-film sapphire as a final calibration step before determining thick-film AlON's thermal conductivity.

Chapter 1

Introduction

1.1 Motivation

When atoms are cooled to near absolute zero - the coldest possible temperature in the universe - they stop behaving like particles and begin behaving more like waves. These cold atom waves are similar to waves found in an ocean and can overlap with one another to create complex interference patterns that retain information about their constituent waves. Physicists are developing techniques to manipulate and interpret these interference patterns to measure tiny gradients in the electromagnetic and gravitational forces. If successful, this technology would be iminently applicable to many high-precision usecases for everything from communications to deep-space navigation [1, 2].

Physicists use devices called microwave atom chips to spatially localize or "trap" atoms for cold atom experiments. These chips generate strong magnetic fields to create stable regions of magnetic "dead-space" called trapping potentials where atoms can fall in and get "stuck" (Figure 1.1). To do this, microwave atom chips must push up to 50 Watts of microwave power (roughly half an Amp) through three antiparallel copper traces only 54 μm wide [1, 2]. That is thinner than a human hair! In such high-power applications, the Joule-heating (Formula 2.3) effect can produce an

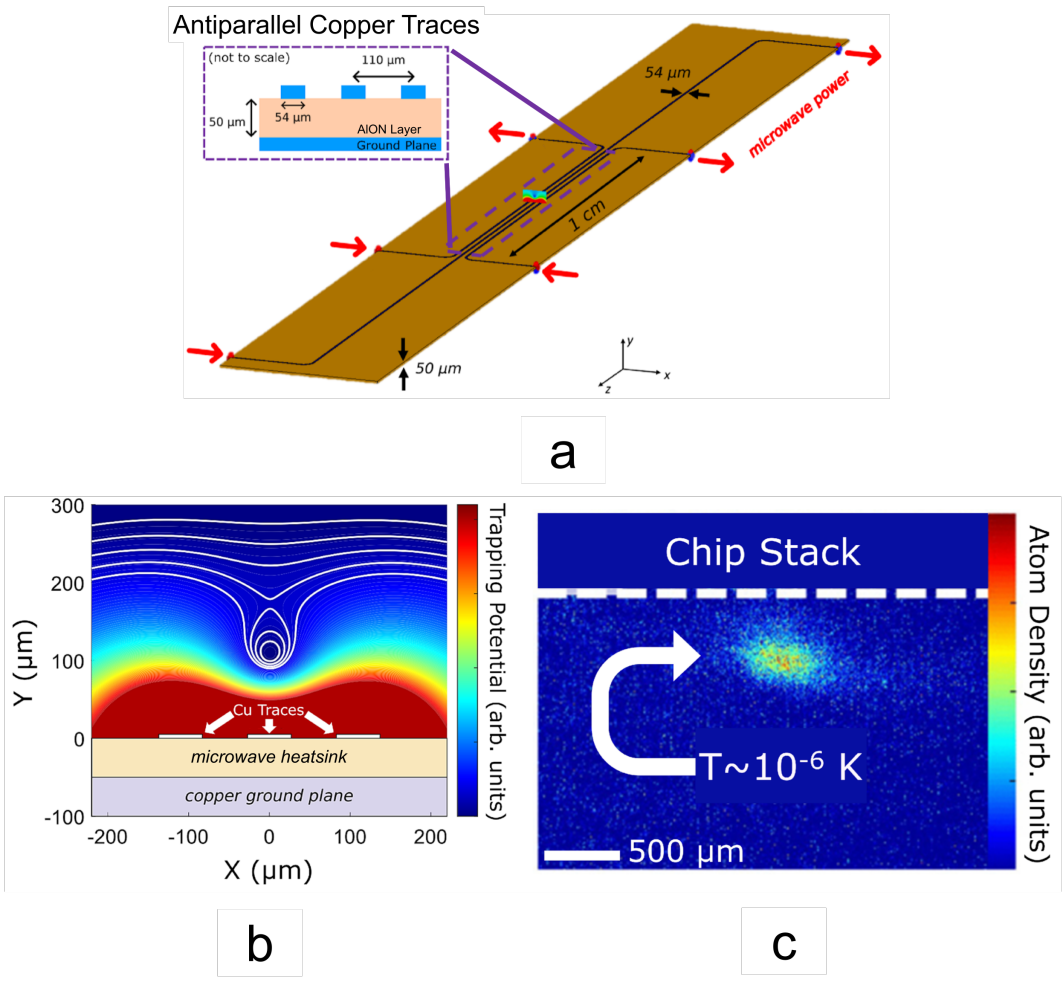

Figure 1.1: (a) "UZU" microwave atom chip design with copper ground plane, AlON microwave heatsink, and three antiparallel copper traces. Adapted from [2].
(b) Cross section of microwave atom chip and simulation of magnetic trapping potential from aluminum nitride (AlN) microwave heatsink. Adapted from [2].
(c) Experimental image of ultracold potassium atoms caught in microwave atom chip's magnetic trapping potential. Adapted from [7].

overwhelming amount of heat that is often sufficient to damage or destroy the chip. The AC Skin effect (Formula 2.4) can also exacerbate this issue by raising localized temperatures along the edges of the trace (Figure 2.1). Moreover, we suspect that heat-dependent changes in copper's resistivity (Formula 2.10) could interfere with

the microwave currents passing through a chip's copper traces, potentially further imparing the chip's performance and operational lifetime. Defective chips must then be replaced, requiring weeks of effort at great opportunity cost to the lab.

Microwave atom chips consist of a thick layer of electrically insulative material called a microwave heatsink that helps guide microwave currents and dissipates heat away from the chip's copper traces [1, 2]. The William & Mary (W&M) Ultra-Cold Atomic, Molecular, and Optical (AMO) Physics Lab uses a "UZU" chip design that consists of a 50 μm aluminum oxynitride (AlON) microwave heatsink. AlON is used because of its high dielectric constant ($\epsilon \approx 11$). This describes AlON's ability to be easily polarized in the presence of electromagnetic fields and to store energy from these fields [2]. AlON is also preferred for its ease of deposition onto copper which can serve as an electrical and thermal ground plane for the chip [1, 2].

An AlON microwave heatsink's performance is determined almost entirely by its thermal conductivty. This property describes AlON's ability to conduct heat away from the microwave atom chip's copper traces. Unfortunately, AlON's thermal conductivity is neither well-known nor well-understood [2]. Recent studies on the very similar material aluminum nitride (AlN) have shown that its thermal conductivity varies dramatically as a function of sample crystallography, thickness, temperature, and morphology [8, 9, 10, 11]. We hypothesize that these considerations are similarly relevant for our thick-film AlON microwave heatsinks. As such, an in-lab method for thermally characterizing UZU microwave atom chips, especially in relation to thick-film AlON's thermal conductivity, would be of great utility to the W&M AMO Physics Lab.

1.2 Background

Unfortunately, some of the most common methods for measuring the temperature and thermal conductivities of thick-film materials (such as time-domain, frequency-domain, and transient thermoreflectance) require ultrafast nanosecond, picosecond, or even femtosecond pulsed lasers, which are prohibitively expensive [12]. As such, this thesis focuses on two simple and cost-effective alternative methods for thermally characterizing thick-film aluminum oxynitride (AlON): steady-state thermoreflectance (SSTR) and the 3ω method [3, 4, 5]:

1.2.1 The Steady-State Thermoreflectance (SSTR) Method

When a material's temperature changes, the amount of light that can be reflected off its surface will slightly change. The material's change in normalized reflectivity is directly proportional to its change in temperature (Formula 2.5). This phenomenon is called the thermoreflectance effect and is the underlying principle behind the SSTR method. Although small, the thermoreflectance effect can be observed by modulating a substrate's temperature while simultaneously measuring the change in power of a reflected laser beam incident on a small metal pad deposited on the substrate's surface. If the metal's thermoreflectance coefficient (β) is known, it is possible to use SSTR to perform highly time-resolved, non-intrusive thermometry measurements of the underlying substrate. And because β is generally a function of a material's thermal conductivity, it is also possible (although often very difficult) to determine a value for the substrate's thermal conductivity [3].

In this thesis, we work backwards and use the SSTR method to determine the thermoreflectance coefficient of a protected aluminum mirror desposited onto a fused silica substrate. Instead of using the mirror's thermoreflectance coefficient to measure

the silica's temperature, we induce a known temperature gradient and determine the mirror's thermoreflectance coefficient. By measuring the normalized change in reflectivity while simultaneously modulating the silica substrate's temperature, we determine a thermoreflectance coefficient of $\beta = (3.38 \pm 0.62) \cdot 10^{-5}$ K^{-1}. Interestingly, this value does not agree with the widely accepted literature value for aluminum's thermoreflectance coefficient of $\beta \approx 1.25 \cdot 10^{-4}$ K^{-1} [3, 6]. One exciting potential interpretation of this unexpected result is discussed later in Section 3.3.

Work applying the SSTR method to the William & Mary Ultra-Cold Atomic, Molecular, and Optical Physics Lab's "UZU" microwave atom chip design is underway, and micrographs of deposited aluminum pads on an AlON substrate sample are provided (Figure 5.1). However, there are several significant challenges which will need to be overcome before SSTR can be viable for our group. Our primary hurdle is that our current SSTR experimental setup estimates our protected aluminum mirror's thermoreflectance coefficient by inducing a roughly 100 K temperature change over the course of tens of minutes (Figure 3.6). However, to determine the thermal conductivity of thick-film AlON, we would need to dramatically reduce these temperature and time scales. In an ideal experiment, our laser and measuring equipment would need to induce and be sensitive to temperature changes occuring on the time scale of tens of microseconds [3, 12]. This is currently impossible for our lab. Other challenges include focusing our laser onto micrometer-scale aluminum pads and the rather complex data analysis necessary to determine thermal conductivity from transient thermometry measurements.

1.2.2 The 3ω Method

For low-power applications at room temperature, a copper trace's electrical resistance is almost exactly constant in time (Formula 2.2). However, when high-

frequency, high-power AC currents pass through the trace, the Joule-heating effect raises its temperature, and its electrical resistance begins to oscillate as a function of time (Formulae 2.1, 2.7, & 2.10). It has been shown (and we derive in Section 2.3) that this oscillating electrical resistance induces a small voltage amplitude that oscillates at a frequency ($3f$) exactly three times greater than the input (fundamental) frequency (f) - the third harmonic ($3\omega = 6\pi f$) [4, 5]. This is the guiding principle behind the 3ω method. By measuring the 3ω component of the voltage drop across our microwave atom chip's copper traces as a function of the input current, we will be able to take highly time-resolved thermometry measurements that inform us about thick-film AlON's thermal conductivity [4, 5]. The higher the amplitude, the poorer the microwave heatisnk's performance. We present our observations of the 3ω effect in a through-hole 100 Ω resistor over an AC ($\frac{\omega}{2\pi} = 10$ Hz) current range from 0.05 to 0.105 Amps and determine maximum oscillatory temperature changes of $\Delta T_{AC} = 5.0K \pm 1.3K$ (Figure 4.6) and $\Delta T_{AC} = 7.4K \pm 1.8K$ (Figure 4.7).

The 3ω method is the most promising method for measuring the thermal conductivity of thick-film AlON in the William & Mary Ultra-Cold Atomic, Molecular, and Optical Physics Lab. It has several advantages over competing techniques, chief of which are that the 3ω method is convenient, modular, simple, and scalable. Our lab has demonstrated its capability to attach wires to even smaller electrical contacts than would be needed to perform 3ω experiments on a microwave atom chip, and Trevor Tingle's experience in similar experimental setups will inform our attempts. Work is currently underway to apply the 3ω method to a sample sapphire substrate with known thermal conductivity. We will use this sapphire sample as a calibration step before finally determining thick-film AlON's thermal conductivity. Design documentation and photos (Figures 5.1 - 5.3) of already deposited pads for measuring

the 3ω component of the voltage drop across our chip's copper traces are provided.

1.3 Relevance to NASA

Ultracold atom interferometry (UAI) is an emerging technology that will directly support NASA's Exploration Systems and Space Technology Mission Directorates [13]. If successful, UAI would enable the development of inertial navigation systems for deep-space travel and precision gravimetric mapping for research satellite initiatives like GRACE-FO [1, 2, 14]. Microwave atom chips are a promising technology for compact UAI that would scale down the technology to fit on a chip. Hence, it is imperative that atom chips be designed to be as long-lived and low-maintenance as possible. This research addresses two methods for thermally characterizing microwave atom chips. Future work using this research's results will enable accurate assessments of microwave atom chips' thermal conductivities and expected operational lifetimes and power ratings. Moreover, this project researches cost-effective methods for thermally characterizing microwave electronics, which is applicable in everything from long-range communication to enhanced radar. This research is sponsored by the Virginia Space Grant Consortium (VSGC) and was presented at the 2026 annual VSGC student research conference. Sections of this thesis are used in the student's VSGC year-long research report.

1.4 Thesis Structure

This thesis is structured as follows. Chapter 2 provides a theory overview of electrical heating effects relevant to microwave atom chips. A description of steady-state thermoreflectance (SSTR) theory and general implementation is provided along with a supporting ANSYS HFSS microwave atom chip computer simulation (Figure 2.1). Finally, a detailed overview and derivation of the theory behind the 3ω method is given along with common values for the temperature coefficient of resistance of electronic materials (Table 2.1) - an important constant needed to perform non-intrusive thermometry measurements using the 3ω method.

Chapter 3 details our SSTR theory calculations to help us build an expectation of what our experimental signal should look like. We present and explain our SSTR experimental setup (Figure 3.1) along with several modifications we added to control for noise, time-resolve our data, and acheive higher resolution signals. Several plots are presented to show the data analysis process (Figure 3.5) for our signal (Figure 3.6), and an exciting potential interpretation for our unexpected result is provided.

Chapter 4 shows our experimental setup for the 3ω method in a 100 Ω through-hole resistor with detailed scope readings in time and frequency space for both the control and experimental signals (Figure 4.2 - 4.5). Our results (Figure 4.6) are presented along with a detailed justification for our fitting parameters and an experimental value for our resistor's temperature coefficient of resistance (α).

Chapter 5 briefly details our recommendations for future work and conclusions along with a design document and several micrographs of deposited experimental setups for the SSTR and 3ω method on chip substrates (Figures 5.1 - 5.4).

Chapter 2

Steady-State & Transient Heating, Steady-State Thermoreflectance (SSTR), & 3ω Theory

In this chapter, we discuss steady-state and transient heating effects relevant to microwave atom chip copper traces. These include the Joule-heating (Section 2.1.1) and AC Skin effects (Section 2.1.2), as well as heating effects due to temperature-dependent resistance as discussed in Section 2.3. We also disuss the general theory and experimental setup of the SSTR method and how the two relate to one another (Section 2.2). We derive and explain the theory behind the 3ω method in full and describe how our thermometry measurments can be linked to AlON's thermal conductivity in future work (Formula 2.16).

2.1 Relevant Heating Effects

Microwave atom chips pass high-power microwave currents through their 54 μm copper traces to produce stable trapping potentials for cold atom research [1, 2]. However, these currents can often be sufficient to overheat the traces and damage or destroy the chip. There are two heating effects that are of primary concern in the "UZU" chip design (Figure 1.1): Joule-heating and the AC Skin effect:

2.1.1 Joule-Heating

Joule-heating is likely the heating effect you first think of when considering overheating in electronics. It is what limits the performance of computer chips and warms appliances like your car's cigarette lighter or a hot iron. Joule-heating is the effect in which current (I) flowing though a material generates heat proportional to that material's electrical resistance (R):

$$P(t) = I^2(t)R(t) \tag{2.1}$$

where $P(t)$ is the heating power applied by the current to the material as a function of time t. For low-power applications at room temperature, $R(t)$ assumes a nearly constant value of R_0, which is the material's resistance at room temperature [4]. And if we let $I(t)$ be proportional to a simple cosine wave, our equation becomes:

$$P(t) = {I_0}^2 cos^2(\omega t)R_0 \tag{2.2}$$

where I_0 is the peak current amplitude, $\omega = 2\pi f$ is the fundamental angular frequency, and f is the fundamental frequency (in Hertz) at which the current oscillates. Averaging Formula 2.2 over time reveals the steady-state expression for the heating power ($\overline{P}$) delivered to the material by the oscillating current [4]:

$$\overline{P} = \frac{1}{2}{I_0}^2 R_0 \tag{2.3}$$

2.1.2 AC Skin Effect

Rather than directly describing the heating power delivered to a material like Joule-heating (Formula 2.3), the AC Skin effect describes the uneven distribution of an oscillating current across a material's surface. The higher the current's natural angular frequency (ω), the more it will tend to gather along the material's edges. The

strength of this effect is determined by the material's skin depth (δ) - the characteristic length over which current density decays inside the material's volume [15]:

$$\delta = \sqrt{\frac{2}{\sigma\mu\omega}} \tag{2.4}$$

where σ and μ are the material's conductivity and magnetic permeability, respectively. For copper, these values are well-known [16]:

$$\sigma \approx 5.95 \cdot 10^7 \text{ S/m and } \mu \approx 1.3 \cdot 10^{-6} \text{ N}/A^2$$

For microwave atom chips, which run at a characteristic frequency of 6.8 GHz, the AC Skin effect poses an enormous problem [1, 2]. Its copper traces' skin depths at room temperature are just 0.78 micrometers on either side. For a 54 μm trace, that means the majority of the microwave currents will be concentrated in just $\sim$ 5.6% of the trace's surface area! However, it should be noted that Formula 2.4 slightly underestimates copper's skin depth for small features like our microwave atom chip's traces. Figure 2.1 demonstrates this uneven current distribution in an ANSYS HFSS computer simulation of a chip's copper trace passing 20 Watts of microwave current at 6.8 GHz. It calculates an estimated surface current density along the trace edges on the order of tens of kiloAmps per meter! In turn, since heating power is proportional to the square of the current (Formula 2.3), this uneven current distribution will yield large temperature gradients across the trace's surface.

2.2 SSTR Theory and the General SSTR Setup

The steady-state thermoreflectance (SSTR) method for performing highly time-resolved, non-instrusive thermometry measurements in thick-film materials relies (perhaps unsurprisingly) on the principle of thermoreflectance [3]:

$$\frac{\Delta\mathcal{R}}{\mathcal{R}} = \beta\Delta T \tag{2.5}$$

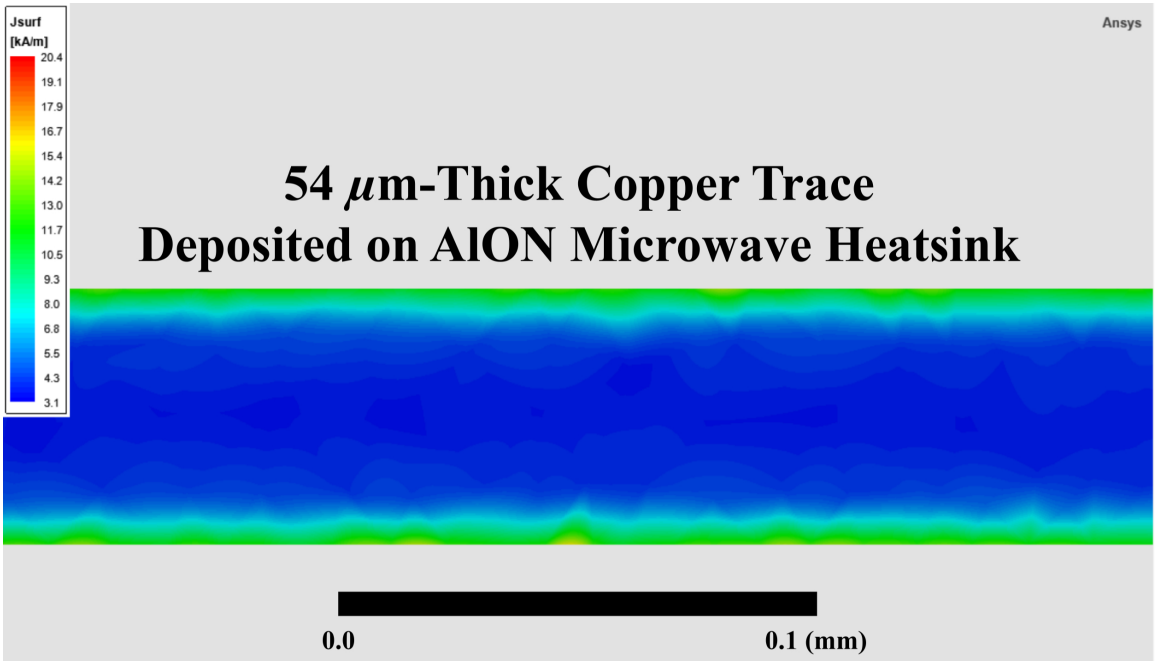

Figure 2.1: ANSYS HFSS current density simulation of a 54 μm copper trace deposited on an AlON microwave heatsink. The simulated signal is a 20 Watt current oscillating at a fundamental frequency of $f = 6.8$ GHz. ANSYS HFSS calculates an estimated surface current density along the trace edges on the order of tens of kilo-Amps per meter (kA/m) with a maximum surface current density range from $\sim$ 3.1 kA/m - $\sim$ 20.4 kA/m.

where $\frac{\Delta\mathcal{R}}{\mathcal{R}}$ is the normalized change in a material's reflectivity, β is the thermoreflectance coefficient of the material, and ΔT is the change in the material's temperature. Put simply, the amount of light that can be reflected off of a material's surface is directly proportional to the material's change in temperature. And the constant that governs this relationship (β) is generally a function of the material's thermal conductivity [3]. So, if you know your material's thermoreflectance coefficient, you can use SSTR to measure your material's change in temperature like a thermometer!

Figure 2.3 shows Braun et al.'s original experimental SSTR setup from 2019 [3]. A dual pulsed-laser system simultaneously heats (green laser $\rightarrow \Delta T$) and measures the normalized change in reflectivity (red laser $\rightarrow \frac{\Delta\mathcal{R}}{\mathcal{R}}$) of a thick-film sample. These

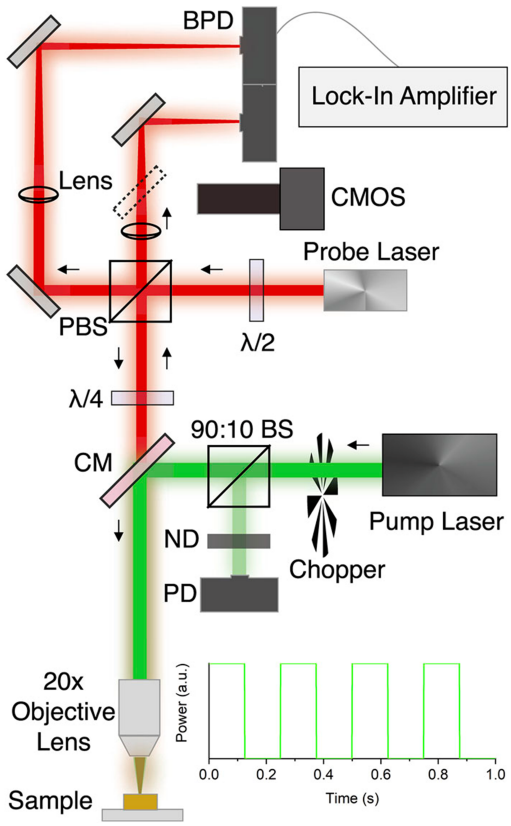

Figure 2.2: Diagram of the general SSTR experimental setup in Braun et al., in 2019 [3]. The pump (green) and probe (red) lasers modulate temperature and measure the sample's normalized change in reflectivity, respectively. The optical chopper pulses the pump laser to generate a square-waveform in which the normalized change in reflectivity climbs to a steady-state peak and decays to zero. If the sample's thermoreflectance coefficient is known, SSTR can perform highly time-resolved, non-intrusive thermometry measurements of the sample (Formula 2.5). Adapted from [3].

systems are called the "pump" and "probe" lasers, respectively. An optical chopper that spins blades like an office fan chops the pump laser into short, periodic pulses that heat (laser on) and cool (laser off) the sample. While the sample is undergoing oscillatory heating, the probe laser is sent through a beam-splitter (called PBS in the diagram) that splits the laser into two identical waves half the intensity of the original beam. One of these two waves is immediately measured to act as a control on the laser source, and the second is reflected off the sample and measured. These

two signals are then subtracted to determine the change in reflectivity as a function of time. Braun et al.'s 2019 experimental setup heavily inspires the "skeleton" of our own SSTR experimental setup (Section 3.2) [3].

2.3 3ω Theory

By the double-angle identity, Formula 2.2 can be expanded as follows:

$$P(t) = \frac{{I_0}^2 R_0}{2} + \frac{{I_0}^2 R_0}{2} cos(2\omega t) \tag{2.6}$$

One component is constant, and the other varies with time, so Joule-heating has both a steady-state ($P_{DC} = \frac{{I_0}^2 R_0}{2}$) and oscillatory ($P_{AC} = \frac{{I_0}^2 R_0}{2} cos(2\omega t)$) component. Thus, the material's change in temperature (ΔT) must also oscillate [4]:

$$P(t) = P_{DC} + P_{AC} \rightarrow \Delta T = \Delta T_{DC} + \Delta T_{AC} \tag{2.7}$$

It is also clear the oscillatory temperature change component (ΔT_{AC}) will vary at twice the input current's fundamental frequency (2ω being the second harmonic of the current's fundamental angular frequency ω). ΔT_{AC} will also acquire some phase shift ϕ to represent the lag time it takes for the material to warm in response to the current [4]:

$$\Delta T_{\mathrm{AC}} = |\Delta T_{\mathrm{AC}}| \cos(2\omega t + \phi) \tag{2.8}$$

The relationship between voltage (V), current (I), and resistance (R) is described by Ohm's law:

$$V(t) = I(t)R(t) \tag{2.9}$$

But at high-power, high-frequency applications, R(t) can no longer be considered constant in time. Formula 2.2 fails. R(t) instead takes the temperature-dependent form:

$$R(t) = R_0(1 + \alpha \Delta T) \tag{2.10}$$

where α is the material's temperature coefficient of resistance [4]:

$$\alpha \approx \frac{1}{R}\frac{dR}{dT} \tag{2.11}$$

The temperature coefficient of resistance (α) for some common materials at room temperature can be found in Table 1:

Material	**Temperature Coefficient of Resistance (α)**
Silver	0.0038
Copper	0.00386
Aluminum	0.00429
Platinum	0.003927
Carbon	-0.0005
Silicon	-0.07

Table 2.1: Temperature coefficient of resistance (α) for common electronics materials at room temperature. Notice α can be positive or negative. Adapted from [17].

Applying Formulae 2.7, 2.8, and 2.10 and again taking $I(t)$ to equal $I_0 cos(\omega t)$, then Formula 2.9 becomes [4]:

$$V(t) = I_0 R_0 cos(\omega t)(1 + \alpha\Delta T_{DC} + \alpha|\Delta T_{\mathrm{AC}}|\cos(2\omega t + \phi)) \tag{2.12}$$

By the cosine product-to-sum identity:

$$\cos(\omega t)\cos(2\omega t + \phi) = \frac{\cos(\omega t + \phi) + \cos(3\omega t + \phi)}{2} \tag{2.13}$$

So, Formula 2.12 expands as follows [4]:

$$V(t) = I_0 R_0[\cos(\omega t) + \alpha\Delta T_{DC}\cos(\omega t) + \frac{1}{2}\alpha|\Delta T_{\mathrm{AC}}|\cos(\omega t + \phi) + \frac{1}{2}\alpha|\Delta T_{\mathrm{AC}}|\cos(3\omega t + \phi)] \tag{2.14}$$

All components in Formula 2.14 vary at the "fundamental" angular frequency (ω) except for the final term:

$$V_{3\omega}(t) = \frac{1}{2} I_0 R_0 \alpha|\Delta T_{\mathrm{AC}}|\cos(3\omega t + \phi) \tag{2.15}$$

This is the essence of the 3ω method. When a current flows through a resistive material at some fundamental angular frequency (ω), the material begins to periodically warm as the current rises and falls. This heating occurs at a frequency three times greater than the fundamental and induces a voltage amplitude at that third harmonic ($V_{3\omega}$). And if we know the material's temperature coefficient of resistance (α), Formula 2.15 can reveal its oscillatory temperature change ($|\Delta T_{AC}|$), enabling us to perform non-intrusive thermometry measurements. By using $V_{3\omega}$ to determine $|\Delta T_{\mathrm{AC}}|$ at multiple values for the fundamental on a microwave atom chip, it will be possible to determine its thermal conductivity [4]:

$$\Delta T_{AC}(2\omega) = -\frac{P_{AC}}{\pi\kappa L}\left[\frac{1}{2}\ln(2\omega) + \frac{1}{2}\ln\left(\frac{b^2}{D}\right) + C\right] - i\frac{P_{AC}}{4\kappa L} \tag{2.16}$$

where $P_{AC} = \frac{{I_0}^2 R_0}{2} cos(2\omega t)$ is the oscillatory Joule-heating power delivered to the chip's copper trace (Formula 2.6 & 2.7); κ is AlON's thermal conductivty; L is the trace's length; $2b$ is the trace's width (54 μm); $D = \frac{\kappa}{\rho C_p}$ is AlON's thermal diffusivity where ρ and C_p are AlON's density (3691 kg/m^3 - 3696 kg/m^3) and specific heat (approximately 920 J/kg-K), respectively; and C is a constant to be calibrated with tests on samples with known thermal conductivties [1, 2, 4, 5, 18].

Chapter 3

SSTR Signal Expectations, Experimental Setup, & Results

In this chapter, we discuss our SSTR signal expectations based on theoretical predictions backed by literature values. We show and provide an in-depth description of our SSTR experimental setup, providing photographs of the setup and laser waveforms in the lab. Finally, we sumarize our SSTR results, determine the thermoreflectance coefficient (β) of a protected aluminum mirror deposited on fused silica, and discuss an interesting interpretation of why our result disagrees with our signal expectations.

3.1 SSTR Signal Expectations

Before constructing a full-fledged SSTR experimental setup like that shown in Figure 2.2, we must verify the principle of thermoreflectance for ourselves to build an informed expectation of our signal's expected magnitude. Formula 2.5 suggests that the higher we raise a material's temperature, the larger the normalized change in reflectivity we can measure. And beccause a material's thermoreflectance coefficient (β) is a function of the incident light's wavelength, we can cleverly choose our material to maximize β at our lab's most easily accessible laser wavelength of 780 nm.

Aluminum has one of the highest thermoreflectance coefficients of common metals at this wavelength with β being roughly equal to $1.25 \cdot 10^{-4}$ K^{-1} [3, 6]. As such, we selected a protected aluminum mirror deposited on fused silica for our SSTR test sample.

By wrapping a copper coil heating element around the perimeter of the mirror and attaching a thermocouple to its back, we are able to raise the aluminum's temperature by $10^2 K$ (Figure 3.2). This heating coil is powered by a CC (constant current) power source that is set to 2.54 Amps when heating and turned off when cooling. This means our expected normalized change in reflectivity at our maximum change in temperature is on the order of 10^{-2} (Formula 2.5):

$$\frac{\Delta \mathcal{R}}{\mathcal{R}} \sim \frac{1.25 \cdot 10^{-4}}{K} \cdot 10^2 K = 1.25 \cdot 10^{-2}$$

However, this scale only applies to the maximum temperature difference we plan to apply to our aluminum mirror. To capture a meaningful signal at $\Delta \mathrm{T} = 0.1K$ (the precision of a standard temperature-reading thermocouole), you would need to be able to read a signal on the order of 10^{-5}!

$$\frac{\Delta R}{R} \sim \frac{1.25 \cdot 10^{-4}}{K} \cdot 10^{-1} K = 1.25 \cdot 10^{-5}$$

Initially, this seems like a huge problem. Most oscilloscopes designed to measure signals like our laser's normalized change in resistivity have 12-bit analog-to-digital converters (ADCs). This means that when we capture the laser's waveform as we modulate our aluminum mirror's temperature, an oscilloscope will only be able to divide the signal into $2^{12} = 4{,}096$ partitions. Signal changes on the order of a part per 100,000 (like our laser's normalized change in reflectivity) will be too small to capture! The scope can only read out to about 3.5 digits of precision (dop) ($\log_{10}(2^{12}) \approx 3.6 \rightarrow$ 3.5 digits after the decimal point). High quality digital multimeters (DMMs) have

16-bit ($\sim$ 4.5 dop) or even 24-bit ($\sim$ 7.5 dop) ADCs that are sufficiently precise to handle these tiny signals. But critically, they have no way of collecting information about these signals as a function of time, which is needed to relate the signal to the mirror's temperature.

3.2 SSTR Experimental Setup

To rectify these precision issues, time-resolve our normalized change in reflectacne measurements, and observe the principle of thermoreflectance, we construct an SSTR experimental setup as seen in Figures 3.1 and 3.4. Our SSTR experimental setup works similarly to that of Braun et al.'s, from 2019 (Figure 2.2) [3]. By measuring the normalized change in reflectivity of a laser beam incident on an aluminum mirror's surface as we modulate the mirror's temperature, we observe the thermoreflectance effect (Formula 2.5) and can determine the mirror's thermoreflectance coefficient.

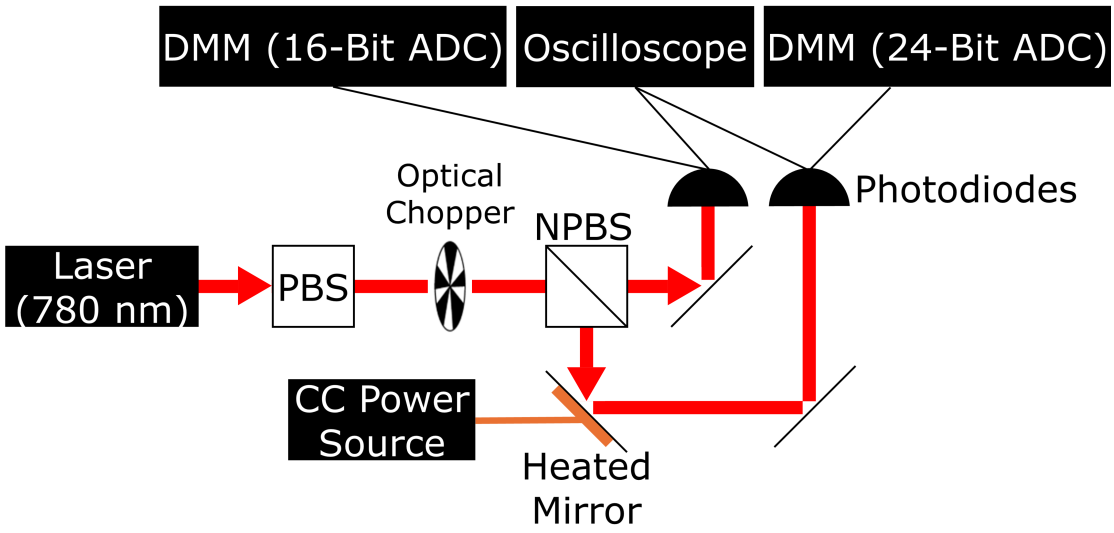

Figure 3.1: Our SSTR experimental setup for observing the principle of thermoreflectance with a protected aluminum mirror deposited onto fused silica. The 16-bit DMM is a Keithley DMM 6500, and the 24-bit DMM is a Keysight 34470A. Optical chopper PNG courtesy of Jack Stawasz.

3.2.1 SSTR Experimental Setup Description

First, a 780 nm laser passes through a polarizing beam splitter (PBS in Figure 3.1) to isolate one of the laser's two polarization axes. It does not matter whether the PBS filters out the vertical or horizontal axis, so long as it only allows one of them to pass. In early tests, we found that over long experimental runtimes, our laser's polarization would shift slightly, which would cause problems due to our mirror's polarization-dependent reflectivities. Filtering out all but a single polarization axis from our laser signal controls for this polarization shift. Next, the laser beam passes through an optical chopper that cuts our continuous laser signal into roughly 4 ms - 6 ms pulses (Figure 3.3). This one clever adaptation enables our experimental setup to simulatenously overcome our precision issues and time-resolve our normalized change in reflectivity measurements. Chopping our signal allows us to bin collected measurements, artificially time stamping our data by converting our continuous laser signal into a pulsed square wave with a set duty cycle (percentage of the time the laser is "on") of approximately 39%. By taking one data point from each pulse and dividing the total runtime by the number of collected binned data points (average time between pulses of approximately 11 ms), we can reconstruct our signal as a function of time (Figure 3.5). This bypasses the previous limitations of high-precision digital multimeters, allowing us to now use them and capture the miniscule changes in our signal we expect to see (Section 3.1)!

After passing through the PBS, optical chopper, and NPBS (non-polarizing beam splitter), our continuous laser signal has become two identical, filtered laser pulses (Figure 3.1). Of these two pulses, one is immediately reflected into a photodiode (light detector) connected to a 16-bit digital multimeter (DMM) with an 1-ms sampling rate. This signal is called our "control" signal and acts as a control on our laser source.

No matter how our source's power or polarization changes over time, we can always compare our second laser pulse to our control. Any changes we see in one are bound to appear in the other, and we can subtract the two signals to remove any bias in the signal due to variations in the source.

Our second laser pulse is called our "experimental" signal. It is the pulse that will carry data about the thermoreflectance effect. After passing through the NPBS, our experimental signal is reflected off of our protected aluminum mirror deposited onto fused silica (Thorlabs: PF05-03-G01). By wrapping a copper coil heating element around the mirror perimeter and attaching a thermocouple to the its back, we can modulate the mirror's temperature over time, which will influence its reflectivity according to Formula 2.5. To reach our target of large normalized changes in reflectivity on the order of 10^{-2} (Section 3.1), we used a constant current (CC) power source set to 2.54 Amps to heat the copper coil. Once a temperature of approximately 125°C was reached, the power source was shut off, and the mirror was allowed to cool back down to room temperature at approximately 25°C. Figure 3.2 shows a photograph of our protected aluminum mirror mounted to an optical breadboard with the copper heating coil wrapped around it and a thermocouple sticking out behind it.

After reflecting off our protected aluminum mirror (Figure 3.2), our experimental signal is reflected through a 780 nm bandpass filter (a screen which only allows 780 nm light to pass) into a photodiode connected to a 24-bit DMM also set to a 1-ms second sampling rate. An oscilloscope is connected to both photodiodes (Thorlabs: FDS 100) to show the general waveforms of the recombined pulsed control and experimental laser signals (see Figure 3.3). Figure 3.5 shows the ouputs from our two digital multimeters over time, from which it can be clearly seen that the control and experimental laser waveforms are of very similar forms thanks to our several methods

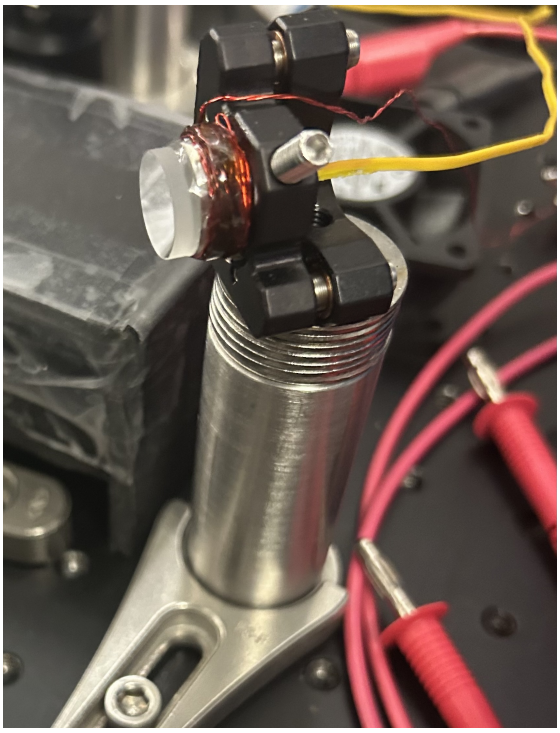

Figure 3.2: Photograph of our protected aluminum mirror (Thorlabs: PF05-03-G01) mounted to an optical breadboard. A copper heating coil is wound about the mirror's perimeter to modulate its temperature. The mirror is mounted to an aluminum peg, inside which a thermocouple (yellow wire) is placed to measure temperature.

to control for laser source variation. During our experiment, we leave the system at rest for the first ten minutes of the run to build a calibration dataset in which the normalized change in reflectivity is effectively zero (Figure 3.6). From this dataset, we use Python code (Appendix A.1) to perform a linear regression on the experimental signal and "scale it up" to the control signal's amplitude (Figure 3.5). Dividing these two signal now reveals our normalized change in reflectivity from which we can observe the thermoreflectance effect in our protected aluminum mirror deposited on fused silica (see Figure 3.6).

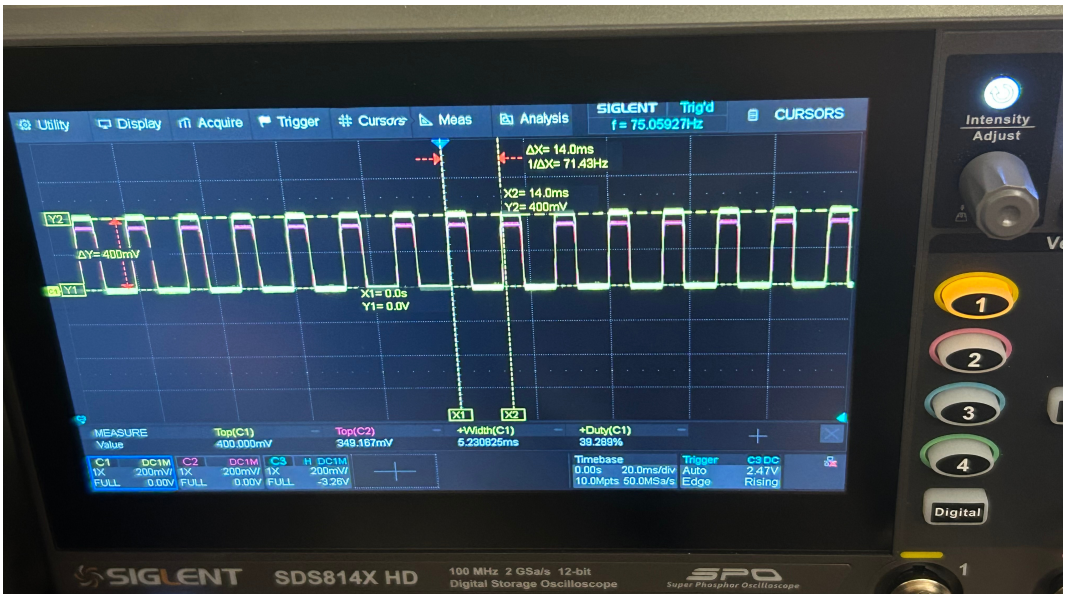

Figure 3.3: Oscilloscope reading of chopped pulse laser control and experimental waveforms in our SSTR experimental setup. The control signal amplitude is 400 mV, and the reflected signal amplitude is approximately 350 mV. The waveform shows a pulse width of 5.2 milliseconds with an approximate duty cycle of 39% and period of 14 milliseconds. Notice the square wave that enables us to "bin" our collected data.

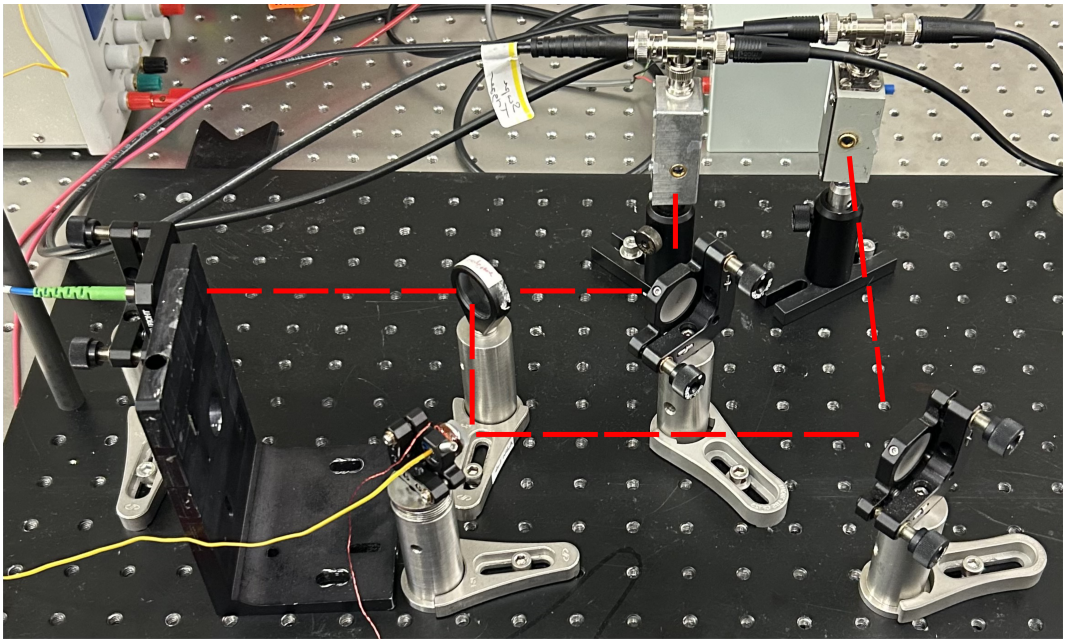

Figure 3.4: Photograph of our SSTR experimental setup for observing the principle of thermoreflectance in a protected aluminum mirror (Figure 3.2).

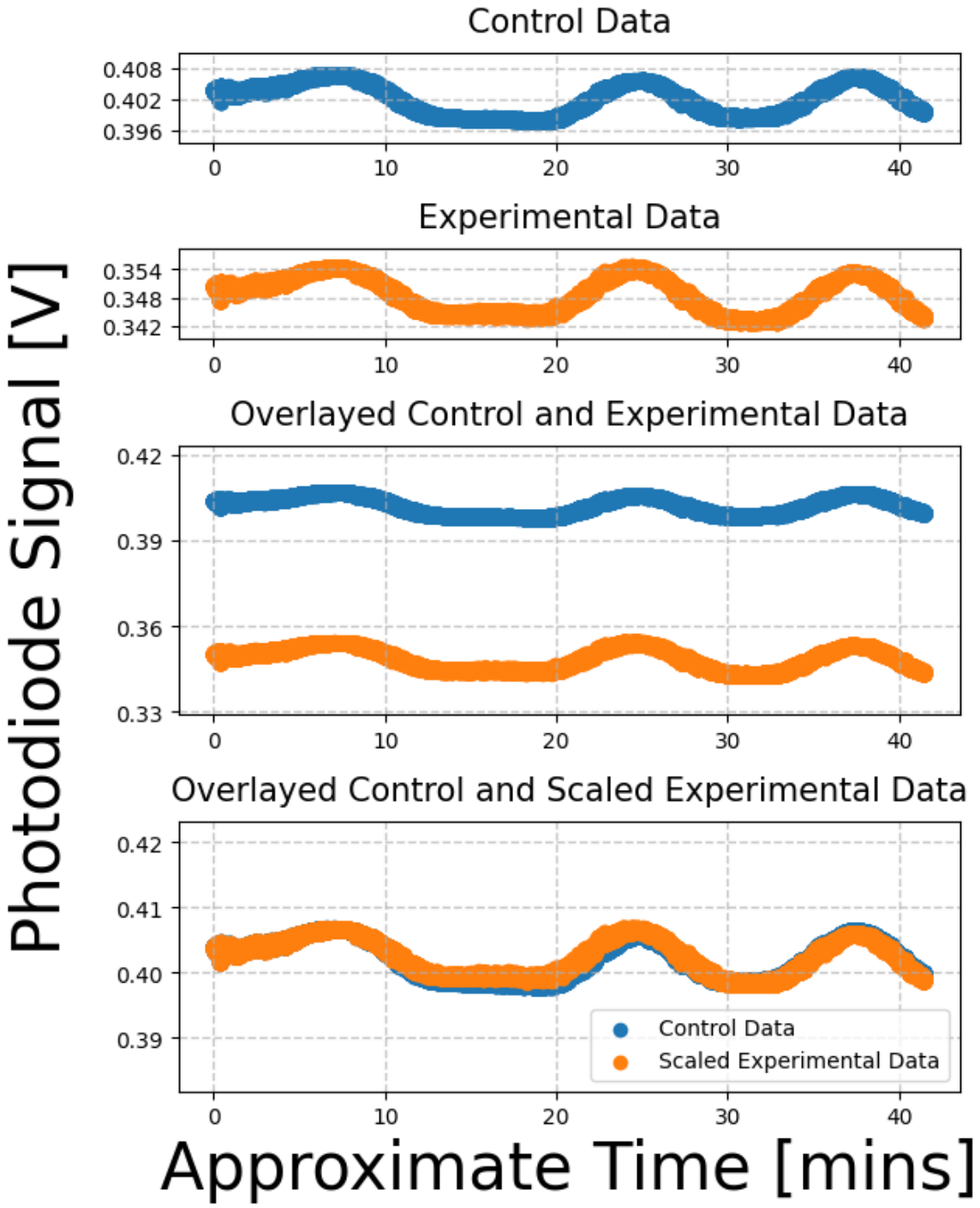

Figure 3.5: The control and experimental signal gathered from our SSTR experimental setup's dual digital multimeters as a function of time. The "overlayed control and scaled experimental data" graph represents our experimental signal overlapping with our control data after undergoing a linear regression (Appendix A.1). Dividing these two signals allows us to observe the thermoreflectance effect in Figure 3.6.

3.3 SSTR Results

From our observations of the thermoreflectance effect in a protected aluminum mirror deposited onto fused silica in Figure 3.6, three distinct regions of behavior can be identified. A ten-minute rest period is implemented at the beginning of the run to allow time for the electronics to settle and to calibrate the machines. During this time, data is also collected to perform a linear regression on the experimental setup (Figure 3.5). A close inspection of Figure 3.3 and 3.5 will reveal that the experimental signal (350mV) has a smaller amplitude than the control signal (400mV). This is expected and is simply due to the fact that the experimental signal must reflect twice and pass through a bandpass filter before reaching the photodiode, whereas the control signal must only reflect once before being measured. However, this amplitude difference is not useful to our experiment, and the first ten minutes of data collection allows me to map the experimental signal onto the control signal by linearly regressing the data set (Appendix A.1 & Figure 3.5).

During this ten-minute calibration period, the mirror's copper heating coil (Figure 3.2) is turned off ($\Delta T = 0$) and the normalized change in reflectivity rests at zero. After the calibration period ends, the copper heating coil is suddenly switched on, and the normalized change in reflectivity begins to rise roughly linearly with the protected aluminum mirror's change in temperature. To be precise, neither the normalized change in reflectivity nor the change in temperature are exactly linear relationships - you would expect the mirror to heat most quickly near the beginning of the "Linear Temperature Rise" and slow down as the normalized change in reflectivity approached its maximum value. However, our linear fit shows it is a good approximation ($R^2 \approx 0.7993$) to our collected data and will allow us to determine the thermoreflectance coefficient of the protected aluminum mirror (Figure 3.6).

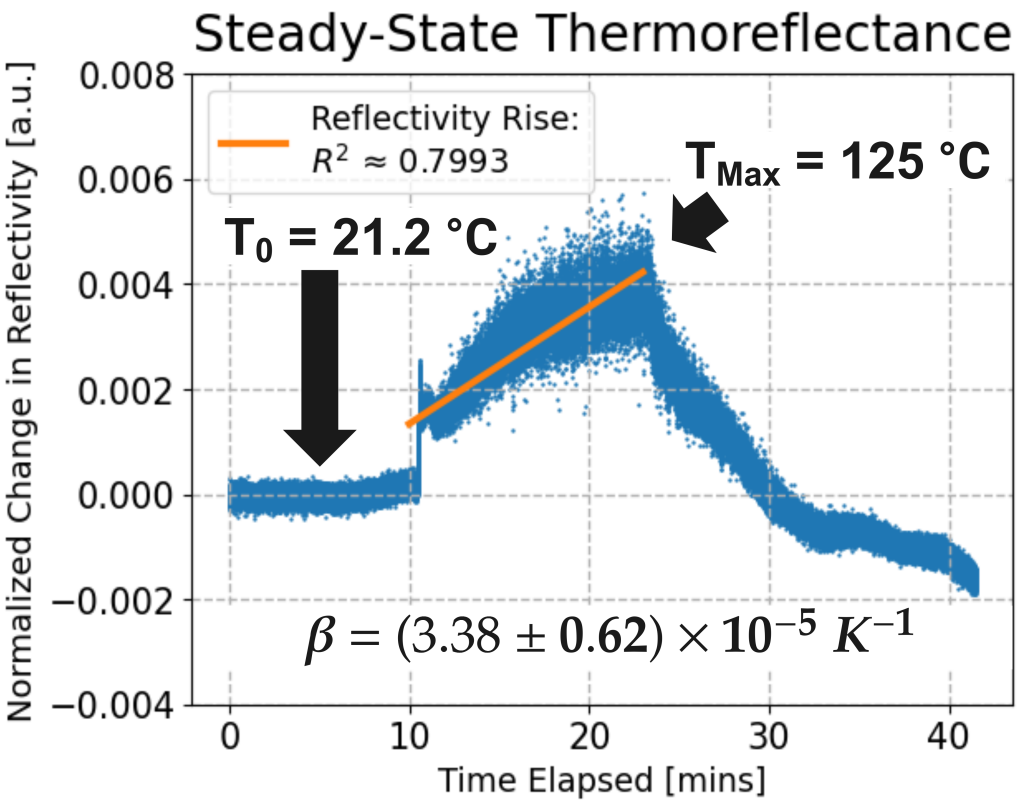

Figure 3.6: Normalized change in reflectivity in protected aluminum mirror deposited onto fused silica. Clear demonstration of the principle of thermoreflectance (Formula 2.5). Notice the three distinct regions of behavior along the time axis. An anomalous hump beginning approximately 32 minutes into the run makes it impossible to determine the thermoreflectance coefficient (β) of our protected aluminum mirror deposited onto fused silica after the copper heating element is turned off. Instead, β can only be determined from the linear temperature rise.

Once the protected aluminum mirror's maximum temperature of 125°C has been reached, the heating coil is turned off, and the model is allowed to cool in an approximately linear temperature decay down to 25°C. Unfortunately, in our collected data, we see that our normalized change in reflectivity dips below zero as our temperature decays back down to 25°C. We are unsure of what the significance of this hump is, and for this reason, rather than performing linear fits for both the temperature rise and fall, it is only possible to determine a thermoreflectance coefficient using data from

the rise in our normalized change in reflectivity from roughly 10 to 23 minutes into the experiment. The total range in normalized reflectivity predicted by our best fit line to the active (heating coil turned on) region (Figure 3.6) is $\frac{\Delta R}{R} = (3.51 \pm 0.64) \cdot 10^{-3}$. Plugging this value into Formula 5 along with our known temperature change at that point ($\Delta T \approx 125°\text{C} - 21.2°\text{C}$ (room temp) $= 103.8°\text{C}$):

$$(3.51 \pm 0.64) \cdot 10^{-3} \approx \beta \cdot 103.8°\text{C} \rightarrow \beta = (3.38 \pm 0.62) \cdot 10^{-5}\ \text{K}^{-1}.$$

Interestingly, this result does not match our initial theoretical expectations outlined in Section 3.1. We expected the protected aluminum mirror's thermoreflectance coefficient (β) to be roughly equal to $1.25 \cdot 10^{-4}$ K^{-1} because that is the broadly accepted literature value for aluminum at 780 nm [3, 6]. But our measured value of $\beta \approx 3.38 \cdot 10^{-5} K^{-1}$ is over 3.5x smaller than the accepted literature value! We struggled with this discrepancy for a while before realizing we may have accidentally stumbled onto an interesting result. Because our protected (thin layer of silica) aluminum mirror is so thin, we suspect it may be possible that our SSTR normalized change in reflectivity is being blurred by the thermal properties of the protective silica film or the fused silica underneath the aluminum. While not mentioned on the manufacturer's (Thorlabs) website, we briefly consulted with one of their engineers and confirmed that the deposited aluminum is only "a few nanometers thick", which makes our suspicion that some of the light is interfering with the silica somewhat plausible. Unfortunately, the literature on silica's thermoreflectance coefficient is sparse, so it is not possible to check whether this interpretation of our result is true.

Chapter 4

3ω Experimental Setup & Results

When an AC electrical signal oscillating at some fundamental frequency f is passed through a resistive material, the 3ω effect induces a non-zero voltage amplitude that oscillates at that fundamental's third harmonic $3f$ (Formula 2.15). This phenomenon can be used to perform non-intrusive thermometry measurements of thick-film materials, which in turn can be used to calculate their thermal conductivities (Formula 2.16)[4, 5]. In this section, we develop a 3ω experimental setup to observe the 3ω effect in a through-hole resistor and determine the oscillatory component of its change in temperature (ΔT_{AC}).

4.1 3ω Experimental Setup

Our experimental setup for observing the 3ω effect is diagrammed in Figure 4.1. A function generator produces a sinusoidal voltage signal that oscillates at a fundamental frequency of $f = 10$ Hz. This signal passes through a "Kepco" AC current source that outputs an identical wave whose current amplitude is constant in time. This ensures that any temperature-dependent changes in resistance later in the setup will not impact the amplitude of the current flowing through the circuit (Formula 2.9). The Kepco's output AC current signal is then read by a multimeter for measurement before being passed into our through-hole resistor setup.

Resistor Setup Approximate Four-Point Measurements ($\pm 0.00050\,\Omega$)		
Thermal Mass Resistor	**Control Resistor**	**Experimental Resistor**
99.83925 Ω	99.8155 Ω	99.78385 Ω

Table 4.1: Approximate Four-Point Resistance Measurements of the Resistor Components in the 3ω Resistor Setup (Figure 4.1).

Our resistor setup consists of three primary components: a thermal mass resistor (TMR) and control resistor, each consisting of four sets of four approximately 100 Ω resistors connected in parallel to create a total effective resistance of approximately 100 Ω; and a single experimental resistor also with a resistance of approximately 100 Ω. The approximate four-point (time-consuming but very accurate) resistance measurements of the thermal mass, control, and experimental resistors at room temperature are provided in Table 4.1. The difference in resistance between the control and experimental resistor is less than 0.05 Ω, minimizing any differences in the 3ω signal due to differences in internal resistance.

A simple measurement of the third harmonic's voltage drop across a single through-hole resistor as a function of current would be insufficient to observe the 3ω effect. It would be impossible to parse how much of the third harmonic's voltage amplitude came from background effects irrelevant to the oscillator temperature change and what was actually a result of the effect described in Formula 2.15. The resistor setup we develop solves this issue by measuring two sets of control signals to provide a background reading that can then be subtracted from the experimental signal. Any changes we see between the control and experimental runs must then be a result of the 3ω effect.

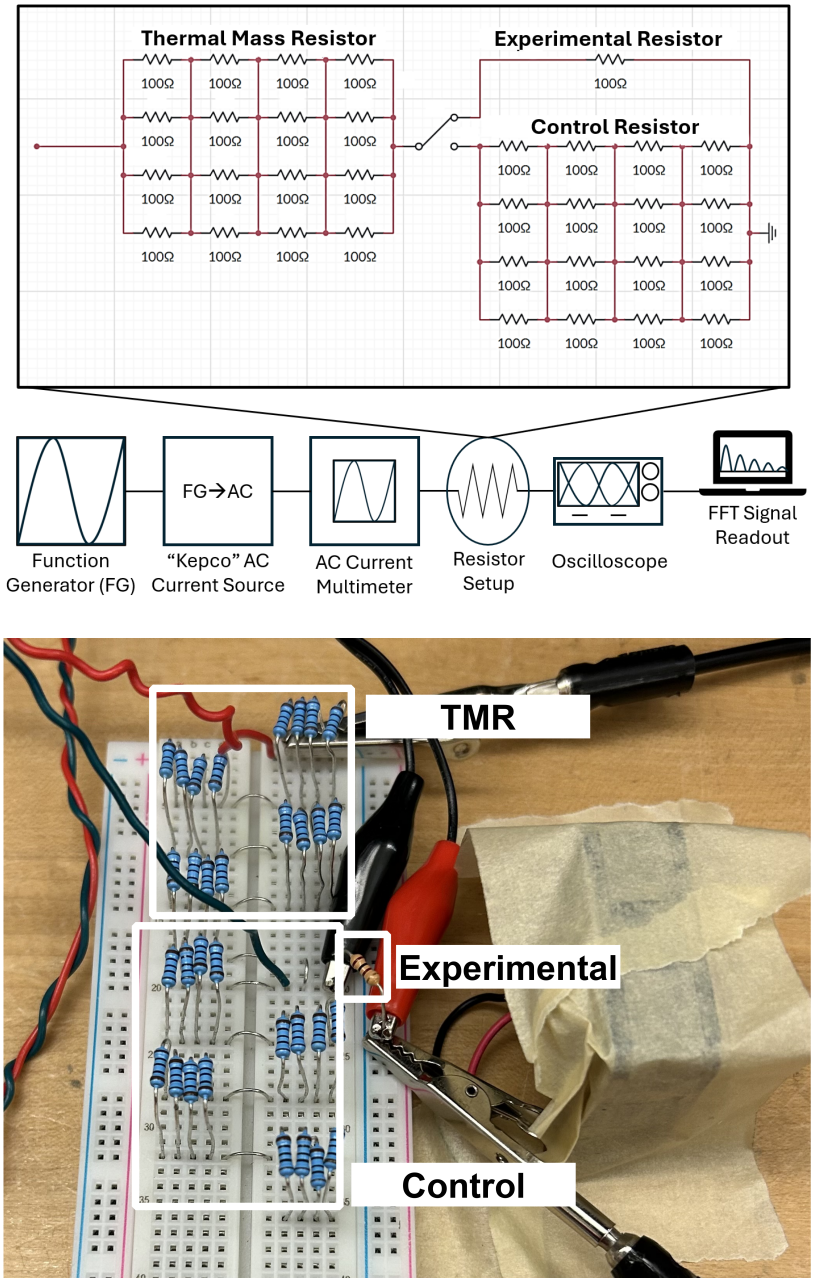

Figure 4.1: (Top) Our 3ω experimental setup for observing the 3ω effect in a 100 Ω through-hole resistor. The input frequency of 10 Hz passes through our resistor setup as a function of current. A custom program prompts an oscilloscope to perform a 15-count Fourier transform on the data and read out the $V_{3\omega}$ amplitude to a .csv file ten times before pausing to wait for the user to change the input current and repeat. (Bottom) Photograph of our 3ω experimental setup. The Experimental resistor is the brown through-hole resistor. The tape to the right is covering up a small fan we used to cool the Experimental resistor so we could reach higher AC current amplitudes without the resistor burning out.

The TMR is connected via a switch to either the control resistor setup or the single experimental resistor. It has a large thermal mass relative to the experimental resistor, ensuring that the voltage drop across it will not significantly change due to changes in temperature as a function of input AC current amplitude. This is best evidenced by the relationship between heating power and the square of the current as presented in Formula 2.3. Since the current is distributed among 16 different resistors, the heating power delivered to each through-hole resistor in the TMR is small compared to that applied to the experimental resistor.

This large thermal mass allows the TMR to serve as a control on both the experimental and control resistors. In control mode, a low-resistance switch connects the TMR to the control resistor, and the voltage drop across the TMR and control resistors are measured over a current range from 0.05 - 0.105 Amps. These signals are added together on an oscilloscope and then analyzed with the built-in Fourier transform function to translate the signal from time space to frequency space and extract the voltage amplitudes at the fundamental's first eight harmonics (10 Hz - 80 Hz in 10 Hz steps). After measuring the control signal, the low-resistance switch connects the TMR to the experimental resistor, and the experiment is performed again over the same current range of 0.05 - 0.105 Amps. Example scope readings showing the subtracted control and experimental signals in both time and frequency space is provided in Figures 4.2 and 4.4. Figures 4.3 and 4.5 also show the corresponding 3ω signals extracted from our SSTR experimental setup in control and experimental mode, respectively, as a function of input current. Finally, the experimental and control third harmonic voltage amplitudes (Figures 4.3 & 4.5) are subtracted from one another to observe the 3ω effect in Figure 4.6.

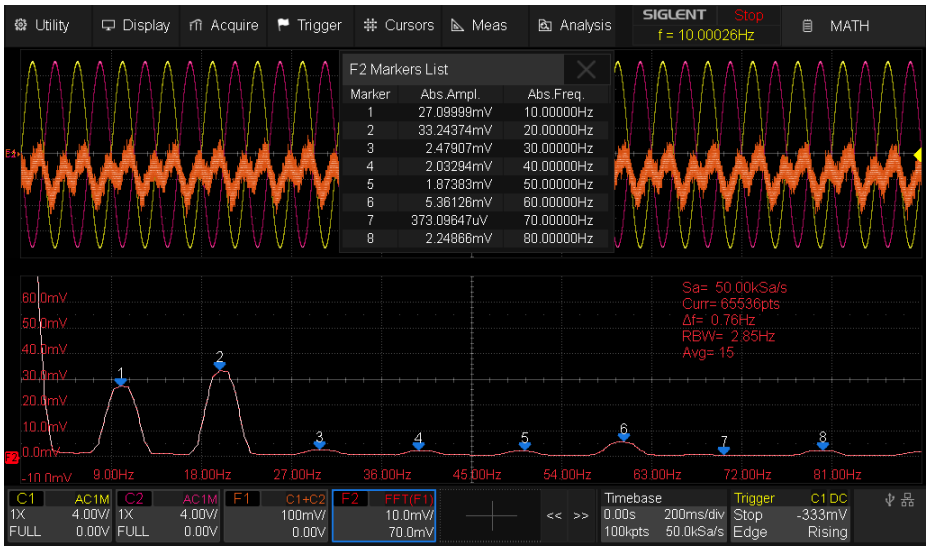

Figure 4.2: Scope reading of our $3\omega = 2\pi \cdot 30Hz$ control signal in "Control" mode; i.e., the TMR is connected to the Control resistor (Figure 4.1). Notice the TMR and Control resistors' signals are roughly equal and opposite. Their subtracted signal (orange), however, is quite irregular, indicating that other frequencies besides the fundamental dominate the signal. The Fourier transform function is extracting the voltage amplitudes of the subtracted signal's first eight harmonics (10 Hz - 80 Hz in 10 Hz steps).

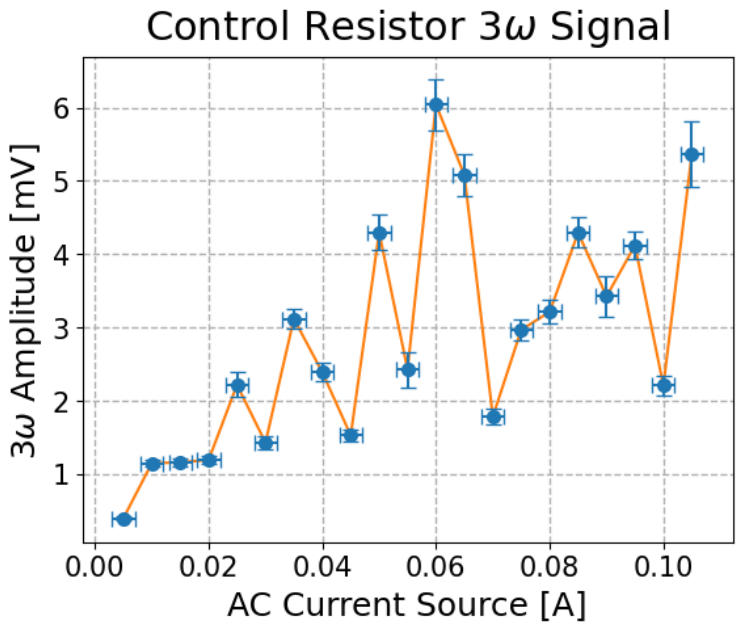

Figure 4.3: Our $3\omega = 2\pi \cdot 30Hz$ voltage amplitude measured as a function of input current in our 3ω setup in "Control" mode (Thermal Mass Resistor is connected to the Control Resistor).

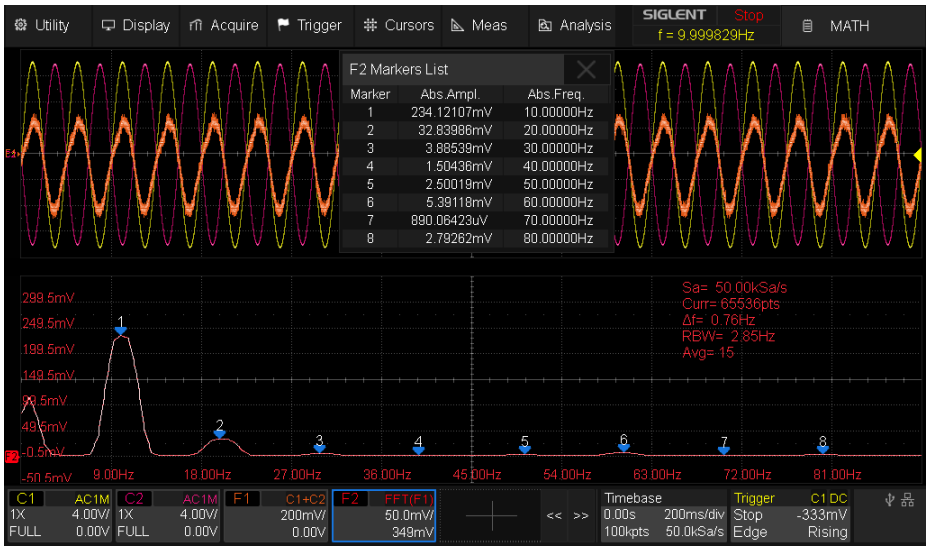

Figure 4.4: Scope reading of our $3\omega = 2\pi \cdot 30Hz$ experimental signal in "Experimental" mode; i.e., the TMR is connected to the Experimental resistor (Figure 4.1). Notice the TMR and Experimental resistors' signals are roughly equal and opposite. Their subtracted signal (orange) is almost sinusoidal, and the Fourier transform function is extracting the voltage amplitudes of the subtracted signal's first eight harmonics (10 Hz - 80 Hz in 10 Hz steps).

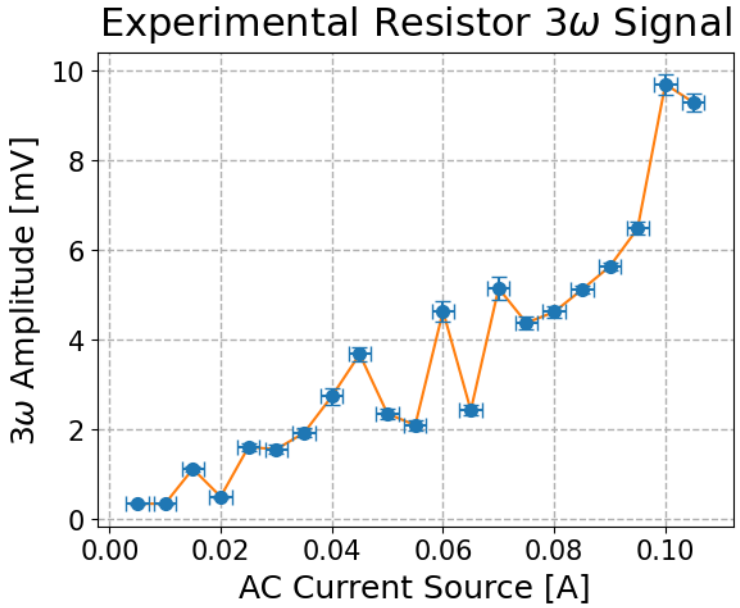

Figure 4.5: Our $3\omega = 2\pi \cdot 30Hz$ voltage amplitude measured as a function of input current in our 3ω setup in "Experimental" mode (Thermal Mass Resistor is connected to the Experimental Resistor) Notice how 3ω's relationship to current appears so much cleaner and smooth than with the control signal (Figure 4.3).

4.2 3ω Results

After measuring our 3ω voltage amplitude in our 3ω experimental setup in both "Control" and "Experimental" mode (Figures 4.3 & 4.5), we can subtract these two signals to observe the 3ω effect in Figure 4.6 as a function of current oscillating at a fundamental frequency of $f = 10Hz$. The raw data for the first eight harmonics ($1\omega - 8\omega$) is in Appendix B. As expected, the 3ω signal starts off small for small input current values and generally grows as current increases. We fit our data to a monomial cubic function (of the form: $V_{3\omega} = A \cdot I_0^3 + B$ where A and B are constants, $V_{3\omega}$ is the 3ω voltage component of the signal, and I_0 is the amplitude of the input AC current). We did this because Formula 2.15 suggests that $V_{3\omega}$ is proportional to the product of I_0 and $|\Delta T_{AC}|$ where $|\Delta T_{AC}|$ is the oscillatory temperature change component due to Joule-heating (Formula 2.6). Formulae 2.6 and 2.7 show $|\Delta T_{AC}|$ itself is proportional to I_0^2, leading us to believe that $V_{3\omega}$ will ultimately be proportional to the third power of the input current ($V_{3\omega} \propto I_0 \cdot |\Delta T_{AC}| \rightarrow V_{3\omega} \propto I_0^3$).

Ultimately, the monomial cubic does not match our collected data in Figure 4.6 very well ($R^2 = 0.532$). It is clear the data generally follows the upwards trend we would expect (Formula 2.15), but we do not understand why there is so much variation in our signal. Some other kind of unaccounted physical processes must be at play that our experimental setup (Figure 4.1) does not successfully control for. Figure 4.7 shows the results of a repeat experiment with similar variability profiles to back this claim up. However, although the data in Figure 4.6 and 4.7 do not perfectly align with our theoretical predictions, the monomial cubic still does have some predictive power. It can be used to determine the oscillatory temperature change induced in our Experimental resistor by our AC current signal at high currents where the monomial cubic better fits the data (Formula 2.15).

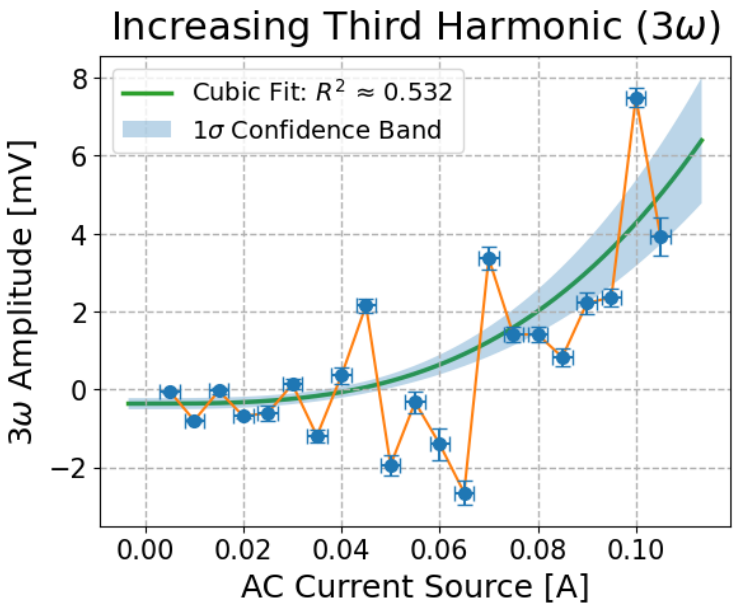

Figure 4.6: Observed 3ω Effect in our 3ω experimental setup's 100 Ω Experimental resistor (Figure 4.1). The trend roughly follows a monomial cubic trend with an R^2 value of approximatey 0.532. A 1σ confidence band is presented to show the standard deviation of the fit. Fit calculated by code in Appendix A.2.

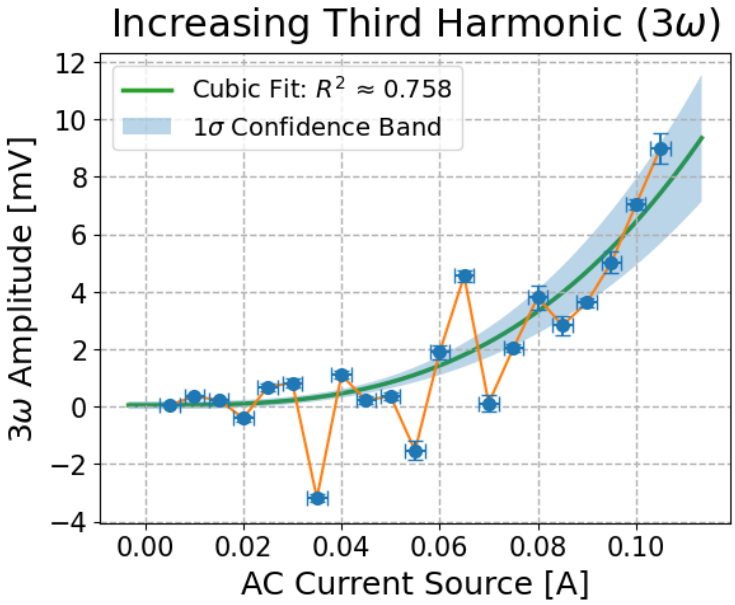

Figure 4.7: Follow-up experiment showing the 3ω Effect in our 3ω experimental setup's 100 Ω Experimental resistor (Figure 4.1). Like in Figure 4.6, the trend roughly follows a monomial cubic trend with a higher R^2 value of approximately 0.758. A 1σ confidence band is presented to show the standard deviation of the fit. Fit calculated by code in Appendix A.2.

To determine the oscillatory temperature change induced in our Experimental resistor by our AC current signal, we need to know the value of our Experimental resistor's temperature coefficient of resistance (α in Formula 2.15). Fortunately, this is quite easy! Formula 2.11 tells us that if we know the resistance of a material as a function of temperature, we can determine its value for α. We constructed a four-point resistance measurement setup inside of a lab oven and measured its resistance as a function of temperature. We heated the Experimental resistor to a peak temperature of 125°C before letting it cool all the way back down to room temperature while taking resistance measurements every 2.5°C. Figure 4.7 shows the data we collected and the linear best fit line's slope we calculated to determine the Experimental Resistor's temperature coefficient of resistance of $\alpha \approx -2.13{\cdot}10^{-4}K^{-1}$ (Appendix A.3). Plugging this value into Formula 2.15 along with our calculated maximum $V_{3\omega}$ from evaluating our cubic fit at 0.105 Amps yields a maximum oscillatory temperature change of $\Delta T_{AC} = 5.0K \pm 1.3K$ for Figure 4.6 and $\Delta T_{AC} = 7.4K \pm 1.8K$ for Figure 4.7.

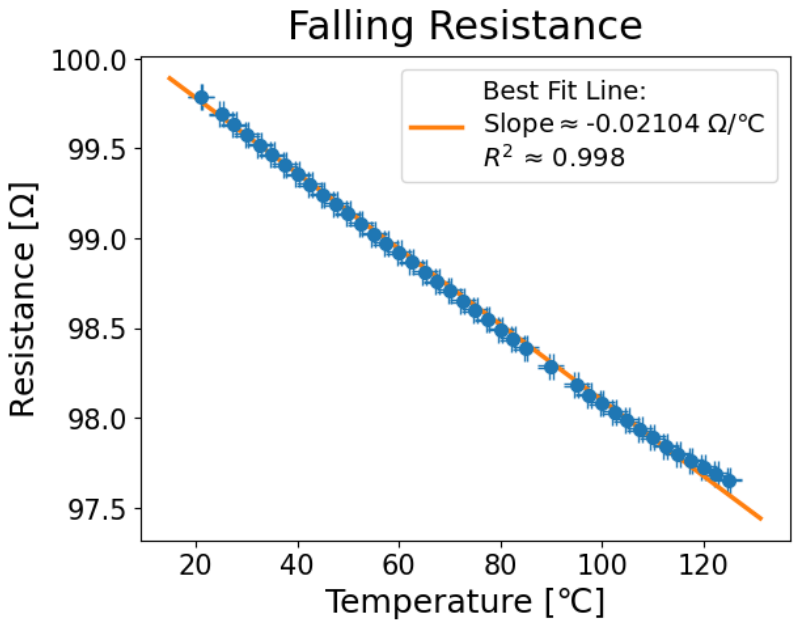

Figure 4.8: Resistance vs Temperature data gathered from a four-point resistance measurment setup inside a lab oven. The maximum temperature reached was 125°C, and measurements were taken as the oven cooled in 2.5°C steps.

Chapter 5

Future Work & Conclusions

5.1 Recommendation of Future Work

Already, work is underway to implement our steady-state thermoreflectance (SSTR) and 3ω method research in the William & Mary (W&M) Ultra-Cold Atomic, Molecular, and Optical (AMO) Physics Lab. Each has their own unique advantages that warrant future work. Two chip samples have been deposited by Brian Hurley (Virginia Commonwelath University) and Trevor Tingle and are the next steps for future work. For SSTR, small aluminum pads have been deposited onto an AlON substrate that will act as a substrate sample for these aluminum "mirrors" off which the thermoreflectance effect should be possible to observe. Future work seeing if it is possible to use our SSTR experimental setup to determine the thermal conductivity of a known substance like our chip sample's copper substrate will provide better insights into whether SSTR will be viable for the lab moving forward. If successful, the next step would be to perform the same experiment again, but on an AlON substrate. Figure 5.1 shows a micrograph of our copper chip's deposited aluminum pads.

Perhaps more promising than SSTR is our current work applying the 3ω method in the W&M AMO Physics Lab. Figure 5.2 shows a design document detailing a 4-point probe copper measurement setup deposited onto AlON for detecting the 3ω

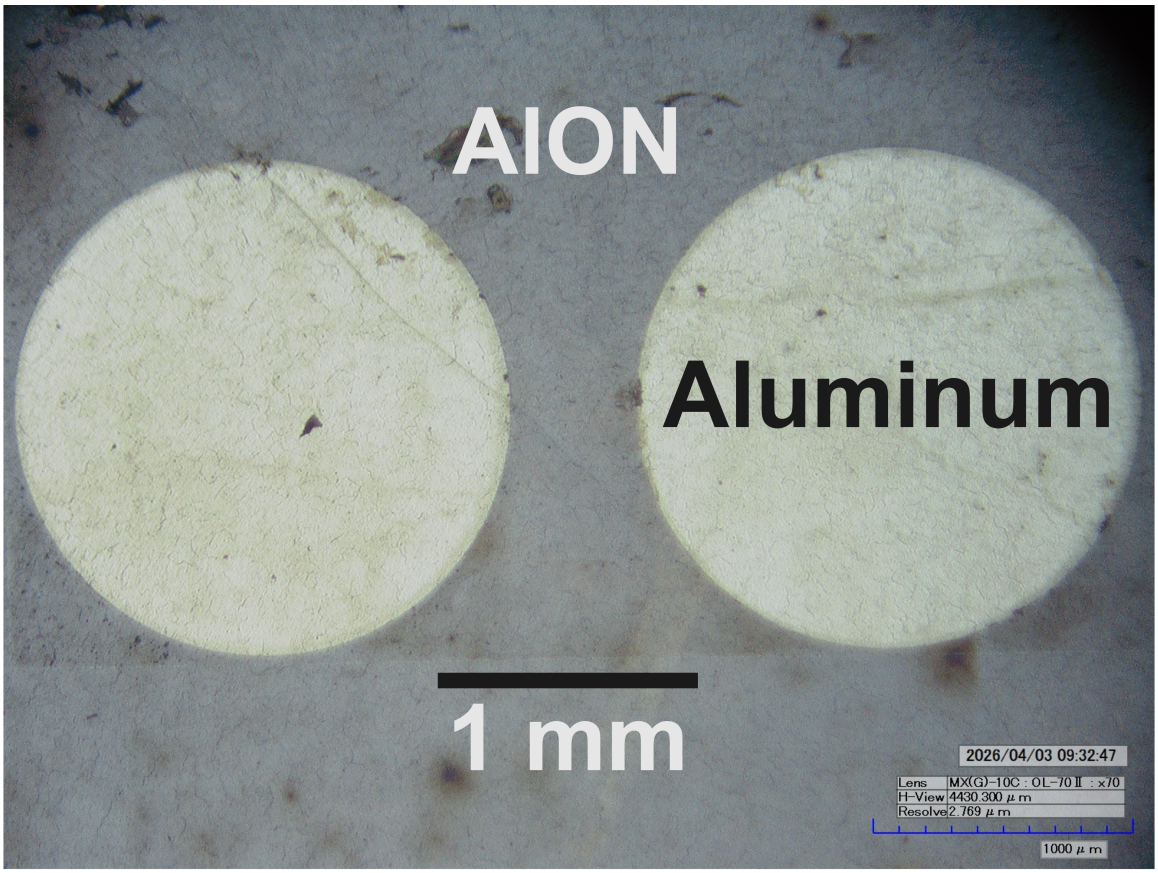

Figure 5.1: 70X Micrograph of deposited aluminum pads onto an AlON substrate. Future SSTR experiments would reflect a probe laser off of these to observe the thermoreflectance effect. The scale bar on the bottom right is 1000 μm. Thank you to Brian Hurley, Trevor Tingle, and Dr. Vitaliy Avrutin for their deposition work.

effect in a copper trace. This is exactly what we need to do to thermally characterize our microwave atom chips! Already this design has been implemented on a sapphire substrate with a known thermal conductivty. Similar to the future work that will need to be conducted on SSTR, if the 3ω effect can be demonstrated to determine the thermal conductivity of sapphire, it will likely also be viable for AlON. Figures 5.3 and 5.4 show 70X and 700X micrographs, respectively of our 4-point probe design deposited onto sapphire.

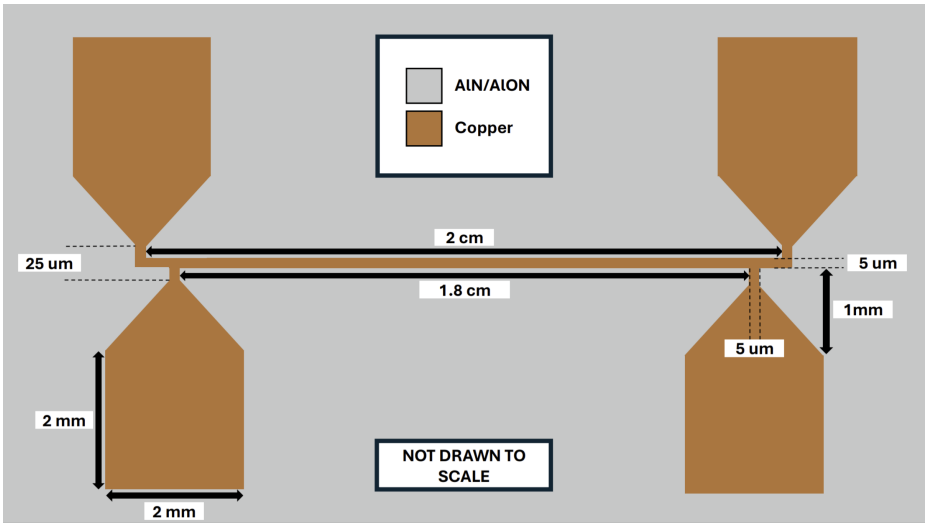

Figure 5.2: Design blueprint for 4-point probe copper measurement setup deposited onto AlON for detecting the 3ω effect in a copper trace.

5.2 Conclusion

Despite their importance to cold atom research, microwave atom chips are liable to be damaged or destroyed due to trace overheating, making them somewhat impractical [1, 2]. And broken chips must be replaced at great opportunity cost to the lab. AlON microwave heatsinks can help mitigate this problem, but their performance is governed by their thermal conductivities, which are not well-known [2]. This thesis investigates two methods (steady-state thermoreflectance and the 3ω method) for thermally characterizing thick-film AlON [3, 4, 5]. These methods are significantly more accessible and cost-effective than more common methodologies and can be easily implemented in our lab [12]. While SSTR poses some significant challenges to being implemented in the lab, our 3ω setup shows great promise for thermally characterizing thick-film AlON. Work is presently underway applying our 3ω experimental setup to thick-film sapphire as a final calibration step before determining thick-film AlON's thermal conductivity.

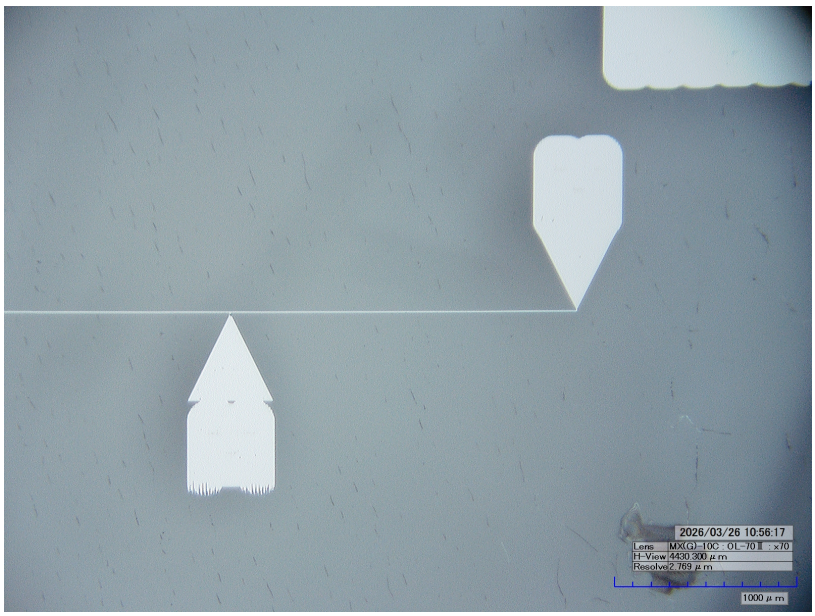

Figure 5.3: 70X Micrograph showing deposited platinum four-point probe pads on sapphire substrate. Thank you to Brian Hurley, Trevor Tingle, and Dr. Vitaliy Avrutin for their deposition work.

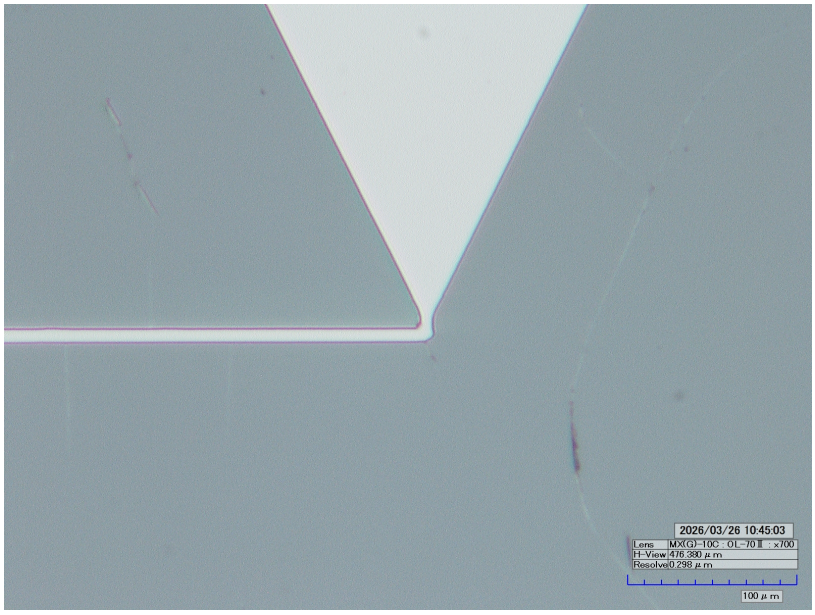

Figure 5.4: 700X Micrograph showing deposited platinum four-point probe pad wedge connecting the pad to the trace on a sapphire substrate. Thank you to Brian Hurley, Trevor Tingle, and Dr. Vitaliy Avrutin for their deposition work.

Appendix A

Data Analysis and Processing Code (Python)

A.1 Script for Analyzing Thermoreflectance Data (Python)

```
#import data analysis libraries

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import uncertainties as u

from IPython.display import HTML
from matplotlib.animation import FuncAnimation
from matplotlib.ticker import MaxNLocator
from scipy.optimize import curve_fit

#load raw data and define the experiment's time parameters

calibration_t = 10 #calibration period [minutes]
run_t = 41.5 #total experiment runtime [minutes]
sampling_t = 0.001 #sampling rate [seconds]

#read csv files
reference_csv = pd.read_csv('kelemen_1119_170335.csv',
                            encoding='latin1',
                            low_memory=False)
reflected_csv = pd.read_csv('DATA.csv',
                            encoding='latin1',
                            low_memory=False)
```

```
# load csv data
reference_raw = reference_csv.to_numpy()
reflected_raw = reflected_csv.to_numpy()
reference_signal = reference_raw[8:]
reflected_signal = reflected_raw[2:]

#concatenate nested data
reference_data = []
reflected_data = []

reference_errorcount = 0
reflected_errorcount = 0

for nested_list in reference_signal:
  try:
    reference_data.append(float(nested_list[0]))
  except ValueError:
    reference_errorcount+=1
    reference_data.append(0.0)

for nested_list in reflected_signal:
  try:
    reflected_data.append(float(nested_list[1]))
  except ValueError:
    reflected_errorcount+=1
    reflected_data.append(0.0)

#print("Data extracted and concatenated with",
#      reference_errorcount,
#      "errors for the reference data and",
#      reflected_errorcount,
#      "errors for the reflected data")

#data binning and max point extraction

reference_average = np.average(reference_data)
reflected_average = np.average(reflected_data)

max_reference_data = []
max_reflected_data = []
reference_temp = []
reflected_temp = []
```

```
  for data_point in reference_data:
  if data_point >= reference_average:
    reference_temp.append(data_point)
  else:
    if reference_temp:
      max_point = max(reference_temp)
      max_reference_data.append(max_point)
    reference_temp = []
if reference_temp:
    max_point = max(reference_temp)
    max_reference_data.append(max_point)

for data_point in reflected_data:
  if data_point >= reflected_average:
    reflected_temp.append(data_point)
  else:
    if reflected_temp:
      max_point = max(reflected_temp)
      max_reflected_data.append(max_point)
    reflected_temp = []
if reflected_temp:
    max_point = max(reflected_temp)
    max_reflected_data.append(max_point)

#compare data set legnths and slice to make equal length

reference_length = len(max_reference_data)
reflected_length = len(max_reflected_data)
delta = abs(reflected_length - reference_length)
abs_delta = abs(delta)

if delta < 0:
  max_reference_data = max_reference_data[:delta]
  #print("The set of max reference data was",
  #      abs_delta,
  #      "sample points longer than the max reflected data set.")
  #print("Excess data was sliced accordingly.")

if delta > 0:
  max_reflected_data = max_reflected_data[:-delta]
  #print("The set of max reflected data was",
  #      abs_delta,
  #      "sample points longer than the max reference data set.")
  #print("Excess was sliced accordingly.")

else:
  #print("The reference and reflected max data sets were equal lengths.")
  #print("No slicing implemented.")
  pass
```

```
#animating data linearity check for clarity

'''
Adapted from:
Source - https://stackoverflow.com/a
Posted by Andrey Sobolev, modified by community.
See post 'Timeline' for change history
Retrieved 2026-01-29, License - CC BY-SA 3.0

Also adapted code from Google Gemini to
implement data grouping feature and an empty frame 0
'''

grouping = 5000

fig = plt.figure()
plt.xlim(min(max_reflected_data)*0.999, max(max_reflected_data)*1.001)
plt.ylim(min(max_reference_data)*0.999, max(max_reference_data)*1.001)
plt.title("Data Linearity Check")
plt.xlabel("Reference Data [V]")
plt.ylabel("Reflected Data [V]")
graph, = plt.plot([], [], 'o')
def animate(i):
    graph.set_data(max_reflected_data[:(i)*grouping],
    max_reference_data[:(i)*grouping])
    return graph,

frame_count = len(max_reference_data) // grouping

ani = FuncAnimation(fig, animate, frames=frame_count, interval=100, blit=True)

plt.close()

HTML(ani.to_jshtml())
```

```
#linear regression to scale experimental data to control data

'''
Adapted from:
Source - https://stackoverflow.com/a
Posted by David Belohrad
Retrieved 2025-12-29, License - CC BY-SA 3.0
'''

average_time_between_binned_data = run_t*60/len(max_reference_data)

final_calibration_point=int(calibration_t*60/average_time_between_binned_data)
max_reference_calibration_data = max_reference_data[:final_calibration_point]
max_reflected_calibration_data = max_reflected_data[:final_calibration_point]

A = np.vstack([max_reflected_calibration_data,
               np.ones(len(max_reflected_calibration_data))]).T
m,b = np.linalg.lstsq(A, max_reference_calibration_data)[0]

max_reflected_scaled_data = [fitted_point for
                             fitted_point in
                             map(lambda y_reflected_point:
                             y_reflected_point*m+b, max_reflected_data)]

#print("Fitting parameters: m =",m,", b =",b)

#plot sclaed data
t = np.linspace(0,run_t,len(max_reference_data))
plt.scatter(t,max_reference_data, label = "Control Data")
plt.scatter(t,max_reflected_scaled_data, label = "Scaled Reflected Data")
plt.title("Reference Data and Fitted Reflected Data")
plt.xlabel("Approximate Time [mins]")
plt.ylabel("PD Signal (V)")
plt.ylim(min(max_reference_data)*0.99, max(max_reference_data)*1.01)
plt.legend(loc="lower right")
```

```
#raw data plotting

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']
fig = plt.gcf()
fig.set_figheight(fig.get_figheight() * 5/3)
gs = fig.add_gridspec(4, 1, height_ratios=[1, 1, 3, 3])

plt.subplot(gs[0, 0])
plt.scatter(t,max_reference_data)
plt.title("Control Data", fontsize = 15, pad=10)
plt.ylim(min(max_reference_data)*0.99, max(max_reference_data)*1.01)
plt.gca().yaxis.set_major_locator(MaxNLocator(nbins=3, min_n_ticks=3))
plt.grid(linestyle = '--', linewidth = 1.0,alpha=0.7)

plt.subplot(gs[1, 0])
plt.scatter(t,max_reflected_data,c="tab:orange")
plt.title("Experimental Data", fontsize = 15, pad=10)
plt.ylim(min(max_reflected_data)*0.99, max(max_reflected_data)*1.01)
plt.gca().yaxis.set_major_locator(MaxNLocator(nbins=3, min_n_ticks=3))
plt.grid(linestyle = '--', linewidth = 1.0,alpha=0.7)

plt.subplot(gs[2, 0])
plt.scatter(t,max_reference_data)
plt.scatter(t,max_reflected_data)
plt.title("Overlayed Control and Experimental Data", fontsize = 15, pad=10)
plt.ylim(min(max_reflected_data)*0.96, max(max_reference_data)*1.04)
plt.gca().yaxis.set_major_locator(MaxNLocator(nbins=3, min_n_ticks=3))
plt.grid(linestyle = '--', linewidth = 1.0,alpha=0.7)

plt.subplot(gs[3, 0])
plt.scatter(t,max_reference_data, label = "Control Data")
plt.scatter(t,max_reflected_scaled_data, label = "Scaled Reflected Data")
plt.title("Overlayed Control and Scaled Experimental Data",fontsize = 15,pad=10)
plt.xlabel("Approximate Time [mins]", fontsize = 30)
plt.ylim(min(max_reference_data)*0.96, max(max_reference_data)*1.04)
plt.grid(linestyle = '--', linewidth = 1.0,alpha=0.7)

plt.legend(loc="lower right")

fig.supylabel("Photodiode Signal [V]", fontsize = 30)

plt.tight_layout()
plt.show()
```

```
#calculate the normalized change in reflecticity
#and fit the temperature rise data

#normalized change in reflecticity
y_normalized = [(b-a)/c for a,b,c in zip(max_reference_data,
max_reflected_scaled_data,max_reflected_scaled_data)]

#find the index where the temperature rise starts
temp = 1
numb = 0
for i in t:
    diff = abs(i - calibration_t)
    if diff < temp:
        temp = diff
        numb = i
    else:
        pass
#print(numb)

times = [i for i in t]
ten_min_index = times.index(numb)

#find the index where the temperature rise stops
max_temp_index = y_normalized.index(max(y_normalized))
#print(ten_min_index)
#print(max_temp_index)

#So, the data now ranges from ten_min_index -> max_temp_index
t_fitting = times[ten_min_index:max_temp_index+1]
y_fitting = y_normalized[ten_min_index:max_temp_index+1]

#use boostrap method to estimate errorbars
def linear(x,m,b):
    return m*x+b

params, cov_matrix = curve_fit(linear,
                               t_fitting,
                               y_fitting,
                               sigma=0.001,
                               absolute_sigma=True)

residuals = np.array(y_fitting) - linear(np.array(t_fitting),
                                         params[0],
                                         params[1])
stdev = np.std(residuals)
#print(stdev)
```

```
params, cov_matrix = curve_fit(linear,
                               t_fitting,
                               y_fitting,
                               sigma=stdev,
                               absolute_sigma=True)

perr = np.sqrt(np.diag(cov_matrix))

#All of those calculations above help to make the linear fit to the
#temperature rise, but the linear fit does not start at
#delR/R = 0, so to calculate the thermoreflectance coefficient,
#I need to take the max from the fit and subtract 0
#(my starting value).

t_zero = times[0:ten_min_index+1]
y_zero = y_normalized[0:ten_min_index+1]

zero_params, zero_cov_matrix = curve_fit(linear,
                               t_zero,
                               y_zero,
                               sigma=0.001,
                               absolute_sigma=True)

residuals_zero = np.array(y_zero) - linear(np.array(t_zero),
zero_params[0],
zero_params[1])
stdev_zero = np.std(residuals_zero)
#print(stdev)

zero_params, zero_cov_matrix = curve_fit(linear,
                               t_zero,
                               y_zero,
                               sigma=stdev_zero,
                               absolute_sigma=True)

perr_zero = np.sqrt(np.diag(zero_cov_matrix))

# calculate the thermoreflectance coefficient from the temperature rise data

def bootstrap_linear(x,m,em,b,eb):
    return u.ufloat(m,em)*x+u.ufloat(b,eb)
```

```
coefficient = (bootstrap_linear(max(t_fitting),
                                params[0],
                                perr[0],
                                params[1],
                                perr[1]) - bootstrap_linear(max(t_zero),
                                                            zero_params[0],
                                                            perr_zero[0],
                                                            zero_params[1],
                                                            perr_zero[1]))/104

print(coefficient)

#This method yields the closer result
But the difference between this value And a second method
where we take the minimum value of the linear fit to the temp rise
(Figure 3.6) as our lowest reflectivity gives us an approximate
errorbar of 0.62

# plot the SSTR signal with best fit

y_fit = np.array(bootstrap_linear(np.array(t_fitting),
                                  params[0],
                                  perr[0],
                                  params[1],
                                  perr[1]))

y_fit_nominal = np.array([i.n for i in y_fit])
y_fit_err = np.array([i.s for i in y_fit])

#calculate R^2
average = np.average(y_fitting)
SS_res = sum((i-j)**2 for i,j in zip(y_fitting,y_fit_nominal))
SS_tot = sum((i-average)**2 for i in y_fitting)
R_square = 1 - (SS_res/SS_tot)
print(R_square)

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']
plt.scatter(t,y_normalized,s=0.5)
plt.title('Steady-State Thermoreflectance', fontsize = 22, pad=10)
plt.xlabel('Time Elapsed [mins]', fontsize = 14)
plt.ylabel('Normalized Change in Reflectivity [a.u.]', fontsize = 14)
a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)
plt.grid(linestyle = '--', linewidth = 1.0)
plt.ylim(-0.006,0.008)

plt.plot(
    t_fitting,
    y_fit_nominal,
    linewidth=3,
    color='C1',
    label=f'Reflectivity Rise:\n$R^2$ \u2248 {np.round(R_square,4)}')

plt.legend(fontsize = 14, loc='upper left')
```

A.2 Script for Analyzing 3ω Data (Python)

```python
# import data libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import uncertainties as u
from scipy.odr import ODR, Model, RealData

# configuration
control_file_name = 'control_04_04_10hz.csv'
experimental_file_name = 'experimental_04_04_10hz.csv'
measurement_count = 10

# resistor measurements [Ohms]
thermal_mass_resistor = 99.83925
experimental_resistor = 99.78385
control_resistor = 99.8155

# read results and initialize control and experimental DataFrames
df_control = pd.read_csv(control_file_name)
df_experimental = pd.read_csv(experimental_file_name)

#print(df_control.to_string())
#print(df_experimental.to_string())

# redefine the DataFrames and skip the first two "Title" rows
df_control = pd.read_csv(control_file_name, skiprows = 2, header = None)
df_experimental = pd.read_csv(experimental_file_name, skiprows = 2,
header = None)

#print(df_control.to_string())
#print(df_experimental.to_string())

# read CC sources from DataFrame (doesn't matter which one)
cc_sources = df_control[df_control.columns[0]].values.tolist()
#print(cc_sources)
```

```python
# construct control and experimental dictionaries that store the measured
# and calculated values (AI-generated)

# data is retrievable as follows: dictionary_name[cc_index = 0-30]
# [harmonic = 1-8][0,1,2 for values, average, and sem, respectively]
# Example: experimental[3][5][2]
def build_measurement_dict(df, cc_sources, measurement_count):
    data = {}

    for i in range(len(cc_sources)):
        data[i] = {}

        for h in range(1, 9):
            values=[df.values[i][k] for k in range(h,
            8*measurement_count+h -1,
            8)]
            avg = np.mean(values)
            sem = np.std(values, ddof=1)/np.sqrt(measurement_count-1)

            data[i][h] = [values, avg, sem]

    return data

control = build_measurement_dict(df_control, cc_sources, measurement_count)
experimental = build_measurement_dict(df_experimental, cc_sources,
measurement_count)

# extract harmonic data (AI-generated)
control_w = [None] + [
    [u.ufloat(control[i][w][1],
    control[i][w][2]) for i in range(len(cc_sources))]
    for w in range(1, 9)
]

experimental_w = [None] + [
    [u.ufloat(experimental[i][w][1],
    experimental[i][w][2]) for i in range(len(cc_sources))]
    for w in range(1, 9)
]
```

```
# plot the 3w control signal and convert from Vrms to mV
plt.errorbar(cc_sources, [i.n*1000*2*np.sqrt(2) for i in control_w[3]],
             xerr = 0.002, yerr = [i.s*1000*2*np.sqrt(2) for i in control_w[3]]
             fmt ='o', capsize = 4, markersize = 7, zorder=10)

plt.plot(cc_sources, [i.n*1000*2*np.sqrt(2) for i in control_w[3]])

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']

plt.grid(linestyle = '--', linewidth = 1.0)
plt.title('Control Resistor 3$\omega$ Signal', fontsize = 22, pad=10)
plt.xlabel('AC Current Source [A]', fontsize = 18)
plt.ylabel('3$\omega$ Amplitude [mV]', fontsize = 18)
a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)

# plot the 3w experimental signal and convert from Vrms to mV
plt.errorbar(cc_sources, [i.n*1000*2*np.sqrt(2) for i in experimental_w[3]],
             xerr = 0.002,
             yerr = [i.s*1000*2*np.sqrt(2) for i in experimental_w[3]],
             fmt ='o', capsize = 4, markersize = 7, zorder=10)

plt.plot(cc_sources, [i.n*1000*2*np.sqrt(2) for i in experimental_w[3]])

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']

plt.grid(linestyle = '--', linewidth = 1.0)
plt.title('Control Resistor 3$\omega$ Signal', fontsize = 22, pad=10)
plt.xlabel('AC Current Source [A]', fontsize = 18)
plt.ylabel('3$\omega$ Amplitude [mV]', fontsize = 18)
a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)

# create separate lists for experimental and control values and errors
experimental_3w_nominal_vals = [i.n for i in experimental_w[3]]
experimental_3w_sem_vals = [i.s for i in experimental_w[3]]
control_3w_nominal_vals = [i.n for i in control_w[3]]
control_3w_sem_vals = [i.s for i in control_w[3]]

# subtract the two signals
difference_3w_signal = [u.ufloat(a,b)*1000*2*np.sqrt(2) -
                        u.ufloat(c,d)*1000*2*np.sqrt(2) for
                        a,b,c,d in zip(experimental_3w_nominal_vals,
                        experimental_3w_sem_vals,
                        control_3w_nominal_vals,
                        control_3w_sem_vals)]

#create separate lists for the difference values and errors
difference_3w_signal_nominal = [i.n for i in difference_3w_signal]
difference_3w_signal_error = [i.s for i in difference_3w_signal]
```

```
#plot the subtracted 3w signal
plt.errorbar(cc_sources, difference_3w_signal_nominal, xerr = 0.002,
yerr = difference_3w_signal_error,
fmt ='o', capsize = 4, markersize = 7, zorder=10)
plt.plot(cc_sources, difference_3w_signal_nominal)
plt.grid(linestyle = '--', linewidth = 1.0)

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']

plt.title('Increasing Third Harmonic (3$\omega$)', fontsize = 22, pad=10)
plt.xlabel('AC Current Source [A]', fontsize = 18)
plt.ylabel('3$\omega$ Amplitude [mV]', fontsize = 18)
a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)

# define a polynomial equation to calculate a function of best fit
def poly_fit(P, x):
    a,b = P
    return a*x**3 + b

difference_data = RealData(cc_sources, difference_3w_signal_nominal,
                           sx = 0.002, sy = difference_3w_signal_error)
model = Model(poly_fit)
P_guess = [4500,-0.25]

difference_fitter = ODR(difference_data,model,beta0=P_guess)
difference_fit = difference_fitter.run()

# plot the best fit line with confidence interval
plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']
plt.errorbar(cc_sources, difference_3w_signal_nominal,
             xerr = 0.002, yerr = difference_3w_signal_error,
             fmt ='o', capsize = 4, markersize = 7, zorder = 10)
plt.plot(cc_sources, difference_3w_signal_nominal, zorder = 2)
plt.grid(linestyle = '--', linewidth = 1.0)

plt.title('Increasing Third Harmonic (3$\omega$)', fontsize = 22, pad=10)
plt.xlabel('AC Current Source [A]', fontsize = 18)
plt.ylabel('3$\omega$ Amplitude [mV]', fontsize = 18)

difference_expected = [poly_fit(difference_fit.beta, i) for i in cc_sources]

SS_res = sum((i-j)**2 for i,j in zip(difference_3w_signal_nominal,
                                     difference_expected))
SS_tot = sum((i-np.average(difference_3w_signal_nominal))**2 for i in
difference_3w_signal_nominal)
R_square = 1 - (SS_res/SS_tot)
```

```
a = np.round(difference_fit.beta[0],5)
b = np.round(difference_fit.beta[1],5)

overshoot = 0.08*max(cc_sources)
x_range = np.linspace(min(cc_sources)-overshoot,
max(cc_sources)+overshoot, 1000)
P = a,b
composite_y_range = poly_fit(P, x_range)

plt.plot(x_range,composite_y_range, linewidth = 2.5, zorder = 0,
 label = (f'Cubic Fit: $R^2$ \u2248 {np.round(R_square,3)}'))

cov_matrix = difference_fit.cov_beta * difference_fit.res_var

##### AI-GENERATED but also worked out by hand by the student #####
# https://en.wikipedia.org/wiki/Propagation_of_uncertainty?utm
sigma = np.sqrt(
    x_range**6 * cov_matrix[0,0]
    + cov_matrix[1,1]
    + 2 * x_range**3 * cov_matrix[0,1]
)
#########################################################################

z=1
plt.fill_between(x_range, composite_y_range - z*sigma,
 composite_y_range + z*sigma, alpha=0.3,
  label = f'{z}$\\sigma$ Confidence Band')
plt.legend(fontsize = 14)

a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)
#calculate max oscillatory temp change
#caluclate error
sigma_max = np.sqrt(
    0.105**6 * cov_matrix[0,0]
    + cov_matrix[1,1]
    + 2 * 0.105**3 * cov_matrix[0,1]
)
#print value
max_temp_change = u.ufloat(poly_fit(P, 0.105),sigma_max)
print(max_temp_change)
```

A.3 Script for Calculating the Experimental Resistor's Temperature Coefficient of Resistance (Python)

```python
#import data analysis libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import uncertainties as u
# Start at room temperature and increase by roughly 10C increments
# until 100C + room temperature is reached.
temperature_resist_measurements = {}
# form: dict[index] = [temperature_reading, temperature_err,
#                      resistance_reading, resistance_err]
temperature_resist_measurements[0] = [21.1, 0.15, 99.78385, 0.0002]
temperature_resist_measurements[1] = [125.0, 0.35, 97.6540, 0.0055]
temperature_resist_measurements[2] = [122.5, 0.35, 97.6880, 0.0055]
temperature_resist_measurements[3] = [120.0, 0.35, 97.7235, 0.0055]
temperature_resist_measurements[4] = [117.5, 0.35, 97.7645, 0.0055]
temperature_resist_measurements[5] = [115.0, 0.35, 97.8008, 0.0055]
temperature_resist_measurements[6] = [112.5, 0.35, 97.8446, 0.0055]
temperature_resist_measurements[7] = [110.0, 0.35, 97.8910, 0.0055]
temperature_resist_measurements[8] = [107.5, 0.35, 97.9377, 0.0055]
temperature_resist_measurements[9] = [105.0, 0.35, 97.9855, 0.0055]
temperature_resist_measurements[10] = [102.5, 0.35, 98.0320, 0.0055]
temperature_resist_measurements[11] = [100.0, 0.35, 98.0798, 0.0055]
temperature_resist_measurements[12] = [97.5, 0.35, 98.1286, 0.0055]
temperature_resist_measurements[13] = [95.0, 0.35, 98.1828, 0.0055]
#temperature_resist_measurements[15] = [92.5, 0.35, , 0.0055] #missed this one
temperature_resist_measurements[14] = [90.0, 0.35, 98.2850, 0.0055]
#temperature_resist_measurements[17] = [87.5, 0.35, 98.3360, 0.0055]
#I didnt remember it well. #The value is a guess
temperature_resist_measurements[15] = [85.0, 0.35, 98.3879, 0.0055]
temperature_resist_measurements[16] = [82.5, 0.35, 98.4400, 0.0055]
temperature_resist_measurements[17] = [80.0, 0.35, 98.4933, 0.0055]
temperature_resist_measurements[18] = [77.5, 0.35, 98.5460, 0.0055]
temperature_resist_measurements[19] = [75.0, 0.35, 98.5990, 0.0055]
temperature_resist_measurements[20] = [72.5, 0.35, 98.6500, 0.0055]
temperature_resist_measurements[21] = [70.0, 0.35, 98.7068, 0.0055]
temperature_resist_measurements[22] = [67.5, 0.35, 98.7560, 0.0055]
temperature_resist_measurements[23] = [65.0, 0.35, 98.8122, 0.0055]
temperature_resist_measurements[24] = [62.5, 0.35, 98.8660, 0.0055]
temperature_resist_measurements[25] = [60.0, 0.35, 98.9199, 0.0055]
temperature_resist_measurements[26] = [57.5, 0.35, 98.9720, 0.0055]
temperature_resist_measurements[27] = [55.0, 0.35, 99.0240, 0.0055]
temperature_resist_measurements[28] = [52.5, 0.35, 99.0802, 0.0055]
temperature_resist_measurements[29] = [50.0, 0.35, 99.1350, 0.0055]
temperature_resist_measurements[30] = [47.5, 0.35, 99.1906, 0.0055]
```

```
temperature_resist_measurements[31] = [45.0, 0.35, 99.2439, 0.0055]
temperature_resist_measurements[32] = [42.5, 0.35, 99.2988, 0.0055]
temperature_resist_measurements[33] = [40.0, 0.35, 99.3537, 0.0055]
temperature_resist_measurements[34] = [37.5, 0.35, 99.4078, 0.0055]
temperature_resist_measurements[35] = [35.0, 0.35, 99.4632, 0.0055]
temperature_resist_measurements[36] = [32.5, 0.35, 99.5196, 0.0055]
temperature_resist_measurements[37] = [30.0, 0.35, 99.5748, 0.0055]
temperature_resist_measurements[38] = [27.5, 0.35, 99.6316, 0.0055]
temperature_resist_measurements[39] = [25.0, 0.35, 99.6894, 0.0055]

#calculate initial linear fit
def linear_fit(x, m, b):
    return m * x + b
x = [temperature_resist_measurements[i][0]
      for i in range(len(temperature_resist_measurements))]
y = [temperature_resist_measurements[i][2]
      for i in range(len(temperature_resist_measurements))]
x_err = [temperature_resist_measurements[i][1]
          for i in range(len(temperature_resist_measurements))]
y_err = [temperature_resist_measurements[i][3]
          for i in range(len(temperature_resist_measurements))]

temp_params, temp_cov_matrix = curve_fit(linear_fit, x, y, sigma = y_err,
                                         absolute_sigma = True)
temp_m = temp_params[0]
#print(temp_m)
#calculate composite_y_err by sum root squaring vertical/horizontal error

horizontal_y_err = []
for i in x_err:
    horizontal_y_err.append(i * temp_m)
#print(horizontal_y_err)

composite_y_err = []
for i in range(len(temperature_resist_measurements)):
    composite_y_err.append(np.sqrt((temperature_resist_measurements[i][3]) **
                                   2 + (horizontal_y_err[i] ** 2)))
#print(composite_y_err)

#perform a new linear fit and extract fit parameters

composite_params, composite_cov_matrix = curve_fit(linear_fit, x, y,
                                                   sigma = composite_y_err,
                                                   absolute_sigma = True)

composite_m, composite_b = composite_params
#print(composite_m)
#print(composite_b)

composite_m_err, composite_b_err = np.sqrt(np.diag(composite_cov_matrix))
#print(composite_m_err)
#print(composite_b_err)
```

```
#calculate R^2

composite_y_range = [linear_fit(i, composite_m, composite_b) for i in x]

SS_res = sum((i-j)**2 for i,j in zip(y,composite_y_range))
SS_tot = sum((i-np.average(y))**2 for i in y)
R_square = 1 - (SS_res/SS_tot)
print(R_square)

# plot resistance vs temp with best fit

overshoot = 0.05*max(x)
x_range = np.linspace(min(x)-overshoot, max(x)+overshoot, 1000)
composite_y_range = linear_fit(x_range, composite_m, composite_b)

plt.rcParams['font.serif'] = ['Times New Roman', 'DejaVu Serif']

plt.errorbar(x, y, xerr = x_err, yerr = y_err, fmt = 'o', capsize = 7.5,
             markersize = 7)
plt.title('Temperature-Dependent Resistance', fontsize = 22, pad=10)
plt.xlabel(f'Temperature [\u2103]', fontsize = 18)
plt.ylabel('Resistance [$\\Omega$]', fontsize = 18)
#plt.plot(x_range, linear_fit(x_range,composite_m,composite_b),
          label = f'Line of Best Fit:
          \nSlope$\\approx\\alpha_R(R)\\approx${np.round(composite_m,4)}')
plt.plot(x_range, linear_fit(x_range,composite_m,composite_b),
        linewidth = 2.5, label = f'Best Fit Line:
       \nSlope$\\approx${np.round(composite_m,5)} $\\Omega$/\u2103  n$R^2$
       \u2248 {np.round(R_square,3)}') # $±$ {np.round(composite_m_err,4)}'

a = 15
plt.xticks(fontsize=a)
plt.yticks(fontsize=a)

plt.legend(fontsize = 14)

# calculate average value for alpha
vals=[-0.02104/i for i in y]
print(np.average(vals))

```

Appendix B

Subtracted Average Harmonic Data for 3ω Experimental Setup

B.1 Subtracted Average 1ω Data

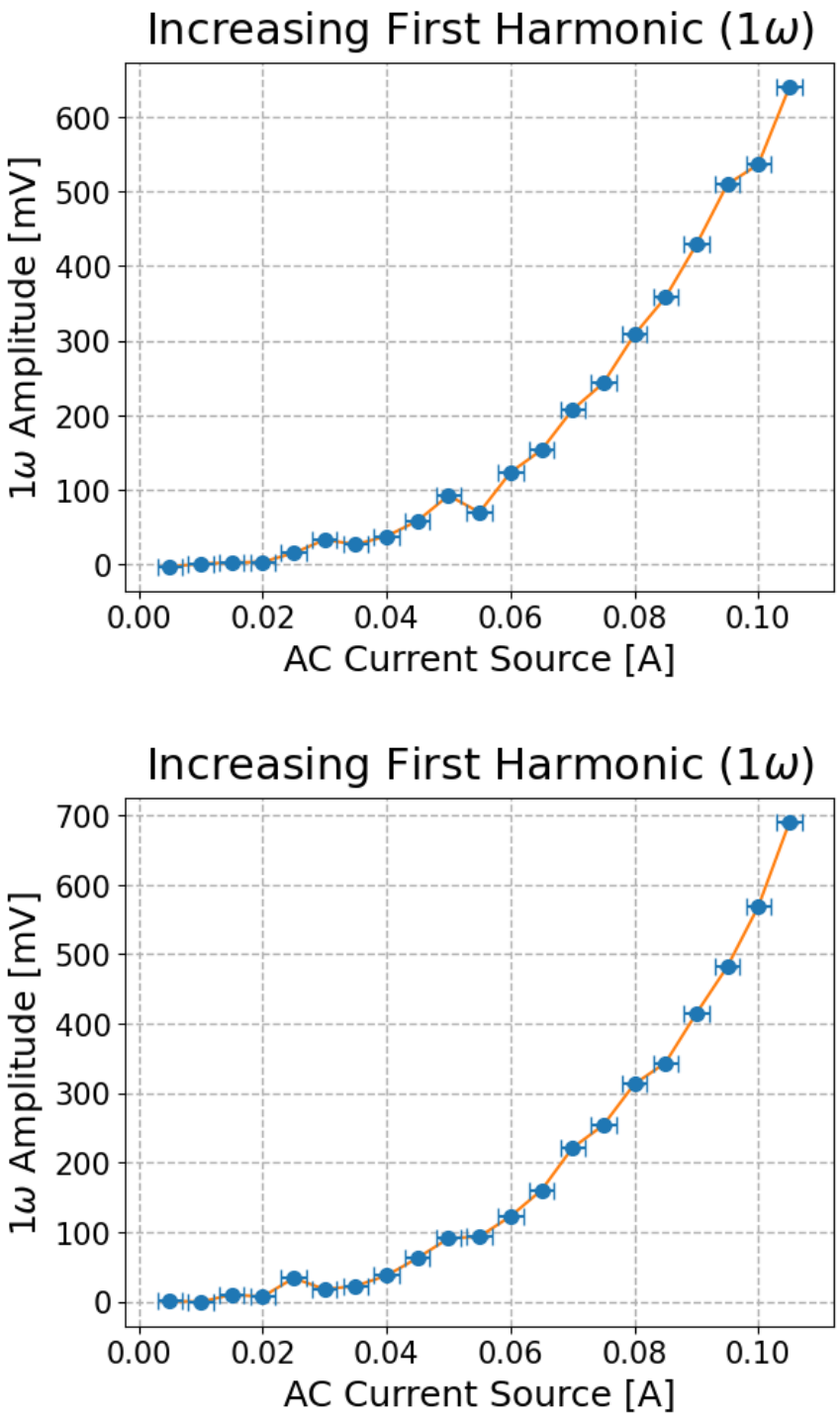

Figure B.1: The difference between our average experimental and control $1\omega = 2\pi \cdot 10$ Hz signals from both experiments.

B.2 Subtracted Average 2ω Data

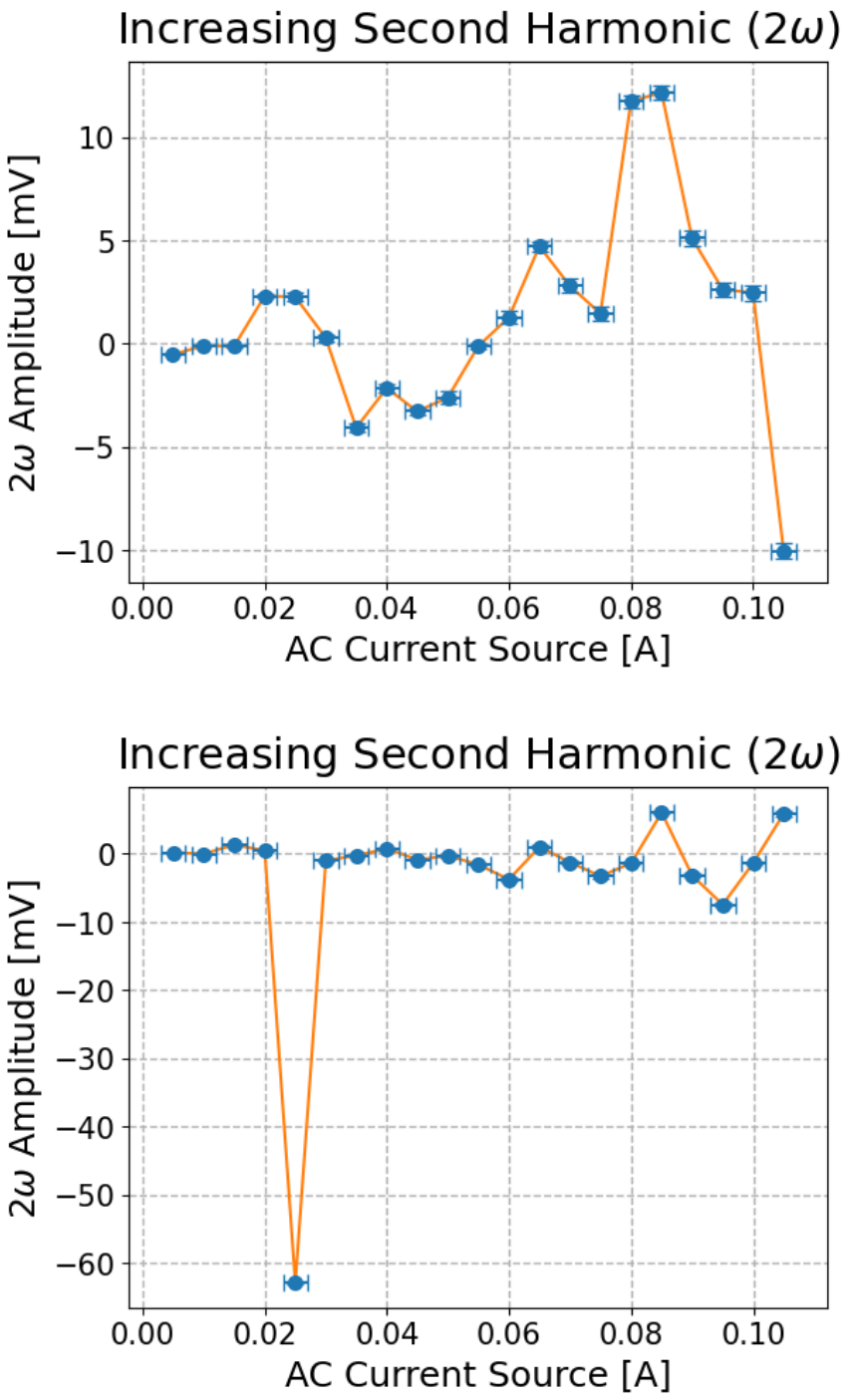

Figure B.2: The difference between our average experimental and control $2\omega = 2\pi \cdot 20$ Hz signals from both experiments.

B.3 Subtracted Average 3ω Data

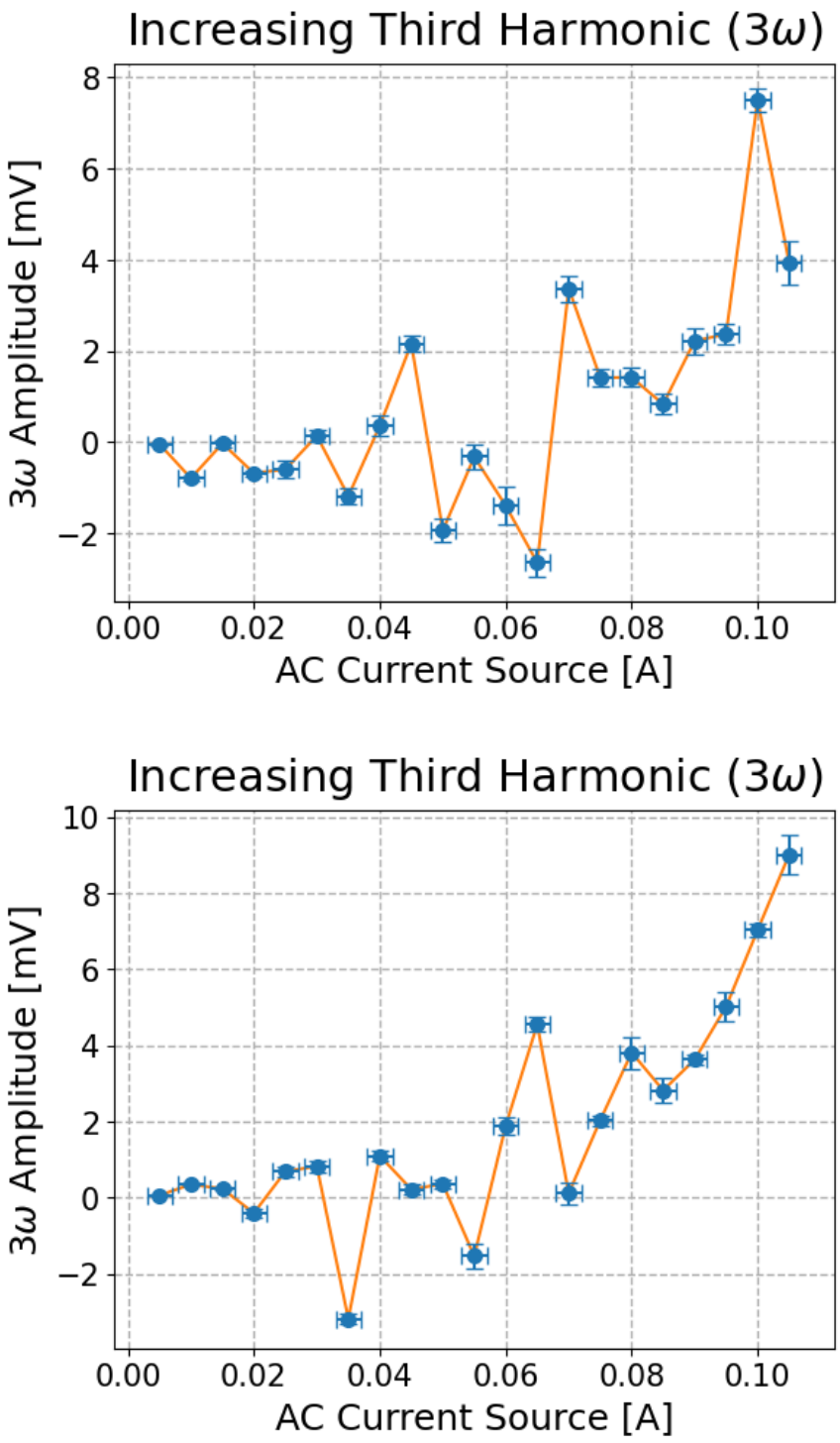

Figure B.3: The difference between our average experimental and control $3\omega = 2\pi \cdot 30$ Hz signals from both experiments.

B.4 Subtracted Average 4ω Data

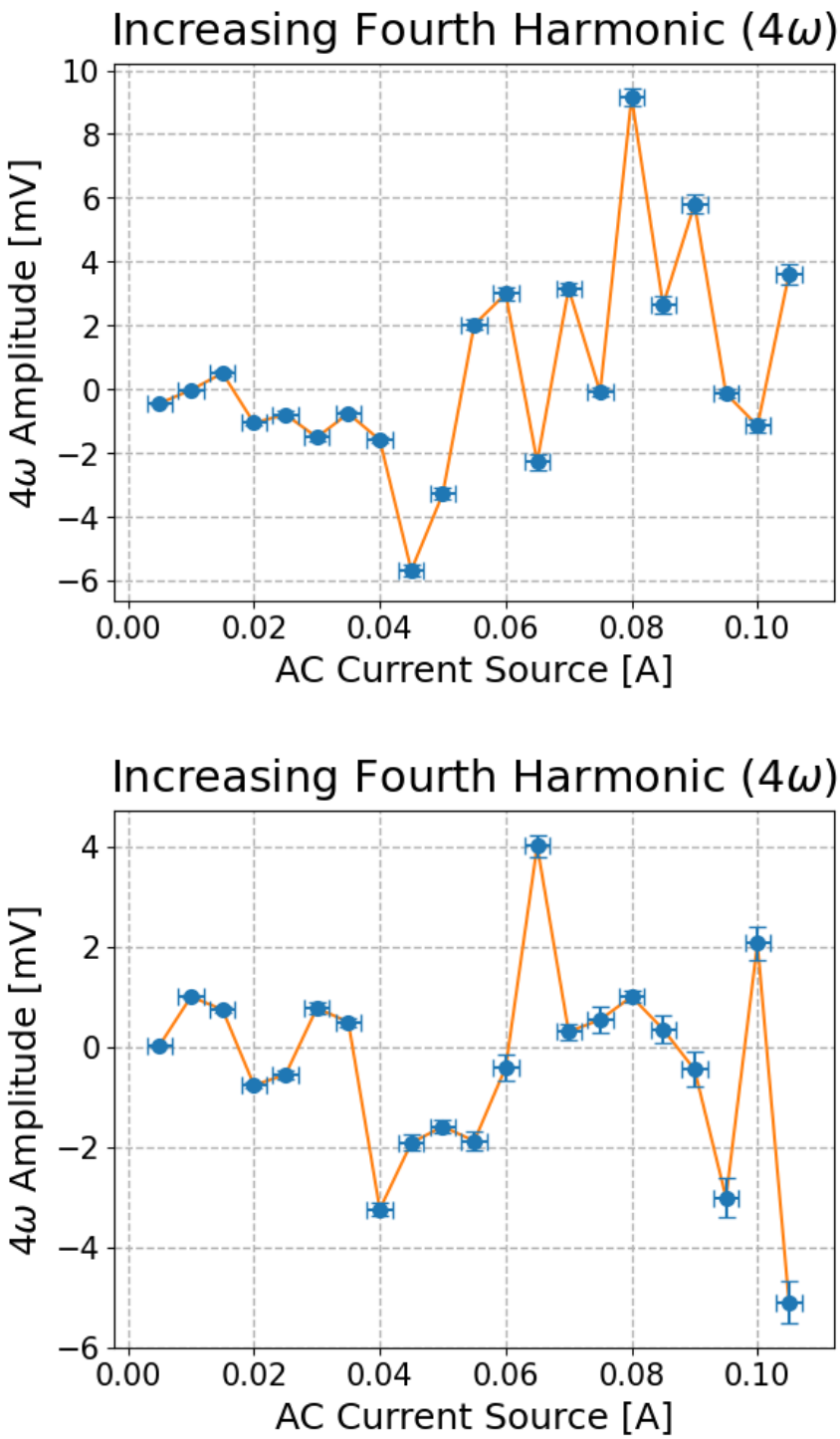

Figure B.4: The difference between our average experimental and control $4\omega = 2\pi \cdot 40$ Hz signals from both experiments.

B.5 Subtracted Average 5ω Data

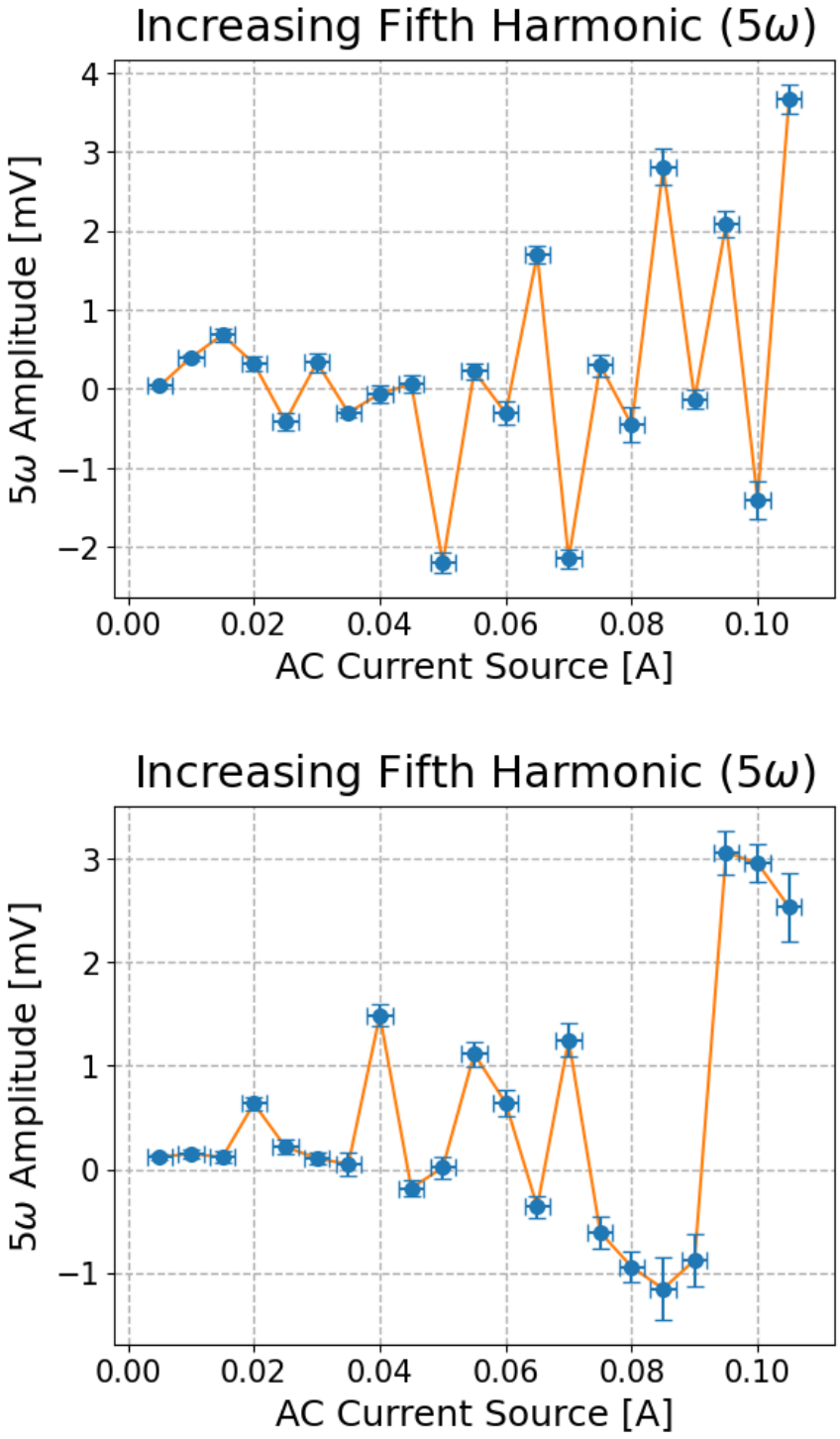

Figure B.5: The difference between our average experimental and control $5\omega = 2\pi \cdot 50$ Hz signals from both experiments.

B.6 Subtracted Average 6ω Data

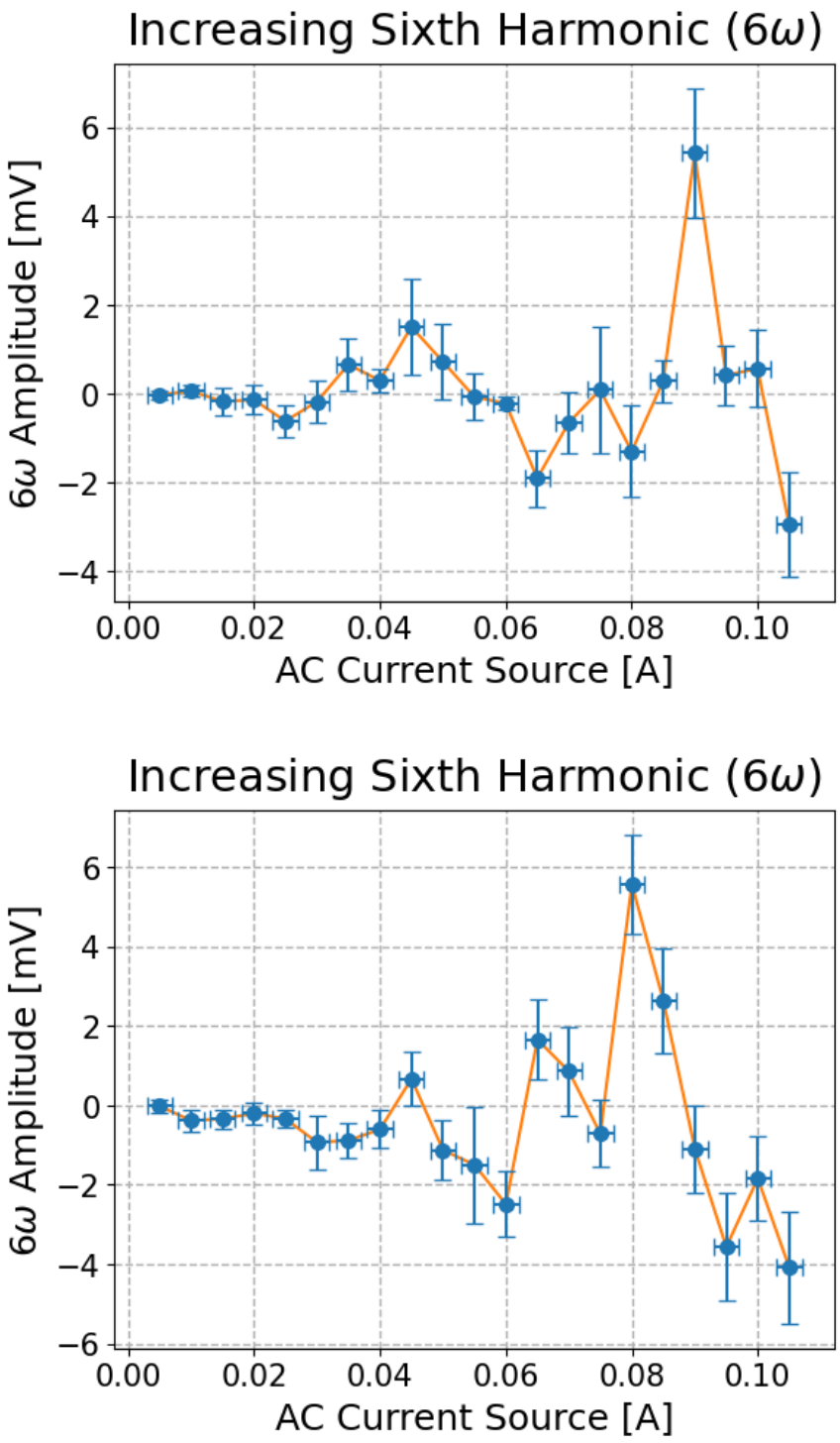

Figure B.6: The difference between our average experimental and control $6\omega = 2\pi \cdot 60$ Hz signals from both experiments.

B.7 Subtracted Average 7ω Data

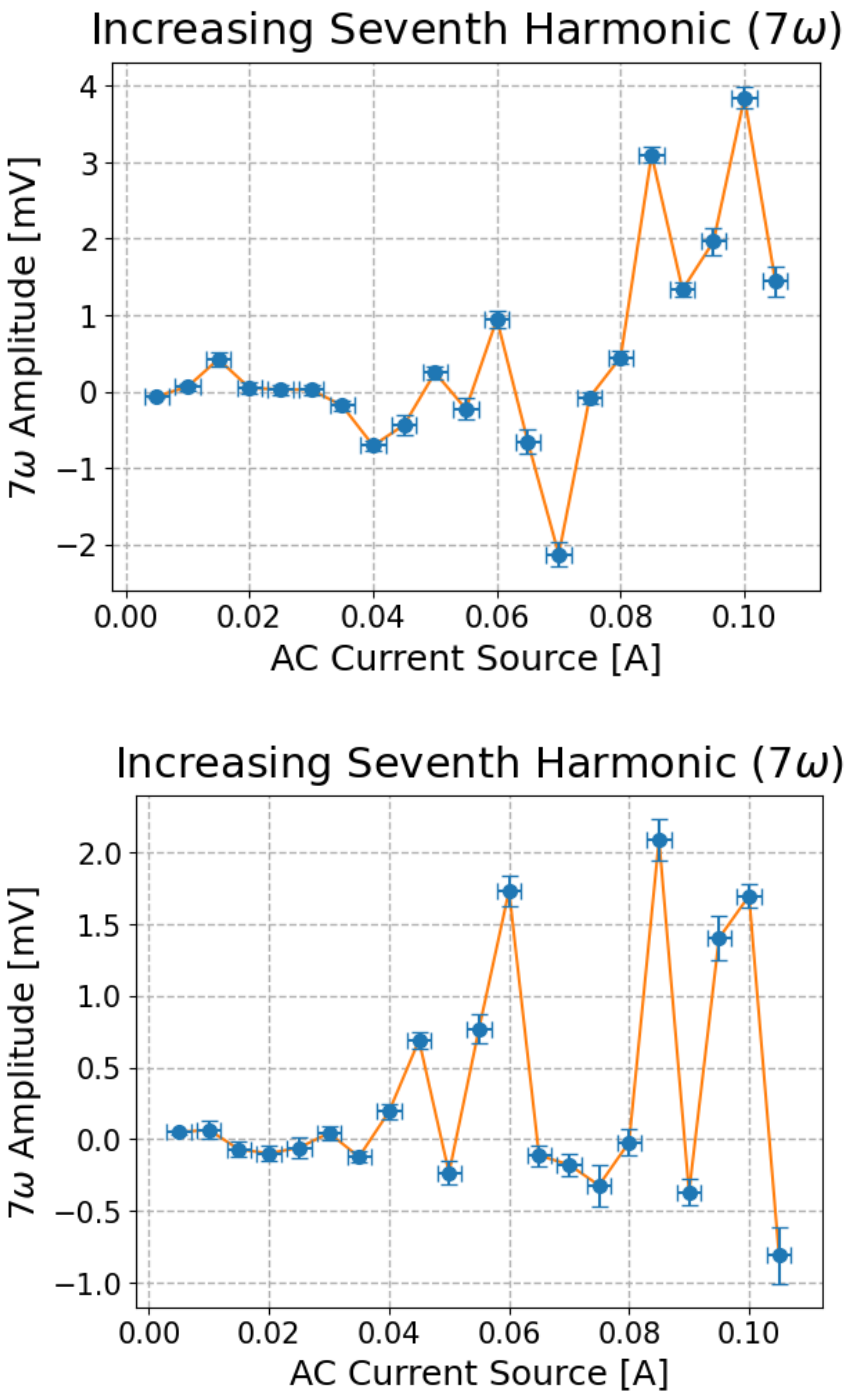

Figure B.7: The difference between our average experimental and control $7\omega = 2\pi \cdot 70$ Hz signals from both experiments.

B.8 Subtracted Average 8ω Data

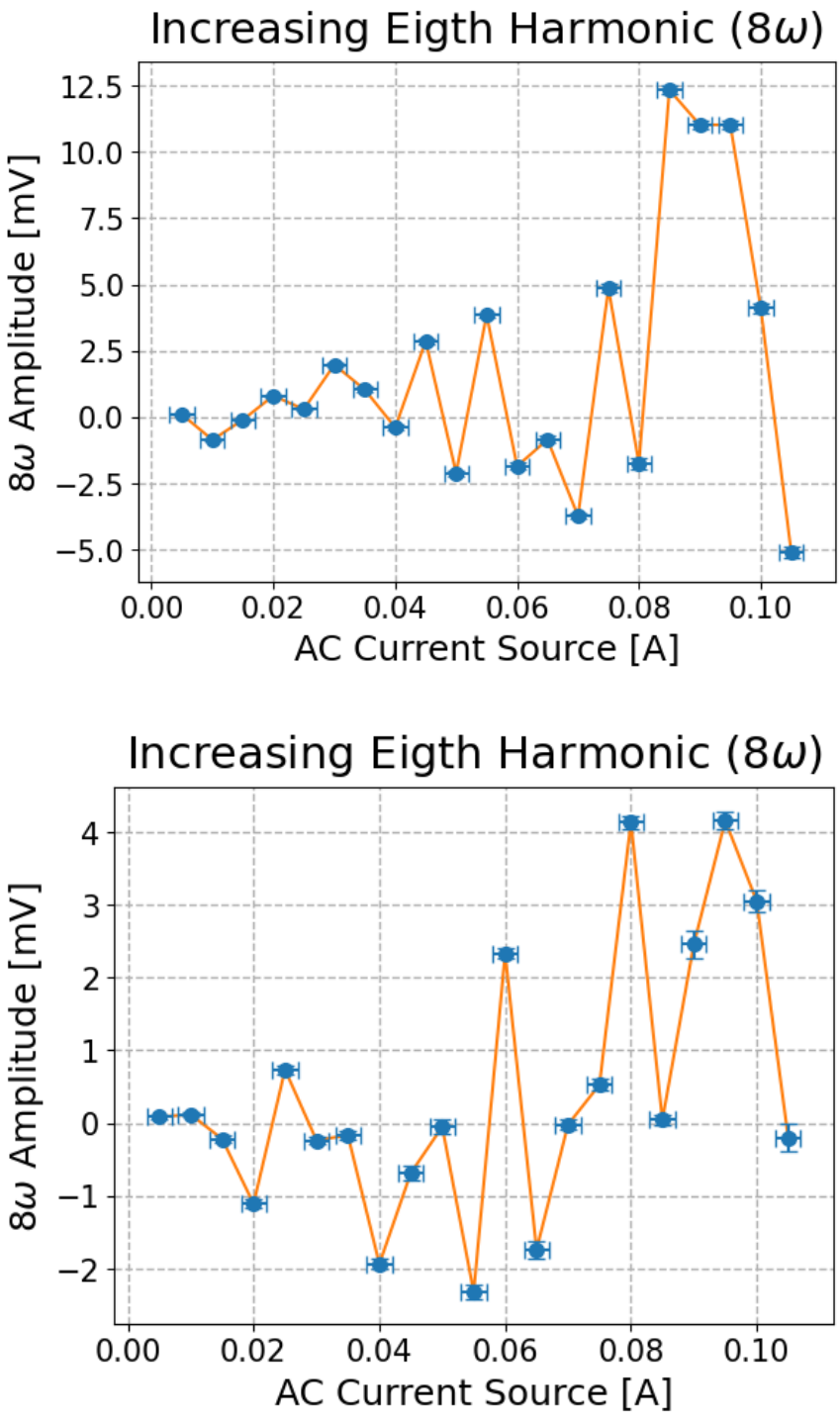

Figure B.8: The difference between our average experimental and control $8\omega = 2\pi \cdot 80$ Hz signals from both experiments.

References

[1] W. Miyahira et al. "Microwave Atom Chip Design". In: *Atoms* 9 (2021). DOI: 10.3390/atoms9030054.

[2] W. Miyahira. "AC Zeeman Potentials for Ultracold Atom Experiments: Suppressing Potential Roughness and the Development of a Microwave Atom Chip". PhD thesis. College of William & Mary, 2026. URL: hhttps://saaubi.people.wm.edu/ResearchGroup/AubinTheses/Miyahira_PhD_2026.pdf (visited on 04/16/2026).

[3] J. L. Braun et al. "A Steady-State Thermoreflectance Method to Measure Thermal Conductivity". In: *Rev. Sci. Instrum* 90 (2019). DOI: 10.1063/1.5056182.

[4] R. G. Bhardwaj and N. Khare. *Review: 3-ω Technique for Thermal Conductivity Measurement—Contemporary and Advancement in Its Methodology*. In: *International Journal of Thermophysics* vol. 43 (2022). DOI: 10.1007/s10765-022-03056-3.

[5] D. G. Cahill. "Thermal Conductivity Measurement from 30 to 750 K: The 3ω Method". In: *Rev. Sci. Instrum.* 61 (1990), pp. 802–808. DOI: 10.1063/1.1141498.

[6] T. Favaloro, J.-H. Bahk, and A. Shakouri. "Characterization of the Temperature Dependence of the Thermoreflectance Coefficient for Conductive Thin Films". In: *Rev. Sci. Instrum.* 86 (2015). DOI: 10.1063/1.4907354.

[7] M. K. Ivory et al. "Atom Chip Apparatus for Experiments with Ultracold Rubidium and Potassium Gases". In: *Rev. Sci. Instrum.* 85 (2014). DOI: 10.1063/1.4869781.

[8] R. L. Xu et al. "Thermal Conductivity of Crystalline AlN and the Influence of Atomic-Scale Defects". In: *J. Appl. Phys.* 126 (2019). DOI: 10.1063/1.5097172.

[9] Y. R. Koh et al. "Bulk-Like Intrinsic Phonon Thermal Conductivity of Micrometer-Thick AlN Films". In: *ACS Applied Materials & Interfaces* 12 (2020), pp. 29443–29450. DOI: 10.1021/acsami.0c03978.

[10] M. S. Bin Hoque et al. "High In-Plane Thermal Conductivity of Aluminum Nitride Thin Films". In: *ACS Nano* 15 (2021), pp. 9588–9599. DOI: 10.1021/acsnano.0c09915.

[11] H. M. Lee, K. Bharathi, and D. K. Kim. "Processing and Characterization of Aluminum Nitride Ceramics for High Thermal Conductivity". In: *Advanced Engineering Materials* 16 (2014), pp. 655–669. DOI: 10.1002/adem.201400078.

[12] C. Yuan, R. Hanus, and S. Graham. *A Review of Thermoreflectance Techniques for Characterizing Wide Bandgap Semiconductors' Thermal Properties and Devices' Temperatures*. In: *J. Appl. Phys.* vol. 132 (2022). DOI: 10.1063/5.0122200.

[13] NASA. *NASA Directorates*. Page Last Updated: 2026-03-16. Page Editor: Massengil, D. URL: https://www.nasa.gov/directorates/ (visited on 04/16/2026).

[14] NASA. *GRACE-FO*. Site Manager: Velev, K. URL: https://grace.jpl.nasa.gov/mission/grace-fo/ (visited on 04/16/2026).

[15] S. Sasinowska. "Suppressing Current Deviations via the AC Skin Effect". B.Sc. Thesis (unpublished). College of William & Mary, 2025.

[16] D. J. Griffiths. *Introduction to Electrodynamics*. Cambridge University Press, 2017. ISBN: 978-1108420419.

[17] Hyperphysics. *Resistivity and Temperature Coefficient at 20 C*. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/Tables/rstiv.html#c1 (visited on 04/16/2026).

[18] Surmet Corporation. *ALON™ Optical Ceramic*. Archived 2013-06-12 from original URL: https://www.surmet.com/docs/Product_sheet_ALON.pdf - no longer available. URL: https://web.archive.org/web/20130612032055/http://www.surmet.com/docs/Product%5C_sheet_ALON.pdf (visited on 04/16/2026).