2020-11-30

Photoacoustic CO2 Detection in Biomass Cookstove Applications

Jacob Matthew Thomas
Brigham Young University

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

Part of the Engineering Commons

BYU ScholarsArchive Citation
Thomas, Jacob Matthew, "Photoacoustic CO2 Detection in Biomass Cookstove Applications" (2020). Theses and Dissertations. 8740.
https://scholarsarchive.byu.edu/etd/8740

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Photoacoustic CO₂ Detection in Biomass Cookstove Applications

Jacob Matthew Thomas

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Matthew R. Jones, Chair
Randy S. Lewis
Troy Munro

Department of Mechanical Engineering
Brigham Young University

Copyright © 2020 Jacob Matthew Thomas
All Rights Reserved
ABSTRACT

Photoacoustic CO\textsubscript{2} Detection in Biomass Cookstove Applications

Jacob Matthew Thomas
Department of Mechanical Engineering, BYU
Master of Science

Billions of people use biomass burning cookstoves in their homes and suffer serious health repercussions. Additionally, global warming is exacerbated by cookstove emissions containing greenhouse gases and particulate matter. Improved cookstoves (ICSs) mitigate the problem, but accurate and affordable emission gas measurements, particularly of Carbon Dioxide (CO\textsubscript{2}) and Carbon Monoxide (CO), are required in order to confidently declare ICSs cleaner burning than traditional cookstoves. The aim of this research is to assess the suitability of photoacoustic (PA) CO\textsubscript{2} detection technology for cookstove emissions monitoring. The designs of several longitudinally resonant, photoacoustic, LED, CO\textsubscript{2} sensors of varying levels of functionality are presented. Three aluminum cell designs allowed the detection of a photoacoustic signal: a 4cm long cylinder with a \(\sim\)1cm diameter (Design 3), a 3.9cm long cylindrical resonator with \(\sim\)1in diameter and quarter-acoustic-wavelength buffer volumes (Designs 4a,b), and a 3.7cm long cylinder with \(\sim\)1in diameter (Design 5). All three cell designs operate in the longitudinal resonant mode via the irradiation of gases inside the PA cell with a 4.3\(\mu\)m wavelength LED, driven at an on-off frequency in the kHz range by a square wave from an Arduino. A rudimentary lock-in amplifier (LIA) based on the AD630 was considered, but the SR830 LIA was actually used to extract the desired MEMS microphone signal from noise. Designs 3-4b produced PA signals dominated by wall-absorption, but the final design (Design 5) yielded a resonant PA signal proportional to CO\textsubscript{2} concentration. It was discovered that photoacoustic gas detection is challenging to design and set up without extensive experience and equipment. Practical lessons learned are shared. Primary limitations with the presented designs are identified as the extremely low power of the 4.3\(\mu\)m LEDs, wall absorption due to insufficient collimation of LED radiation, dependence on temperature, and reliance on an expensive, high performance, lock-in amplifier. Further testing and development of designs like Design 5 (short cylinder with large diameter-to-length ratio) is necessary to evaluate their potential for in-field, real-time CO\textsubscript{2} concentration measurement. Though LED PA CO\textsubscript{2} sensing was demonstrated to be possible, it is concluded that NDIR CO\textsubscript{2} sensors are currently better suited for cookstove use. In addition to photoacoustic detection, a method of detecting CO\textsubscript{2} concentration by measuring resonant frequency of the gas cell (The Acoustic Method) is presented.

Keywords: longitudinal, resonant, LED, photoacoustic, gas detection, CO\textsubscript{2}, resonant frequency, mid IR, LED, combustion emissions monitoring, biomass cookstoves
ACKNOWLEDGMENTS

Working through what seemed like endless unforeseen challenges in a multi-disciplinary research project was no cakewalk for me, and I could not have done it alone. Attempting to keep my acknowledgments “simple and in good taste,” I express my gratitude to the following.

My Committee  Dr. Jones for his years of mentorship and help with thermodynamic modeling and Drs. Randy Lewis and Troy Munro for agreeing to be on my committee and for sharing personal feedback and their research equipment.

Resources at BYU  The BYU Electrical Engineering Shop, BYU Prototyping Lab, BYU Mechanical Engineering Checkout Room, and BYU Chemical Engineering Stockroom for providing convenient and affordable access to supplies, equipment, training, and insight.

Hamamatsu Photonics  Hamamatsu Photonics for giving several free LED samples.

My Engineering Friends  Jonny Meldrum for his tireless tutoring in writing C++ Arduino code, Erich Nygaard for his valuable electronics wisdom, Alex Hansen for being my hands on campus during my 28 days in quarantine, and my other engineering comrades for supplying friendship and fresh ideas.

My Family  My parents for consistently loving and encouraging from afar and my wife, Taylor, for enduring my seemingly endless discussion of plots, being patient with my long work days, and laughing with me at my jokes.
# TABLE OF CONTENTS

## LIST OF TABLES


## LIST OF FIGURES


## NOMENCLATURE


# Chapter 1 Introduction

1.1 Understanding the Problem ........................................... 1
1.2 Objective ....................................................................... 4
1.3 Background ..................................................................... 4
  1.3.1 Bell’s Discovery ......................................................... 4
  1.3.2 Photoacoustic Applications ............................................ 5
  1.3.3 Mathematical Modeling ............................................... 6
  1.3.4 Gas Cell ................................................................. 8
  1.3.5 Radiation Source ....................................................... 11
  1.3.6 Signal Detection ....................................................... 14
1.4 Introduction Summary ................................................... 18

# Chapter 2 Methods

2.1 Modeling PA Signal Strength ........................................... 20
  2.1.1 Irradiation as a function of path length ......................... 20
  2.1.2 Initial Pressure Condition .......................................... 22
  2.1.3 Initial Pressure Time Derivative Condition .................. 23
2.2 Acoustic Method of CO\textsubscript{2} Detection ....................... 24
2.3 Manufacturing Processes ................................................ 26
  2.3.1 Design 1 ............................................................... 26
  2.3.2 Design 2a,b ........................................................... 26
  2.3.3 Design 3 ............................................................... 27
  2.3.4 Design 4a,b ........................................................... 27
  2.3.5 Design 5 ............................................................... 29
2.4 Data Acquisition, Data Processing, and Electronics ............... 30
  2.4.1 Lock-in Amplifier Evaluation ........................................ 31
  2.4.2 Verification of LED operation with Photodiode ............... 34
  2.4.3 Design 1 ............................................................... 37
  2.4.4 Design 2a ............................................................ 39
  2.4.5 Design 2b ............................................................ 39
  2.4.6 Design 3 ............................................................... 42
  2.4.7 Design 4a ............................................................ 42
  2.4.8 Design 4b ............................................................ 42
  2.4.9 Design 5 ............................................................... 42
2.5 CO\textsubscript{2} Sensor Testing ............................................ 42
  2.5.1 Test Set-up and Procedures ......................................... 43
Chapter 3  Results and Discussion  ......................................................... 47
  3.1 Acoustic Method of CO$_2$ Detection  ............................................. 47
  3.2 Designs  .............................................................................. 47
    3.2.1 Design 1  ........................................................................ 49
    3.2.2 Design 2a ......................................................................... 51
    3.2.3 Design 2b ......................................................................... 55
    3.2.4 Design 3 ........................................................................ 58
    3.2.5 Design 4a ......................................................................... 63
    3.2.6 Design 4b ......................................................................... 68
    3.2.7 Design 5 ........................................................................ 73
  3.3 Results and Discussion Summary ...................................................... 77

Chapter 4  Conclusions and Future Work .................................................... 79
  4.1 Conclusions  ........................................................................ 79
    4.1.1 Photo-Acoustic CO$_2$ Detection  ......................................... 79
    4.1.2 Acoustic CO$_2$ Detection .................................................. 80
  4.2 Future Work  ...................................................................... 80
    4.2.1 Photo-Acoustic CO$_2$ Detection ......................................... 81
    4.2.2 Acoustic CO$_2$ Detection .................................................. 83

REFERENCES ............................................................................. 85

Appendix A  Bill of Materials ................................................................. 92

Appendix B  Data Logging and Generation of Rectangle Wave ....................... 93
  B.1 Higher Frequency Data Logging with PWM .................................. 93
  B.2 Lower Frequency Data Logging with PWM ................................. 99
  B.3 Semi-automatically Changing Rectangle Wave Frequencies with Datalogging ... 106
  B.4 Automatically Changing Square Wave Frequencies with Datalogging ....... 113
  B.5 Automatic Square Wave Frequency Sweep .................................. 119

Appendix C  Post-processing Code .......................................................... 120
  C.1 MATLAB/Octave Code: Post-processing Audio Data ...................... 120
  C.2 Post-processing LIA Data ......................................................... 122

Appendix D  Design and Computation Code .............................................. 124
  D.1 Resonant Cell Design .............................................................. 124
    D.1.1 Function to Compute Resonant Frequency .............................. 124
    D.1.2 Function to Compute Resonant Length .................................. 125
  D.2 LED Circuit Design ................................................................. 126
  D.3 AD630 Based Lock-in Amplifier Design ...................................... 128
  D.4 Acoustic Method ................................................................. 131
Appendix E  CO₂ Optical Filter Analysis ........................................... 133
Appendix F  Design 5 Process Sheet ..................................................... 134
LIST OF TABLES

3.1 Complete Design Table. ....................................................... 49
3.2 Design 1 Table. ................................................................. 49
3.3 Design 2a Table. ............................................................... 52
3.4 Design 2b Table. ............................................................... 55
3.5 Design 3 Table. ................................................................. 59
3.6 Design 4a Table. ............................................................... 63
3.7 Design 4b Table. ............................................................... 69
3.8 Design 5 Table. ................................................................. 73
LIST OF FIGURES

1.1 A Peruvian woman lighting a traditional cookstove (Photo by Matthew Jones in Salkantay, Peru, 2010). ...................................................... 2
1.2 IR spectrum for CO$_2$ [1]. Note the strong absorption band around 4.3$\mu$m. ................................................................. 3
1.3 The photoacoustic effect. ........................................................................ 3
1.4 One of Bell’s early PA systems, made with the help of a Mr. Tainter. [2] .......... 4
1.5 1D cylindrical resonator. ...................................................................... 7
1.6 A cross-section of a 1D pipe resonator design with quarter-wavelength buffer volumes and the standing pressure wave represented by a dotted sinusoid. ......................... 10
1.7 Laser based PA sensor schematic [3]. .................................................. 12
1.8 LED operating principle [4]. ................................................................. 13

2.1 Cylindrical cross section system and interactions for the absorption energy balance. .................................................. 21
2.2 Absorption Coefficient of CO$_2$ plotted over wavelength in the 4.3$\mu$m wavelength region [5]. ....................................................... 22
2.3 Cylindrical cross section system and interactions to obtain the initial pressure condition. .................................................. 22
2.4 3D printed Design 1 cell. ..................................................................... 26
2.5 Top: CAD model of Design 2a. Bottom: Machined and 3D printed Design 2a. .... 27
2.6 Machined Design 3 cell. ....................................................................... 28
2.7 Machined Design 4 cell. ....................................................................... 28
2.8 Machined Design 5 cell. Square LED hole on left, large sapphire window hole on right. 30
2.9 An Arduino with a micro SD card module connected on the left and MEMS microphone connected on top, typical for all of the PA designs. .................................................. 31
2.10 Electronic block diagram of 1kHz test with Arduino and SD card. .......... 32
2.11 Purchased AD630-based LIA output plotted against various amplitude Arduino PWM input signals. .................................................. 33
2.12 Electronic block diagram of 1kHz test with function generator and o-scope. ...... 33
2.13 Purchased AD630-based LIA output plotted against various amplitude function generator input signals. .................................................. 34
2.14 SR830 measurement of the microphone signal at increasing distances. ................................. 34
2.15 Circuit schematic for PD in photovoltaic mode. ........................................ 35
2.16 LED and PD set-up. ........................................................................... 35
2.17 Lms43LED-CG test with PD. The peaks show when the LED was aligned with the PD at the following distances: 0.5, 1, 2, 3, and 4 cm. ........................ 36
2.18 L13201-0430M with x3.3 Package test with PD. The peaks show when the LED was aligned with the PD at the following distances: 0.5, 2, and 4 cm. ........................ 37
2.19 LED43-R test with PD. The peaks show when the LED was aligned with the PD at the following distances: 0.5, 2, 4, and 6 cm. .................................................. 38
2.20 The Roithner LED43-R (left) with QCW driver board. ............................. 38
2.21 Electronic block diagram for the set-up of Design 2b and later designs. ......... 39
2.22 Circuit used to drive LED with +8V through a MOSFET. PCB layout for custom LED circuit, as modeled in Dip Trace (units in mm) shown on right. ......................... 40
2.23 The microphone PCB, as modeled in Dip Trace (units in mm). .................... 41
2.24 CO$_2$ injection system, comprised of a CO$_2$ cartridge, ball needle, and PVC tubing. ........ 43
2.25 Resonant profile for Design 2b in room air, typical of all resonance profiles. 44
2.26 Left: Resonance peaks due to buzzer from 2774 to 3374Hz for Design 4. Right: Resonant peak due to PA effect at 3000Hz for Design 4. 45

3.1 Concentration plotted over resonant frequency for three cell lengths using the Acoustic Method at T=294K. 48
3.2 A generic schematic for PA designs. 48
3.3 Schematic of PA sensor Design 1. Note that it was not functional. 50
3.4 Design 2a, noted for its adjustable length and 3D printed end caps. 51
3.5 FFT comparing LED driver frequency initially (cold) and after warming up a while (hot). 53
3.6 Resonant profile for Design 2a in room air. 54
3.7 A commercial driver board for the LED43 experienced frequency change over time, likely due to the heating of electronics. 55
3.8 Same cell as PAS2, but with the ends pushed in for a minimum resonator length of 13.5cm. 56
3.9 Resonant profile for Design 2b in room air. 57
3.10 The third design, showed with the full set-up on the left and alone on the right. 59
3.11 Resonant profile for Design 3 in room air. 60
3.12 Microphone signal with no acoustic source on the left and with the LED driver frequency at 3.6, 3.9, and then 4.2kHz on the right as CO$_2$ leaked in Design 3. 61
3.13 Microphone signal with buzzer frequency at 3.6, 3.9, and then 4.2kHz as CO$_2$ escaped in Design 3. 62
3.14 Concentration of CO$_2$ plotted against resonant frequency by the Acoustic Method for Design 3. Large concentrations on the left; ppm levels on the right. 62
3.15 Design 4a, showed with the full set-up on the left and the resonator alone on the right. 64
3.16 Resonant profile for Design 4 in room air. 64
3.17 Microphone signal with buzzer frequency at 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, and 3.48kHz as CO$_2$ was pumped out of Design 4a. 65
3.18 Microphone signal with buzzer frequency at 2.8, 2.9, and 3.0kHz as CO$_2$ naturally leaked in Design 4a. 65
3.19 PA peaks from LED driven at 2.9, 3.1, 3.3, and 3.48kHz as CO$_2$ was pumped out of Design 4a. 66
3.20 Concentration of CO$_2$ plotted against resonant frequency by the Acoustic Method for Design 4a. Also applies to Design 4b. 66
3.21 Microphone signal response to LED heating (decrease at $\sim$2 min) and cooling (increase at $\sim$7 min). 67
3.22 PA signal drops as air is pumped into a cell initially full of CO$_2$ (Left) and DFE (Right). 68
3.23 Resonance profiles due to the PA effect in Design 4a are shown for 2.8, 3.0, and 3.2kHz, respectively as CO$_2$ levels were decreased. 68
3.24 Erratic microphone signal produced with leaking CO$_2$ in non-resonant mode. 70
3.25 LED warm-up with fans (right) and without (left) in room air. 70
3.26 Resonant profiles, from left to right, for 2774, 2837, 2903, 2972, 3045, 3201, and 3374Hz respectively from one of five buzzer tests for Design 4b. 71
3.27 Time of resonant peaks with error bars of ± two standard deviations from the mean of the five buzzer tests with Design 4b. 72
3.28 PA resonant peaks at 2840, 3048, and 3205Hz modulation frequency for Design 4b.

3.29 Design 5, with full set-up view on the left and close-up of cell on the right.

3.30 Resonant profile for Design 5 in room air.

3.31 Acoustic Method for Design 5.

3.32 Buzzer resonant profiles at 3784, 4197, and 4603Hz (left) and 3784, 4095, 4401, and 4603Hz (right) as CO\textsubscript{2} naturally leaked from Design 5.

3.33 LED resonant profiles at 3787, 4201Hz, and 4608Hz (left) and 3787, 4098, 4405, and 4608Hz (right) as CO\textsubscript{2} naturally leaked from Design 5. Note that resonant profiles shorten over time.

3.34 Time of resonant peaks with error bars of ± two standard deviations from the mean of the six buzzer tests with Design 5.

3.35 Calibration curve for Design 5, relating peak resonant signal obtained from leak testing to concentration of CO\textsubscript{2}.

A.1 Not exhaustive. Does not include lab equipment such as lathe, SR830, etc.

E.1 Overlapping FWHM spectral bandwidths of CO\textsubscript{2} absorption, LED43 emission, and 4.26µm optical bandpass filter.

F.1 Process sheet for obtaining resonant profiles as CO\textsubscript{2} leaked in Design 5.
# NOMENCLATURE

## Latin Alphabet

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>amplitude</td>
</tr>
<tr>
<td>a</td>
<td>argument</td>
</tr>
<tr>
<td>C</td>
<td>cell constant, set-up constant</td>
</tr>
<tr>
<td>c</td>
<td>speed of sound</td>
</tr>
<tr>
<td>D</td>
<td>DC component</td>
</tr>
<tr>
<td>f</td>
<td>frequency</td>
</tr>
<tr>
<td>G</td>
<td>irradiation flux</td>
</tr>
<tr>
<td>H</td>
<td>deposited heat density</td>
</tr>
<tr>
<td>L</td>
<td>length</td>
</tr>
<tr>
<td>m</td>
<td>mass</td>
</tr>
<tr>
<td>n</td>
<td>numerator</td>
</tr>
<tr>
<td>P</td>
<td>pressure</td>
</tr>
<tr>
<td>Q</td>
<td>resonance quality factor</td>
</tr>
<tr>
<td>q</td>
<td>rate of heat transfer</td>
</tr>
<tr>
<td>R</td>
<td>specific gas constant</td>
</tr>
<tr>
<td>S</td>
<td>source term</td>
</tr>
<tr>
<td>T</td>
<td>temperature</td>
</tr>
<tr>
<td>t</td>
<td>time</td>
</tr>
<tr>
<td>x</td>
<td>mass fraction</td>
</tr>
<tr>
<td>u</td>
<td>uncertainty</td>
</tr>
<tr>
<td>V</td>
<td>volume</td>
</tr>
<tr>
<td>v</td>
<td>voltage, electrical potential</td>
</tr>
<tr>
<td>Z</td>
<td>impedance</td>
</tr>
<tr>
<td>z</td>
<td>longitudinal distance</td>
</tr>
</tbody>
</table>

## Greek Alphabet

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>γ</td>
<td>ratio of specific heats</td>
</tr>
<tr>
<td>κ</td>
<td>absorption coefficient</td>
</tr>
<tr>
<td>λ</td>
<td>wavelength</td>
</tr>
<tr>
<td>μ</td>
<td>one millionth of</td>
</tr>
<tr>
<td>\nabla</td>
<td>gradient</td>
</tr>
<tr>
<td>\nabla^2</td>
<td>laplacian</td>
</tr>
<tr>
<td>ρ</td>
<td>density</td>
</tr>
<tr>
<td>Σ</td>
<td>sum</td>
</tr>
<tr>
<td>θ</td>
<td>phase lag</td>
</tr>
<tr>
<td>ω</td>
<td>angular frequency</td>
</tr>
</tbody>
</table>
### Symbols with Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_n$</td>
<td>amplitude of $n^{th}$ eigenmode component</td>
</tr>
<tr>
<td>CO$_2$</td>
<td>carbon dioxide</td>
</tr>
<tr>
<td>$c_v$</td>
<td>specific heat at constant volume</td>
</tr>
<tr>
<td>$c_n$</td>
<td>$n^{th}$ constant offset</td>
</tr>
<tr>
<td>$d_c$</td>
<td>duty cycle</td>
</tr>
<tr>
<td>$f_r$</td>
<td>resonant frequency</td>
</tr>
<tr>
<td>$G_\lambda$</td>
<td>irradiation per unit wavelength</td>
</tr>
<tr>
<td>$\kappa_\lambda$</td>
<td>absorption coefficient at a wavelength</td>
</tr>
<tr>
<td>$\kappa_{\Delta\lambda}$</td>
<td>average absorption coefficient of a wavelength band</td>
</tr>
<tr>
<td>$S_n$</td>
<td>$n^{th}$ AC signal</td>
</tr>
<tr>
<td>SO$_2$</td>
<td>sulfur dioxide</td>
</tr>
<tr>
<td>$t_p$</td>
<td>pulse time</td>
</tr>
<tr>
<td>$x_+$</td>
<td>mass fraction above ambient concentration</td>
</tr>
<tr>
<td>$\phi_n$</td>
<td>dimensionless eigenmode distribution function</td>
</tr>
</tbody>
</table>

### Other symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>carbon monoxide</td>
</tr>
<tr>
<td>$W$</td>
<td>power, work rate</td>
</tr>
<tr>
<td>$\sum$</td>
<td>summation with index $n$</td>
</tr>
<tr>
<td>$\infty$</td>
<td>infinity</td>
</tr>
<tr>
<td>$\Sigma_{n=a}^b$</td>
<td>summation with index $n$ from $a$ to $b$</td>
</tr>
</tbody>
</table>

### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>computer-aided design</td>
</tr>
<tr>
<td>DFE</td>
<td>difluoroethane</td>
</tr>
<tr>
<td>DNE</td>
<td>does not exist</td>
</tr>
<tr>
<td>EIN</td>
<td>equivalent input noise</td>
</tr>
<tr>
<td>ECM</td>
<td>electret condenser microphone</td>
</tr>
<tr>
<td>FID</td>
<td>flame ionization detector</td>
</tr>
<tr>
<td>FWHM</td>
<td>full width at half maximum</td>
</tr>
<tr>
<td>ICS</td>
<td>improved cookstove</td>
</tr>
<tr>
<td>IR</td>
<td>infrared</td>
</tr>
<tr>
<td>LED</td>
<td>light-emitting diode</td>
</tr>
<tr>
<td>LIA</td>
<td>lock-in amplifier</td>
</tr>
<tr>
<td>MEMS</td>
<td>microelectromechanical system</td>
</tr>
<tr>
<td>NDIR</td>
<td>nondispersive infrared</td>
</tr>
<tr>
<td>PA</td>
<td>photoacoustic</td>
</tr>
<tr>
<td>PAS</td>
<td>photoacoustic spectroscopy</td>
</tr>
<tr>
<td>PCB</td>
<td>printed circuit board</td>
</tr>
</tbody>
</table>
PLA | polylactic acid
PM | particulate matter
ppm | parts per million
ppt | parts per trillion
PWM | pulse-width modulation
SMD | surface-mount device
SNR | signal-to-noise ratio

**Glossary**

**Acoustic Method**  Method of CO₂ detection using an acoustic source and sensor to determine resonant frequency, as presented in thesis.

**aliasing**  Misidentification of an AC signal frequency.

**buffer volumes**  Hollow cylinders of relatively large diameter placed on either side of the cell resonator meant to reduce acoustic noise and increase resonant quality.

**cell**  Structure containing the gas sample to be analyzed.

**end corrections**  Fraction of the radius of a pipe added to its length to improve the accuracy of the analytical length for a pipe open on each end. Typically .6R.

**mid IR**  Portion of the IR range of the electromagnetic spectrum between the near and far IR. For this thesis, wavelengths from \( \sim 3-12 \mu m \).

**modulate, drive**  To periodically alter the properties of an AC signal. In this thesis, to power on and off at a frequency.

**1D pipe**  Theoretical hollow cylinder of infinite length-to-radius ratio.

**rectangle wave**  Signal that periodically alternates between two fixed values at any duty cycle.

**resonant frequency**  Frequency that produces a resonant maximum. In this thesis, the required acoustic frequency to excite a standing wave in the resonator.

**resonator**  Portion of resonant cell in which resonance occurs.

**square wave**  Rectangle wave with 50% duty cycle.

**wall absorption**  Absorption of radiation by cell walls instead of cell gases, causing PA noise.

**window**  Thin, solid material transparent to the optical wavelength of interest (4.3\( \mu \)m).
CHAPTER 1. INTRODUCTION

1.1 Understanding the Problem

Half of the world’s population uses biomass cookstoves to cook and heat their homes, such as is in Figure 1.1 [6]. When these stoves are used, particulate matter (PM), commonly known as smoke, and potentially harmful greenhouse gases such as carbon monoxide (CO) and carbon dioxide (CO$_2$) are emitted into the surroundings, causing both local health problems and worldwide climate changes [7–9].

Biomass burning cookstove users are at increased risk of heart and lung diseases such as pneumonia, ischaemic heart disease, and chronic obstructive pulmonary disease because of stove emissions. Women and children are particularly affected because they generally spend more time at home. It is estimated that 3.8 million people per year die prematurely due to illness attributable to cookstove emissions such as CO and PM. [6]

In addition, greenhouse gases and black carbon are believed to increase global temperatures. Both CO$_2$ and CO absorb in the infrared (IR), making them greenhouse gases (see Figure 1.2). When biomass is completely burned, carbon, hydrogen, and oxygen in the biomass react with oxygen in the air to create CO$_2$ and water, both non-toxic gases [10]. When there is insufficient oxygen in the air, the toxic gas CO is formed in addition to CO$_2$, carbon (soot), and water. Improved cookstoves (ICS) aim to burn more efficiently by increasing airflow and reducing heat loss through the ground. More efficient combustion results in reduced CO emissions and increased CO$_2$ emissions. This is desirable because CO is a lethal gas and CO$_2$ is a natural product of the human and plant life interaction. [8, 11–13]

In order to reduce toxic gas and PM emissions, research has been done to design ICS that burn cleanly to replace current stoves. Standardized performance metrics have been recommended by James Jetter and include the modified combustion efficiency, overall thermal efficiency, specific energy consumption, and specific emission rate [14]. To quantifiably compare cookstoves with
these metrics, concentrations of toxic gas and PM emissions must be known. Common methods for measuring pollutant gases in cookstove emissions are Nondispersive Infrared (NDIR) technology for CO$_2$, electrochemical technology for CO, and Flame Ionization Detector (FID) technology for both CO and CO$_2$ [14–16]. The water boiling test, a standard cookstove research metric, currently recommends NDIR technology for CO$_2$ detection and electrochemical or NDIR technology for CO detection [17]. These gas detection methods all provide accurate and near-linear calibration curves. However, electrochemical sensors must be regularly calibrated and replaced every few years and FID jets require regular cleaning or replacing [18]. CO$_2$ NDIR sensors are fairly low-maintenance and low-cost, but are limited in measuring very small concentrations [19].

A promising alternative technology for accurately detecting CO and CO$_2$ is PA spectroscopy (PAS). The photoacoustic (PA) effect dictates that when a substance is irradiated with modulated light, a sound wave is produced as the substance periodically expands (see Figure 1.3). PA gas detection involves irradiating a gas sample with intensity-modulated light at a strong and unique absorbing wavelength of the target pollutant gas (see Figure 1.2), and measuring the sound with a pressure sensor. The concentration of the target gas is proportional to the amplitude of the sound wave.
Figure 1.2: IR spectrum for CO₂ [1]. Note the strong absorption band around 4.3µm.

Figure 1.3: The photoacoustic effect.

Unlike electrochemical sensors and FIDs, PA technology is not consumable and would potentially allow consolidation to a single sensor that detects both CO and CO₂ concentrations. Unlike NDIR signals (which are proportional to a background value minus the amount of gas absorption), PA signals are directly proportional to the amount of gas absorption. Therefore, PA sensors can be highly sensitive and stable [20]. Photoacoustic CO and CO₂ sensors have already been implemented in such applications as fire-detection, obtaining absorption and scattering coefficients, and traffic emissions measurement [21–24], but have not yet become common in the cookstove research community. It is also very rare for contemporary cookstove researchers to report estimates for uncertainty in gas concentration measurements, an essential practice for evaluating ICS [25].
1.2 Objective

The objective of the presented research is to assess the suitability of photoacoustic (PA) CO$_2$ detection in the monitoring of biomass cookstove emissions. To this end, several PA CO$_2$ sensors were designed, constructed and tested.

1.3 Background

1.3.1 Bell’s Discovery

The PA effect was first discovered by Alexander Graham Bell around 1881 [2]. Bell’s first experiments were performed by modulating sunlight with a chopper wheel (Interrupting Disk) and directing the light onto a substance contained within a glass-like cavity. Sound at the chopper wheel frequency could be heard via a hearing tube, as shown in Figure 1.4. Bell discovered the effect of resonance by allowing the chopper wheel frequency to slow until it matched the fundamental, or resonant, frequency of the cell, forming a standing acoustic wave. He described the sound as transitioning from “feeble” to “so loud that it might have been heard by an audience of hundreds of people [2].”
1.3.2 Photoacoustic Applications

Since its discovery, PAS has been employed for various purposes including fire detection, indoor air monitoring, harsh outdoor monitoring, military chemical detection, miniature cell-phone components, automotive emissions, and combustion emissions.

**Fire Detection** Since both CO and CO\textsubscript{2} are products of combustion, researchers have developed CO and CO\textsubscript{2} photoacoustic sensors to indicate the presence of fires. Chen and Jiang [21, 26] have both successfully used CO measurements at the parts-per-million (ppm) level for fire detection and Nebiker [22] has used CO\textsubscript{2} measurement for fire detection. Note that these researchers used lasers and broadband IR emitters as the radiation source, but not LEDs.

**Ambient Monitoring** Huber and Ambs have studied the detection of CO\textsubscript{2} for indoor monitoring using custom-built PA sensors [27–29] while Eberhardt and Scholz have explored PAS in extreme climates such as permafrost areas [30]. Their unique approach was to develop a modified NDIR sensor design that uses a PA detector in place of a typical photo detector to measure CO\textsubscript{2} and methane. Sholz and Eberhardt refer to this design as an NDIR sensor and use LEDs while Huber and Ambs call the design a “two cell” setup and use. [27–35]

**Miniaturized Applications** Traditionally, PA sensors were limited to laboratory settings due to expensive and bulky equipment such as optics table set-ups, LIAs, and laser controllers [36]. In recent years, efforts have been made to miniaturize PA sensors for various applications. For example, Rouxel built a centimeter sized PA sensor capable of accurately measuring methane [37, 38]. Huber’s and Amb’s room monitoring sensors [27–29, 39] were compact, as was Willer’s winston cone-shaped absorption detector [40]. In addition to indoor monitoring, Huber investigated the use of a miniature PA CO\textsubscript{2} sensor in automotive applications [39]. Bauer created a miniature PA sensor by 3D printing a tiny gas cell [41,42] whereas Scholz miniaturized her NDIR photoacoustic CO\textsubscript{2} sensor for implementation in mobile devices. The Army Research Laboratory also conducted research on PAS miniaturization [3]. In order to operate in resonant mode, many systems used high radiation modulation frequencies to allow for a very short resonant cell. Cell diameters were also made as small as possible while still maintaining low wall absorption [43]. Other systems
used non-resonant PA systems, such as the NDIR design, so as to avoid the relationship between cell length and modulation frequency altogether.

**Military Chemical Detection**  For over 14 years, the Army Research Laboratory has worked on implementing PAS in their chemical detection technologies due to its broad potential for application. They successfully used PAS on both gaseous and condensed phase materials, measuring as low as parts-per-trillion (ppt) levels. Additionally, they verified the ability to miniaturize PA technology without a loss of performance and detect at a stand-off distance. Note that high-end, expensive equipment was used. [3]

**Combustion Emissions Monitoring**  Perhaps most relevant to the cookstove field, El-Safoury developed a PA sulfur dioxide (SO$_2$) sensor with a resolution less than 1 ppm specifically for use with combustion emissions [44]. Kirchstetter [23] has also used PAS within cookstove applications, but to measure absorption and scattering coefficients rather than pollutant concentrations [23].

1.3.3 Mathematical Modeling

**The Wave Equation**

Since a PA signal is an acoustic wave, it can be modeled by the wave equation. The wave equation for PAS can be derived from the Navier-Stokes, heat diffusion, continuity, and thermodynamic state equations. Thus, laws governing pressure wave propagation include gas pressure, temperature, density, and velocity variables, but properties other than pressure are mathematically eliminated. For a more thorough treatment of photoacoustic modeling, see Miklós or Besson. [45, 46] Beginning with the inhomogenous pressure wave equation,

$$\frac{\partial^2 P}{\partial t^2} - c^2 \nabla^2 P = S$$

where the source term, $S$, accounts for what drives the wave and $c$ is the speed of sound. In PAS, $S$ represents the absorbed radiant power that heats the gas. Effectively all of this absorbed energy is released in the gas through expansion, or wave propagation [46]. For a 1D resonator in the $z$
direction (see Figure 1.5),

$$\frac{\partial^2 P}{\partial t^2} - c^2 \frac{\partial^2 P}{\partial z^2} = (\gamma - 1) \frac{\partial H}{\partial t} \tag{1.1}$$

where $H$ is the “deposited heat density” and $\gamma$ is the ratio of specific heats [46].

**Direct Solution to Wave Equation**

The solutions to the homogeneous form of this equation with no losses are orthogonal, and are the eigenfunctions of a closed, lossless resonator. The solution to the lossless inhomogenous equation (Equation 1.1) can be expressed as a series expansion of the lossless eigenfunctions.

$$P = A_0 + \sum_n A_n \phi_n$$

The expansion coefficient, $A_n$, is the amplitude of the nth eigenmode component and $\phi_n$ is the dimensionless eigenmode distribution function.

Within the expression for $A_n$, a resonance quality factor is included to account for total loss

$$Q = \frac{2\pi \times \text{accumulated energy}}{\text{energy lost in one period}}$$

or, for high Q systems,

$$Q = \frac{f_r}{\text{FWHM}} \tag{1.2}$$

where $f_r$ is the cell resonant frequency of interest and FWHM is the full-width at half maximum, or the frequency band bounded by the two frequencies where half the maximum sound is detected. The quality factor, $Q$, is a measure of the ratio of resonant amplitude to non-resonant amplitude and is a standard throughout PAS literature. Cells with $Q$ factors of up to 1000 exist, but typical PA systems have medium $Q$ values well under 100. [46]
Another factor defined within $A_n$ is the cell constant, or set-up constant, often represented as $C$. It is a measure of the sensitivity of a PAS system at a given resonance frequency and is also common in PAS research publications. Typical values are on the order of a few hundred Pa-cm/W [46–48]. As a high cell constant means a lower $Q$ factor, Luo and Tavakoli assert that finding an appropriate balance between the two is the key issue in PA resonator design [47,49,50]. As with all sensors, signal-to-noise ratio (SNR) is also an important parameter for PA gas detectors [46].

Other Modeling Approaches

The approximations made in the direct solution above are not always well-suited for the properties of a cylindrical resonator of high length-to-diameter ratio. Though not used in this thesis, high length-to-diameter resonators (commonly called “pipe” or “1D” resonators) are often modeled by mathematical analogy to the electromagnetic equations of a waveguide, or the “transmission line equations.” [46]. For the “two cell” PA-NDIR sensor design, Huber presents a simpler approach for modeling the pressure signal using the ideal gas law [28].

1.3.4 Gas Cell

An essential component in any PAS sensor is a cell, or chamber, for containing the gas sample to be irradiated. PAS systems can operate in either resonant or non-resonant modes.

Non-resonant Mode

When the radiation source is modulated at a frequency lower than the first resonant frequency of the cell, the system is non-resonant. Modulation frequencies in non-resonant systems are usually less than 100 Hz [46]. In addition to a substantially weaker acoustic signal, a major drawback to non-resonant operation is that the gas sample must be contained in a closed system (no flow) during measurement [45]. This makes continuous monitoring complicated at best. A third problem at low modulation frequencies is that most noise sources increase linearly with decreasing frequency (1/f noise), resulting in a very low SNR [46].
Resonant Mode

As demonstrated by Bell, a primary advantage of resonance in PAS systems is the ability to increase the acoustic signal to be measured [2]. Due to this and the disadvantages of non-resonant amplification in PA systems, resonant systems are most common. The portion of a resonant gas cell containing a standing wave in resonant mode is called the resonator.

Cell Size and Shape  Resonant cells of various geometries have been developed [46, 51], but some variation of the cylinder seems to be dominant in PAS [36, 46]. Common designs include Helmholtz resonators, cylinders, and cylinders with larger radius sections (buffer volumes) on either end [46].

The shorter a resonator is, the higher the radiation modulation frequency needs to be to form a longitudinal standing wave within the resonator. For a cylindrical cell closed or open on both ends, the cylinder length is ideally equal to half of the wavelength when resonant. Recognizing that \( \lambda f = c \) (where \( c \) is the speed of sound in ambient air, \( \lambda \) is the wavelength, and \( f \) is the frequency) the approximate cylinder length is obtained.

\[
L = \frac{c}{2f}
\]  

(1.3)

Subtracting length corrections (typically 60% of resonator radius) from each cylinder end improves the accuracy of Equation 1.3 for open-open cylinders. According to Equation 1.3, reasonably short resonant cells (several centimeters) require high radiation source modulation frequencies, usually in the kHz range.

Many design considerations are made in the construction of a resonant cell for PAS. Resonant cells with small volume are desirable because they create a larger acoustic signal and have faster response time, but should not be so thin that a significant portion of energy from the radiation source is absorbed by the cell walls (wall absorption) instead of the gas sample [43].

Cell Fundamental Frequency  Cylindrical cells have multiple fundamental frequencies in longitudinal, radial, and azimuthal modes. Modulation frequencies are typically tuned to match one or more resonant modes of the cell. Though cells with low fundamental frequencies produce higher
signals and $Q$ factors, they have low SNR due to high levels of noise. [46] Longitudinal resonance modes tend to have lower $Q$ factors than radial modes, but have higher sensitivity and avoid problems associated with high $Q$ factors such as high sensitivity to modulation frequency drift and temperature [48]. Acoustic losses from poor surface finish, inlet and outlet holes, microphone holes, and window and wall absorption cause an increase in noise and a drop in resonant amplification. As such, holes are minimized, inner surfaces are polished, and acoustic filters are often employed [43, 47].

![Diagram of 1D pipe resonator design with quarter-wavelength buffer volumes](image)

**Figure 1.6:** A cross-section of a 1D pipe resonator design with quarter-wavelength buffer volumes and the standing pressure wave represented by a dotted sinusoid.

### 1D Pipe Resonators with Buffer Volumes

Cylinders with a high length-to-diameter ratio are typically designed for the first longitudinal resonance frequency and are sometimes called 1D pipe resonators in literature [46]. Bijnen, Tavakoli, Luo, and others [43, 47, 49, 50] have studied in detail the optimization of this kind of resonator. They all used polished brass cylindrical resonators with a radius of a few millimeters and with larger radius “buffer volumes” on either side. These buffer volumes (see Figure 1.6), now widespread in PAS [46], act to insulate the microphone from unwanted acoustic background signals produced by absorption in the windows and cell walls. Bijnen experimentally determined the optimal buffer geometry for destructive acoustic interference (i.e. noise cancellation) to be a quarter of the acoustic wavelength long with a large buffer radius. Buffer lengths less than an eighth of the wavelength cause a significant loss in resonant quality. He suggests a buffer radius at least three times larger than the radius of the resonator. [43, 47]

The resonator within a cylinder with buffer volumes behaves similarly to an open ended pipe [46]. Bijnen’s experimental resonant frequency is less than 3% from the ideal open-open
cylinder. In a COMSOL simulation, Pernau compares a closed-closed pipe to a cylinder with eighth wavelength long buffer volumes on either end. His simulation shows the barbell geometry produces acoustic signals 10 times stronger than that of the unmodified cylinder [52]. Thus, buffers both reduce noise and increase resonant quality.

**Cell Material** Though brass has been frequently used [43, 47, 49, 50], likely for simplicity of polishing and low absorption in the mid IR, it is a relatively expensive material [53]. Aluminum has similarly low absorption at 4.3\(\mu\)m, but its lower expense makes it a promising alternative. 3D printed resonators are even lower cost, and can be easily manufactured by additive manufacturing (3D printing). Bauer and Yang have used 3D printed resonant cells [41, 42, 54, 55].

### 1.3.5 Radiation Source

Depending on the type of equipment used, PA sensors can vary greatly in cost. Researchers have historically built PA systems with some type of laser (quantum cascade, distributed feedback, CO\(_2\), etc.) [36] in order to create a high power energy source of a desired wavelength. Lasers require additional equipment including a laser temperature controller and a signal generator. This method is effective, but very expensive [41, 56, 57]. More affordable emission sources include the broadband IR emitter and the LED, though they suffer from lower power and wider bandwidths.

**Laser**

As illustrated in Figure 1.7, a traditional laser based PA sensor generally works as follows.

1. A signal generator modulates the intensity of tunable laser radiation at a resonant frequency of the gas cell
2. The laser irradiates cell gases with high energy radiation within a small wavelength band
3. A microphone detects the pressure wave signal resulting from periodic gas expansion
4. A lock-in amplifier (LIA) selectively amplifies sound within the cell at the laser modulation frequency
Broadband IR Emitter

Broadband IR sources, sometimes known as lamps or thermal emitters, are affordable, but emit radiation over a broad spectrum (several $\mu$m) as the name suggests. Where high selectivity of the target gas is desired, the use of a costly optical band-pass filter is required [48]. They are also generally lower power than a comparable laser source. However, the greater limitation of thermal IR sources lies in their inability to oscillate at a frequency much greater than $\sim$100 Hz due to their relatively slow cool down time [36,58]. A modulation frequency this low can only feasibly be used in a non-resonant PA system. As mentioned in Section 1.3.4, non-resonant systems do not allow for real-time flow, limiting the utility of broadband IR sources in the field [45].

LED

A recently emerging radiation source for PAS is the LED. Though LEDs that emit between 3 and 12$\mu$m (mid-IR) are orders of magnitude lower power [36, 59] than diode lasers of similar wavelength, they are very affordable [31, 59] in comparison. Historically, mid-IR LED’s have not been available but are now commercially available in wavelengths up to nearly 5$\mu$m [59, 60]. As compared to low-cost IR broadband and laser sources, LED’s consume little power and are simple to frequency modulate [59]. Both CO and CO$_2$ have absorption bands between 4 and 5 $\mu$m [31,60], allowing the use of low-cost LEDs in sensor design. Additionally, the spectral emission band of an LED is small enough in many applications to be approximated as a monochromatic source, avoiding the need for an optical filter [58].
Figure 1.8: LED operating principle [4].

The working principle behind an LED is electroluminescence, or the emission of light when a current is passed through a material. An LED contains a semiconductor material with a high concentration of electrons on one side and holes on the other (i.e. material is doped). The boundary is known as a P-N junction. When a current is passed through a P-N junction, high energy electrons recombine with holes, releasing photons in the process. See Figure 1.8. The color, or wavelength, of emitted light is determined entirely by the semiconductor material. [61]

Mid-IR LEDs have only been available on the market since around 1990 and are still generally limited to research applications [59]. Nevertheless, there are several companies from which such components can be ordered at unit prices ranging from $\sim 40$ to a couple hundred dollars. Microsensor Tech in Florida, IBSG and Microsensor NT in Russia, Roithner in Austria, and Hamamatsu in Japan all offer LEDs in the mid-IR range, including $\sim 4.3 \mu m$. Two LED head attachments are commonly offered to focus emission: Chalcogenide Glass (CG) coverings and parabolic reflectors. Additionally, mid-IR photodiodes, LED arrays, and printed circuit board (PCB) LED drivers are available. Mid-IR LEDs are sensitive to electro-static discharge and require careful set-up and handling to ensure functionality. Due to customs and distance, shipping prices can exceed the cost of an LED and minimum order quantities of several hundred dollars are often enforced, favoring bulk orders. International shipping can take over a month. Locally, Terahertz Device Corporation in Salt Lake City sells quantum interband-cascaded superlattice mid-IR LEDs (QuiC SLEDs), but only offers competitive unit prices for bulk orders.

The first LED photoacoustic system was published by the Optical Society of America in 1987 [62]. As a result of the recent availability and affordability of mid-IR LED’s, researchers
over the past 10 years have revived the study of LED sources in PA sensors. Kuusela successfully used a mid-IR LED in place of a laser with a cantilever microphone in a non-resonant cell to measure CO$_2$ and propane concentrations [58]. Over the next few years, Scholz developed her non-resonant PA NDIR sensors using a 4.2$\mu$m and 3.4$\mu$m LED as the radiation source for CO$_2$ and methane, respectively [31–35]. Scholz employed the use of a microelectromechanical system (MEMS) microphone and a sealed cell of the target gas. A year later, both Pernau and Ishaku published the development of the first PA detectors with an LED source that operated in resonant mode with open-flow. However, sensitivity is only presented for very large concentrations of CO$_2$ [52, 63]. Both Pernau and Ishaku used a cylindrical cell with a MEMS microphone, but Ishaku included buffer volumes. Recently, Pernau has contributed to the work of El-Safoury and Weber in developing PAS systems with an LED radiation source for measuring sulfur dioxide and nitric dioxide concentrations [44, 64]. Suitable for combustion emissions monitoring, both sensors operate in resonant mode and use a MEMS microphone.

### 1.3.6 Signal Detection

PA gas sensors are based on a relationship between gas concentration and acoustic signal. In most applications, the strength of the acoustic signal is very low and inaudible to the human ear. For this reason, a sensitive acoustic transducer must be used, most commonly a microphone. Data logging at relatively high frequencies (kHz range) is a necessity to fully capture acoustic wave information in a resonant sensor without misinterpreting the signal (a.k.a. aliasing). Besides being small, the PA signal is also in the midst of literal noise from the surroundings and signal noise inherent to the electronics in use. When the PA signal is completely buried in noise (i.e. the noise floor is stronger than the PA signal), a special technique must be used to extract the signal, namely lock-in amplification.

### Microphone

Capacitive MEMS microphones are commonly used in PAS because of their small size, low-cost, temperature independence, and ever-decreasing self-noise. Many microphone specifications are presented in decibels (dBs), a logarithmic unit equal to a tenth of a bel. The decibel
actually represents the change in power, but is often used with a reference value to give a power magnitude. The decibel change in power is calculated in the following manner.

$$10 \log_{10} \left( \frac{\dot{W}_2}{\dot{W}_1} \right)$$

Since power is proportional to the square of voltage ($\dot{W} = \frac{v^2}{Z}$), the decibel change in voltage is calculated by bringing the exponent value in the logarithm to the multiplier outside the logarithm.

$$20 \log_{10} \left( \frac{v_2}{v_1} \right)$$

The human ear hears loudness on a scale similar to the dB scale. To account for differences at various frequencies, a correction factor is sometimes applied. For example, a dBA is a dB multiplied by the A correction factor at a given frequency.

Self-noise (a.k.a. the noise floor) is the amount of noise from a device when no other sounds are present, and is inherent to the device. Two common specifications describe the self-noise of a microphone: Equivalent Input Noise (EIN) and Signal to Noise Ratio (SNR). They are related through a reference pressure and frequency of 1Pa (94dB) and 1kHz.

$$\text{SNR} = 94\text{dB} - \text{EIN}$$

Historically, MEMS microphones have been limited in their acoustic performance, particularly SNR, making the full-sized Electret Condenser Microphone (ECM) necessary in many applications. However, MEMS microphones of recent years have SNRs as high as 70dBA, comparable to ECM SNRs, while maintaining a much smaller size. [65]

It should be noted that MEMS microphones are surface-mount devices (SMDs), and thus practically require the creation of a PCB in order to use them. Since the sensitivity (size of voltage change produced given a certain pressure change) of a microphone is very small on its own, MEMS microphone signals often also require external amplification.
**Data Logger**

The Nyquist sampling theorem states that data must be sampled at a minimum of twice the frequency of the AC signal to be measured in order to avoid aliasing. For a PA resonant cell with a given fundamental frequency, sampling must occur at greater than twice that frequency in order to capture the full AC signal. Thus, care must be taken to select a data logging setup with enough volatile memory to read microphone data and enough non-volatile memory to store data. When an AC signal is not measured, as with the use of a lock-in amplifier, a simple averaging technique can be used.

**Lock-in Amplifier**

When noise is greater than the PA signal, lock-in amplification (LIA) is a technique to extract the signal. The principle behind LIA is that when two signals are multiplied, the mean value of the resultant signal (DC component) is nonzero only if the the signals have exactly the same frequency (are phase-locked) and the relative phase is not ±90°. A LIA requires a reference signal that is phase-locked with the signal of interest. In practice, the reference signal must usually be taken from the wave driving the PA signal to ensure no relative phase change. In addition to amplification and filtering processes, a LIA multiplies the PA signal with the reference signal and outputs the DC component. The component that multiplies the two signals is known as a phase-sensitive detector (PSD), demodulator, or mixer [66, 67]. A simple LIA generally performs the following operations on a noisy analog signal [68, 69].

1. **Pre-amplification.** Signal is amplified

2. **Anti-aliasing filtration.** Band-pass or low-pass filter removes the majority of unwanted frequencies from signal

3. **Phase-locking.** Phase-locked Loop (PLL) puts reference signal in-phase with signal

4. **Phase-sensitive detection.** PSD multiplies signal and reference signal together

5. **AC filtration.** AC component of product wave is filtered out with a low-pass filter, leaving only a DC component
6. **DC amplification.** DC component is amplified

The reference signal wave can have any waveform as long as it is phase-locked with the PA signal. The PA signal resembles a sine wave \([31]\) for high frequencies and a pulse wave for low frequencies.

**Multiplication of Sine Waves**  The product of two phase-locked \((\omega_1 = \omega_2 = 2\pi f)\) sinusoidal waves of different amplitudes, different offsets, and a relative phase lag of \(\theta\) can be written as

\[
S_1S_2 = [A_1 \sin(\omega t) + c_1] \times [A_2 \sin(\omega t - \theta) + c_2]
\]

Expanding the expression,

\[
S_1S_2 = A_1 A_2 \sin(\omega t) \times \sin(\omega t - \theta) + A_1 c_2 \sin(\omega t) + A_2 c_1 \sin(\omega t - \theta) + c_1 c_2
\]

Applying the trigonometric product formula,

\[
S_1S_2 = \frac{A_1 A_2}{2} [\cos(\theta) - \cos(2\omega t - \theta)] + A_1 c_2 \sin(\omega t) + A_2 c_1 \sin(\omega t - \theta) + c_1 c_2
\]

Recognizing that any sinusoid term containing \(t\) has a mean value of zero, we obtain a DC component \((D)\) for the product wave,

\[
D = \frac{A_1 A_2}{2} \cos(\theta) + c_1 c_2 \quad (1.4)
\]

**Multiplication of Rectangular Waves**  By experimentation in Octave/MATLAB, it was determined that two rectangular waves with different amplitudes, offsets, and duty cycles \((d_c)\) have a maximum product of

\[
D_{\text{max}} = \frac{dc_2}{dc_1} (A_1 A_2 + c_1 c_2) \quad (1.5)
\]

**Multiplication of Sine and Rectangular Waves**  The product of a rectangular wave and a sinusoidal wave of same frequency, different amplitudes, different offsets, and a relative phase lag of
\( \theta \) can be written as

\[
S_1 S_2 = \left[ 2A_1 \left( d_c + \sum_{n=1}^{\infty} \left( \frac{2}{n \pi} \sin(n \pi d_c) \cos(2n \pi f t - n \pi d_c) \right) \right) + c_1 \right] \times \left[ A_2 \sin(\omega t - \theta) + c_2 \right]
\]

Where \( S_1 \) is the Fourier series expression of a rectangular wave, \( S_2 \) is a sine wave with no offset, and \( d_c \) is the rectangular wave duty cycle. Multiplying the \( 2A_1 \) through, expanding the expression, and using the trigonometric product formula, two terms have nonzero DC components:

Term 1 : \( \frac{2A_1}{\pi} \sum_{n=1}^{\infty} \left( \frac{1}{n} \left( \sin(2n \pi f t) + \sin(2n \pi d_c - 2n \pi f t) \right) \right) \times (A_2 \sin(2\pi f t - \theta)) \)

Term 2 : \( c_2 (2A_1 d_c + c_1) \)

Term 2 is a constant, but finding the DC component of Term 1 requires more work. Distributing the sine term, using the trigonometric product formula to express as a sum of sinusoids, and recognizing the pattern results in just two nonzero terms. The DC component can be written as

\[
D_1 = \frac{A_1 A_2}{\pi} (\cos(\theta) - \cos(\theta - 2\pi d_c))
\]

Thus, the final DC component is

\[
D = \frac{A_1 A_2}{\pi} (\cos(\theta) - \cos(\theta - 2\pi d_c)) + c_2 (2A_1 d_c + c_1) \quad (1.6)
\]

It has been verified in Octave/MATLAB that the first term of Equation 1.6 is correct, but the constant offset does not match up. It will be left as an exercise for the reader to correct the offset.

### 1.4 Introduction Summary

In sum, ICSs are a solution to worldwide health and environmental problems due to cookstove emissions, but require accurate and low-cost CO\(_2\) detection. NDIR CO\(_2\) sensors are well-suited for ICS evaluation, but generally cannot accurately measure very low concentrations. PA technology has been used to detect several gases with accuracy and sensitivity in various applications. The cylindrical gas cell with buffer volumes has been used extensively in PA detection due to its practicality and efficacy. Typical PA set-ups are lab-based and are composed of expen-
sive equipment, such as a laser set-up and LIA. However, researchers have had some success with lower-cost components including the mid IR LED and MEMS microphone. The objective of this thesis is to evaluate the suitability of PA CO$_2$ detection for cookstove emissions monitoring.
CHAPTER 2. METHODS

First, a relatively simple model for estimating PA signal strength order of magnitude is derived. Then, an analytical relationship between resonant frequency of a cylinder and concentration of CO\textsubscript{2} is presented. Throughout the duration of this research, several designs were constructed and tested (1, 2a, 2b, 3, 4a, 4b). The manufacturing techniques and data acquisition systems used on each design are discussed. A complete summary of each design is presented in Table 3.1 of Chapter 3: Results and Discussion. Finally, various testing methods and procedures are described.

2.1 Modeling PA Signal Strength

In order to confidently design all subsystems of a PA sensor, an order-of-magnitude estimate for the PA signal strength is necessary. Modeling the PA effect with the 1D wave equation (Equation 1.1) requires two boundary conditions and two initial conditions. The boundary conditions were taken as ambient pressure at either end of the resonator. To find the initial conditions, a thermodynamic analysis was carried out.

2.1.1 Irradiation as a function of path length

Treating the emitted LED radiation as fully collimated, attenuation of the LED power can be modeled with the absorption coefficient, $\kappa_\lambda$. Taking an arbitrary cylindrical cross section of the radiation path with thickness $\Delta z$ as the system (see Figure 2.1) and neglecting any radial heat transfer, an energy balance is taken.

$$\frac{G_\lambda(z + \Delta z) - G_\lambda(z)}{\Delta z} = -\kappa_\lambda G_\lambda(z + \frac{\Delta z}{2})$$
Taking the limit as \( \Delta z \) goes to zero, the rate at which spectral irradiation changes in the \( z \) direction is obtained.

\[
\frac{dG_\lambda}{dz} = -\kappa_\lambda G_\lambda(z)
\]

Separating, integrating with respect to \( z \) from \( z = 0 \) to \( z \), and solving for \( G_\lambda(z) \),

\[
G_\lambda(z) = G_\lambda(0)e^{-\kappa_\lambda z}
\]

Note that this irradiation process occurs at the speed of light, and is thus orders of magnitude greater than the sound speed. Integrating over a spectral band and taking \( \kappa_\lambda \) as an average value across that spectral band, the total irradiation as a function of \( z \) is obtained.

\[
G(z) = G_0e^{\kappa_\lambda \Delta \lambda z}
\]  (2.1)

\( G_0 \) is found from the radiation source emission power and area. The average absorption coefficient is found from a line-by-line simulation from HITRAN on the Web, an online software based on the HITRAN2012 database [5]. The absorption coefficient for the most prominent isotope of CO\(_2\) at 85kPa (atmospheric pressure in Provo, Utah), 296K, and 5000ppm CO\(_2\) (a typical concentration in cookstove emissions) is plotted \( \sim 4.3\mu m \) in Figure 2.2. The integrated average is shown to be \( \sim 0.1\text{cm}^{-1} \).
Figure 2.2: Absorption Coefficient of CO$_2$ plotted over wavelength in the 4.3$\mu$m wavelength region [5].

2.1.2 Initial Pressure Condition

To get the initial pressure ($P_i$), another energy balance is performed on the same control volume as in Figure 2.1, though the interactions are expressed differently (see Figure 2.3).

![Figure 2.3: Cylindrical cross section system and interactions to obtain the initial pressure condition.](image)

Specific heats are considered constant and the irradiation process is approximated as instantaneous because the speed of light is much greater than sound speed in a gas. Additionally, the absorption time is considered instantaneous.

\[
\rho c_v \frac{dT}{dt} = \frac{q}{V}
\]  

(2.2)
Treating the gas as ideal and introducing the absorption coefficient,

\[ \frac{P}{RT}c_v \frac{dT}{dt} = \kappa_{\Delta\lambda} G(z) \]

Expressing \( R \) and \( c_v \) in terms of \( \gamma \) and separating,

\[ \frac{1}{T} \frac{dT}{dt} = \frac{(\gamma - 1) \kappa_{\Delta\lambda} G(z)}{P} \]

Integrating with respect to time across the instantaneous pulse time \( t_p \) and solving for pressure, the pressure distribution immediately after the radiation pulse is obtained.

\[ P_i = \frac{(\gamma - 1)t_p \kappa_{\Delta\lambda} G(z)}{\ln\left(\frac{T_{\text{max}}}{T_0}\right)} \]

Note that an expression for \( G(z) \) is already found (Equation 2.1). To solve for \( P_i \), \( T_{\text{max}} \) must also be obtained. Starting with Equation 2.2 and integrating with respect to time, treating \( \rho \) as constant at ambient density, and solving for \( T_{\text{max}} \),

\[ T_{\text{max}} = T_0 + \frac{t_p \kappa_{\Delta\lambda} G(z)}{c_v \rho_0} \]

Finally, substituting this expression for \( T_{\text{max}} \) into \( P_i \), using the ideal gas law, and expressing \( R \) and \( c_v \) in terms of \( \gamma \),

\[ P_i = \frac{(\gamma - 1) \kappa_{\Delta\lambda} G(z)t_p}{\ln\left(1 + \frac{(\gamma - 1) \kappa_{\Delta\lambda} G(z)t_p}{P_0}\right)} \quad (2.3) \]

For reasonably small lengths and typical 4.3\( \mu \)m LED power, Equation 2.3 gives a pressure distribution on the order of nPa. With resonant amplification, a pressure signal on the order of \( \mu \)Pa is expected.

### 2.1.3 Initial Pressure Time Derivative Condition

Equation 2.3 provides one initial condition, but the initial time derivative of pressure is also needed. This condition is obtained by differentiating (2.3) with respect to pulse time and setting \( t_p = 0 \).
To make Equation 2.3 easier to work with, the numerator is temporarily defined as \( n \) and the argument of the natural log in the denominator is defined as \( a \) (i.e. \( P_i = \frac{n}{\ln(a)} \)). Bringing the denominator of Equation 2.3 to the numerator and raising to the power of -1, the product rule expresses the derivative as

\[
\frac{\partial P_i}{\partial t} = \frac{\partial n}{\partial t} (\ln a)^{-1} + n \frac{\partial (\ln a)}{\partial t}^{-1}
\]

Expressing \( \frac{\partial n}{\partial t} = (\gamma - 1)\kappa_{\Delta \lambda} G(z) \) as \( n' \), \( a = 1 + \frac{n't_p}{P_0} \). Substituting these expressions into the equation for \( \frac{\partial P_i}{\partial t} \), evaluating with the chain rule where necessary, and simplifying,

\[
\frac{\partial P_i}{\partial t} = \frac{n'}{\ln \left( 1 + \frac{n't_p}{P_0} \right)} \left( 1 - \frac{n't_p}{\ln \left( 1 + \frac{n't_p}{P_0} \right) (P_0 + n't_p)} \right)
\]

where \( n' = (\gamma - 1)\kappa_{\Delta \lambda} G(z) \). Setting \( t_p = 0 \), the second initial condition is undefined, but likely equals infinity, which makes sense considering the approximation of an instantaneous pulse. L’Hospital’s rule applied twice still provides an undefined function.

\[
\frac{\partial P_i}{\partial t} = \text{DNE}
\] (2.4)

Though further thought is necessary to identify useful boundary conditions, the initial pressure condition (Equation 2.3) gives reason to expect a pressure signal no lower than \( n \)Pa for a concentration of 5000ppm CO\(_2\).

2.2 Acoustic Method of CO\(_2\) Detection

As has been previously studied [70,71], concentration can be found by determination of the resonant frequency. For a cylinder operating in resonant mode, as shown in Equation 1.3, resonant cell length, resonant frequency, and the speed of sound are related. By substituting in the lossless expression for the speed of sound (\( \sqrt{\gamma RT} \)) and solving for the gas-dependent properties (\( \gamma R \)),

\[
\gamma R = \frac{4L^2 f_r^2}{T}
\] (2.5)

24
Approximating the gas mixture as composed of only air ($\gamma = 1.40, R = 287 \frac{J}{kgK}$) and additional CO₂ ($\gamma = 1.29, R = 189 \frac{J}{kgK}$), the sum of the mass fractions of air and CO₂ above the ambient level (CO₂⁺) must be unity [72].

\[
\frac{m_{air}}{m_{mix}} + \frac{m_{CO₂⁺}}{m_{mix}} = 1
\]

Defining $x_+$ as the fraction of CO₂ above the ambient CO₂ level ($x_+ = \frac{m_{CO₂⁺}}{m_{mix}}$) and expressing $\gamma$ and $R$ of the gas mixture as weighted averages of that of pure air and CO₂,

\[
\gamma_{mix} = (x_+)\gamma_{CO₂} + (1 - x_+)\gamma_{air}
\]
\[
R_{mix} = (x_+)R_{CO₂} + (1 - x_+)R_{air}
\]

Multiplying $\gamma_{mix}$ by $R_{mix}$, simplifying, rearranging, and substituting in Equation 2.5 for $\gamma_{mix}R_{mix}$, the equation is presented in quadratic form.

\[
a x_+^2 + b x_+ + d = 0
\]
\[
a = \left[\gamma_{CO₂}R_{CO₂} - \gamma_{CO₂}R_{air} - \gamma_{air}R_{CO₂} + \gamma_{air}R_{air}\right]
\]
\[
b = \left[\gamma_{CO₂}R_{air} + \gamma_{air}R_{CO₂} - 2\gamma_{air}R_{air}\right]
\]
\[
d = \left[\gamma_{air}R_{air} - \left(\frac{4L^2f_r^2}{T}\right)_{mix}\right]
\]

Solving for $x_+$ with the quadratic formula and selecting the useful solution,

\[
x_+ = -\frac{b - \sqrt{b^2 - 4ad}}{2a} \quad (2.6)
\]

Thus, for a fixed geometry and known temperature, concentration of CO₂ can be calculated as a function of resonance frequency alone. This method, as presented in Equation 2.6, is called the “Acoustic Method of CO₂ Detection” by the author, or just the “Acoustic Method” for short. To obtain total CO₂ concentration, one must add the ambient concentration. For accuracy, the length of the resonator should be found by experimentally determining the resonant frequency with room-air and using Equation 1.3 to back-solve for length. If the resonator is an open-open cylinder, end corrections should be used.
2.3 Manufacturing Processes

This section provides details of the manufacturing process for each design.

2.3.1 Design 1

The cell for Design 1 was modeled with CAD and 3D printed in PLA plastic to contain a cylindrical cavity with a resonant frequency matching the LED modulation frequency of 1kHz. See Figure 2.4. It was composed of two pieces: one that formed a cell end and held the LED driver board and another that contained the cylindrical resonator, inlet, and outlet. The two pieces held an optical filter in place via four bolts and nuts. The inlet and outlet were tapped for pipe fittings.

![3D printed Design 1 cell.](image)

2.3.2 Design 2a,b

The shared cell design for Designs 2a and 2b was modeled with CAD, then manufactured as seen in Figure 2.5. The end caps and microphone holder were 3D printed and respectively bolted together. A half-inch sapphire window and O-ring was placed between the two bolted pieces of each end cap. The buffer holes were drilled (d=1.5in, 8.0cm deep) in stock aluminum (d=1.75in, L=13.5cm). A through-hole was reamed to d=5/8in. The resonator tube was made from a stock aluminum tube of 0.495in inner diameter and 0.065in thickness. The outside was polished with a metal grinding compound to fit. O-rings were used to keep a gas-tight seal and oil was applied to the resonator to ensure a smooth slide.
2.3.3 Design 3

First, a cut of aluminum tubing (1.0cm ID, 1.2cm OD) was turned down to a length of 4.5cm on a manual lathe. Next, two holes for mounting the microphone PCB and one hole for the microphone were milled. The microphone hole was located 0.3cm from the end and was 0.15cm in diameter. The two mounting holes were tapped for M3 screws. Yellow gas-grade PTFE tape was used with the screws to try and achieve a CO₂-proof seal. Two 0.28cm holes were drilled \(\sim\)0.7cm from either end for the inlet and exit, which PVC tubing was fit into. The LED parabolic reflector was sealed inside the resonator with a section of 1/32in PVC tubing. The other window was bonded to the resonator end with epoxy. Yellow gas-grade Teflon tape was used with the M3 screws to provide a seal. See Figure 2.6.

2.3.4 Design 4a,b

The shared cell for Design 4a and 4b was composed primarily of a resonator piece and two buffer volume pieces. One cut of \(\sim\)4cm and another of \(\sim\)8cm was made from stock aluminum (d=7.6cm). See Figure 2.7.
Lathe

The longer piece was drilled (15/16in) and reamed (1in) to a depth of $\sim 4.5\text{cm}$. A relief groove was cut $\sim 1.3$ to $\sim 1.5\text{cm}$ from the edge, $\sim 0.2\text{cm}$ deep. The manual lathe (in threading gear) was used to thread the outside up to the relief groove at 16 threads-per-inch (tpi) until the threads were fully formed. The longer stock was then cut into two pieces: one for the resonator and one for a buffer. The other end of the resonator was faced to a total length of 4cm and threaded like the other end.

Next, a buffer piece was turned down to $d=6.94\text{cm}$ and faced to a 3.8cm length. It was then drilled ($d=1\text{in}$) and bored (up to $d=6.575\text{cm}$) to a depth of 3.3cm. Additional boring (up to $d=7.75\text{cm}$) was done to a depth of 1.3cm. A relief groove was cut on the inside from $\sim 1.1$ to $\sim 1.25\text{cm}$, 0.2cm deep. The manual lathe (in threading gear) was used to thread the inside at 16tpi up to the relief groove. Passes were repeated until the threads were deep enough to fit the resonator. This process was repeated for the other buffer piece.

For one buffer piece, a 0.8cm hole for the LED was drilled through, a 1cm hole was drilled 0.25cm
deep. For the other buffer, a 1.24cm hole was drilled through, a 1.3cm hole was drilled for the Sapphire window to a depth of 0.45cm, and a 1.75cm hole was bored 0.35cm deep for an O-ring.

For Design 4b, a Sapphire window was installed on the LED buffer side in the same way as the exit buffer side.

**Mill**

A 7/64in drill was used about 0.7cm from the edge of each buffer piece for the inlet and exit. A sunken rectangle was milled for the microphone PCB. Corner holes were mill drilled (d=1/8in) to a quarter-inch depth to eliminate corner radii, but only a 0.1cm clearance and 0.2in depth was given for the rest of the indent. The microphone hole was drilled through, and the two screw holes were drilled partway, all with a #40 drill. The screw holes were then tapped with an M3x0.5 tap.

Yellow gas-grade Teflon tape was used with the threads, an O-ring with the window, and the LED PR itself wrapped in PVC tubing to seal the vessel.

### 2.3.5 Design 5

The cell for Design 5 consisted of only an aluminum resonator and a sapphire exit window. The LED itself provided the inlet window. The aluminum cell was machined from stock aluminum round (d=1.5in) using both a manual lathe and mill. See Figure 2.8.

**Lathe**

First, a piece of Aluminum round (d=1.5in) was faced to a length of 4.5cm. A 29/32in hole was drilled to a depth of 1.75in, then reamed (15/16in). A boring bar was used to create a hole for the 1in sapphire window to a depth of .15in.

**Mill**

After the lathe was used to machine the sapphire window hole, the mill was used to machine the LED slot. A 3/16in end mill bit was used to mill a 1.20cm near-square to a depth of .3cm and a 2.42cm near-square through. The term near-square is used because the round geometry
Figure 2.8: Machined Design 5 cell. Square LED hole on left, large sapphire window hole on right.

of the end mill bit prevented the squares from having perfect 90deg corners. Two 27/64in holes were drilled about a quarter inch from either end of the cell for an inlet and exit. To mount the microphone, a 27/64in through-hole was drilled .23in from the Sapphire Window side of the cell and two 27/64in holes were drilled a quarter inch deep in the PCB locations shown in Figure 2.23. The two mounting holes were then M3x0.5 tapped.

Finally, the Sapphire window was epoxied in its place and the square LED with reflector package was inserted into the cell with a flexible PVC tubing seal. In response to the LED package blowing out from the cell and the LED window coming loose, the LED package and PVC seal were epoxied to the cell and the LED window was epoxied to its package.

2.4 Data Acquisition, Data Processing, and Electronics

First, tests evaluating the operation of two LIAs (a low-cost AD630-based LIA and the SR830) are presented. Then, tests demonstrating the correct operation of three mid IR LEDs (Roithner LED43-RW, Microsensor Technology Lms43LED-CG, and Hamamatsu L13201-0430M with x3.3 Package) with a mid IR photodiode (Lms43PD-03-CG) are detailed. The results of these tests were not obtained until Design 2b, but helped clarify design decisions involving LEDs and LIAs in the later designs.
Since greater understanding of PA data acquisition came with each new design, the PA data acquisition set-up evolved over time. In addition to the LIA and LEDs tests, the data acquisition approach and its evolution with each design is discussed. For every design, an Arduino Uno or Mega was used to power a MEMs microphone board and log data with a micro SD module, as shown in Figure 2.9. The Arduino was also used to drive the LED and buzzer after Design 2a.

![Arduino with micro SD card module and MEMS microphone](image)

Figure 2.9: An Arduino with a micro SD card module connected on the left and MEMS microphone connected on top, typical for all of the PA designs.

### 2.4.1 Lock-in Amplifier Evaluation

Two LIAs were evaluated: an inexpensive LIA sold online (based on the AD630) and a high-end LIA for lab-use (SR830).

**Lock-in Amplifier: AD630 based**

Relatively inexpensive LIAs based on the AD630 PSD with almost no documentation were ordered from Chinese sellers. As a simple test, a 1kHz square wave was supplied as the reference signal and a voltage division of the reference was used as the input signal. By changing a resistor
value in the voltage divider, the input signal was varied. In theory, the product of two phase-locked square waves is known (see Section 1.3.6). This 1kHz test was performed first with Arduino PWM and SD datalogging and then with a function generator and o-scope.

**Arduino PWM and SD**  The 1kHz test was first performed with an Arduino PWM square wave and SD data logger. An Arduino Mega provided the reference (2.5V amplitude square wave), the input (voltage division of reference), and datalogging (micro SD card). A block diagram of the set-up is shown in Figure 2.10. See Figure 2.11 for results.

![Figure 2.10: Electronic block diagram of 1kHz test with Arduino and SD card.](image)

**Function Generator and O-scope**  The 1kHz square wave test was also performed with the 3310A HP function generator and an o-scope. The function generator provided the reference (2.73V amplitude square wave), the input (voltage division of 0.17V square wave), and measurements were obtained with an o-scope. Note that the o-scope could not measure an amplitude less than 0.6mV. A block diagram of the set-up is shown in Figure 2.12. See Figure 2.13 for results.

It was found that the lower limit for the input signal of the purchased LIA with a reference amplitude of around 2.7V was in the µV range. However, due to the inconsistent and low gain and unknown target operating frequency, it was decided that designing and building an inexpensive
Figure 2.11: Purchased AD630-based LIA output plotted against various amplitude Arduino PWM input signals.

Figure 2.12: Electronic block diagram of 1kHz test with function generator and o-scope.

LIA based on literature was the better low-cost route to take [69]. To save time, however, the lab grade SR830 LIA was actually used in all PA set-ups.

**Lock-in Amplifier: SR830**

A simple test, measuring microphone signal with a piezo buzzer at various distances from the microphone, was first performed to increase confidence that the SR830 unit was working (see Figure 2.14). The low-cost piezo buzzer was driven by Arduino PWM at 1.5kHz.
2.4.2 Verification of LED operation with Photodiode

It became important to verify that the LEDs were working. Since they operate in the mid IR, radiation is invisible and very low intensity. Previous attempts at measuring the spectral emission of the LEDs in a Fourier Transform Infrared (FTIR) spectrometer failed to produce any discernible signal, likely due to the small signal strength. Similarly, simple attempts to measure the LED output with a mid IR photodiode (PD43-03-3) proved futile. Through correspondence
with the LED and PD manufacturers, it was determined that the PD signal should be on the \( \mu \)V order. To mitigate noise and amplify the signal, a PD with an amplifying glass cover was set-up in a photovoltaic circuit (see Figure 2.15) and the SR830 LIA was used to detect the PD signal. Since the circuit is inverting, +8V was used both to power the op-amp (OP07) and as ground for the PD circuit. This left a possible signal output range of 0 to -8V. The third PD pin was connected to ground. The selected resistor was \( \sim 1k\Omega \) and the SR830 further amplified the potential. To reduce noise, almost all connections were soldered and a single power supply of +8V was used to power the LED driver, the PD circuit, and the Arduino Uno.

The LED and PD set-up included the Arduino Uno for PWM and data logging, the LED and driver with a 36\( \Omega \) resistor/heatsink, a hand-soldered PD board with the OP07 op-amp and 1k\( \Omega \) resistor, and the SR830 LIA. See Figure 2.16. Three LEDs were tested: LmsLED43-CG, LED43-R, and L13201-0430M.
LmsLED43-CG

The functionality of the LED (Microsensor Technology Lms43LED-CG) was tested at 1.5kHz and 50\% duty cycle with a PD (Lms43PD-03-CG), as shown in Figure 2.17. The \( \sim 1 \) minute long periods at \( \sim 2 \mu V \) show when the LED was pointed away from the PD. The peaks (left to right) show when the LED was aligned with the PD at the following distances: 0.5, 1, 2, 3, and 4 cm. Note that little to no signal is detected at the 4cm distance.

![Figure 2.17: Lms43LED-CG test with PD. The peaks show when the LED was aligned with the PD at the following distances: 0.5, 1, 2, 3, and 4 cm.](image)

L13201-0430M

The functionality of the LED (Hamamatsu L13201-0430M with x3.3 Package) was tested at 1.5kHz and 25\% duty cycle with a PD (Lms43PD-03-CG), as shown in Figure 2.18. The \( \sim 1 \) minute long periods at \( \sim 2 \mu V \) represent when the LED was pointed away from the PD. The peaks (left to right) show when the LED was aligned with the PD at the following distances: 0.5, 2, and 4 cm. Again, note that the data at 4cm is practically indistinguishable from no signal.
The functionality of the LED (Roithner LED43-R) was tested at 1.5kHz and 50% duty cycle with a PD (Lms43PD-03-CG), as shown in Figure 2.19. The ∼1 minute long periods at ∼2µV represent when the LED was pointed away from the PD. The peaks (left to right) show when the LED was aligned with the PD at the following distances: 0.5, 2, 4, and 6 cm. Due to the difficulty of perfectly orienting the LED at large distances, the data at 6cm is inconsistent, but is still substantial compared to the 0.5cm signal. Therefore, the parabolic reflector significantly improves the long range power of the LED radiation, as expected.

Recognizing that the PA cell for Design 2a was much longer than 4cm, an LED with a parabolic reflector package (i.e. LED43-R) was chosen for subsequent designs because collimated radiation provides the most radiant power to the gases in the cell instead of the cell walls. Also, PAS Design 2a was designed for the LED43-R, and thus physically fit Design 2b with no alterations.

2.4.3 Design 1

The radiation source consisted of an LED of 4.3µm central wavelength with a parabolic reflector (Roithner LED43-R) and an LED driver (Roithner PCB-mdriver-QCW) for powering the
LED with a $\sim 1$kHz square wave (see Figure 2.20). The signal driver board eliminated the need for a chopper wheel.

The Sparkfun ADMP401 MEMs microphone and breakout board were used to detect pressure. The LED driver frequency of $\sim 1$kHz drove the rate at which sound waves were produced in the photoacoustic cell. Because audio data represents a wave, the sampling frequency must be at least double the sound frequency (i.e. the Nyquist Frequency) to avoid aliasing. Thus, the sound wave frequency of Design 1 was $\sim 1$kHz and the Nyquist frequency was $\sim 2$kHz. Data was collected via an Arduino Uno and micro SD card module, but at a frequency much too low to avoid aliasing.
2.4.4 Design 2a

The same electronics from Design 1 were used with Design 2a. However, a custom Arduino Uno C++ script (see Appendix B) used a two-buffer data collection technique to read data from the microphone and write raw binary data to the SD card at a sampling frequency of 4kHz to avoid aliasing.

Once raw binary data was logged, a MATLAB script (see Appendix C) converted it from binary to ASCII. An audio amplitude was then obtained for every few seconds of data by performing an FFT and finding the maximum amplitude near the photoacoustic sound wave frequency. Finally, the microphone FFT magnitudes were saved and plotted over time.

2.4.5 Design 2b

The data acquisition system for Design 2b was significantly upgraded from Design 2a. The sensor is comprised of four basic electronic subsystems: LED, Microcontroller, Microphone, and LIA. A block diagram of the full set-up is shown in Figure 2.21. Note that the only difference for tests involving the piezo buzzer is that the LIA reference comes from the buzzer Arduino, which is powered independently through USB, rather than the LED Arduino.

![Figure 2.21: Electronic block diagram for the set-up of Design 2b and later designs.](image-url)
LED

Instead of using the Roithner driver board, the 4.3µm LED was driven by Arduino PWM on a custom PCB. Note that the LED and requires low-temperature (<180°C) solder. The LED was driven from a +8V source through a P-channel MOSFET (SI2325DS-T1-GE3) whose gate was connected to the +5V PWM signal from the Arduino. The circuit used to operate the LED is shown in Figure 2.22. A typical quarter-watt resistor was not sufficient for the power through this circuit, so higher power resistors were used. See Appendix C for LED circuit calculations.

![Figure 2.22: Circuit used to drive LED with +8V through a MOSFET. PCB layout for custom LED circuit, as modeled in Dip Trace (units in mm) shown on right.](image)

Microcontroller

Again, the selected microcontroller was an Arduino. Custom Arduino C++ code was used both to log data from the LIA and to produce a square wave via PWM to drive the LED with a duty cycle anywhere between 1 and 50 % (see Appendix B). The raw binary data was unpacked and plotted in Octave/MATLAB (see Appendix C).

Microphone

To reduce microphone signal noise, the ADMP401 (SNR = 62dBA) and Sparkfun amplification board were replaced with the the Knowles SPM0687LR5H-1 MEMS microphone (SNR = 70dBA) and a custom-designed breakout board. The PCB was designed in DipTrace software
(free version) and did not include any amplification circuitry, leaving all necessary amplification to the LIA. See Figure 2.23.

**Lock-in Amplifier**

A LIA can often be used to measure an AC signal that is small relative to noise (see Section 1.3.6). Instead of filtering raw AC data in software, a LIA uses active hardware filters, amplifiers, and a PSD to filter out noise and amplify a DC signal proportional to the input amplitude. As the original goal of this research was to develop an inexpensive PA detector, an inexpensive LIA was highly desired. Low-cost LIAs ordered from China were tested and rejected and a low-cost LIA based on Liu’s article was partially designed, but not constructed due to time limitations [68]. Instead, the SR830 was used to acquire the microphone signal for convenience and reliability. Since the output of a LIA is a DC signal, the Nyquist sampling theorem is not relevant. However, the LIA data was collected at \(\sim 100\)Hz so that it could be averaged every few seconds in post-processing. Note that a voltage divider was used to bring the CH1 output of the SR830 (up to 10.5V) down to Arduino-friendly levels (less than 5V) so that the phase-independent DC output (R) could be logged safely.
2.4.6 Design 3

The same Arduino set-up, C++ code, microphone circuit, and LIA (SR830) used in Design 2b were used with Design 3, but at higher LED driving frequency. The Roithner LED43-RW was employed in this design.

2.4.7 Design 4a

The same Arduino set-up, C++ code, microphone circuit, LIA (SR830), and LED (LED43-RW) used in Design 3 were used with Design 4a, but at lower LED driving frequency. To reduce noise, BNC cable lengths were significantly reduced.

2.4.8 Design 4b

The same Arduino set-up, C++ code, microphone circuit, and LIA (SR830) used in Design 4a were used with Design 4b, but the Roithner LED43-R was employed instead of the LED43-RW.

2.4.9 Design 5

The same Arduino set-up design, C++ code, microphone circuit, and LIA (SR830) used in Design 4a were used with Design 5, but the Hamamatsu L15895-0430ML-SAMPLE was employed instead of a Roithner LED43 unit. Note that the L15895-0430ML-SAMPLE is the same as the Hamamatsu L13201-0430M, but with a parabolic reflector and square window package. Additionally, the SD datalogging module and voltage divider were soldered to the Arduino in an effort to further minimize electrical noise.

2.5 CO₂ Sensor Testing

The set-up and procedures for both acoustic and photoacoustic tests for each design are described. Finally, calibration techniques are discussed.
2.5.1 Test Set-up and Procedures

Though the ultimate goal was to measure PA signals, several purely acoustic tests were also important. Since CO\textsubscript{2} was the target gas for measurement, the method of injecting and ejecting CO\textsubscript{2} into the resonant cell is described first.

**CO\textsubscript{2} Injection and Ejection**

A simple set-up comprised of CO\textsubscript{2} cartridges designed for bicycle tire inflation, an associated valve, and a ball pump needle provided the test gas needed to evaluate sensor operation. See Figure 2.24.

In a typical test,

1. The valve is opened and CO\textsubscript{2} flushes the cell to near 100\% concentration

2. The inlet and outlet are plugged for a period of time, then opened

3. An acoustic source (buzzer or LED) produces a microphone signal

4. CO\textsubscript{2} concentration is decreased by either natural leakage or manual air pumping
Resonance Profile by Frequency Sweep

In order to determine resonance frequency of a resonator under relatively steady conditions, a series of acoustic signals at differing frequencies were measured by the resonant cell microphone. The resonant frequency is the frequency with the highest measured acoustic signal, and a profile is observed around this frequency, such as shown in Figure 2.25. This test was performed differently for later designs than early designs, but the concept remains unchanged.

**Designs 1 - 2a** Before a high-performance LIA was employed, as in later designs, resonant frequency profiles were determined by use of an inexpensive earbud headphone and tones produced by videos on YouTube. The earbud was secured near the end of the cell and the amplified microphone breakout board signal was logged via Arduino and SD card. In Octave/MATLAB, an FFT plot would reveal signal strength at the tone frequency. By obtaining FFT magnitudes at various tone frequencies, resonant profiles were obtained.

**Designs 2b - 4b** With Design 2b and later, the amplified microphone breakout board was replaced by a lower-noise, but un-amplified microphone and board, combined with Lock-in Amplification via the SR830. Thus, FFT magnitudes were replaced by LIA DC signals. Additionally, the YouTube tone-playing earbud was replaced by a piezo buzzer driven at the desired frequency with
an Arduino. Aside from using different electronics, the method for obtaining resonant profiles was fundamentally the same: the resonant cell microphone was exposed to various frequencies and the corresponding signal strength was recorded.

**Design 5** A code was written for Design 5 that automatically swept through buzzer frequencies (see Appendix B). Resonance was identified, as usual, by an increased signal profile around the resonant frequency.

**Resonant Profiles by Decreasing Concentration**

In addition to obtaining resonance profiles by frequency sweeps, profiles were obtained by holding an acoustic signal at a constant frequency above the original resonant frequency and decreasing concentration of CO$_2$ in the cell, either by manual air pumping or by natural leakage, until a resonant profile was observed. Profiles were seen because as concentration of CO$_2$ decreases in air, resonant frequency increases. By changing the acoustic frequency once a resonance profile was passed, multiple profiles could be obtained in one test, as seen in Figure 2.26. Note that these tests were only performed for Design 3 and later and that both piezo buzzers and 4.3μm LEDs were used as acoustic sources. Though not often thought of as an acoustic source, the LED radiation created a PA sound that was amplified when resonance was reached.

![Figure 2.26: Left: Resonance peaks due to buzzer from 2774 to 3374Hz for Design 4. Right: Resonant peak due to PA effect at 3000Hz for Design 4.](image-url)
With Design 5, resonant profiles were obtained by decreasing concentration CO$_2$ through natural leakage. Extra care was given to performing each test in a repeatable manner. The process sheet is shown in Appendix F.

2.5.2 Calibration

A typical calibration of a gas sensor can be done by diluting a gas of high CO$_2$ content with a gas void of CO$_2$ to produce various levels of CO$_2$. This is typically done with a gas tank containing a known concentration of CO$_2$ and a tank of synthetic air or N$_2$. An alternative method of calibration is to measure the resonant frequency and use the Acoustic Method to determine CO$_2$ concentration at various levels.
CHAPTER 3. RESULTS AND DISCUSSION

3.1 Acoustic Method of CO$_2$ Detection

As derived in Section 2.2, Equation 2.6 (shown again below) provides a way to calculate concentration of CO$_2$ above ambient levels ($x_+$) in air based on the resonance frequency ($f_r$) of a given cylindrical cell. The concept of determining concentration by the resonant frequency is not new [70, 71], but as far as the author is aware, hasn’t been presented in the same way before. Note that in the presented figures, temperature was assumed constant, but that temperature variation can easily be accounted for if temperature data are logged during testing.

$$x_+ = -b - \frac{\sqrt{b^2 - 4ad}}{2a}$$

$$a = [\gamma_{CO_2}R_{CO_2} - \gamma_{CO_2}R_{air} - \gamma_{air}R_{CO_2} + \gamma_{air}R_{air}]$$

$$b = [\gamma_{CO_2}R_{air} + \gamma_{air}R_{CO_2} - 2\gamma_{air}R_{air}]$$

$$d = [\gamma_{air}R_{air} - \left(\frac{4L^2f_r^2}{T}\right)_{mix}]$$

In Figure 3.1, concentration is plotted against resonant frequency for three resonator lengths using the Acoustic Method (Equation 2.6). The slope steepens with increasing resonator length, meaning that longer cell designs are highly sensitive to resonant frequency changes. For this reason, it appears that the Acoustic Method is better suited for shorter cells. Note that for absolute concentration, ambient CO$_2$ concentration must be added.

3.2 Designs

All of the designs operate generally in the following manner, as illustrated in Figure 3.2. Note that some of the cell designs include buffer volumes of a quarter-acoustic wavelength on either end of the resonator, as shown in Figure 1.6.
Figure 3.1: Concentration plotted over resonant frequency for three cell lengths using the Acoustic Method at T=294K.

Figure 3.2: A generic schematic for PA designs.

- Radiant power from an LED is driven at a fixed frequency and 50% duty cycle by a driver board
- The LED, equipped with a parabolic reflector, emits radiation with wavelengths ranging from 3.2-5µm and peaking around 4.2µm
- The radiation passes through an entrance window (1), cell gases, and an exit window (2)
- A MEMS microphone PCB positioned along the resonator detects the pressure waves within the resonator
- Microphone data is logged to an SD card at a fixed frequency with an Arduino and processed
A summary of all of the designs is presented in Table 3.1. See Figure 3.2 for reference. For additional detail on the manufacturing and electronics, see Sections 2.3 and 2.4, respectively.

Table 3.1: Complete Design Table.

<table>
<thead>
<tr>
<th>Design</th>
<th>1</th>
<th>2a</th>
<th>2b</th>
<th>3</th>
<th>4a</th>
<th>4b</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td>L x D (cm)</td>
<td>17.5 x 1.6</td>
<td>14-18 x 2.5,3.2</td>
<td>13.5 x 2.5,3.2</td>
<td>4.09 x 1.0</td>
<td>3.90 x &lt;2.54</td>
<td>3.90 x &lt;2.54</td>
</tr>
<tr>
<td>Material</td>
<td>PLA</td>
<td>~1000 AL</td>
<td>Al</td>
<td>Al</td>
<td>4200 AL</td>
<td>Al</td>
<td>3480 Al</td>
</tr>
<tr>
<td>Buffers</td>
<td>L x D (cm)</td>
<td>NA</td>
<td>8.0 x 7.6</td>
<td>NA</td>
<td>2.0 x 6.8</td>
<td>NA</td>
<td>2.0 x 6.8</td>
</tr>
<tr>
<td>Material</td>
<td>NA</td>
<td>PLA x2</td>
<td>PLA x2</td>
<td>NA</td>
<td>Al x2</td>
<td>Al x2</td>
<td>Al x2</td>
</tr>
<tr>
<td>Windows</td>
<td>D1 (cm)</td>
<td>&lt;2.54</td>
<td>&lt;1.27</td>
<td>NA**</td>
<td>NA**</td>
<td>&lt;1.27</td>
<td>NA**</td>
</tr>
<tr>
<td>Material 1</td>
<td>Optical Filter</td>
<td>Sapphire</td>
<td>Sapphire</td>
<td>NA**</td>
<td>NA**</td>
<td>&lt;1.27</td>
<td>NA**</td>
</tr>
<tr>
<td>Material 2</td>
<td>Solid PLA*</td>
<td>Sapphire</td>
<td>Sapphire</td>
<td>Sapphire</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
</tr>
<tr>
<td>Radiation</td>
<td>LED</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-R</td>
</tr>
<tr>
<td>Driver</td>
<td>Roithner QCW</td>
<td>Roithner QCW</td>
<td>Custom, PWM</td>
<td>Custom, PWM</td>
<td>Custom, PWM</td>
<td>Custom, PWM</td>
<td>Custom, PWM</td>
</tr>
<tr>
<td>Microphone</td>
<td>Mic.</td>
<td>ADMP401</td>
<td>BOB-09868</td>
<td>BOB-09868</td>
<td>BOB-09868</td>
<td>BOB-09868</td>
<td>BOB-09868</td>
</tr>
<tr>
<td>Location</td>
<td>Center</td>
<td>ADMP401</td>
<td>SPM0687.</td>
<td>SPM0687.</td>
<td>SPM0687.</td>
<td>SPM0687.</td>
<td>SPM0687.</td>
</tr>
<tr>
<td>Data</td>
<td>flog (Hz)</td>
<td>&lt;10Hz</td>
<td>4000</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Processing</td>
<td>NA</td>
<td>Avg FFT</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
</tr>
<tr>
<td>Notes</td>
<td>-</td>
<td>Inoperative</td>
<td>Need LIA</td>
<td>Too big</td>
<td>PA wall abs.</td>
<td>PA wall abs.</td>
<td>PA wall abs.</td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall

**LED package contained a window, which formed the cell wall and first window

3.2.1 Design 1

Table 3.2: Design 1 Table.

<table>
<thead>
<tr>
<th>Design</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td>L x D (cm)</td>
</tr>
<tr>
<td>Material</td>
<td>PLA</td>
</tr>
<tr>
<td>Air f_{res} (Hz)</td>
<td>~1000</td>
</tr>
<tr>
<td>Buffers</td>
<td>L x D (cm)</td>
</tr>
<tr>
<td>Material</td>
<td>NA</td>
</tr>
<tr>
<td>Windows</td>
<td>D1 (cm)</td>
</tr>
<tr>
<td>Material 1</td>
<td>Optical Filter</td>
</tr>
<tr>
<td>Material 2</td>
<td>Solid PLA*</td>
</tr>
<tr>
<td>Radiation</td>
<td>LED</td>
</tr>
<tr>
<td>Driver</td>
<td>Roithner QCW</td>
</tr>
<tr>
<td>Microphone</td>
<td>Mic.</td>
</tr>
<tr>
<td>Location</td>
<td>Center</td>
</tr>
<tr>
<td>Data</td>
<td>flog (Hz)</td>
</tr>
<tr>
<td>Processing</td>
<td>NA</td>
</tr>
<tr>
<td>Notes</td>
<td>-</td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall
The first photoacoustic system design is described in Table 3.2 and illustrated in Figure 3.3.

![Figure 3.3: Schematic of PA sensor Design 1. Note that it was not functional.](image)

**Design Considerations**

An LED of 4.3μm central wavelength with a parabolic reflector (Roithner LED43-R) and an LED driver (Roithner QCW-mdriver-QCW) that powers the LED with a square wave (50% duty cycle) were obtained. The LED was selected due to its relatively low cost and the driver board was desired for convenience and because it eliminated the need for a chopper wheel. An expensive band-pass filter was used to limit LED radiation to only the CO$_2$ absorption spectrum. The resonant cell was 3D printed to contain a cylindrical cavity with a resonant frequency matching the LED modulation frequency of 1kHz (similar to Bauer’s work [41, 42]). Advantages of 3D printing include simplicity and low cost. Also for convenience, a MEMs microphone and breakout board from Sparkfun were used.

**Test Results**

Design 1 was subject to several discrete concentrations of a gas mixture containing CO and CO$_2$ to verify that the acoustic signal was proportional to gas concentration. Unfortunately, the signal was either entirely buried under noise or not present at all, and the data was useless. There are several significant problems with this design.

**Unsuitable Data Acquisition**  The microphone (ADMP401), mounted on an associated breakout and amplification board (SparkFun), was positioned halfway along the length of the cylinder. This
is a pressure node in a simple cylinder, and thus the point of weakest sound to detect. Additionally, the pressure data from the microphone was simply used to estimate a sound wave amplitude over a small period of time and was totally unfiltered, allowing noise to dominate the signal of interest. Since no exit window was included, there would also have been significant absorption noise at the LED driver frequency.

**Trouble Exploiting Resonance**  The entire cell was 3D printed (Fused Deposition Modeling) in PLA plastic with no polishing. The rough inside surface, combined with relatively poor acoustic reflection due to soft material, hinder the formation of a standing wave for resonance. The fundamental frequency of the cylinder was not experimentally determined, and could have varied significantly from the LED modulation frequency, further hindering acoustic resonance.

**Weak Signal**  Perhaps if resonance was achieved and the audio data was properly collected and processed, a signal could have been detected. However, the low power LED (∼10µW), large resonant cell (>300cm³), and narrow band optical filter (see Appendix E) would have resulted in a very low signal difficult to detect even with an appropriate set-up.

**3.2.2 Design 2a**

Considering the problems associated with Design 1, a second design was made as described in Table 3.3 and shown in Figure 3.4.
Table 3.3: Design 2a Table.

<table>
<thead>
<tr>
<th>Design</th>
<th>-</th>
<th>1</th>
<th>2a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L x D (cm)</td>
<td></td>
<td>17.5 x 1.6</td>
<td>14-18 x 2.5,3.2</td>
</tr>
<tr>
<td>Air $f_{res}$ (Hz)</td>
<td>~1000</td>
<td>~1000</td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>PLA</td>
<td>PLA</td>
<td></td>
</tr>
<tr>
<td>Buffers</td>
<td></td>
<td>NA</td>
<td>8.0 x 7.6</td>
</tr>
<tr>
<td>L x D (cm)</td>
<td></td>
<td>NA</td>
<td>PLA x2</td>
</tr>
<tr>
<td>Material</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_1$ (cm)</td>
<td>&lt;2.54</td>
<td>&lt;1.27</td>
<td></td>
</tr>
<tr>
<td>Material 1</td>
<td>Optical Filter</td>
<td>Sapphire</td>
<td></td>
</tr>
<tr>
<td>$D_2$ (cm)</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material 2</td>
<td>Solid PLA*</td>
<td>&lt;1.27</td>
<td></td>
</tr>
<tr>
<td>Radiation</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td></td>
</tr>
<tr>
<td>LED Driver</td>
<td>Roithner QCW</td>
<td>Roithner QCW</td>
<td></td>
</tr>
<tr>
<td>Microphone</td>
<td>ADMP401</td>
<td>ADMP401</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>BOB-09868 Center</td>
<td>BOB-09868 Center</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_{log}$ (Hz)</td>
<td>&lt;10Hz</td>
<td>4000</td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td>NA</td>
<td>Avg FFT</td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td></td>
<td>Inoperative</td>
<td>Need LIA</td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall

Design Considerations

The second design is essentially a 1D pipe resonator with buffer volumes on either side (see Figure 1.6). The first longitudinal resonant frequency is exploited. The buffers serve two purposes: minimize acoustic noise and increase resonant quality [46,52]. The design includes the same LED and ~1kHz driver board as Design 1 because of its low cost, narrow bandwidth, and relatively high frequency. High frequencies avoid lower frequency noise sources and allow a reasonably short resonant cell (see Equation 1.3). Noise is filtered out in the frequency domain via software (see Appendix C). A parabolic reflector partially collimates the radiation in order to irradiate the gas sample rather than absorbing resonator walls.

As discussed in the background section, smaller chambers produce higher signals [46]. However, the walls should be large enough to minimize wall absorption. Therefore, the chosen geometry includes a resonator radius of 1.26cm (1.59cm once pipe ends), adjustable length of 14-18cm, and cylindrical buffer volumes of 8cm length and 3.81cm radius. The buffer volume length is a quarter of the acoustic wavelength to minimize background noise from window absorption and inlet/outlet acoustic noise via destructive interference. Though Bijnen recommends a resonator radius at least 300% of the radiation source waist, a resonator radius 141% the source radius is
employed to keep cell volume down and to maintain a length-to-radius ratio on the order of 10:1 as recommended by Luo [43, 49, 50]. To combat increased wall absorption, the cell was made of aluminum and the end-caps made of PLA, neither of which absorb strongly in the mid IR [53]. Approximating the cell as an open-open cylinder of cell resonator length with 60% radius end corrections gives a resonant frequency close to 1kHz. To tune the fundamental frequency of the cell to match exactly the LED driver frequency, the resonator length is variable through the use of a hermetic sliding mechanism. Additionally, the adjustable resonator length allows for resonant operation at different ambient temperatures [49, 73].

The microphone is positioned near the center of the resonator because it is a pressure antinode of the standing wave, where maximum acoustic signal is obtained [47]. Inlet and outlet holes of 3.175mm diameter are located at the outer acoustic signal of the buffer volumes.

Raw binary data is unpacked, transformed to the frequency domain using the Fast Fourier Transform (FFT), filtered, and iteratively used to obtain an FFT magnitude proportional to gas concentration at the LED driver frequency to avoid the use of a costly LIA.

The FFT function was also invaluable in testing audio output and visualizing entire data sets. For example, it was first found that the LED driver frequency shifted over time by plotting the FFT of the LED driver frequency signal (see Figure 3.5).

![Figure 3.5: FFT comparing LED driver frequency initially (cold) and after warming up a while (hot).](image)
Test Results

The length of the cell required for resonance at 1056Hz was experimentally determined to be 17cm by subjecting the cell to 1056Hz at constant volume level via a simple earbud headphone and changing the slider distance (i.e. a resonance profile by manual frequency sweep). See Figure 3.6. Though undoubtedly a substantial improvement from the first design, Design 2a still suffered from an irretrievable signal and required reconsideration.

Time-variant LED Driver Frequency A problem discovered during the testing of Design 2a was that the LED driver board frequency changed substantially over time, likely due to increasing temperature of the electrical components. This made exploiting resonance very difficult. See Figure 3.7.

Weak Signal The same low-power LED in Design 1 was used in Design 2a. Additionally, the resonator volume (>40cm$^3$) was still much larger than the recommended cell size (<10cm$^3$), reducing signal strength.

Unsuitable Data Acquisition The FFT filtering method was still insufficient for finding the PA signal, necessitating a LIA.
Figure 3.7: A commercial driver board for the LED43 experienced frequency change over time, likely due to the heating of electronics.

3.2.3 Design 2b

<table>
<thead>
<tr>
<th>Design</th>
<th>1</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td>L x D (cm)</td>
<td>17.5 x 1.6</td>
<td>14-18 x 2.5,3.2</td>
</tr>
<tr>
<td>Air</td>
<td>f_{res} (Hz)</td>
<td>~1000</td>
<td>~1000</td>
</tr>
<tr>
<td>Material</td>
<td>PLA</td>
<td>Al</td>
<td>Al</td>
</tr>
<tr>
<td>Buffers</td>
<td>L x D (cm)</td>
<td>NA</td>
<td>8.0 x 7.6</td>
</tr>
<tr>
<td>Material</td>
<td>NA</td>
<td>PLA x2</td>
<td>PLA x2</td>
</tr>
<tr>
<td>Windows</td>
<td>D_{1} (cm)</td>
<td>&lt;2.54</td>
<td>&lt;1.27</td>
</tr>
<tr>
<td>Material 1</td>
<td>Optical Filter</td>
<td>Sapphire</td>
<td>Sapphire</td>
</tr>
<tr>
<td>D_{2} (cm)</td>
<td>1.6</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
</tr>
<tr>
<td>Material 2</td>
<td>Solid PLA*</td>
<td>Sapphire</td>
<td>Sapphire</td>
</tr>
<tr>
<td>Radiation</td>
<td>LED Driver</td>
<td>LED43-R</td>
<td>LED43-R</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roithner QCW</td>
<td>Roithner QCW</td>
</tr>
<tr>
<td>Microphone</td>
<td>Mic.</td>
<td>ADMP401</td>
<td>ADMP401</td>
</tr>
<tr>
<td>PCB</td>
<td>BOB-09868</td>
<td>BOB-09868</td>
<td>Custom</td>
</tr>
<tr>
<td>Location</td>
<td>Center</td>
<td>Center</td>
<td>Center</td>
</tr>
<tr>
<td>Data</td>
<td>f_{log} (Hz)</td>
<td>&lt;10Hz</td>
<td>4000</td>
</tr>
<tr>
<td>Processing</td>
<td></td>
<td>NA</td>
<td>Avg FFT</td>
</tr>
<tr>
<td>Notes</td>
<td></td>
<td>Inoperative</td>
<td>Need LIA</td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall.

Considering the problems associated primarily with the electronics and data acquisition of Design 2a, significant upgrades were made as described in Table 3.4 and shown in Figure 3.8.
Design Considerations

Design 2b shared with Design 2a the same aluminum cylinder with quarter wavelength buffer volumes, 3D printed end caps, sapphire windows, MEMS microphone in the center of the resonator, mid-IR LED centered around 4.2\(\mu\)m as radiation source, and Arduino with micro SD card for datalogging. Almost all of the physical set-up from Design 2a was used unaltered.

![Figure 3.8: Same cell as PAS2, but with the ends pushed in for a minimum resonator length of 13.5cm.](image)

However, Design 2b had some significant differences. To slightly decrease cell volume and increase geometric repeatability, the ends of the sliding cell were pushed together as far as possible for a resonator length of 13.5cm (see Figure 3.8), increasing the fundamental frequency of the cell. An Arduino not only logged data, but also drove the LED through a MOSFET at this fundamental frequency in quasi-continuous-wave (QCW) mode (50% duty cycle) so as to avoid the time-variant frequency of the smaller commercial driver board. The C++ code used (see Appendix B) was also capable of driving the LED with more current in pulsed mode (1% duty cycle), though no testing was done in this mode. The frequency of the PWM driving signal was tunable, allowing a resonant cell of fixed length. Additionally, a relatively inexpensive LIA based on the AD630 PSD was considered (see Appendix C). However, for convenience and reliability, a lab-grade LIA (SR830) is locked with the LED driver frequency to filter out noise and amplify the target signal. Finally, the Sparkfun microphone was replaced with the Knowles microphone with a higher SNR. For simplicity sake, the same PA cell for Design 2a was tested with the improved data acquisition system.
Test Results

The resonant frequency when filled with room temperature air was experimentally determined to be $\sim 1194\text{Hz}$ by the use of 30 second audio tones from YouTube, microphone and board from Sparkfun, data acquisition from an Arduino via SD card, and frequency analysis with a FFT (i.e. manual frequency sweep test). See Figure 3.9. To verify the repeat-ability of this set-up, the apparatus was taken apart, put back together, and tested again at similar temperature. Using the definition of the quality factor (see Equation 1.2), $Q = 57$ for this cell and resonant frequency test.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{Figure39.png}
\caption{Resonant profile for Design 2b in room air.}
\end{figure}

In the first PA tests, a CO\textsubscript{2} cartridge flushed the PA cell with CO\textsubscript{2}, LEDs were driven at 1.5kHz into the cell, and LIA data was collected in hopes that a non-resonant PA signal would be detected when the LEDs were turned on. A dramatic signal change was seen when the LED circuit was powered on and off, but not because of the PA effect. The signal of Design 2b was dominated by noise. However, several important lessons were learned.

1. To avoid damaging data-acquisition hardware, it is important to know signal output range. The Arduino was potentially damaged because a signal up to 10.5V was fed into an analog pin with an absolute maximum of 5.5V. This also clipped the data that was collected to 5V.
2. Resonance is affected by gas concentration and temperature. The speed of sound in a calorically perfect gas, expressed by \( \sqrt{\gamma RT} \), is affected by both gas composition and temperature [74]. The resonance frequency determined with room-temperature air (Figures 3.9) was meaningless when large concentrations of cold CO\(_2\) were injected into the PA chamber. Combustion processes rarely produce CO\(_2\) concentrations greater than around 1% [75], so approximating the gas as air is reasonable in most field cases. However, lab tests require more consideration because not only does gas temperature vary as a pressurized tank empties, but CO\(_2\) concentrations can far exceed naturally occurring CO\(_2\) levels.

3. Circuitry noise is significant when measuring very small signals. The dramatic signal change that was first observed was actually due to the LED circuit being powered on and off, not because of a changing PA signal. While typical electronics do not require more than some back-of-the-envelope calculations and a multi-meter, small noise signals require critical thinking to minimize noise. For example, the LIA noise floor was reduced by a factor of 10 simply by measuring the PA signal from the differential microphone outputs rather than the positive output and ground. To improve electrical connections, PCBs and soldered proto-boards replaced breadboards and alligator clips. Ground-loops and long-wire noise were minimized by using a single power source and shorter wires. In case of electro-magnetic interference, signal and ground cables were twisted where possible.

3.2.4 Design 3

Design 3 was made with the intent to maximize PA signal strength, and is described in Table 3.5 and shown in Figure 3.10.

Design Considerations

Similar to the other designs, a cylindrical resonator was chosen for simplicity’s sake. However, buffer volumes were omitted and cell diameter reduced to 10mm to allow as much LED power as possible absorbed in the resonator itself. The intent of this design was to get a recognizable PA signal without continuous flow. For a much smaller length (4cm), a resonant frequency around 4kHz was selected. Both ends were acoustically closed, but optically open; one end with
Table 3.5: Design 3 Table.

<table>
<thead>
<tr>
<th>Design</th>
<th>1</th>
<th>2a</th>
<th>2b</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td>L x D (cm)</td>
<td>17.5 x 1.6</td>
<td>14-18 x 2.5,3.2</td>
<td>13.5 x 2.5,3.2</td>
</tr>
<tr>
<td>Air $f_{res}$ (Hz)</td>
<td>~1000</td>
<td>~1000</td>
<td>1194</td>
<td>4200</td>
</tr>
<tr>
<td>Material</td>
<td>PLA</td>
<td>Al</td>
<td>Al</td>
<td></td>
</tr>
<tr>
<td>Buffers</td>
<td>L x D (cm)</td>
<td>NA</td>
<td>8.0 x 7.6</td>
<td>8.0 x 7.6</td>
</tr>
<tr>
<td>Material</td>
<td>NA</td>
<td>PLA x2</td>
<td>PLA x2</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>$D_1$ (cm)</td>
<td>&lt;2.54</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
</tr>
<tr>
<td>Material 1</td>
<td>Optical Filter</td>
<td>Sapphire</td>
<td>Sapphire</td>
<td></td>
</tr>
<tr>
<td>$D_2$ (cm)</td>
<td>1.6</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
</tr>
<tr>
<td>Material 2</td>
<td>Solid PLA*</td>
<td>Sapphire</td>
<td>Sapphire</td>
<td></td>
</tr>
<tr>
<td>Radiation LED Driver</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-R</td>
<td>LED43-RW</td>
</tr>
<tr>
<td>Roither QCW</td>
<td>Roither QCW</td>
<td>Custom, PWM</td>
<td>Custom, PWM</td>
<td></td>
</tr>
<tr>
<td>Microphone Mic.</td>
<td>ADMP401</td>
<td>ADMP401</td>
<td>SPM0687.</td>
<td>SPM0687.</td>
</tr>
<tr>
<td>Location</td>
<td>BOB-99868</td>
<td>BOB-99868</td>
<td>Custom</td>
<td>Custom</td>
</tr>
<tr>
<td>PCB Center</td>
<td>Center</td>
<td>Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data $f_{log}$ (Hz) Processing</td>
<td>&lt;10Hz</td>
<td>4000</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Notes</td>
<td>Inoperative</td>
<td>Need LIA</td>
<td>Too big</td>
<td>PA wall abs.</td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall
**LED package contained a window, which formed the cell wall and first window

the Roithner LED43-RW window and the other with a sapphire window. As the pressure antinodes are at either end of the resonator, the microphone was placed near the sapphire window end. The only thing done to minimize wall-absorption was to make the cell out of aluminum.

Figure 3.10: The third design, showed with the full set-up on the left and alone on the right.
Test Results

First, the resonant frequency of the cell in room-temperature air was determined by subjecting the cell and microphone to various piezo buzzer frequencies (see Figure 3.11). Though there were slight variations due to room temperature fluctuations and geometrical inconsistencies, the resonant frequency was consistently $\sim 4.2$kHz.

The plan was originally to fill the cell with CO$_2$ via bicycle cartridges, plug the inlet and exit, and let the gas warm up to room temperature so that the CO$_2$ concentration would be $\sim 100\%$ with a constant resonant frequency. This would facilitate the highest possible PA signal. However, resonant tests with the CO$_2$-filled cell revealed that the resonant frequency did not remain constant, and always reached a steady-state resonant frequency equal to that of the room-temperature air. Thus, the cell was leaking CO$_2$. Recognizing that a consistent resonant PA signal was not feasible with such a volatile resonant frequency, tests were performed at several discrete LED driving frequencies below that of pure air to demonstrate a PA signal profile as the resonant frequency gradually aligned with the driver frequency (i.e. resonance profiles by decreasing concentration test). It was during the testing of Design 3 that the Acoustic Method of determining concentration of CO$_2$ was discovered and first employed.
Figure 3.12: Microphone signal with no acoustic source on the left and with the LED driver frequency at 3.6, 3.9, and then 4.2kHz on the right as CO$_2$ leaked in Design 3.

Figure 3.12 displays resonance profiles obtained by decreasing concentration from $\sim 100\%$ CO$_2$. First, the LIA was locked on to 4kHz, but neither LED or buzzer was driven by the frequency. The only acoustic source was ambient noise. Results are shown on the left of Figure 3.12. Notice that no resonant peaks occurred. Next, the test was repeated except the LED43-RW was driven at 3.6, 3.9, and then 4.2kHz. Results are shown on the right of Figure 3.12. The three peaks correspond to the resonance of the cell matching these driving frequencies. The measured signal must either be a PA signal or an acoustic signal caused by the heating of electronics.

Finally, the test was repeated with 3.6, 3.9, and 4.2kHz, but with an acoustic buzzer instead of the LED. Results are shown in Figure 3.13. The three peaks correspond to the resonance of the cell matching these driving frequencies. Since no resonant peaks are determined without either the LED or direct sound source, the peaks in Figure 3.12 are due either to a thermal or PA effect, and not random noise. It is quite possible that the observed signal changes are caused by a combination of wall absorption, CO$_2$ absorption, and LED warming and cooling due to electric current. However, a test performed with Design 4a indicates that the resonant peaks are due to the PA effect rather than a thermal effect (see Figure 3.21).

By the Acoustic Method used with the geometry of Design 3 at T=294K, gauge concentration of CO$_2$ is related to resonant frequency. Figure 3.14 shows concentration of CO$_2$ as a function of resonance frequency corresponding to the experimental set-up in Design 3 (see Figures 3.13 and 3.12). Thus, when the resonance frequency reached 3.6, 3.9, and 4.2kHz, the CO$_2$ concentration
Figure 3.13: Microphone signal with buzzer frequency at 3.6, 3.9, and then 4.2kHz as CO$_2$ escaped in Design 3.

Figure 3.14: Concentration of CO$_2$ plotted against resonant frequency by the Acoustic Method for Design 3. Large concentrations on the left; ppm levels on the right.

reached about 66%, 34%, and zero, respectively. Note that for absolute concentration, ambient CO$_2$ concentration must be added.
3.2.5 Design 4a

Design 4a takes elements from both Design 2b and Design 3, aiming for a stronger signal than Design 2b and for less wall absorption than Design 3. Design 4a is described in Table 3.6 and shown in Figure 3.15.

Design Considerations

Design 3 may have produced a detectable PA signal, but not one dependent on CO$_2$ concentration, likely due to wall-absorption. Thus, Design 4a incorporates a similar resonator length (~3.9cm), but with a larger diameter (~2.54cm) and quarter-wavelength buffer volumes of diameter (~6.8cm), almost triple that of the resonator diameter.

Test Results

First, it was found that the resonant frequency of Design 4 with room-temperature air was ~3.48kHz, as shown in Figure 3.16. Then, buzzer leak tests were performed. In Figure 3.17, an air pump is used to reduce the concentration of CO$_2$ in the gas cell from ~100%, and thus increase the resonant frequency. Each peak represents when the resonant frequency of the cell matches the buzzer
frequency. Buzzer frequencies are, from left to right, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, and finally 3.48kHz.

The leak rate was tested by leaving both ends of the cell open and observing when various resonant peaks occurred as CO\textsubscript{2} leaked (see Figure 3.18). By the Acoustic Method, the 2.8, 2.9, and 3.0kHz peaks shown correspond to 89, 76, and 64\% CO\textsubscript{2}, respectively. Thus, it took over 90 minutes to lose 36\% CO\textsubscript{2}. 

Figure 3.15: Design 4a, showed with the full set-up on the left and the resonator alone on the right.

Figure 3.16: Resonant profile for Design 4 in room air.
Figure 3.17: Microphone signal with buzzer frequency at 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, and 3.48kHz as CO₂ was pumped out of Design 4a.

Figure 3.18: Microphone signal with buzzer frequency at 2.8, 2.9, and 3.0kHz as CO₂ naturally leaked in Design 4a.

In Figure 3.19, weak resonant peaks due to the PA effect are shown for 2.9, 3.1, 3.3, and 3.48kHz. No resonant peak was seen without a buzzer or LED.

By the Acoustic Method for the geometry of Design 4a at T=294K and 40% resonator radius end corrections, resonant peaks are directly related to concentration of CO₂. Figure 3.20
Figure 3.19: PA peaks from LED driven at 2.9, 3.1, 3.3, and 3.48kHz as CO$_2$ was pumped out of Design 4a.

Figure 3.20: Concentration of CO$_2$ plotted against resonant frequency by the Acoustic Method for Design 4a. Also applies to Design 4b.

corresponds to the experimental set-up for Design 4a (see Figures 3.17, 3.19, and 3.23). Thus, when the resonance frequency reached 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, and 3.48kHz, the CO$_2$ concentration reached about 89%, 76%, 64%, 51%, 38%, 24%, 11%, and zero, respectively.

**PA Verification: Warming and Cooling Test** To verify that the resonant profiles observed in plots such as Figures 3.12 and 3.19 were due to the PA effect and not simply thermal variations
causing pressure changes, a simple test was performed. The microphone signal was measured as the LED was warmed by a heated mass placed beneath the LED and then cooled by a chilled mass placed beneath the LED. An increase in microphone signal was observed when the LED was cooled and a decrease observed when heated (see Figure 3.21). Since LED power increases when cooled, the measured signal must have been due primarily to the LED radiation, and not thermal waves from the LED. Thus, the signal is PA.

**PA Signal Independent of Concentration: DFE Test**  It was originally thought that a PA signal dependent on CO$_2$ concentration was found with Design 4a, but through testing, it was discovered that the decreasing PA signal over time was more likely due to temperature increase of the LED.

Figure 3.22 shows a clear relationship between microphone signal and time as CO$_2$ was pumped out of the cell. Note the temporary effect of resonance (~7 min in) at 3kHz, the LED driver frequency in these tests. Since concentration decreased steadily with time, the microphone signal was thought to depend on concentration in the non-resonant mode. However, a similar plot was produced by replicating the test with pure Difluorethane (DFE), a gas with almost no absorption around 4.3µm (see Figure 3.22) [76]. Therefore, the PA signal decrease must have been caused by sensitivity to something else (likely temperature) rather than CO$_2$ absorption. Also, there must
Figure 3.22: PA signal drops as air is pumped into a cell initially full of CO$_2$ (Left) and DFE (Right).

have been significant wall or window absorption to produce a PA signal even with almost no CO$_2$ in the cell.

The effects of resonance in Design 4a are clearly seen in Figure 3.23. Note that there is no clear trend in resonant gain from these data, but there is a decreasing baseline voltage over time, as explained by decreasing LED power with increasing temperature.

Figure 3.23: Resonance profiles due to the PA effect in Design 4a are shown for 2.8, 3.0, and 3.2kHz, respectively as CO$_2$ levels were decreased.

3.2.6 Design 4b

The intent of Design 4b was to modify Design 4a in a way that minimized the interference of temperature effects. See Table 3.7.
Table 3.7: Design 4b Table.

<table>
<thead>
<tr>
<th>Design</th>
<th>L x D (cm)</th>
<th>Air $f_{res}$ (Hz)</th>
<th>Material</th>
<th>L x D (cm)</th>
<th>Material</th>
<th>Material</th>
<th>L x D (cm)</th>
<th>Material</th>
<th>Material</th>
<th>L x D (cm)</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resonator</td>
<td>17.5 x 1.6</td>
<td>~1000</td>
<td>PLA</td>
<td>14-18 x 2.5,3.2</td>
<td>~1000</td>
<td>Al</td>
<td>13.5 x 2.5,3.2</td>
<td>Al</td>
<td>Al</td>
<td>3.9 x &lt;2.54</td>
<td>Al</td>
</tr>
<tr>
<td>Buffers</td>
<td>8.0 x 7.6</td>
<td>8.0 x 7.6</td>
<td>PLA x 2</td>
<td>3.9 x &lt;2.54</td>
<td>Al x 2</td>
<td>Al x 2</td>
<td>4.0 x 1.0</td>
<td>3480</td>
<td>3480</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>Sapphire</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>Sapphire</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>&lt;1.27</td>
<td>Sapphire</td>
<td></td>
</tr>
<tr>
<td>Microphone</td>
<td>ADMF401</td>
<td>ADMF401</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td>SPM0687</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>NA</td>
<td>Avg FFT</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td>SR830 LIA</td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td>Inoperative</td>
<td>Need LIA</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

*Design 1 had no exit window; the resonator ended in a solid PLA wall
**LED package contained a window, which formed the cell wall and first window

Design Considerations

Since LED temperature variation was causing a significant PA signal response, some modifications were made to Design 4a. First, a sapphire window was installed on the LED buffer side in the same way as the exit buffer. This allowed the LED to be less dependent on cell gas temperature and to reject more heat to the environment. The LED43-R replaced the LED43-RW to preclude having an unnecessary window. To help cool the LED and maintain consistent temperature, two simple fans were used to provide forced convection cooling.

Test Results

First, a non-resonant test was performed on Design 4b. After filling the cell with CO$_2$ and opening the inlet and outlets, the LED was driven at 3kHz and the PA signal was measured as the concentration of CO$_2$ diminished. Unfortunately, no relationship was found (see Figure 3.24). Instead of a decreased microphone signal with decreased concentration, erratic data was produced. It is expected that wall and window absorption are primarily to blame, but temperature variation, low signal strength, and imperfect LED orientation are also likely explanations. Since a non-resonant signal dependent on CO$_2$ was not found, efforts were focused on detecting resonant signals.
Considering that temperature effects LED power, it was experimentally determined that the LED took about an hour to reach a steady state temperature with fan cooling (see Figure 3.25). Note that the addition of fans not only increased the LED power, but also added signal stability. To determine the leak rate of Design 4b, five buzzer tests were performed in which the cell was filled with CO$_2$, plugged, then opened, allowing various resonant peaks to be exploited via a piezo buzzer (see Figure 3.26) as CO$_2$ leakage changed the resonant frequency in the cell. Consistent with the observed steady state time, the inlet and outlet were plugged for an hour after the cell
was filled to allow a steady state to be nearly approached. An Arduino C++ code automatically changed buzzer frequency in between resonant peaks according to pre-set times (see Appendix B).

Figure 3.26: Resonant profiles, from left to right, for 2774, 2837, 2903, 2972, 3045, 3201, and 3374Hz respectively from one of five buzzer tests for Design 4b.

Figure 3.27 displays the mean time to reach each resonant peak ± two standard deviations according to the five data sets that were obtained. Assuming a normal distribution, 95% of test data would fall within this range.

Note that the uncertainty in time it takes for a given resonant peak to occur increases substantially with increasing resonant frequency, and by the Acoustic Method (see Figure 3.20), decreasing concentration CO₂. Thus, the leak time it takes to reach very low concentrations is much less repeatable than for high concentrations. According to the leak data (Figure 3.27) and the Acoustic Method (Figure 3.20), the uncertainty in time to reach 3335Hz corresponds to an uncertainty in concentration as high as ±5% CO₂ or 25% of the mean concentration. Since cookstove CO₂ levels are typically very low (ppm range), using a resonant peak leak timing method of calibration would require a more repeatable leak set-up (i.e. controlled tubing position, compensation for ambient temperature and pressure, etc.) than presented here.

The same leak test as in Figure 3.26 was performed again, but with LED radiation in place of the buzzer as acoustic source. However, some influence other than the PA effect (likely temper-
Figure 3.27: Time of resonant peaks with error bars of ± two standard deviations from the mean of the five buzzer tests with Design 4b.

ature) caused the data to have many unwanted increases and decreases, making it difficult clearly identify the resonant profiles as CO$_2$ leaked. To help clarify the location of resonant peaks, the test was repeated for several frequencies, one frequency at a time, instead of exploiting multiple resonant peaks in one test. Though the signals were still rocky, Figure 3.28 shows the resonant profiles at 2840, 3048, and 3205Hz, corresponding to 84, 57, and 37%, respectively. Note that the times of these peaks agree fairly well with the buzzer leak tests (Figure 3.27), though tend to be a little longer. Perhaps this is due to decreased leakage due to either flow obstruction or heating by the LED. Also worth noting is that a signal decrease occurred before resonant amplification in every case, an effect not seen with the buzzer. One possibility is that the PA signal first cancelled out noise at the target frequency, causing a signal reduction, before causing the signal to slightly increase. Though there is an overall microphone signal over time, it is uncertain whether this is due to the concentration drop or temperature increase over time.
3.2.7 Design 5

Since Designs 3-4b all produced PA signals with no distinguishable dependence on CO\textsubscript{2} concentration, the primary aim of Design 5 was to minimize wall absorption. See table 3.8 for a design summary and Figure 3.29 for pictures.

### Design Considerations

Since Designs 3-4b all produced PA signals with no distinguishable dependence on CO\textsubscript{2} concentration, the primary aim of Design 5 was to minimize wall absorption. Design 4 did have a relatively large resonator radius, but had a smaller exit window and had quarter acoustic wavelength buffer volumes. This required LED radiation to be very well aligned and collimated in order for radiation to pass through the buffer and into the resonator. Similar to Design 3, Design 5 was...
short (∼4cm) for a large signal and had no buffer volumes for a short path length. However, the resonator and exit window radius of Design 5 nearly doubled that of Design 3 in effort to irradiate the sapphire exit instead of the cell walls. For convenience, a Hamamatsu LED packaged in a square parabolic reflector was used instead of ordering another Roithner LED43-RW.

Figure 3.30: Resonant profile for Design 5 in room air.
Test Results

First, the resonant frequency in room-temperature air was determined to be \( \sim 4.7 \text{kHz} \), as shown in Figure 3.30. This resonant frequency was verified to apply to the cell both before and after epoxy was used to secure the LED. An important discovery was that the CO\(_2\) injection system slowly leaked CO\(_2\) even when the valve was closed. Therefore, the injection valve was only used to plug the inlet during the CO\(_2\) warm up time (see Appendix F). The Acoustic Method for Design 5 is shown in Figure 3.31. The resonant frequency right after being filled with CO\(_2\) was experimentally determined to agree with the Acoustic Method’s estimate of between 3.6 and 3.7kHz.

Next, the CO\(_2\) leak rate of Design 5 was quantified in the same manner as with Design 4b. The testing procedure is documented in Appendix F. A piezo buzzer excited resonant profiles of either three frequencies (3784, 4197, and 4603Hz) or four frequencies (3784, 4095, 4401, and 4603Hz) as CO\(_2\) naturally leaked from the cell, as shown in Figure 3.32. By the Acoustic Method (Figure 3.31), the three frequencies correspond to \( \sim 89, 50, \) and 10% CO\(_2\) respectively and the four frequencies correspond to \( \sim 89, 60, 30, \) and 10% CO\(_2\) respectively.

PA resonant profiles were obtained by the same process as with buzzer profiles, but with LED in place of buzzer and very slightly different frequencies, as shown in Figure 3.33. Unlike in previous designs, the PA resonant profiles of Design 5 were clean and strong. Also, note that the resonant peak magnitudes decrease with decreased CO\(_2\) concentration, indicating that Design
5 is capable of measuring CO₂ concentration with the PA method. In other words, there is a relationship between resonant profile height and concentration of CO₂.

Figure 3.33: LED resonant profiles at 3787, 4201Hz, and 4608Hz (left) and 3787, 4098, 4405, and 4608Hz (right) as CO₂ naturally leaked from Design 5. Note that resonant profiles shorten over time.

Figure 3.34 shows the leak rate data for both buzzer and LED leak tests. Though substantial variation exists, particularly in the later data points, the times at which LED resonant peaks occurred match well with the timing of buzzer resonant peaks. Thus, the LED profiles are not random, but due to the loss of CO₂.
Figure 3.34: Time of resonant peaks with error bars of ± two standard deviations from the mean of the six buzzer tests with Design 5.

As shown in Figure 3.33, there is a relationship between peak resonant signal and time of CO₂ leakage. By the Acoustic Method, a calibration curve for design 5 is shown in Figure 3.35. Notice that this calibration curve is less sensitive at high concentrations. By use of a curve fit of this calibration plot, one can obtain concentration of CO₂ given the strength of the resonant PA signal.

3.3 Results and Discussion Summary

In sum, seven designs were constructed and tested: 1, 2a, 2b, 3, 4a, 4b, and 5. Lessons learned from each design guided the design considerations of the next. Design 1 was simple, but had serious problems such as incompetent datalogging, incorrect microphone placement, and no exit window. Design 2a had a datalogging rate that satisfied the Nyquist sampling theorem, correct microphone placement, and radiation windows, but was quite large and did not make use of a LIA. Design 2b used the same cell as 2a, but introduced lower noise electronics and utilized a LIA. To increase PA signal strength as much as possible, Design 3 was much smaller than any previous design and had a resonator diameter barely larger than the LED reflector package diameter to
ensure all the LED power contributed to the PA signal. Unfortunately, this all but guaranteed significant wall absorption. Design 3 was the first system to produce a measurable PA signal. Design 4a included a short resonator, but with a larger diameter to minimize wall absorption. Buffer volumes were intended to increase resonant quality, but likely hindered LED power from effectively entering the resonator. Additionally, temperature dependence of the LED and set-up obscured the PA effect. Design 4b used the same cell as 4a, but an inlet window was added so that the LED could be separated from the cell and fan-cooled during operation. Both versions of Design 4 produced a PA signal, but like Design 3, did not clearly differentiate between different concentrations of CO$_2$. Design 5 combined a short resonant cell with a relatively large cell radius to maximize the PA signal from CO$_2$ absorption. Unlike previous designs, resonant PA signals were measured that depended on CO$_2$ concentration, demonstrating that LED PA technology can be used to measure CO$_2$ concentrations.

In addition to the PA data measured, a method of measuring CO$_2$ concentration based on resonant frequency was developed and used (i.e. The Acoustic Method). Further study is required to assess its accuracy.
CHAPTER 4. CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

Several conclusions were reached as a result of this research.

4.1.1 Photo-Acoustic CO₂ Detection

PA signals were obtained with Designs 3, 4a, 4b, and 5 (see Figures 3.12, 3.19, 3.23, 3.28, and 3.33). PA signals from Designs 1-4b did not depend on concentration of CO₂ because of wall absorption, but resonant PA signals from Design 5 dropped with decreased concentration CO₂. A calibration was carried out via the Acoustic Method (see Figure 3.35). It is concluded that PA signals excited by LED radiation can be used to detect concentration of CO₂, but it is unclear how well the technology could be applied to cookstove emissions in the field. To the author’s understanding, the combination of several limitations prevent PA CO₂ sensors from currently offering a practical alternative to NDIR sensors.

- LEDs in the mid IR are very low power, precluding adequate sensitivity

- LEDs do not emit in a focused, collimated manner as lasers do, even when a parabolic reflector package is used. Therefore, either resonator diameter must be kept very large with respect to LED reflector diameter to prevent wall absorption (reducing signal strength), or a set of relatively expensive mid IR lenses must be used to focus and collimate all radiant power (see [44]).

- With such low signal strength, high quality Lock-in-Amplification may be required. Lab-grade LIAs are very expensive and bulky, hindering practical field use.
Additional complications are that LEDs in the mid IR and resonant frequency in the cell are both sensitive to temperature. Resonant frequency is also sensitive to gas composition, but less so for trace gas detection. Therefore, two things must occur for accurate sensing.

1. LED temperature must be measured and either compensated for or controlled.

2. Either gas temperature must be measured or, preferably, resonant frequency must be actively tracked in order to update radiation driver frequency in real time to ensure resonant operation.

The author acknowledges that further research is necessary to assess the suitability of Design 5 for trace gas detection and that other PA designs could yield better results.

4.1.2 Acoustic CO$_2$ Detection

In addition to PA CO$_2$ detection, the Acoustic Method can also be used to determine concentration of CO$_2$, but further research is needed to assess its accuracy and potential for trace gas detection. For accuracy, gas temperature must be measured in real time and resonant frequency must be determined accurately and quickly. A major advantage of this technique is that no optics are required, reducing complexity and cost.

Use of leak testing to determine concentration as a function of time via the Acoustic Method is simple enough, but uncertainty notably increases with decreasing concentration, as seen in Figures 3.27 and 3.34. For Design 4b at a resonant frequency of 3335Hz (±13% CO$_2$), an uncertainty in concentration CO$_2$ of ±25% of the mean was determined from leak rate variation alone. Thus, accuracy in determining low concentrations is limited.

4.2 Future Work

As this thesis research was conducted, new areas of potential research were uncovered, both relating to the Acoustic and PA methods of gas detection.
4.2.1 Photo-Acoustic CO\textsubscript{2} Detection

Development of Design 5

Design 5 was shown to be capable of measuring CO\textsubscript{2} concentration in the resonant mode, but little testing was performed. Further study is necessary to determine the suitability of large diameter-to-length ratio designs like Design 5 for open-flow, in-field, trace gas detection. A resonant tracking system would need to be developed in order to ensure continuous resonant operation. Additionally, signals obtained with Design 5 should be thoroughly calibrated with known gas concentrations and compared with trusted CO\textsubscript{2} sensors, such as NDIR devices.

Higher Power LEDs

Since a major limitation of current LED PA sensors is low radiant power, developing and testing higher power LEDs is an obvious solution to pursue. The presented research exclusively studied low-cost 4.3\textmu m LEDs sold in low quantities. However, alternative 4.3\textmu m LEDs such as the QuiC SLED (Terahertz Device Corporation) or LED array (Microsensor Technology) offer higher power. Further development in mid IR technology may result in higher power and lower cost 4.3\textmu m LEDs in the future.

Improved LED Collimators

Another major limitation of LED PA sensors is that even LEDs equipped with a simple parabolic reflector package do not produce truly collimated radiation. An effective mid IR collimator is critical to avoid significant wall absorption noise. A promising solution is to use mid IR lenses to focus and collimate LED radiation [44]. However, lenses in the mid IR are costly, so the challenge is to design an affordable mid IR collimator.

Optimization of LED Duty Cycle

4.3\textmu m LEDs are typically operated in either QCW mode (50% duty cycle) or pulsed mode (1% duty cycle). Since QCW mode offers higher average power and pulsed mode offers higher peak power, it is uncertain which mode produces a stronger PA signal. Taking it one step further, it
would be beneficial to find the optimal duty cycle for exciting a PA wave. For Arduino code capable of duty cycles varying from pulsed to QCW, see Appendix B.

Wall Absorption with Different Materials

Wall absorption is suspected to have been the primary impediment to finding a PA signal dependent on CO$_2$ concentration in the presented research. It would be helpful to see a comparison of wall absorption from various materials. For example, how does the wall absorption of a 3D printed PLA cell compare to a brass or aluminum cell?

Resonant Quality with Different Materials

A convenient and affordable manufacturing technique is 3D printing. A valuable question to answer would be how the resonant quality of plastic cells compare to that of metals. Additionally, what effect does polishing have on resonant quality?

Exploit Different Absorption Band

As shown in Figure 1.2, CO$_2$ has three major absorption bands in the IR. As a rule, LED power decreases with increasing wavelength in the mid IR. Thus, though the absorption band $\sim$2.7$\mu$m is weaker, the power of a $\sim$2.7$\mu$m LED is higher and potentially worth exploring. Note that water also absorbs $\sim$2.7$\mu$m, and would need to be somehow accounted for.

Inexpensive and Portable LIA

A substantial barrier to the affordability of LED PAS is the need for a LIA. Typically, lab LIAs are quite bulky and very expensive. A smaller and lower-cost LIA can be designed at the expense of versatility, such as the one designed by Liu [69]. Also see Appendix D.
Pre-amplification of Microphone

Out of convenience, the microphone PCB designed in this thesis contains no amplification, relying wholly upon the SR830 LIA to amplify the small signal. Perhaps if the PCB had included a stage of amplification via a low-noise op-amp (OP07), the LIA signal would have been more detectable.

QEPAS

Instead of increasing the PA signal strength, Quartz-enhanced Photoacoustic Spectroscopy (QEPAS) replaces the microphone with a quartz tuning fork, advantageous for its excellent resonant quality. The high resonant quality of the quartz tuning fork allows for very sensitive and selective detection of trace gases. Thus, resonance is used with the acoustic sensor to amplify the desired PA signal. Perhaps the extra-sensitive PA detection from the quartz tuning fork would sufficiently combat the low LED power in the mid IR. [77]

NDIR, 2-chamber PA Design

An alternative method of detecting CO$_2$ with the PA effect is the “two cell” design discussed in Section 1.3.2. Though it is fundamentally more of an NDIR than a PA sensor, it has shown promise in several LED systems already, and could be a good alternative to the existing NDIR sensors.

4.2.2 Acoustic CO$_2$ Detection

Accuracy of Acoustic Method

Since the Acoustic Method is analytical by nature, the accuracy of the Acoustic Method should be determined experimentally. One way to do this would be to compare the Acoustic Method with a traditional calibration using known concentrations of CO$_2$.

Suitability of Acoustic Method with Low CO$_2$ Concentrations

The Acoustic Method of CO$_2$ detection should be tested for suitability at low concentrations. For cookstove concentrations of CO$_2$ (up to ~10,000ppm), a set-up capable of determining resonant
frequency to at least four significant figures would be necessary (see Figure 3.14). This may require an acoustic source other than a low-cost piezo buzzer.

Suitability of Leak Test for In-field Calibration

The idea of quantifying natural CO$_2$ leakage to calibrate is attractive for in-field calibrations where lab equipment is not available. The leak-rate and associated error was quantified for Design 4b and found to be unsuitable for any calibration other than high percentage CO$_2$. However, more attention should be given to ensuring a repeatable leak rate. This would involve designing an invariable cell geometry (i.e. no long tubing), compensating for temperature and pressure, and reducing unwanted leakage. With these actions taken, perhaps the leak rate variation could be small enough for even low-concentration calibrations.

Suitability of Acoustic Method for Detecting other Gases

Since the Acoustic Method presented in this thesis is only for CO$_2$ detection, it could be worthwhile to extend this approach to detect other gases such as CO or Methane. One important consideration is that the presented approach approximated the sample as composed of only air and CO$_2$. A more thorough approach would be necessary to detect other trace gases.
REFERENCES


[9] United States Environmental Protection Agency (EPA). Particulate Matter (PM) Basics. 1


[19] Trieu Vuong Dinh, In Young Choi, Youn Suk Son, and Jo Chun Kim. A review on non-dispersive infrared gas sensors: Improvement of sensor detection limit and interference correction, 8 2016. 2


[23] Thomas Kirchstetter, Chelsea Preble, Odelle Hadley, and Ashok Gadgil. Quantification of Black Carbon and Other Pollutant Emissions from a Traditional and an Improved Cookstove. 2010. 3, 6


86


[46] Andrs Miklós, Peter Hess, and Zoltn Bozóki. Application of acoustic resonators in photoacoustic trace gas analysis and metrology, 2001. 6, 7, 8, 9, 10, 52


[55] Xiaotao Yang, Youhong Xiao, Yufei Ma, Ying He, and Frank K. Tittel. A miniaturized QEPAS trace gas sensor with a 3D-printed acoustic detection module. Sensors (Switzerland), 17(8), 8 2017. 11


[63] Lucky A Ishaku and David Hutson. A Resonant Photoacoustic Sensor Based on MID-IR LED and MEMS Microphone Technology Operating at 4.3m. *Online*, 7(8):2222–1727, 2016. 14


[77] Yufei Ma. Review of recent advances in qepas-based trace gas sensing, 2018. 83
### APPENDIX A. BILL OF MATERIALS

**Gas Cell**

<table>
<thead>
<tr>
<th>Part Description</th>
<th>Part ID</th>
<th>Seller</th>
<th>Cost($) /unit</th>
<th>Quantity</th>
<th>Total ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.26um optical bandpass filter</td>
<td>4.26umAg25-25</td>
<td>Anderson Corp</td>
<td>$283.00</td>
<td>1</td>
<td>$283.00</td>
</tr>
<tr>
<td>10um optical bandpass filter</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Minimum round (6in, 7.75&quot;)</td>
<td>-</td>
<td>BU4 Prototyping Lab</td>
<td>$12.00</td>
<td>6</td>
<td>$72.00</td>
</tr>
<tr>
<td>Aluminum tube (ID=4.05in, 1.000in)</td>
<td>9056R8</td>
<td>McMaster Carr</td>
<td>$0.74</td>
<td>11</td>
<td>$7.74</td>
</tr>
<tr>
<td>Inexpensive, 24g threaded (ID=1.000in)</td>
<td>6.025060</td>
<td>Digi-Key</td>
<td>$0.20</td>
<td>5</td>
<td>$1.00</td>
</tr>
<tr>
<td>PVC tube (ID=2.54in, 1/16in)</td>
<td>-</td>
<td>BU4 CHEN stockroom</td>
<td>$0.08</td>
<td>102</td>
<td>$8.16</td>
</tr>
<tr>
<td>30 printed mec. mount and end caps</td>
<td>-</td>
<td>BU4 Prototyping Lab</td>
<td>$0.32</td>
<td>102</td>
<td>$33.12</td>
</tr>
<tr>
<td>Other resistors</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Key ohm, 24k resistor</td>
<td>-</td>
<td>Digi-Key</td>
<td>$0.33</td>
<td>5</td>
<td>$5.00</td>
</tr>
<tr>
<td>312 pF, 220V ceramic capacitor</td>
<td>-</td>
<td>Digi-Key</td>
<td>$0.20</td>
<td>102</td>
<td>$20.40</td>
</tr>
<tr>
<td>1000 ohm, 1/2W trim (ID=1.000in)</td>
<td>6.025060</td>
<td>Digi-Key</td>
<td>$0.20</td>
<td>5</td>
<td>$1.00</td>
</tr>
<tr>
<td>PVC tube (ID=2.54in, 1/16in)</td>
<td>-</td>
<td>BU4 CHEN stockroom</td>
<td>$0.08</td>
<td>102</td>
<td>$8.16</td>
</tr>
<tr>
<td>Inexpensive LIA</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Notes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Other Resistors**

<table>
<thead>
<tr>
<th>Part ID</th>
<th>Seller</th>
<th>Cost($) /unit</th>
<th>Quantity</th>
<th>Total ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Sparkfun</td>
<td>$3.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>Sparkfun</td>
<td>$0.85</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>BYU ChEn stockroom</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Esco</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>Inexpensive LIA</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Inexpensive LIA**

- Selected according to design

**Purpose**

- Design(s)

**Notes**

- Cost is approximate
APPENDIX B. DATA LOGGING AND GENERATION OF RECTANGLE WAVE

Various C++ code were used to program the Arduino (Uno and/or Mega) to log data to a micro SD card and/or generate rectangular waves.

B.1 Higher Frequency Data Logging with PWM

C++ code for the Arduino Mega, used both to log microphone data and to produce a rectangle wave to drive LED radiation at frequencies in the kHz range. Duty cycle limited to about 5% to 95%.

```
#include <SdFat.h>
#include <EEPROM.h> // to open new file each startup

// INPUT desired PWM frequency, duty cycle, and prescaler
#define dc .5 // desired LED duty cycle. Limited to .05 - .95...
#define freq 1000 // desired frequency (Hz). INCLUDE .0 to declare as float.
#define P1 // sets prescaler in TCCR1B. (1, 8, 64, 256, 1024). Low Prescaler good for 1-4kHz tuning

// INPUT desired datalogging frequency
#define f_log 4000 // should be greater than 2*f. INCLUDE .0 to declare as float.
#define P_log 1 // int. High for low frequencies, low for high frequencies. Can't go below ~100Hz

// Calculate PWM and datalogging TOP values
#define f (freq + 1) // freq tends to be ~1Hz too low in practice
#define dcP (1.0 - dc) // Using a P-channel MOSFET...
#define F 16e6 // system clock frequency ("the crystal oscillator")
#define TOP round(F / (P * f) - 1) // determines desired frequency. TOP limited to 65535.
#define TOPH round(dcP * TOP)
#define TOPL (TOP - TOPH)
#define TOP_log round(1.0 + F / (1.0 + P_log * 1.0 + f_log) - 1)

// Calculate closest we can get to desired PWM frequency and duty cycle
#define TOP_act (TOPH + TOPL)
#define f_act (F / (P * (1 + TOP_act)))
```
```
#define dc_act (1.0+TOPL / TOP_act)  // float type
#define f_log_act (F / (P_log + (1+ TOP_log)) )

// Other definitions for buffering, analogue reading, and datalogging
#define PWMpin 9  // PMW pin for Timer 1
#define inputSig A0
#define BUF_A_FULL 0x1  // hexdecimal for a specific bit in a byte
#define BUF_B_FULL 0x2
#define USE_BUF_AB_N 0x4
#define BUF_LEN 256  // 512 byte buffers. For an Arduino, int = 2 bytes.

volatile unsigned int bufferA[BUF_LEN];  // array of 256 elements
volatile unsigned int bufferB[BUF_LEN];
volatile unsigned int idx = 0;  // Where to write the next piece of data within a buffer
// volatile unsigned int count;
volatile byte flags = USE_BUF_AB_N;

//SdFatEX SD; // SD Fat setup
SdFat SD; // SD Fat setup
//SdFile myFile;
File myFile;

// Interrupt: Runs when the TCNT2 register reaches TOP (Reads analogue data)
ISR(TIMER2_COMPA_vect, ISR_BLOCK) {
    ADCSRA |= 0x40;  // Start ADC conversion
}

// Interrupt: runs when an ADC conversion is complete
ISR(ADC_vect, ISR_BLOCK) {
    unsigned char aDCl = ADCL;  // required to read the ADC data in the order: low byte, high byte
    unsigned char aDCh = ADCH;
    unsigned int val = (aDCh << 8) | aDCl;
    ADCSRA &= ~(0x40);  // TESTING – doesn’t seem to help
    if (flags & USE_BUF_AB_N) {
        bufferA[idx++] = val;
    }
    else {
        bufferB[idx++] = val;
    }
    if (idx >= BUF_LEN) {  // We’ve filled a buffer
        if (flags & USE_BUF_AB_N) {
            flags |= BUF_A_FULL;
        }
        else {
            flags |= BUF_B_FULL;
        }
    }
}
```
flags |= USE_BUF_AB_N; // Switch to the other buffer
idx = 0;
}

ISR(TIMER1_OVF_vect, ISR_BLOCK) {
  int new_val = !digitalRead(PWMpin); // new_val = opposite of LED status
  digitalWrite(LED_BUILTIN, new_val); // Toggle the LED
digitalWrite(PWMpin, new_val); // Toggle the PWM pin
  if (new_val) { // if LED is now HIGH
    OCR1A = TOPL; // set OCR1A to TOPL for next time when LED is LOW
  } else { // if LED is now LOW
    OCR1A = TOPH; // set OCR1A to TOPH for next time when LED is HIGH
  }
}

/*
 * Stop whatever the Arduino is doing and run the code in the
 * ISR(TIMER1_OVF_vect, ISR_BLOCK) section every 1/f seconds:
 * f = 16MHz / (prescale * (1 + TOP)).
 * See ATmega2560 datasheet "16-bit Timer/Counter (Timer/Counter 1, 3, 4, and 5)", "Fast PWM".
 */

// Timer 2 set-up for datalogging
void timer2_start() {
  noInterrupts();
  TCCR2A = 0xC2;

  // Prescale set by Timer/Counter2 Control Register B (last 3 digits of TCCR2)
  switch (P_log) // (datasheet: Clock Select Bit Description table)
  {
    case 1: TCCR2B = 0x01; // prescale = 1;
      break;
    case 8: TCCR2B = 0x02; // prescale = 8
      break;
    case 32: TCCR2B = 0x03; // prescale = 32
      break;
    case 64: TCCR2B = 0x04; // prescale = 64
      break;
    case 128: TCCR2B = 0x05; // prescale = 128
      break;
    case 256: TCCR2B = 0x06; // prescale = 256
      break;
  }
}
break;
case 1024: TCCR2B = 0x07; // prescale = 1024
break;
default: Serial.println("Invalid Prescaler for Timer 2");
}

OCR2A = TOP_log; // Set count between 0 and 255 (must be integer).
TIMSK2 = 0x02;
interruptions();

// Timer 1 set-up for PWM frequency
void timer1_start(void) {
    noInterruptions();

    // Timer/Counter1 Control Register A set to mode 15 (Fast PWM, TOP=OCR1A)
    TCCR1A = 0b0011; // WGM11, WGM10 from data sheet

    // Prescale set by Timer/Counter1 Control Register B (last 3 digits of TCCR1)
    switch (P) // (datasheet: Clock Select Bit Description table)
    {
    case 1: TCCR1B = 0x19; // 0b11001; prescale = 1; WGM13:12, CS12:10
        break;
    case 8: TCCR1B = 0x1A; // 0b11010; prescale = 8
        break;
    case 64: TCCR1B = 0x1B; // 0b11011; prescale = 64
        break;
    case 256: TCCR1B = 0x1C; // 0b11100; prescale = 256
        break;
    case 1024: TCCR1B = 0x1D; // 0b11101; prescale = 1024
        break;
    default: Serial.println("Invalid Prescaler for Timer 1");
    }

    OCR1A = TOPL; // Set TOP count between 0 and 2^16-1 (must be integer)
    TIMSK1 = 0x01; // Overflow Interrupt Enable for Timer/Counter1 (TIMER1_OVF)
    interrruptions();
}

// Function that writes integer to EEPROM at p_address and (p_address+1)
void EEPROMWriteInt(int p_address, int p_value) {
    byte lowByte = ((p_value >> 0) & 0xFF);
    byte highByte = ((p_value >> 8) & 0xFF);
    EEPROM.write(p_address, lowByte);
    EEPROM.write(p_address + 1, highByte);
/ Function that reads integer from EEPROM at p_address and (p_address+1)
unsigned int EEPROMReadInt(int p_address) {
    byte lowByte = EEPROM.read(p_address);
    byte highByte = EEPROM.read(p_address + 1);
    return ((lowByte << 0) & 0xFF) + ((highByte << 8) & 0xFF00);
}

// ADC function
void adc_setup() {
    ADMUX = 0xCO; // Use 5V reference, use analog pin 0
    // ADCSRA with Prescalers: 0x8F for 128, 0x8E for 64, 0x8D for 32, 0x8C for 16, 0x8B for 8, 0
    // XRA for 4, 0x89 for 2. (0b10001___)
    ADCSRA = 0x8D; // Enable ADC. Enable interrupt, set prescale to 32. Prescale set by ADPS (last
    // 3 bits of ADCSRA. Datasheet: 26.4, 5, and 8
    ADCSRB = 0x00; // Could tie ADC to timer1 or timer0 with this register

    // Disable all analog pins except for analog pin 0 (reduces power consumption and noise)
    DIDR2 = 0xFF;
    DIDR0 = 0xFE;
}

void setup() {
    // Serial set up
    Serial.begin(9600);
    if (TOP_act > 65535) {
        Serial.println("Timer1 TOP can't exceed 65535");
    }
    if (TOP_log > 255) {
        Serial.println("Timer2 TOP can't exceed 255");
    }
    Serial.println(f_act, 5); // actual PWM frequency to 5 decimal places
    Serial.println(dc_act, 5); // actual duty cycle 
    Serial.println(f_log_act, 5); // actual datalogging frequency 

    // PWM and input signal pins set-up
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(PWMpin, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);
    digitalWrite(PWMpin, LOW);
    pinMode(inputSig, INPUT);

    // SD set up
    pinMode(53, OUTPUT); // Set pin 10 on Arduino as an output for CS pin on micro SD adaptor
SD.begin(53); // Put this line before any SD commands are made

// EEPROM set up
volatile unsigned int N = EEPROMReadInt(0); // MAX(unsigned int) = 65535
char fname[12]; // PA (2 chars) + #### (5 chars) + .bin (4 chars) + \0 [null] terminator (1 char)
sprintf(fname, "AD%05u.bin", N); // New filename
EEPWriteInt(0, ++N); // set N to new value
// myFile = SD.open(fname, O_CREAT | O_TRUNC | O_WRITE); // open SD card for writing to new file
myFile = SD.open(fname, FILE_WRITE); // open SD card for writing to new file

adc_setup();
timer2_start(); // start timer 2 function according to set datalogging values
timer1_start(); // start timer 1 function according to set PWM values
}

// Log data from full buffer to SD card
void loop() {
if (flags & BUF_A_FULL) { // if Buffer A is full
  // Write bufferA
  myFile.write((char*)bufferA, sizeof(int)*BUF_LEN);
  myFile.flush(); // writes buf A bytes to SD card permanently and resets buf A
  flags &= ~BUF_A_FULL; // tell code Buffer A is no longer full
  //Serial.println("BAW");
}
if (flags & BUF_B_FULL) { // if Buffer B is full
  // Write bufferB
  myFile.write((char*)bufferB, sizeof(int)*BUF_LEN);
  myFile.flush();
  flags &= ~BUF_B_FULL;
  //Serial.println("BBW");
}
}

/*Notes:
PWM duty cycle is limited to between 5% and 95%
Outputs an on-off signal from pin 9 at the selected frequency and duty cycle.
Also logs input data to SD card via pin 53 or 10 at selected logging frequency. Limited.
t = SD Memory*(2 bytes/int)/f_log.
14.7GB on SD card allows for 8.5 days of data collection at 10kHz.
Time between data points should be 1/f_log.
Written for the Arduino Mega. Change SD pin to 10 and EEPROM address to make Uno compatible.
ADC with interrupts replaces analogRead because it's more computationally efficient.
Written by Jacob Thomas with much help from Jonny Meldrum.*/
B.2 Lower Frequency Data Logging with PWM

C++ code for the Arduino Uno, used to log LIA data at frequencies in the hundreds of Hz or below. Capable of lower duty cycle than the higher frequency data logging code.

```cpp
#include <SdFat.h>
#include <EEPROM.h> // to open new file each startup

// INPUT desired PWM frequency, duty cycle, and prescaler
#define dutyCycle .5 // desired LED duty cycle. Verified for .01 and .5.
#define freq 4000 // desired frequency (Hz). INCLUDE .0 to declare as float.
#define P 1 // sets prescaler in TCCR1B. (1, 8, 64, 256, 1024). Low Prescaler good for 1−4kHz tuning

// INPUT desired datalogging frequency
#define f_log 100 // high enough for averaging to take place every few seconds. Verified for 100−500Hz on Mega.
#define P_log 1024 // high for low frequencies, low for high frequencies. Can't go below ~100Hz

// INPUT Arduino model
#define Arduino 1 // 1 for uno, anything else for Mega

// Calculate PWM and datalogging TOP values
#define f (freq + 1) // freq tends to be ~1Hz too low in practice
#define dc (dutyCycle) // duty cycle tends to be .1% low −.001
#define dcP (1−1.0*dc) // using a P-channel MOSFET...
#define F 1666 // system clock frequency ("the crystal oscillator")
#define TOP round(F / (P * 1.0=f) − 1) // determines PWM frequency. TOP limited to 65535.
#define tp 1.0*dc / (1.0*f) // pulse time
#define TOP_log round(1.0*F / (1.0*P_log + 1.0*f_log) − 1)

// Calculate closest we can get to desired PWM frequency and duty cycle
#define f1 (F / (P + (1 + TOP))) // clock 1 frequency (PWM)
#define f2 (F / (P_log + (1 + TOP_log))) // clock 2 frequency (datalogging)

// PWM definitions and initializations
#define PWMpin 9 // PWM pin for Timer 1
// volatile boolean on; // PWM
int tdif = 2.0 / 16; // 1/16MH; time to run portx to turn pin back on
```
// Arduino model–dependent parameters
#if Arduino == 1
#define CSpin 10
#define BUF_LEN 256 // 256 byte buffers. 512 required too much memory for uno. For an Arduino, int = 2 bytes.
#define ATMEGAport PORTB // Uno pinout: dgtl pin 9 = PB1. Thus, PORT B and pin 1 on ATMEGA328P.
#define ATMEGApin 0x2 // pin 1 = b0000 0010 = 0x2.
#define DIDR0hex 0x3E //Uno: Enables analog pins 1–5, but leaves A0 enabled
#define ArduinoModel "Model: Uno"
#else
#define CSpin 53
#define BUF_LEN 256 // 512 byte buffers. For an Arduino, int = 2 bytes.
#define ATMEGAport PORTH // Mega pinout: dgtl pin 9 = PH6. Thus, port H and pin 6 on ATMEGA2560.
#define ATMEGApin 0x40 // pin 6 = b0100 0000 = 0x40.
#define DIDR0hex 0xFE //Mega: Enables analog pins 1–7, but leaves A0 enabled
#define ArduinoModel "Model: Mega"
#endif

// Data collection definitions and initializations
#define inputSig A0
#define BUF_A_FULL 0x1 //hexadecimal for a specific bit in a byte
#define BUF_B_FULL 0x2
#define USE_BUF_AB_N 0x4
volatile unsigned int bufferA[BUF_LEN]; //array of 256 elements
volatile unsigned int bufferB[BUF_LEN];
volatile unsigned int idx = 0; // Where to write the next piece of data within a buffer
volatile byte flags = USE_BUF_AB_N;

// SD Fat setup
SdFat SD;
File myFile;

// Interrupt: Runs when the TCNT2 register reaches TOP (Reads analogue data instead of analogRead)
ISR(TIMER2_COMPA_vect, ISR_BLOCK) {
    ADCSRA |= 0x40; // Start ADC conversion
}

// Interrupt: runs when an ADC conversion is complete
ISR(ADC_vect, ISR_BLOCK) {
    unsigned char aDCl = ADCL; // required to read the ADC data in the order: low byte, high byte
    unsigned char aDCh = ADCH;
    unsigned int val = (aDCh << 8) | aDCl;
    if (flags & USE_BUF_AB_N) {
bufferA[idx++] = val;
}
else {
  bufferB[idx++] = val;
}
if (idx >= BUF_LEN) { // We've filled a buffer
  if (flags & USE_BUF_AB_N) {
    flags |= BUF_A_FULL;
  }
else {
    flags |= BUF_B_FULL;
  }
  flags ^= USE_BUF_AB_N; // Switch to the other buffer
  idx = 0;
}

ISR(TIMER1_OVF_vect, ISR_BLOCK) {
  // on = !digitalRead(LED_BUILTIN); // LED starts off
  // digitalWrite(LED_BUILTIN, on);
  ATMEGAport &= ~ATMEGApin; // Pin turned off, LED turned on. Replaced digitalWrite(PWMpin, !on);
  delay_us(tp*pow(10,6)-tdif); // Keep LED on for pulse time. tdif to account for time digitalWrite takes.
  ATMEGAport |= ATMEGApin; // Pin turned on, LED turned off. Replaced digitalWrite(PWMpin, on);
  // digitalWrite(LED_BUILTIN, !on);
}
/**
 * Stop whatever the Arduino is doing and run the code in the
 * ISR(TIMER1_OVF_vect, ISR_BLOCK) section every 1/f seconds:
 * f = 16MHz / (prescale + (1 + TOP)).
 * See ATmega2560 datasheet "16-bit Timer/Counter (Timer/Counter 1, 3, 4, and 5)", "Fast PWM".
 */

// Timer 1 set-up for PWM frequency
void timer1_start(void) {
  noInterrupts();

  // Timer/Counter1 Control Register A set to mode 15(Fast PWM, TOP=OCR1A)
  TCCR1A = 0b0011; // WGM11, WGM10 from data sheet

  // Prescale set by Timer/Counter1 Control Register B (last 3 digits of TCCR1)
  switch (P) // (datasheet: Clock Select Bit Description table)
  {

case 1: TCCR1B = 0x19; // 0b11001; prescale = 1; WGM13:12, CS12:10
    break;
case 8: TCCR1B = 0x1A; // 0b11010; prescale = 8
    break;
case 64: TCCR1B = 0x1B; // 0b11011; prescale = 64
    break;
case 256: TCCR1B = 0x1C; // 0b11100; prescale = 256
    break;
case 1024: TCCR1B = 0x1D; // 0b11101; prescale = 1024
    break;
default: Serial.println("Invalid Prescaler for Timer1");
}

OCR1A = TOP; // Set TOP between 0 and 2^16 - 1 (must be integer)
TIMSK1 = 0x01; // Overflow Interrupt Enable for Timer/Counter1 (TIMER1_OVF)
interruption();

// Timer 2 set-up for datalogging
void timer2_start() {
    noInterrupts();
    TCCR2A = 0xC2;

    // Prescale set by Timer/Counter2 Control Register B (last 3 digits of TCCR2)
    switch (P_log) // (datasheet: Clock Select Bit Description table)
    {
    case 1: TCCR2B = 0x01; // prescale = 1;
        break;
case 8: TCCR2B = 0x02; // prescale = 8
        break;
case 32: TCCR2B = 0x03; // prescale = 32
        break;
case 64: TCCR2B = 0x04; // prescale = 64
        break;
case 128: TCCR2B = 0x05; // prescale = 128
        break;
case 256: TCCR2B = 0x06; // prescale = 256
        break;
case 1024: TCCR2B = 0x07; // prescale = 1024
        break;
default: Serial.println("Invalid Prescaler for Timer2");
    }

OCR2A = TOP_log; // Set count between 0 and 255 (must be integer).
TIMSK2 = 0x02;
interrupts();

void EEPROMWriteInt(int p_address, int p_value) {
    byte lowByte = ((p_value >> 0) & 0xFF);
    byte highByte = ((p_value >> 8) & 0xFF);
    EEPROM.write(p_address, lowByte);
    EEPROM.write(p_address + 1, highByte);
}

unsigned int EEPROMReadInt(int p_address) {
    byte lowByte = EEPROM.read(p_address);
    byte highByte = EEPROM.read(p_address + 1);
    return ((lowByte << 0) & 0xFF) + ((highByte << 8) & 0xFF00);
}

void adc_setup() {
    ADMUX = 0x40; // Use 5V reference, use analog pin 0
    // ADCSRA with Prescalers: 0x8F for 128, 0x8E for 64, 0x8D for 32, 0x8C for 16, 0x8B for 8, 0
    // X8A for 4, 0x89 for 2. (0b10001___)
    ADCSRA = 0x8D; // Enable ADC, Enable interrupt, set prescale to 32. Prescale set by ADPS (last
    // 3 bits of ADCSRA. Datasheet: 26, 4, 5, and 8
    ADCSRA = 0x00; // Could tie ADC to timer1 or timer0 with this register
    // Disable all analog pins except for analog pin 0 (reduces power consumption and noise)
    DIDR0 = DIDR0hex; //Disables analog pins up to 7 except 0
    #if Arduino != 1
    DIDR2 = 0xFF; //Disables analog pins 8–15. Only for Mega.
    #endif
}

void setup() {
    //Serial setup and prints
    Serial.begin(9600);
    Serial.println(" "); // first line is short and lame
    if (TOP > 65535) {
        Serial.println("Timer 1 TOP can't exceed 65535");
    }
    if (TOP_log > 255) {
        Serial.println("Timer 2 TOP can't exceed 255. Top_log is ");
        Serial.println(TOP_log, 5);
    }
}
Serial.println(ArduinoModel);
Serial.print("PWM Freq (Hz): ");
Serial.println(f1 - 1, 5);
Serial.print("Datalogging Freq (Hz): ");
Serial.println(f2 - 1, 5);
Serial.print("Pulse time (us): ");
Serial.println(tp*pow(10,6), 5); *

// PWM and input signal pins set-up
// pinMode(LED_BUILTIN, OUTPUT);
pinMode(PWMpin, OUTPUT);
// digitalWrite(LED_BUILTIN, LOW); //LED starts off
digitalWrite(PWMpin, HIGH); //PWM pin starts on, so LED starts off. Change to LOW for N-channel
MOSFET.

pinMode(inputSig, INPUT);

// SD set up
// pinMode(53, OUTPUT); // Mega: Set pin 53 as an output for CS pin on micro SD adaptor
// SD.begin(53); // Mega: Put this line before any SD commands are made
pinMode(10, OUTPUT); // Uno: Set pin 10 as an output for CS pin on micro SD adaptor
SD.begin(10); // Uno: Put this line before any SD commands are made

// EEPROM set up
volatile unsigned int N = EEPROMReadInt(0); // MAX(unsigned int) = 65535
char fname[12]; // PA (2 chars) + #### (5 chars) + .bin (4 chars) + \0 [null] terminator (1
char )
sprintf(fname, "PS%05u.bin", N); // New filename
EEPWriteInt(0, ++N); // set N to new value
myFile = SD.open(fname, FILE_WRITE); // open SD card for writing to new file

adc_setup();
timer2_start(); // start timer 2 function according to set datalogging values
timer1_start(); // start timer 1 function according to set PWM values

} // Log data from full buffer to SD card

void loop() {
  if (flags & BUF_A_FULL) { // if Buffer A is full
    // Write bufferA
    myFile.write((char*)bufferA, sizeof(int)*BUF_LEN);
    myFile.flush(); // writes buf A bytes to SD card permanently and resets buf A
    flags &= ~BUF_A_FULL; // tell code Buffer A is no longer full
    //Serial.println("BAW");
  } if (flags & BUF_B_FULL) { // if Buffer B is full

104
// Write bufferB
myFile.write((char*)bufferB, sizeof(int)*BUF_LEN);
myFile.flush();
flags &= ~BUF_B_FULL;
// Serial.println("BBW");
}

/* Notes:
Outputs an on–off signal from pin 9 at the selected frequency and duty cycle (good for at least .01 – .95).
Also logs input data to SD card via pin 53 or 10 at selected logging frequency.
Total memory: t = SD Memory*(2 bytes/int)/f_log. (14.7GB on SD card allows for 8.5 days of data collection at 10kHz)
Time between data points should be 1/f_log
ADC with interrupts replaces analogRead and Portx commands replace digitalWrite for speed.
Written for the Arduino Mega or Uno. All changes automatically made by setting "Arduino" value.
Written by Jacob Thomas with much help from Jonny Meldrum.
*/
B.3 Semi-automatically Changing Rectangle Wave Frequencies with Datalogging

C++ code for the Arduino Uno that cycles through seven selected PWM frequencies by manually resetting the Arduino. It is capable of varying duty cycle and logs data at 100Hz. Though this code was not used for any experiments, it has potential for driving LEDs with non-square, rectangle waves (not 50% duty cycle).

```cpp
#include <SdFat.h>
#include <EEPROM.h> // to open new file each startup

// INPUT desired PWM frequencies, duty cycle, and prescaler
#define f0 2774 // 92% ~0.5hr .13hr
#define f1 2837 // 84% ~0.8hr
#define f2 2903 // 76% ~1.25hr
#define f3 2972 // 67% ~2.1hr
#define f4 3045 // 58% ~2.9hr
#define f5 3201 // 37% ~5.5hr
#define f6 3374 // 14% ~13hr

#define dc .5 //desired LED duty cycle

// INPUT desired datalogging frequency
#define f_log 100 // high enough for averaging every few seconds. Verified for 100–500Hz on Mega
#define P_log 1024 // high for low frequencies, low for high frequencies. Can’t go below ~100Hz

// INPUT Arduino model
#define Arduino 1 //1 for uno, anything else for Mega

// Calculate PWM and datalogging TOP values
#define dcP (1−1.0*dc) // using a P-channel MOSFET...
#define F 16e6 // system clock frequency ("the crystal oscillator")
#define TOP_log round(1.0*F / (1.0*P_log + 1.0*f_log)) − 1)

// Calculate closest we can get to desired PWM frequency and duty cycle
#define f_data (F / (P_log + (1+ TOP_log))) // clock 2 frequency (datalogging)
// PWM definitions and initializations
#define PWMpin 9 // PWM pin for Timer 1
#define tdiff 2.0 / 16 // 1/16MHz time to run portx to turn pin back on

// Arduino model-dependent parameters
#if Arduino == 1
#define CSpin 10
#define ATMEGApin PORTB // Uno pinout: digital pin 9 = PB1. Thus, PORT B and pin 1 on ATMEGA328P.
#define ATMEGApin 0x2 // pin 1 = b0000 0010 = 0x2.
#define DIDR0hex 0x3E // Uno: Disables analog pins 1–5, but leaves A0 enabled
#define ArduinoModel "Model: _Uno"
#else
#define CSpin 53
#define ATMEGApin PORTH // Mega pinout: digital pin 9 = PH6. Thus, port H and pin 6 on ATMEGA2560.
#define ATMEGApin 0x40 // pin 6 = b0100 0000 = 0x40.
#define DIDR0hex 0xFE // Mega: Disables analog pins 1–7, but leaves A0 enabled
#define ArduinoModel "Model: _Mega"
#endif

// Data collection definitions and initializations
#define inputSig A0
#define BUF_A_FULL 0x1 // hexadecimal for a specific bit in a byte
#define BUF_B_FULL 0x2
#define USE_BUF_AB_N 0x4
#define BUF_LEN 256 // 256 byte buffers. 512 required too much memory for uno. For an Arduino,
                   int = 2 bytes.
volatile unsigned int bufferA[BUF_LEN]; // array of 256 elements
volatile unsigned int bufferB[BUF_LEN];
volatile unsigned int idx = 0; // Where to write the next piece of data within a buffer
volatile byte flags = USE_BUF_AB_N;

// Function to calculate TOP value for desired frequency
int CalcTop(int f) {
    int TOP = round(F / (P * 1.0*f) - 1);
    return TOP;
}

// Read EEPROM to determine PWM frequency
int TOPS[] = {CalcTop(f0), CalcTop(f1), CalcTop(f2), CalcTop(f3), CalcTop(f4), CalcTop(f5),
               CalcTop(f6)};
int freqs[] = {f0, f1, f2, f3, f4, f5, f6};
byte idx = EEPROM.read(2);
int f = 1.0*freqs[idx];
int tp = 1.0*dc / (1.0*f); // pulse time
// Calculate closest we can get to desired PWM frequency and duty cycle
// int f_LED ( F / ( P * ( 1 + TOPS[fidx]) ) // clock 1 frequency (PWM)

// SD Fat setup
SdFat SD;
File myFile;

// Interrupt: Runs when the TCNT2 register reaches TOP (Reads analogue data instead of analogRead )
ISR (TIMER2_COMPA_vect, ISR_BLOCK) {
    ADCSRA |= 0x40; // Start ADC conversion
}

// Interrupt: runs when an ADC conversion is complete
ISR (ADC_vect, ISR_BLOCK) {
    unsigned char aDCl = ADCL; // required to read the ADC data in the order: low byte, high byte
    unsigned char aDCh = ADCH;
    unsigned int val = (aDCh << 8) | aDCl;
    if (flags & USE_BUF_AB_N) {
        bufferA[idx++] = val;
    } else {
        bufferB[idx++] = val;
    }
    if (idx >= BUF_LEN) { // We’ve filled a buffer
        if (flags & USE_BUF_AB_N) {
            flags |= BUF_A_FULL;
        } else {
            flags |= BUF_B_FULL;
        }
        flags ^= USE_BUF_AB_N; // Switch to the other buffer
        idx = 0;
    }
}

// Interrupt: runs when the TCNT1 register reaches TOP. Turns pin off (LED on) for pulse time.
ISR (TIMER1_OVF_vect, ISR_BLOCK) {
    switch (f)
    {
    case f0: ATMEGApport &= ~ATMEGApin; // Pin off, LED on. Replaced digitalWrite(PWMpin, !on);
        delay_us(dc/10 *pow(10,6)− tdif);
        ATMEGApport |= ATMEGApin; // Pin on, LED off. Replaced digitalWrite(PWMpin, on).
        break;
    }
case f1: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f1 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
case f2: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f2 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
case f3: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f3 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
case f4: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f4 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
case f5: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f5 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
case f6: ATMEGAport &= ~ATMEGApin;
    _delay_us(dc/f6 *pow(10,6)− tdif);
    ATMEGAport |= ATMEGApin;
    break;
}
}
/**
 * Stop whatever the Arduino is doing and run the code in the
 * ISR(TIMER1_OVF_vect, ISR_BLOCK) section every 1/fs seconds:
 * f = 16MHz / (prescale * (1 + TOP)).
 * See ATmega2560 datasheet "16−bit Timer/Counter (Timer/Counter 1, 3, 4, and 5)", "Fast PWM".
 */

// Timer 1 setup for PWM frequency
void timer1_start(int TOP) {
    noInterrupts();

    // Timer/Counter1 Control Register A set to mode 15(Fast PWM, TOP=OCR1A)
    TCCR1A = 0b0011; // WGM11, WGM10 from data sheet

    // Prescale set byTimer/Counter1 Control Register B(last 3 digits of TCCR1)
    TCCR1B = 0x19; // 0b11001; prescale = 1; WGM13:12, CS12:10

    OCR1A = TOP; // Set TOP between 0 and 2^16−1 (must be integer)
    TIMSK1 = 0x01; // Overflow Interrupt Enable for Timer/Counter1(TIMER1_OVF)
interrupts();
}

// Timer 2 set-up for datalogging

void timer2_start() {
    noInterrupts();
    TCCR2A = 0xC2;
}

// Prescale set by Timer/Counter2 Control Register B (last 3 digits of TCCR2)

switch (P_log) // (datasheet: Clock Select Bit Description table)
{
    case 1: TCCR2B = 0x01; // prescale = 1;
        break;
    case 8: TCCR2B = 0x02; // prescale = 8
        break;
    case 32: TCCR2B = 0x03; // prescale = 32
        break;
    case 64: TCCR2B = 0x04; // prescale = 64
        break;
    case 128: TCCR2B = 0x05; // prescale = 128
        break;
    case 256: TCCR2B = 0x06; // prescale = 256
        break;
    case 1024: TCCR2B = 0x07; // prescale = 1024
        break;
    default: Serial.println("Invalid Prescaler for Timer 2");
}

OCR2A = TOP_log; // Set count between 0 and 255 (must be integer).
TIMSK2 = 0x02;
interrupts();
}

// Function that writes integer to EEPROM at p_address and (p_address+1)

void EEPROMWriteInt(int p_address, int p_value) {
    byte lowByte = ((p_value >> 0) & 0xFF);
    byte highByte = ((p_value >> 8) & 0xFF);
    EEPROM.write(p_address, lowByte);
    EEPROM.write(p_address + 1, highByte);
}

// Function that reads integer from EEPROM at p_address and (p_address+1)

unsigned int EEPROMReadInt(int p_address) {
    byte lowByte = EEPROM.read(p_address);
    byte highByte = EEPROM.read(p_address + 1);
}
return ((lowByte << 0) & 0xFF) + ((highByte << 8) & 0xFF00);
}

// ADC function
void adc_setup() {
  ADMUX = 0x40; // Use 5V reference, use analog pin 0
  // ADCSRA with Prescalers: 0x8F for 128, 0x8E for 64, 0x8D for 32, 0x8C for 16, 0x8B for 8, 0
  // X8A for 4, 0x89 for 2. (0b10001...)
  ADCSRA = 0x8D; // Enable ADC, Enable interrupt, set prescale to 32. Prescale set by ADPS (last
  // 3 bits of ADCSRA. Datasheet: 26.4, 5, and 8
  ADCSRB = 0x00; // Could tie ADC to timer1 or timer0 with this register
  // Disable all analog pins except for analog pin 0 (reduces power consumption and noise)
  DIDR0 = DIDR0hex; //Disable analog pins up to 7 except 0
  // if (Arduino != 1) {
  // DIDR2 = 0xFF; //Disable analog pins 8−15. Only for Mega.
  // } FIX LATER
}

void setup() {
  // PWM and input signal pins set−up
  pinMode(PWMpin, OUTPUT);
  digitalWrite(PWMpin, HIGH); //PWM pin starts on, so LED starts off. Change to LOW for N−channel
  // MOSFET.
  pinMode(inputSig, INPUT);

  // SD set up
  pinMode(CSpin, OUTPUT);
  SD.begin(CSpin);

  // EEPROM − datalogging
  volatile unsigned int N = EEPROMReadInt(0); // MAX(unsigned int) = 65535
  char fname[12]; // PA (2 chars) + ##### (5 chars) + .bin (4 chars) + \0 [null] terminator (1
  // char)
  sprintf(fname, "PS%05u.bin", N); // New filename
  EEPROMWriteInt(0, ++N); // set N to new value
  myFile = SD.open(fname, FILE_WRITE); // open SD card for writing to new file

  // EEPROM − frequency index
  int TOP = TOPS[fidx];
  if (fidx < sizeof(TOPS) / sizeof(int) − 1) {
    EEPROM.write(2, ++fidx); // increment fidx for next startup
  } else {
    EEPROM.write(2, 0); // reset fidx for next startup if max index reached
  }
}
// Run Functions
adc_setup();
timer2_start(); // start timer 2 function according to set datalogging values
timer1_start(TOP); // start timer 1 function according to set PWM values
}

// Log data from full buffer to SD card
void loop() {
  if (flags & BUF_A_FULL) { // if Buffer A is full
    // Write bufferA
    myFile.write((char*)bufferA, sizeof(int)*BUF_LEN);
    myFile.flush(); // writes buf A bytes to SD card permanently and resets buf A
    flags &= ~BUF_A_FULL; // tell code Buffer A is no longer full
    //Serial.println("BAW");
  }
  if (flags & BUF_B_FULL) { // if Buffer B is full
    // Write bufferB
    myFile.write((char*)bufferB, sizeof(int)*BUF_LEN);
    myFile.flush();
    flags &= ~BUF_B_FULL;
    //Serial.println("BBW");
  }
}

/*Notes:
Outputs an on−off signal from pin 9 at a selected frequency and duty cycle (good for at least .01 − .95).
Reset Arduino to change frequency. Cycles through 8 defined frequencies.
Also logs input data to SD card via pin 53 or 10 at selected (low) logging frequency.
Total memory: t = SD Memory*(2 bytes/int)/f_log. (14.7GB on SD card allows for 8.5 days of data
collection at 10kHz)
Time between data points should be 1/f_log
ADC with interrupts replaces analogRead and Portx commands replace digitalWrite for speed.
Written for the Arduino Mega or Uno. All changes automatically made by setting "Arduino" value.
Written by Jacob Thomas with much help from Jonny Meldrum.
*/
B.4 Automatically Changing Square Wave Frequencies with Datalogging

C++ code for the Arduino Uno that automatically cycles through seven selected tone() frequencies (50% duty cycle only) according to selected timing and logs data at 100Hz. This code was used for the natural leak tests on Designs 4b and 5.

```c++
#include <SdFat.h>
#include <EEPROM.h> // to open new file each startup

// INPUT desired datalogging frequency
#define f_log 100 // INCLUDE .0 to declare as float.
#define P_log 1024 // int. High for low frequencies, low for high frequencies. Can't go below ~100Hz;

// INPUT Arduino model
#define Arduino 1 // 1 for uno, anything else for Mega

// INPUT tone() frequencies
#define f0 2774 // 92%
#define f1 2837 // 84%
#define f2 2903 // 76%
#define f3 2972 // 67%
#define f4 3045 // 58%
#define f5 3201 // 37%
#define f6 3374 // 14%

// INPUT tone() frequency switch times
#define tscale 60 // 1 for seconds, 60 for minutes, 3600 for hours
#define t0 13 *1.0*tscale // seconds
#define t1 34 *1.0*tscale
#define t2 62 *1.0*tscale // *1.0s are necessary
#define t3 96 *1.0*tscale
#define t4 185 *1.0*tscale
#define t5 400 *1.0*tscale

// time changes
#define dt0 t0 //seconds
#define dt1 (t1−t0)
#define dt2 (t2−t1)
#define dt3 (t3−t2)
#define dt4 (t4−t3)
#define dt5 (t5−t4)
```
// Calculate PWM and datalogging TOP values
#define F 16e6 //system clock frequency ("the crystal oscillator")
#define TOP_log round(1.0+F / (1.0+P_log + 1.0+f_log) - 1)

// Calculate closest we can get to datalogging frequency
#define f_log_act (F / (P_log + (1+ TOP_log)) )

// Other definitions for buffering, analogue reading, datalogging, and tone
#define inputSig A0
#define tonePin 9
#define BUF_A_FULL 0x1 //hexadecimal for a specific bit in a byte
#define BUF_B_FULL 0x2
#define USE_BUF_AB_N 0x4
#define BUF_LEN 256 // 512 byte buffers. For an Arduino, int = 2 bytes.

// Arduino model-dependent parameters
#if Arduino == 1
#define CSpin 10
#define DIDR0hex 0x3E //Uno: Disables analog pins 1-5, but leaves A0 enabled
#define ArduinoModel "Model: Uno"
#else
#define CSpin 53
#define DIDR0hex 0xFE //Mega: Disables analog pins 1-7, but leaves A0 enabled
#define ArduinoModel "Model: Mega"
#endif

// Variable Declarations and Initializations
volatile unsigned int bufferA[BUF_LEN]; //array of 256 elements
volatile unsigned int bufferB[BUF_LEN];
volatile unsigned int idx = 0; // Where to write the next piece of data within a buffer
volatile byte flags = USE_BUF_AB_N;
int freqs[] = {f0, f1, f2, f3, f4, f5, f6};
long dts[] = {dt0+f_log , dt1+f_log , dt2+f_log , dt3+f_log , dt4+f_log , dt5+f_log}; //count units, not seconds
volatile byte fidx = 0; // Determines which element of freqs[] and dts[] to use
volatile long countf = 0; // Determines timing of tone() frequency changes

// SD Fat setup
SdFat SD; // SD Fat setup
File myFile;

// Interrupt: Runs when the TCNT0 register reaches TOP (Reads analogue data)
ISR(TIMER0_COMPA_vect, ISR_BLOCK) {
    ADCSRA |= 0x40; // Start ADC conversion
    countf++;
}
if (countf == dts[fidx]) {
    fidx++;
    countf = 0;
    tone(tonePin, freqs[fidx]);
}

// Interrupt: runs when an ADC conversion is complete
ISR(ADC_vect, ISR_BLOCK) {
    unsigned char aDCl = ADCL; // required to read the ADC data in the order: low byte, high byte
    unsigned char aDCh = ADCH;
    unsigned int val = (aDCh << 8) | aDCl;
    if ((flags & USE_BUF_AB_N) {
        bufferA[idx++] = val;
    } else {
        bufferB[idx++] = val;
    }
    if (idx >= BUF_LEN) { // We've filled a buffer
        if ((flags & USE_BUF_AB_N) {
            flags |= BUF_A_FULL;
        } else {
            flags |= BUF_B_FULL;
        }
        flags ^= USE_BUF_AB_N; // Switch to the other buffer
        idx = 0;
    }
}

// Timer 0 set-up for datalogging
void timer0_start() {
    noInterrupts();
    TCCR0A = 0xC2;

    // Prescale set by Timer/Counter0 Control Register B (last 3 digits of TCCR0)
    switch (P_log) // (datasheet: Clock Select Bit Description table: 15–9)
    {  
        case 1: TCCR0B = 0x01; // prescale = 1, 0b001
            break;
        case 8: TCCR0B = 0x02; // prescale = 8, 0b010
            break;
        case 64: TCCR0B = 0x03; // prescale = 64, 0b011
            break;
        case 256: TCCR0B = 0x04; // prescale = 256, 0b100
            break;
    }
}
break;
case 1024: TCCR0B = 0x05; // prescale = 1024, 0b101
break;
default: Serial.println("Invalid Prescaler for Timer 0");
}

OCR0A = TOP_log; // Set count between 0 and 255 (must be integer).
TIMSK0 = 0x02; // b10
interrupts();

// Function that writes integer to EEPROM at p_address and (p_address+1)
void EEPROMWriteInt(int p_address, int p_value) {
 byte lowByte = ((p_value >> 0) & 0xFF);
 byte highByte = ((p_value >> 8) & 0xFF);
 EEPROM.write(p_address, lowByte);
 EEPROM.write(p_address + 1, highByte);
}

// Function that reads integer from EEPROM at p_address and (p_address+1)
unsigned int EEPROMReadInt(int p_address) {
 byte lowByte = EEPROM.read(p_address);
 byte highByte = EEPROM.read(p_address + 1);
 return ((lowByte << 0) & 0xFF) + ((highByte << 8) & 0xFF00);
}

// ADC function
void adc_setup() {
 ADMUX = 0x40; // Use 5V reference, use analog pin 0
 // ADCSRA with Prescalers: 0xF for 128, 0x8E for 64, 0x8D for 32, 0x8C for 16, 0x8B for 8,
 // 0x8A for 4, 0x89 for 2. (0b10001...)
 ADCSRA = 0x8D; // Enable ADC, Enable interrupt, set prescale to 32. Prescale set by
 // ADPS (last 3) bits of ADCSRA. Datasheet: 26.4, 5, and 8
 ADCSRB = 0x00; // Could tie ADC to timer1 or timer0 with this register

 // Disable all analog pins except for analog pin 0 (reduces power consumption and noise)
 DIDR0 = DIDR0hex; //Disables analog pins up to 7 except 0
 // if (Arduino != 1) {
 // DIDR2 = 0xFF; //Disables analog pins 8–15. Only for Mega.
 // } FIX LATER
}

void setup() {
 // Serial set up
 // Serial.begin(9600);
// if (TOP_log > 255) {
//     Serial.println("Timer 0 TOP can’t exceed 255");
// }
// Serial.println(f_log_act, 5); //actual datalogging frequency

// Input and Output signal pin set-up
pinMode(inputSig, INPUT);
pinMode(tonePin, OUTPUT);

// SD set up
pinMode(CSpin, OUTPUT); // Set pin 10 on Arduino as an output for CS pin on micro SD adaptor
SD.begin(CSpin); // Put this line before any SD commands are made

// EEPROM set up
volatile unsigned int N = EEPROMReadInt(0);  // MAX(unsigned int) = 65535
char fname[12]; // PA (2 chars) + ###### (5 chars) + .bin (4 chars) + \0 [null] terminator (1 char)
sprintf(fname, "AD%05u.bin", N); // New filename
EEPWriteInt(0, ++N); // set N to new value
myFile = SD.open(fname, FILE_WRITE); // open SD card for writing to new file

tone(tonePin, f0); // start first tone
adc_setup();
timer0_start(); // start timer 0 function according to set datalogging values
}

// Log data from full buffer to SD card
void loop() {
    if (flags & BUF_A_FULL) { // if Buffer A is full
        myFile.write((char*)bufferA, sizeof(int)*BUF_LEN);  // Write bufferA
        myFile.flush();  // writes buf A bytes to SD card permanently and resets buf A
        flags &= "BUF_A_FULL"; // tell code Buffer A is no longer full
        //Serial.println("BAW");
    }

    if (flags & BUF_B_FULL) { // if Buffer B is full
        myFile.write((char*)bufferB, sizeof(int)*BUF_LEN); // Write bufferB
        myFile.flush();
        flags &= "BUF_B_FULL";
        //Serial.println("BBW");
    }
}

//Notes:
Frequency changes within interrupt version.
Logs input data to SD card via pin 53 or 10 at selected logging frequency and produces square wave from tonePin with selected frequencies at selected times.

\[ t = \text{SD Memory} \times (2 \text{ bytes/int}) / f \_\text{log} \].

14.7GB on SD card allows for 8.5 days of data collection at 10kHz.

Time between data points should be \( 1/f \_\text{log} \).

Written for the Arduino Mega or Uno.

ADC with interrupts replaces analogRead because it's more computationally efficient.

Written by Jacob Thomas with much help from Jonny Meldrum.

**Pseudo-code**:  
Set tone(f0)  
Write data in buffers to SD card  
Interrupt at 100Hz to read A0 and check count for frequency switching  
Interrupt when ADC of A0 is complete to store read data in buffers  
*/
B.5 Automatic Square Wave Frequency Sweep

C++ code for the Arduino Mega that automatically cycles through an array of available tone() frequencies (50% duty cycle only) according to selected time increment. This code was used for determining resonant frequencies of Design 5.

// Set initial frequency, frequency increment, and time increment of resonance sweep loop
// INPUT freqs initial and final index and time step size
#define idxi 0 //must be in the freqs[] array. 17
#define idxf 50 //must be in the freqs[] array. (sizeof(freqs) / sizeof(int)) = 58
#define dt 10 //s

// Frequencies Available using tone() on Mega
int freqs[] = { 2497, 2522, 2548, 2574, 2601, 2628, 2656, 2685, 2714, 2744, 2774, 2806, 2837, 2870, 2903, 2937, 2973, 3008, 3045, 3083, 3121, 3161, 3201, 3243, 3286, 3329, 3374, 3420, 3468, 3517, 3567, 3619, 3672, 3727, 3783, 3842, 3902, 3917, 3932, 3948, 3964, 3979, 3995, 4011, 4027, 4044, 4060, 4077, 4093, 4110, 4127, 4145, 4162, 4179, 4197, 4214, 4232, 4250, 4269, 4287, 4305, 4324, 4343, 4362, 4381, 4400, 4420, 4439, 4459, 4479, 4499, 4520, 4540, 4561, 4582, 4603, 4624, 4646, 4667, 4689, 4711, 4734, 4756, 4779, 4802, 4825, 4849, 4872, 4896, 4920, 4945, 4969, 4994, 5019, 5045, 5070, 5096, 5120};

// Initialize
int f = freqs[idxi];
byte idx = idxi;

void setup() {}

void loop() {
tone(9, f);
delay(dt*1000.0);
if (idx > idxf) {
  noTone(9); //idx = 0
}
else {
  idx++;
}
f = freqs[idx];
}
APPENDIX C. POST-PROCESSING CODE

C.1 MATLAB/Octave Code: Post-processing Audio Data

MATLAB/Octave code used to unpack binary raw audio data, transform to the frequency domain, and plot.

% Jacob Thomas
% Sound Processing script
% Performs Fourier Transform on entire data set to reveal prominent freq.
clc; clear all;

% Set up
fs = 8000; % Sampling Frequency (Hz)
delT = 1/fs; % s

% Import audio data
fileID = fopen('AD00223.bin'); % Raw binary data.
data = fread(fileID, 'uint16'); % Reads binary data in units of 16 unsigned bits

% Split up data
% t1 = 1; % s
% t2 = 31; % s
% idx1 = t1*fs;
% idx2 = t2*fs;
data = data((idx1+1):(idx2+1));

% Plot Audio Data
N = length(data);
t_total = (N-1)*delT;
t = (0:delT:t_total)';
p = data; % - median(data)

% figure (1);
% plot(t,p)
% xlabel('Time (s)');
% ylabel('Microphone Raw Output');
%Plot Fourier Transform
ftp = fft(p)/N; % length(ftp) = N
ftp(1) = 0; %get rid of DC component
ftp = fftshift(ftp);
ftp_mag = abs(ftp);
f = (-N/2:N/2-1)*f_s/N; %max frequency at Nyquist, centered around 0Hz. 4.2Hz too high?
hal2 = (N/2+1:N);

figure (2);
plot( f(hal2), ftp_mag(hal2) ); %Because fft mirrors across 0Hz, plot 2nd half only
title ('Fourier Transform')
xlabel('Frequency (Hz)');
ylabel('Fourier Transform Magnitude');
hold on

%Plot just around 1000Hz, or wherever you want
f_th = 1050; % CHANGE TO ADJUST TARGET FREQ
n = 200; % Hz CHANGE TO ADJUST FREQ SPAN
ind_th_vec = find(f >= f_th);
ind_th = ind_th_vec(1);
ind_lo_vec = find(f >= (f_th - n));
ind_lo = ind_lo_vec(1);
ind_hi_vec = find(f >= (f_th + n));
%ind_hi_vec = find(f >= 998); % CHANGE
ind_hi = ind_hi_vec(1);
band_th = (ind_lo):(ind_hi); %theoretical index range around 1000Hz
[maxftp, ind] = max(ftp_mag(band_th)); %find max ftp around 1000Hz
ind_max = ind + (ind_th - n);
f_max = f(ind_max);

%figure (3)
%plot( f(band_th), ftp_mag(band_th) );
%hold on
C.2 Post-processing LIA Data

MATLAB/Octave code used to unpack, average, and plot binary data from the SR830 LIA.

```matlab
% Jacob Thomas
% Sound Processing script
clear all;

% Set up
fs = 100; % Sampling frequency (Hz)
Ts = 1/fs; % s. Time between raw data points
Ta = 10; % Averaged period. Time between each plotted data point.

% Import voltage divided, ADC'd, LIA output data
fileID = fopen('PS00813.bin'); % Raw binary data.
data = fread(fileID, 'uint16'); % Reads bin. data in blocks of 16 unsigned bits

% For when the datalogging system (PS00xxx) splits data into multiple files
Fnumi = 813; % initial file number
Fnumf = Fnumi; % final file number
data = [];
for Fnum = Fnumi:Fnumf
    FName = ['PS00', num2str(Fnum), '.bin '];
    fileID = fopen(FName); % Raw binary data.
    if fileID > 0
        datum = fread(fileID, 'uint16'); % Reads bin. data in blocks of 16 uns. bits
        data = [data; datum];
    end
end

% Split up data
t1 = 5 * 60; % s
t2 = 17 * 60; % s
idx1 = t1 * fs;
idx2 = t2 * fs;
data = data((idx1+1): (idx2+1)); % include data between t1 and t2
data = data((idx1+1): end); % include all data after t1

% Plot raw and averaged data
tf = length(data) * Ts; % final time
ts = (Ts: Ts: tf)'; % sampling time array

tfa = floor(tf/Ta) * Ta; % round to nearest Ta + Ta/2
ta = (Ta/2: Ta: tfa)'; % averaged time array
```
avgdata = zeros(length(ta),1);
for i = (1:length(ta))
    ind1 = ceil(i*Ta/Ts - (Ta-1)/Ts);
    ind2 = ceil(i*Ta/Ts);
    avg = mean( data(ind1:ind2) );
    avgdata(i) = avg;
end

figure(1, 'DefaultAxesFontSize', 15, 'DefaultLineWidth', 1, 'DefaultMarkerSize', 15)
plot(ts/60, data, 'color', [.8 .8 .8], '--', ta/60, avgdata, '-k')
title('Raw Data over Time')
xlabel('Time (min)');
ylabel('Raw Data (a.u.)');
legend('Raw', 'Averaged Raw', 'Location', 'northeast')

Plot raw and averaged LIA output voltage
R1 = 1.193e3;
R2 = .981e3;
ADCconv = 5.2/1023; % convert from ADC scale to volts, outputting 4.15V...
VDIVconv = (R1+R2)/R2; % convert from voltage divided to actual SR830 output
GAINconv = 100/10; % in uV. SR830 sensitivity of #uV / 10V.
%21.84/10.92 = 10/20 etc.
PA = data *ADCconv*VDIVconv*GAINconv;
PAa = avgdata *ADCconv*VDIVconv*GAINconv;

figure(2, 'DefaultAxesFontSize', 15, 'DefaultLineWidth', 1, 'DefaultMarkerSize', 15)
plot(ts/60, PA, 'color', [.8 .8 .8], '--', ta/60, PAa, '-k')
title('LIA Output over Time')
xlabel('Time (min)');
ylabel('LIA output (uV)');
legend('Raw_LIA', 'Averaged_LIA', 'Location', 'northeast')

% and Lms43PD-03-CG, Lms43LED-CG #57, LED43-PR, L13201-0430M with x3.3 Package
%hold on

figure(3, 'DefaultAxesFontSize', 15, 'DefaultLineWidth', 1, 'DefaultMarkerSize', 15)
plot(ta/60, PAa, '-k')
title('Averaged LIA Output over Time')
xlabel('Time (min)');
ylabel('Averaged LIA Output (uV)');

fclose all %fixed a bug

123
APPENDIX D.  DESIGN AND COMPUTATION CODE

D.1 Resonant Cell Design

For a 1D cylinder operating in longitudinal resonance mode, Equation 1.3 can be used to compute either the required length given a frequency or to compute the required frequency given a length. The following Octave/MATLAB functions were used both to select an appropriate cell length when designing resonators and to locate the approximate resonant frequency when experimentally determining resonant frequency. Note that end corrections should be added to both ends of an open-open cylinder for accuracy.

D.1.1 Function to Compute Resonant Frequency

Octave/MATLAB function used to compute longitudinal resonant frequency of a cylinder given a resonator length, operating temperature, and gas.

```matlab
% Jacob Thomas
% Calculates resonant frequency of open-open or closed-closed pipe
function ResFreq = CalcResFreq (L,T,G)

if G==1 % Air
    gamma = 1.4;
    R = 287;
end
if G==2 % CO2
    gamma = 1.29;
    R = 189;
end

    c = sqrt(gamma*R*T);
    ResFreq = c / (2*L);
end
```
D.1.2 Function to Compute Resonant Length

Octave/MATLAB function used to compute longitudinal resonant length of a cylinder given a frequency, operating temperature, and gas.

```
% Jacob Thomas
% Calculates resonant length of open-open or closed-closed pipe

function ResLength = CalcResLength(F,T,G)
    if G==1 % Air
        gamma = 1.4;
        R = 287;
    end
    if G==2 % CO2
        gamma = 1.29;
        R = 189;
    end
    c = sqrt(gamma*R*T);
    ResLength = c / (2*F);
end
```
D.2 LED Circuit Design

Octave/MATLAB code used to estimate appropriate resistor value for LED circuit (see Figure 2.22). When experimentally determining the resistor value, a significantly higher resistor was always first chosen to ensure safe current level.

% Jacob Thomas
% LED, MOSFET circuit, all units base SI
cle; clear all;

% Power and MOSFET specs
% https://www.sparkfun.com/datasheets/Components/General/FQP27P06.pdf
Vcc = 8;
%Rds = .06; %Resistance from Drain to Source when current flows. MOSFET data Sheet.

% LED43 and MS Lms43LED–CG
% http://www.roithner-laser.com/datasheets/led_midir/led43.pdf
% http://lmsnt.com/datasheets/Standard%20chip/Lms43LED–CG/Lms43LED–CG_TO18_rev240317.pdf
Iq = .25; %Max QCW current. r for roithner.
Ip = 2; %Max pulsed current at dc = .01
Vq = .3; %LED data sheet, If vs Vf .65 before???
Vp = 1.8; %LED data sheet, If vs Vf, extrapolation guess
Rq = (Vcc – Vq)/Iq %KVL
Rp = (Vcc – Vp)/Ip

%Vq,m = Iq*Rds; %also for Lms43LED–CG
%Vp,m = Ip*Rds;
%Rq,m = (Vcc – Vq – Vq,m)/Iq %KVL
%Rp,m = (Vcc – Vp – Vp,m)/Ip

% Hamamatsu L13201–0430
% https://www.hamamatsu.com/resources/pdf/ssl/113201_series_kled1069e.pdf
Ihq = .1;
Ihp = .5;
Vhq = 1.8; %LED data sheet, If vs Vf
Vhp = 3.45; %LED data sheet, If vs Vf
Rhq = (Vcc – Vhq)/Ihq %KVL
Rhp = (Vcc – Vhp)/Ihp

%Vhq,m = Ihq*Rds;
%Vhp,m = Ihp*Rds;
%Rhq,m = (Vcc – Vhq – Vhq,m)/Ihq %KVL
%Rhp,m = (Vcc – Vhp – Vhp,m)/Ihp
These are the minimum estimated resistor values. Thus, a safety factor
(\sim 2 \text{ for QCW and } \sim 3 \text{ for pulsed}) should be applied for the first guess
and the minimum actual resistor value should be experimentally determined.

The results show that the MOSFET can be approximated as zero current when on,
as expected.

Higher operating voltage \implies higher power dissipation \implies hot and damaged
LED. Use at lower voltage or add heatsink.
D.3 AD630 Based Lock-in Amplifier Design

Octave/MATLAB code for selecting resistor, capacitor, and inductor values for a low-cost LIA, as designed by Liu [69]. Note that this design is incomplete.

% Jacob Thomas
% LIA based on AD630 PSD
% "Design of Lock-in Amplifier used on the Photoacoustic Spectroscopy" (Liu)
clc; clear all;

%Set-up
i = (-1)\.5;

%Pre-amplifier ( 2 stage, active, non-inverting amplifiers)
R1 = 1e3; R2 = 1e3; R4 = 1e3; R5 = 1e3;
L1 = 1e-3; C1 = 1e-7; C2 = 1e-7; C3 = 1e-7; C4 = 1e-7;

R3 = 1e5; %set gain
R6 = R3;

G1 = (1 + R3/R1)*(1 + R6/R4) %^10,000

%Band-pass Filter (active low-pass and high-pass filters in succession)
% Test PAS2 with SR830 before designing for 1kHz
% Ideal to build new cell and test resonant freq before choosing
% When selecting components, add tolerance error bars, factor of safety, etc
f0 = 2000; % CENTER PASS FREQUENCY
C5 = 1e-7; C6 = C5; % tolerance?
R7 = (2*pi*f0*C5)^-1;
%R7 = .99*794;
%f0 = (2*pi*R7*C5)^-1;
R8 = R7;
R10 = 2*R7;

B = 100; %PASSBAND WIDTH (Hz), 20Hz change = 10ohms change in R11
R9 = 1e3;
R11 = R9*(2 - B/f0); %VARIABLE, High Tolerance

%R9 = 1e3;
%R11 = 1990;
%B = (2-R11/R9)*f0;

f = 0:2*f0;
%B = (2 - R11/R9)*f0;
\[ Q = \frac{f_0}{B}; \quad \text{\textit{quality of BP filter}} \]

\[ G_{2f} = \text{abs}\left( \frac{1 + R_{11}/R_9}{(2 - R_{11}/R_9 + i*(f/f_0 - f_0/f)} \right); \]

\[ \text{lo = find}(f >= f_0 - B/2, 1); \]

\[ \text{hi = find}(f >= f_0 + B/2, 1); \]

\[ G_2 = \text{mean}(G_{2f}(\text{lo:hi})); \quad \text{\textit{BP FILTER GAIN. Depends on B. \textasciitilde 100}} \]

\[ G_{2vec} = \text{ones(length(lo:hi), 1)}*G_2; \]

\texttt{figure (1)}

\texttt{plot(f, G_{2f}, f(lo:hi), G_{2vec})}

\texttt{xlabel('Frequency (Hz)')}

\texttt{ylabel('Band-pass Filter Gain')}

\texttt{legend('Gain as a function of frequency', 'Average gain in pass-band')}

\%Phase Shifter

\[ R_{12} = 1e3; R_{13} = 1e3; R_{14} = 1e3; R_{15} = 1e3; R_{16} = 1e3; R_{16} = 1e3; \]

\[ R_{17} = 1e3; R_{18} = 1e3; C_7 = 1e-7; C_8 = 1e-7; C_9 = 1e-7; \]

\[ G_3 = 1 \text{ no gain?} \]

\%Phase-sensitive Detector (AD630)

\[ R_{19} = 1e3; R_{20} = 1e3; C_{10} = 1e3; C_{11} = 1e3; \]

\[ \text{Aref} = 5/2; \]

\[ G_4 = 2/\pi \text{ assuming sinusoidal PA signal} \]

\%Low-pass Filter (active LP filter)

\[ R_{21} = 1e3; R_{22} = 1e3; \quad R_{23} = 1e3; \]

\[ C_{12} = 1e-7; C_{13} = 1e-7; \]

\[ R_{24} = 1e5; \quad \text{\textit{SET GAIN}} \]

\[ s = i*2*\pi*f; \]

\[ \text{Vratio5} = ( \frac{(s^2*R_{21}*R_{22}*C_{12}*C_{13} + s*(R_{21}*C_{12} + R_{22}*C_{13}) + 1)}{(1 + R_{24}/R_{23}) \ldots - R_{21}*C_{12}*s}).^-1; \]

\[ G_{5f} = \text{abs}(\text{Vratio5}); \]

\[ G_5 = G_{5f}(1) \text{ \textit{Vratio5 with s=0, 1 + R_{24}/R_{23}} \]

\[ \text{indc = find}(G_{5f} <= G_5/2, 1); \]

\texttt{figure (2)}

\texttt{plot(f, G_{5f})}

\texttt{xlabel('Frequency (Hz)')}

\texttt{ylabel('Low-pass Filter Gain')}

\%DC Amplifier (active, non-inv. amp)

\[ R_{25} = 1e3; R_{26} = 1e3; \]

\[ R_{27} = 1e3; \]

\[ G_6 = (1 + R_{27}/R_{25}); \]
Overall Gain

$G_{tot} = G_1 \times G_2 \times G_3 \times G_4 \times G_5 \times 10^{-9}$ \%Want $^*10^{-9}$
D.4 Acoustic Method

Octave/MATLAB code used to calculate and plot concentration as a function of resonant frequency (see Section 2.2).

```matlab
% Jacob Thomas
% Resonant Frequency Method to determine Concentration CO2
clc; clear all;

% Input
T = 294; % measured during experiment
L = 3.90e-2; % back-solve by performing resonant test with room-temp air.
Mode = 2; % 1 = Closed-Closed, 2 = open-open
% PAS 4, L = 4.09 cm; PAS 5, L = 3.90
R = 1.3e-2; % resonator radius

% End corrections
ec = 2*.4*R; % end corrections. Gives more accurate result for open cyl.
if Mode == 1
    Lc = L; % closed-closed
elseif Mode == 2
    Lc = L + ec; % open-open end corrections applied
end

fa = CalcResFreq(Lc,T,1); % res. freq of room temp air
fc = CalcResFreq(Lc,T,2); % res. freq of room temp CO2
f = fc:fa; % possible range assuming constant T
%f = 4193:.001:fa; % ppm range
%f = 2700;

c_a = .1; % CO2 percent in indoor air
gama_a = 1.40;
R_a = 287;
gama_c = 1.29;
R_c = 189;

% Calculations
gamR_m = 4*Lc.^2.*f.^2/T;
A = gama_c*R_c - gama_c*R_a - gama_a*R_c + gama_a*R_a;
B = gama_c*R_a + gama_a*R_c - 2*gama_a*R_a;
D = gama_a*R_a - gamR_m;

c1 = (-B + (B.^2 - 4*A*D).^0.5) ./ (2*A); % not between 0 and 100%
c2 = (-B - (B.^2 - 4*A*D).^0.5) ./ (2*A); % the reasonable solution
```

131
\[ c_g = c_2 \times 100; \text{ } \textit{gauge, percent of CO2 above ambient level} \]
\[ c_{abs} = c_g + c_a; \text{ } \textit{percent, including CO2 already in air} \]

```matlab
figure(1, 'DefaultAxesFontSize', 15, 'DefaultLineLineWidth', .1, ...
      'DefaultLineMarkerSize', 15)
plot(f/1000, c_g, '-k')
title('Resonance Frequency Method')
xlabel('Resonant Frequency (kHz)')
ylabel('Gauge Carbon Dioxide Concentration (%)')
legend('L = 1cm', 'L = 5cm', 'L = 25cm', 'Location', 'northeast')
hold on
```
APPENDIX E. CO₂ OPTICAL FILTER ANALYSIS

A simple analysis demonstrating that the selected optical filter (Andover 4.26GA05-25) for Design 1 passes a large fraction, but a fraction nonetheless, of LED radiation in the 4.3\(\mu\)m CO₂ absorption band [1]. All values were averages obtained from the components’ data sheets. Note that in addition to minor signal attenuation due to over-filtration (see Figure E.1), the optical filter only transmits 75\% of LED power within the allowed bandwidth. Thus, over a quarter of useful radiated LED power is lost with the use of the filter.

Figure E.1: Overlapping FWHM spectral bandwidths of CO₂ absorption, LED43 emission, and 4.26\(\mu\)m optical bandpass filter.
APPENDIX F. DESIGN 5 PROCESS SHEET

1. Ensure set-up is correct
   - Components should be in their places as indicated on the foam mat
   - All three LIA terminals should be connected to the set-up
   - Correct codes should be uploaded to LED/datalogging Arduino and buzzer Arduino
   - CO2 cartridge should not be empty

2. Warm up equipment for at least 45 minutes
   - Turn on LIA, power supply, and fans (fans on high; buzzer Arduino not powered yet)
   - Securely connect LED Arduino power cord to power supply (SD card not inserted yet)
   - Place pink foam layer on set-up
   - Let sit for 45+ minutes (set a timer)

3. Flush cell with CO2 and let warm up for 45 minutes
   - Remove pink foam layer
   - Insert SD card
   - Insert CO2 injector needle into inlet tube
   - Position probe plug near outlet tube for quick plugging
   - Open CO2 valve enough for moderate flow for 20 seconds
   - Immediately plug cell outlet tube (tubes should connect to cell horizontally)
   - Reset LED Arduino (verify that orange light on SD card module periodically blinks)
   - Replace pink foam layer
   - Let sit for 45 minutes (set a timer)

4. Start Leak Rate Test
   - Plug in buzzer Arduino with wall USB adaptor (buzzer should be audible)
   - Unplug cell inlet and outlet tubes (tubes should connect to cell horizontally)
   - Let sit until last resonant peak is passed

5. End Leak Rate Test
   - Remove pink foam layer
   - Disconnect LED Arduino from power
   - Turn off power supply, LIA, and fans and unplug buzzer Arduino from USB wall adaptor
   - Remove SD card

6. Post Processing
   - Copy SD file to laptop
   - Plot data using AudioProcessing_LIAaveraging.m (Octave/MATLAB)
   - Save a picture (.png) of whole data set
   - Record times where each resonant peak occurred and add to LeakRatesPlot_Design5.m

Figure F.1: Process sheet for obtaining resonant profiles as CO2 leaked in Design 5.