2012-12-12

Methods for Identifying Acoustic Emissions From the Front Face of a Small Piezoelectric Blower

Brad K. Solomon
Brigham Young University - Provo

Methods for Identifying Acoustic Emissions

from the Front Face of a Small

Piezoelectric Blower

Brad Solomon

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

Master of Science

Jonathan D. Blotter, Chair
Scott D. Sommerfeldt
Kent L. Gee

Department of Mechanical Engineering

Brigham Young University

December 2012

ABSTRACT

Methods for Identifying Acoustic Emissions
from the Front Face of a Small
Piezoelectric Blower

Brad Solomon
Department of Mechanical Engineering, BYU
Master of Science

This thesis focuses on identifying acoustic noise generating components in piezoelectric blowers through transverse velocity measurements and the development of a numerical fluid model. Piezoelectric ceramics have proven useful for many industries and areas of research involving: high precision actuators, noise control, ultrasonic devices, and many other areas. As of late, a unique adaptation of piezoelectric ceramics is surfacing in the area of pumping and cooling. Air pumps that use these ceramics replace the traditional electric motor, resulting in lower power consumption, less moving parts, constant pressure gradients, lower overall weight, and a low profile. The current drawback of this application is the acoustic radiation produced by the blowers. Since these blowers are new to market, little research or development has been done to characterize the noise emissions. This thesis studies the acoustic emissions from the front face of a Murata piezoelectric blower.

Jet noise and structural vibrations are two acoustic sources of interest that are studied in this research. A Direct Numerical Simulation (DNS) of the fluid flow through a Murata blower is developed to better identify noise generating mechanisms. The model solutions predict trends in sound pressure levels (SPL) of the jet noise and volumetric flow rates. Both the SPL and flow rate are shown to be functions of critical geometrical dimensions within the flow path of a Murata blower. Important dimensional components are identified as well as non-influential ones. Design guidelines are given to reduce noise emission from the front side of a blower and increase the volumetric flow rate. The results of this research have a direct impact on the piezoelectric blower industry and future blower designs.

ACKNOWLEDGMENTS

know the Lord blessed me because of my family's prayers. I will forever be grateful for this learning opportunity and the life skills I obtained from completing this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

xiii

1   INTRODUCTION

1.1   The Problem and Motivation

The use of piezoelectric ceramics has broadened and been adapted to a variety of circumstances and applications in recent years.   Such applications range from use in loudspeakers to atomic force microscopes.   Recently, an application has been found for piezoelectric ceramics in air pumps.   These air pumps are referred to as piezoelectric blowers. The blowers are designed to have a small profile when compared to a traditional motor-driven cooling fan.   One of the smaller blowers manufactured by Murata Inc. measures 10 mm x 10 mm and has a height of only 1.3 mm.   Figure 1-1 demonstrates the size of a slightly larger blower also manufactured by Murata Inc.
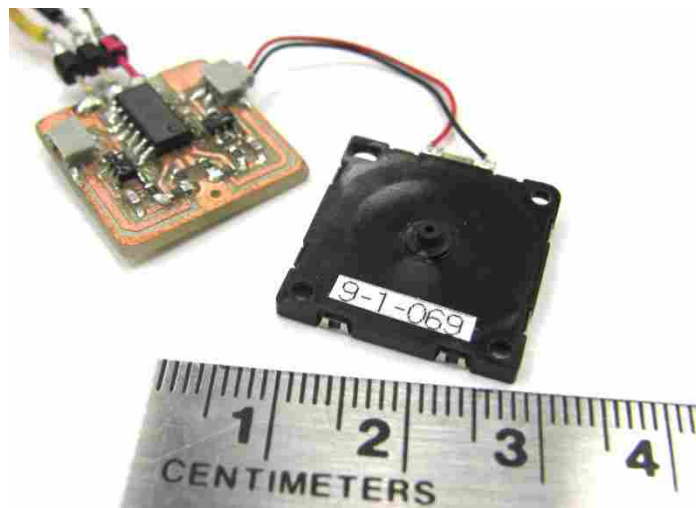
Figure 1-1: A size comparison of a Murata piezoelectric blower.

One advantage to piezoelectric blowers is they generally consume less power than a comparable motorized fan because the blower has no need to spin fan blades.[1] The traditional motor mechanism is replaced by a piezoelectric ceramic. This ceramic is excited by an oscillating voltage which causes the ceramic to vibrate rapidly. The ceramic is generally attached to a pumping chamber which pushes air out of an exit nozzle. In the case of the Murata blower, the ceramic is driven at 25.4 kHz. Figure 1-2 labels the key functional components within a Murata blower.
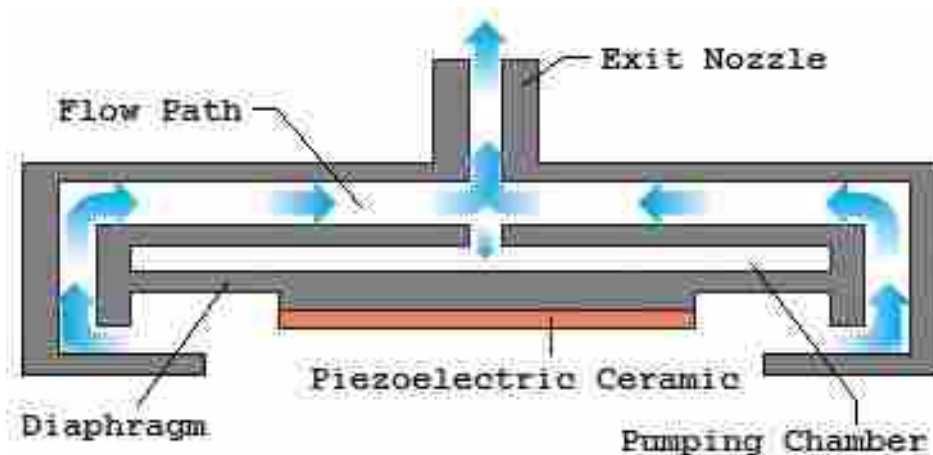
Figure 1-2: A cross sectional view of the blower with blue arrows showing path of air.

The diagram in Figure 1-3 demonstrates the positions of the ceramic during a single oscillation. Views 1 and 2 in the figure show the ceramic's location during a vacuum and a discharge action. This dynamic motion results in a constant flow out the exit nozzle. The exact physics behind the flow generation is not a point of topic in this thesis but it is important to understand that air enters the back side of the blower and is ejected out the opposite side through a nozzle. The Murata blower seen in Figure 1-1 can produce a maximum flow rate of 1.0 L/min and a maximum static pressure of 2.0 kPa. In the case of the Murata blowers, increasing the

flow rate also increases the static pressure. Whereas, traditional cooling fans generally compromise a higher flow rate for a lower static pressure.[1]
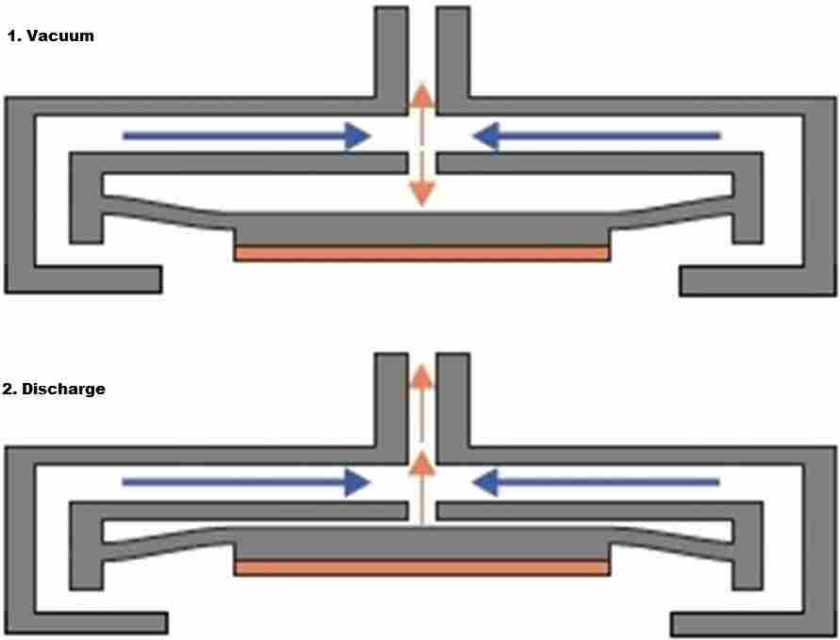
Figure 1-3: A cross sectional view of a blower demonstrating the operating principle.

The small profile and efficient cooling capabilities open new possibilities for heat dissipation and cooling electronics. Generally, the processing power in small mobile devices is limited by the cooling capabilities. A larger processor produces more heat and requires a more complex and powerful cooling system. By using these blowers, small electronic devices can implement larger processors, be cooled efficiently, and not compromise the size or power requirements of the system.

A common problem with piezoelectric blowers is the emission of acoustic radiation. The noise they produce is not damaging to the human ear but is undesirable and irritating during

prolonged exposures due to some higher frequency content. The overall sound pressure level for the blower in Figure 1-1 measured approximately 52 dB[a] (P_{ref} = 20μPa) at 0.5m and 0° off the jet axis. If piezoelectric blowers are to be used in electronic devices such as phones, tablets, laptops or any other electronics that are in close proximity to humans, the source of the noise generation needs to be understood and minimized. Passive attenuation methods such as adding acoustic mufflers are effective and useful at minimizing the noise. But these methods of attenuation add to the profile and size of the blower. This is not desirable in some situations. Thus, it is important that acoustic radiation emitted by piezoelectric blowers be studied. Measuring the acoustic noise radiated from a blower can help characterize the noise. Specific blower components that generate noise can be better understood by numerically modeling the flow path and pressure field of a blower. If critical components of the sound field are identified, guidelines for improving sound pressure levels and volumetric flow rates can be developed. Future blowers could then be useful in new applications that have previously been impossible.

1.2   Objective and Goals

The main objective of this research is to characterize the acoustic emissions generated from the front face of a Murata piezoelectric blower. To accomplish this, accurate measurements of the blower's sound power and acoustic frequency spectrum are taken and analyzed. Characteristics of jet noise are studied. The contribution of structural vibrations to the overall sound power is determined. Another critical step is numerically modeling the pressure field generated at the exit nozzle of a Murata piezoelectric blower. The model is used as a tool to identify which geometrical components in a blower contribute the most to turbulent mixing or jet

---

[a] The SPL measurement spanned from 20 Hz to 20 kHz

noise. The model is a Direct Numerical Simulation (DNS). It predicts flow rates and sound pressure levels for varying flow paths within a blower. Geometrical parameters for altering critical dimensions of the flow path are incorporated into the model such that the user can efficiently make changes to model geometry. The main objective of this research and the steps needed to achieve it are outlined in Table 1-1.

Table 1-1: Research objective and steps

OBJECTIVE	Characterize the acoustic emissions from the front side of a Murata piezoelectric blower and identify any specific sources of emission
STEP 1	Determine the contribution that structural vibrations have on the front side sound power
STEP 2	Identify any jet noise contributions to the front side sound power
STEP 3	Develop a DNS model that predicts trends in SPL near the nozzle exit and flow rates
STEP 4	Identify any geometrical components that are contributors to jet noise at the nozzle exit

1.3 Hypothesis

The goal of this thesis is to understand the acoustic noise generated from the front side of a Murata piezoelectric blower. A simple understanding of how sound waves are produced helped define the steps taken in this research to achieve this goal. When an object shifts or vibrates in a medium such as air, it causes a pressure differential to form near the object's surface. [2] This pressure change can travel through the air in the form of a wave. If a pressure wave is driven at specific frequencies, they can be detected by the human ear as sound. Thus, the source of acoustic noise can be a motion or vibration.

In the case of the blower there are at least two significant sources of motion and/or vibration of particular interest. One is structural vibrations and the other is turbulent mixing. Both sources can result in sound generation and can be identified by experimental means. The structural vibrations are due to the rapid velocity and motion of the piezoelectric ceramic. Some

of this energy is transferred through the physical structure of the blower, causing other components in the blower to also vibrate. It is possible that structural modes on the outer surface of the blower are formed during operation and radiate some of this energy as acoustic noise.[3] The second source of motion or vibration mentioned above happens at a smaller scale in free space as air travels through the blower. Turbulent mixing of the air occurs as it is pushed through the chambers and channels of the blower and out the exit nozzle. This is known as jet noise.[4] The formation and characteristics of jet noise are discussed in greater detail in Chapter 2 of this thesis. Structural vibrations and jet noise are two of many possible sources that contribute to the sound power spectrum of a blower.

It is hypothesized that the contribution of structural vibrations to the sound power spectrum of a blower can be measured and quantified using a Scanning Laser Doppler Vibrometer (SLDV) and plate radiation theory. If dominant frequencies in the sound power spectrum of the front side of a blower are shown to be caused by structural vibrations, then mechanical design changes to the structure of the blower can be made to minimize the acoustic radiation of those frequencies.[5] Experimental testing can be done to identify if jet noise is another contributor to the sound power spectrum near the nozzle exit. However, jet noise is harder to understand and mitigate than purely structural vibrations. Therefore, a DNS model of air as it passes through the free space in a blower could be a useful tool in identifying blower components that contribute to or influence jet noise. With a robust model, parameters such as nozzle height or nozzle radius can be changed and corresponding sound pressure levels (SPL) near the nozzle exit can be resolved along with resulting flow rates. A DNS model of a blower will help identify any critical geometrical components that contribute to both these properties. Results from such a model will help develop guidelines for designing a blower that increases

flow rate and lowers jet noise at the nozzle exit. Any geometrical components that have a significant impact on the blower's performance regarding flow rates and jet noise can be identified with a model. The results will allow future blower designers to better understand the noise generation and focus only on the components that have a significant impact on overall blower performance.

The model can be a Direct Numerical Simulation (DNS) model. This method of resolving flow fields is generally avoided due to excessive computational costs. However, the small nature and simple design of the Murata blower makes a DNS solution feasible. The case for using a DNS method to resolve the flow field of a Murata blower is made in Chapter 3.

In conclusion, the sound generation of a piezoelectric blower is unique and therefore has had little research and study. Using plate radiation theory, the structural vibration contribution of sound power for the front side of a Murata blower can be identified and quantified. With the development of a DNS model, critical mechanical components within the blower that influence the SPL near the nozzle exit and the flow rate can also be identified. The results of the research can provide important design information for future prototypes of the Murata blower.

1.4   Work Scope and Thesis Outline

The goals and objective given in Table 1-1 have been achieved using the testing equipment available at Brigham Young University. All experimental sound measurements have been conducted in the Large Anechoic Chamber at BYU to reduce reflections of sound in any recorded data. The audible frequency range is the most important band width looked at during this research. Recorded data can be sampled at a rate as high as 204,800 Hz. This sample rate is more than sufficient for measuring the frequencies of interest in this research. Blowers are being provided by Murata Inc. The test setups and flow modeling will be based on the structural

design of Murata blowers which function on the principles shown previously in Figure 1-3. The blower will be powered by 18 V peak to peak through the provided Murata circuit board for all experimental testing. The circuit board provides a 25.4 kHz square wave to drive the piezoelectric ceramic. Before any test data is taken, the wave signal and voltage will be verified to ensure standardization across all test measurements.

The thesis will follow the recommended outline as prescribed by the BYU Mechanical Engineering Department Guidelines. The end of this chapter contains a literature review and a brief background in the area of piezoelectric ceramics and acoustics. Chapter 2 describes two sources of acoustical noise generated from the front side of a piezoelectric blower; the first being structural vibrations, and the second being jet noise. The contribution of both sources are quantified and validated in the chapter. Chapter 3 discusses the Direct Numerical Simulation (DNS) model of air flow through a Murata blower. Steps to construct and solve the DNS model in ADINA are outlined. The validation techniques, limitations, and benefits of the DNS model are presented. Chapter 4 gives the results of changing various component dimensions within the model. The parameters of interest will include: nozzle radius, nozzle height, inlet radius, inlet gap distance, pump-chamber dimensions, and chamfer size. Lastly, Chapter 5 summarizes the conclusions of this research and includes recommendations for future work. The Appendix gives information for recreating the results of the research.

1.5   Background

In recent years, highly innovative uses of piezoelectric devices have been researched and developed. Such uses range from piezoelectric actuators and transducers in active noise control to high-power ultrasonic devices and even underwater acoustic microphones. The idea to use a piezoelectric actuator as a small air pump is new to the market and consequently, the acoustic

effect generated by such a pump has not been fully researched or disclosed by their designers. Yet, because of modern interest in new piezoelectric applications, a lot of research has been done on topics relating to acoustics and piezoelectric ceramics. In 2006, Lematre[6] modeled the propagation of guided waves in piezoelectric structures while subjected to a pre-stressed gradient. Lematre highlights some key features of dispersion curves appearing in the presence of the gradient. In 2004, Shuyu[7] studied and optimized the performance of a sandwich piezoelectric ultrasonic transducer. The transducer is now widely used in high-power ultrasonic devices and underwater acoustics. Shuyu showed how the length and position of the piezoelectric elements affect the performance of the transducer. In 2003, Geng[8] developed a new modeling method for active noise control in a three-dimensional irregular enclosure using piezoelectric actuators.

One challenging aspect of this research is gaining an understanding of the interaction between sound and fluid flow in small spaces. Many published articles have relevant research in this regard. An older article written in 1971 by Beechmans and Sen-Gupta[9] looks at flow through porous media in the presence of sound. The research found that the pressure gradient of a specified flow of air through a porous material increases with the presence of sound. In a more recent article published in 1994, Lacour[10] studied the effect of fluid compressibility and permeability on porous piezoelectric ceramics. He concluded that it is possible to retain the desirable property of a high piezoelectric coefficient along with good mechanical properties by applying a high hydrostatic pressure to the ceramic.

Another area that presents a challenge in this research is characterizing the noise generation of the Murata blower. The noise can come from many sources. One source is jet noise. Tam[11] wrote an article in 2010 that describes the characteristics of jet noise by presenting

directivity and spectral characteristics. Published research by Lush[12] specifically measures subsonic jets and compares his results to low-speed jet theory. His publications are particularly useful for characterizing the blower noise as jet noise in this thesis. Further details in this area are presented in sections of Chapter 2. Background information and theory on Computational Fluid Dynamics (CFD) and DNS modeling is also needed for completion of this research. These topics play a large role in the results of this thesis. Therefore, a more thorough review on these topics is presented in Chapter 3. It is hoped that the results of this research will help the developers at Murata improve their blower design and be instrumental in helping identify important components that influence SPL and flow rate.

2 ACOUSTICAL NOISE CHARACTERISTICS OF A BLOWER

This chapter describes and discusses two theories for possible acoustic noise in piezoelectric blowers. One is structurally induced noise from internal vibrations and the other is jet noise. These two sources of noise can contribute to the total acoustic radiation emitted by a blower. Experiments in this chapter use plate radiation theory to show how much internal vibrations actually contribute to a measured acoustic field. The characteristics of jet noise are discussed and methods for identifying it are presented. Characterizing the noise emission from the front face of a Murata blower is the focus of this chapter.

2.1 Noise Induced from Structural Vibrations

Many sources of sound radiate their energy by means of a vibrating solid surface that impacts a surrounding fluid. Some examples of this phenomenon include bells, guitars, and loudspeakers. It is rational to think that structural vibrations in a Murata blower could be a driving source of acoustic radiation since the piezoelectric is oscillating at 25.4 kHz! This rapid driving of the ceramic, coupled with broadband noise impacting the blower, excite structural modes on the outside surfaces of the blower. Such modes will be referred to as operational modes. In this case the blower would be acting like a loudspeaker producing and radiating sound. It therefore becomes necessary to identify any such operational modes that might be

efficiently radiating acoustical noise. A look into plate radiation theory can help identify any modes that might exist and quantify their resulting sound power.

2.1.1 Plate Radiation Theory

Plate radiation theory suggests that radiated sound power from a plate can be predicted at a given frequency. When a plate is excited by broadband noise, it can vibrate at multiple frequencies, some of which can effectively radiate acoustic noise. The efficiency depends on the amount of resistance to acoustic radiation at the given frequency. Thus given the resistance of the plate, the amount of radiated sound power from a baffled plate can be calculated.[13]

Imagine a vibrating plate that can be broken into R number of small elements. The total resistance to radiation of the entire plate can be expressed in the radiation resistance matrix $[R]$ in Eq. (2.1). The matrix is dependent upon the physical properties of the plate and fluid. With R number of elements in the plate, the matrix will be an R x R matrix, where ω is the frequency, ρ_o is the medium density, A_e is the elemental area, c is the speed of sound in the medium, and kd_{ab} is the wave number multiplied by the distance from element d_a to d_b.

$$[R] = \frac{\omega^2 \rho_o A_e^2}{4\pi c} \begin{bmatrix} 1 & \frac{\sin kd_{12}}{kd_{12}} & \cdots & \frac{\sin kd_{1R}}{kd_{1R}} \\ \frac{\sin kd_{21}}{kd_{21}} & 1 & & \vdots \\ \vdots & & & \\ \frac{\sin kd_{R1}}{kd_{R1}} & \cdots & & 1 \end{bmatrix} \tag{2.1}$$

Figure 2-1 shows how each sub-element of the divided plate has a unique normal velocity $(\tilde{v}_{er})$. Each of these velocities depends on the driving frequency of the plate. A column vector

$\{\tilde{V}_e\}$ can be formed that contains the transverse velocity for each individual sub plate as defined by Eq. (2.2).
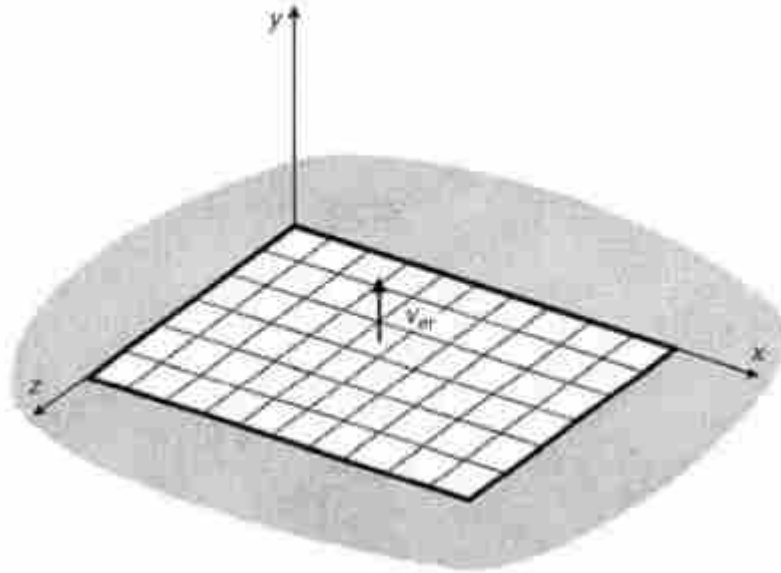
Figure 2-1: Subdivision of a panel into elemental radiators.

$$\{\tilde{V}_e\} = \lfloor \tilde{v}_{e1}\ \tilde{v}_{e2} \cdots \tilde{v}_{eR} \rfloor^T \qquad (2.2)$$

The total radiated sound power is calculated using the radiation resistance matrix in Eq. (2.1) and the transverse velocity column vector in Eq. (2.2). The sound power of the plate is now defined in Eq. (2.3) and shows a dependence on frequency.

$$\bar{P}(\omega) = \{\tilde{V}_e\}^H [\boldsymbol{R}]\{\tilde{V}_e\} \qquad (2.3)$$

A powered Murata blower has a front face that can be assumed to be a vibrating plate. Eq. (2.3) can be used to predict the amount of sound power at a given frequency that theoretically should radiate from the blower surface. This value can then be compared to a measured value of sound power at the same frequency. If the predicted value is equal to the

13

measured value then it could be concluded that the sound source is structural in nature. However, if the predicted power is much less than the measured power, then it can be concluded that the measured power derives mostly from other sources. The structural vibrations in such a case would be insignificant contributors to the sound power. The result would suggest that operational modes are only a small factor that contributes to the acoustics of the blower.

2.1.2 Experimental Setup and Results

A calculation of the radiated sound power in Eq. (2.3) from an operational mode of a blower requires identifying the transverse velocity column vector of Eq. (2.2). This vector is measured using a Scanning Laser Doppler Vibrometer (SLDV). The front face of the blower is broken into a grid of 2,401 points. The blower is mounted in a baffle that only exposes the front face of the blower. While the blower is powered, the laser scans each of the 2,401 points and records the corresponding normal velocity and phase across a specified frequency spectrum. A 1/8 inch microphone is placed near the surface of the blower during the scan to capture a reference signal. These measurements are used for the transverse velocities required for calculating the sound power. This method is used to calculate the radiated sound power from the front surface of a Murata blower at given frequencies. Figure 2-2 shows the test setup that was used during this process.
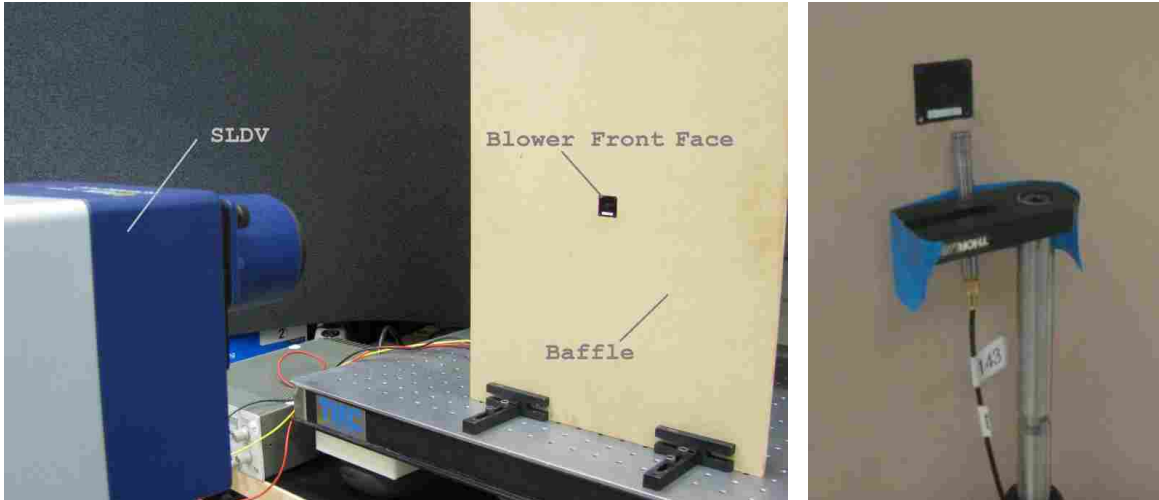
Figure 2-2: SLDV measurement test setup (LEFT), Microphone placement (RIGHT).

The resulting velocity spectrum of the blower's face shows a large spike at 25.39 kHz near -27 dB (V_{ref} = 1m/s). This is the driving frequency of the blower and is the dominant vibration frequency as would be expected. This is shown by Figure 2-3.
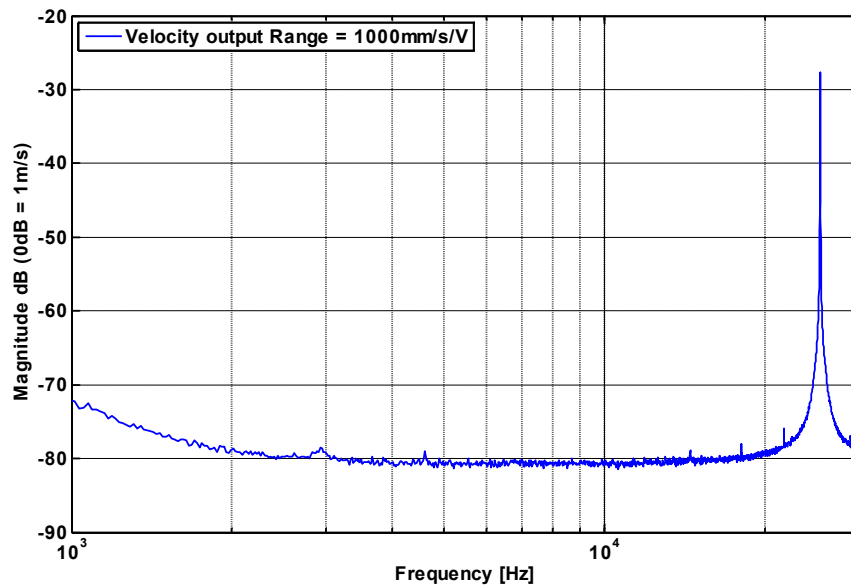
Figure 2-3: Transverse velocity spectrum of the front blower face.

15

However, the driving frequency is outside the audible range. Looking at frequencies from 1000 Hz to 20 kHz, there is nothing near the magnitude of the driving frequency. But by changing the voltage and resolution settings[b] on the SLDV the noise floor of the measurement can be lowered. Retesting with adjusted settings gives a better idea of what is happening in the audible range. Figure 2-4 displays three prominent spikes above -100 dB ($V_{ref} = 1$m/s). This velocity spectrum indicates that many operational modes, although much less in magnitude, are excited by the driving frequency. The three most prominent are at 14.36, 6.75, and 2.93 kHz.
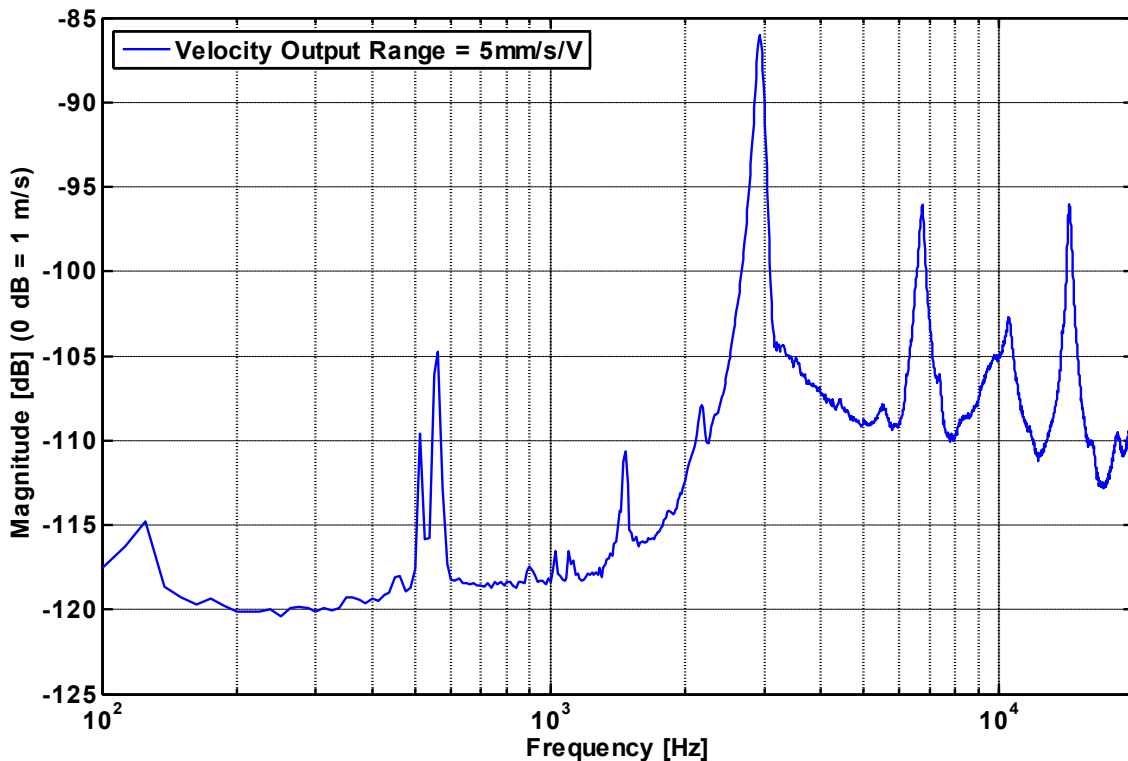
Figure 2-4: Audible transverse velocity spectrum of the front blower face.

---

[b] Refer to Appendix A.1 for SLVD settings used

Figure 2-5 through Figure 2-7 show these operational mode shapes as measured by the SLDV. The results indicate that the blower face acts like a circular membrane with a fixed rim.14 The 14.36 kHz operational mode shape in Figure 2-5 has zero radial node lines and two nodal circles. This vibration mode can thus be denoted as the (0, 2) vibration mode. The 6.75 kHz frequency vibrates in the (1, 1) mode as seen in Figure 2-6. Lastly, the 2.93 kHz frequency vibrates in the simplest mode of (0, 1) as Figure 2-7 demonstrates.
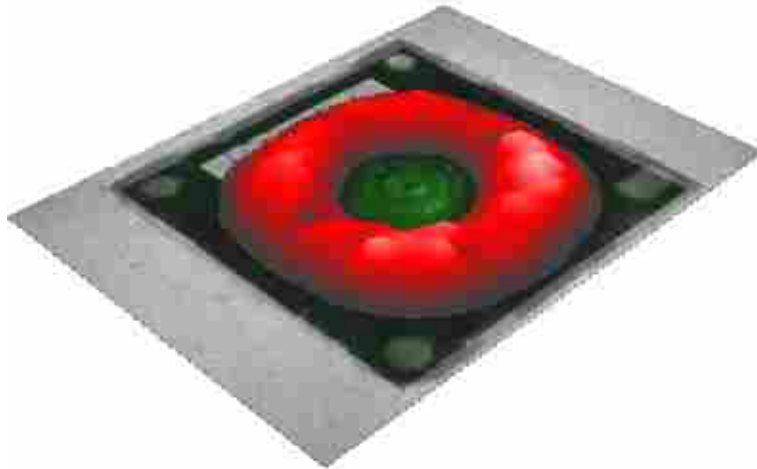
Figure 2-5: Operational mode shape of blower face at 14.36 kHz.

Figure 2-6: Operational mode shape of blower face at 6.75 kHz

17
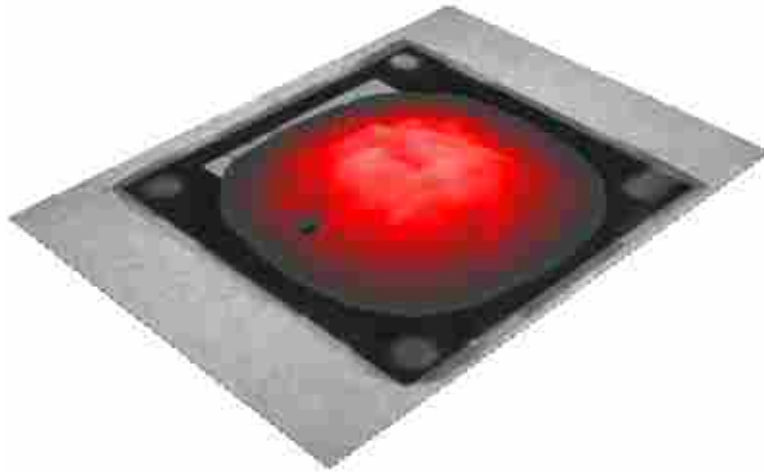
Figure 2-7: Operational mode shape of blower face at 2.93 kHz

Given these test results, the amount of sound power radiated by each operational mode can be calculated. The calculated values of radiated sound power can then be directly compared to an actual sound power measurement. The comparison will reveal how much of the measured sound power is a result of the operational modes induced by the driving frequency of the blower.

Measuring the sound power is done in an anechoic chamber to minimize the effect of acoustic reflections on the measured data. Because the measurement will be compared to the sound power radiating only from the front face of the blower, the exact same blower and mount used in the SLDV test is used during this measurement. A semi-circular boom with an array of free-field microphones was used to measure the noise emitted by the blower. Measurements were acquired at boom rotations from 0º to 180º in 10º intervals. Sufficient extensions to the baffle were added to reduce noise emitted from the back side of the blower. The test setup is seen in Figure 2-8.
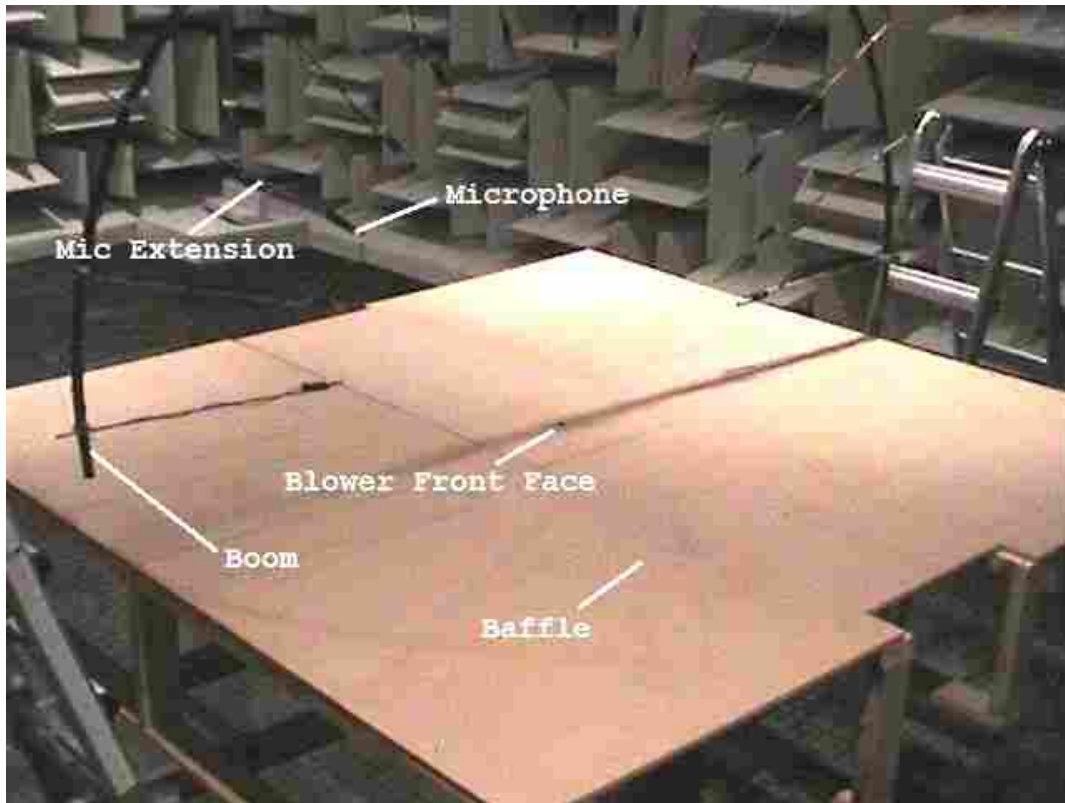
Figure 2-8: Sound power measurement setup for front face of blower.

From this data collection the sound power spectrum for the Murata piezoelectric blower is calculated and plotted in Figure 2-9. The overall sound power for the front face of the blower was computed to be 44.2 dB$(ref\ P = 10^{-12}\ W)$[c]. Spikes in sound power appear at the same frequencies where operational modes exist.

---

[c] Calculated from 1 to 20 kHz using the code in Appendix A.2
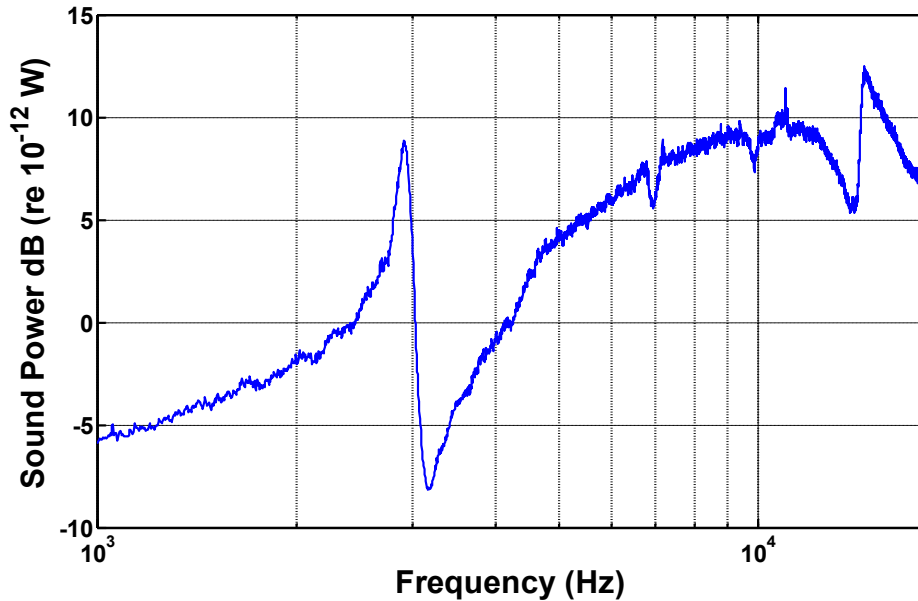
Figure 2-9: Sound power spectrum measured from the front face of a blower.

The radiated sound power of the three most prominent operational modes in the audible frequency range can be compared to the measured sound power obtained from Figure 2-9. Table 2-1 shows the comparisons for the three modes and includes a comparison of the driving frequency as well.[d]

Table 2-1: Structural vibration contribution to radiated sound power.

Operational Mode	Measured Power in Anechoic Chamber		Calculated Power from SLDV		Decibel Spread	% Structural Mode Sound Power
kHz	dB (ref P=10^-12 W)	picoWatts	dB (ref P=10^-12W)	picoWatts	dB (Measured-Calc.)	(Calc. Watts/ Measured Watts)
2.93	7.68	5.86	0.56	1.27	7.12	21.67%
6.75	7.54	5.67	1.03	3.05	5.72	54.76%
14.36	12.5	17.8	12.38	17.3	0.50	97.19%
25.39	73.87	24.38 µW	71.33	13.57 µW	2.54	55.66%

The last column in Table 2-1 gives the percentage of the structural vibration contribution to the measured sound power by dividing the calculated watts by the measured watts. The

---

[d] Refer to Appendix A.3 for the sound power calculations from the SLDV data

percentage of structurally induced sound power is greater than 21% for all three modes in the audible frequency range. The higher the frequency, the more it appears to contribute to the measured noise level. This comparison shows that the majority of radiated sound power at 6.75 and 14.36 kHz is produced by the operational modes at those frequencies.

The last row of the table gives the results for the operating frequency mode of 25.39 kHz. Since the blower is driven at this frequency, it should be expected that a large portion of the sound power at this frequency is due to structural vibration. 71.33 dB of sound power was calculated to be radiating due to the vibration at the driving frequency. The measured power was 73.87 dB. The results have a decibel spread of 2.54 dB. This is a -3.4 % error from the measured value.

The results presented in this section show that the contribution of structural vibrations to the audible sound power of a blower should not ignored. The spikes in the measured sound power in Figure 2-9 at 2.93, 6.75, and 14.36 kHz are produced in part by the operational modes of the blower. Minimizing this vibration would reduce the sound power spikes in the spectrum and reduce the overall sound power emitted by the front face of the blower.

2.2   Characteristics of Jet Noise

Structural vibrations contribute to single audible frequencies but broadband noise is also emitted from the front face of the blower as Figure 2-9 shows. Now the research looks at the flow of air as it exits the blower and the resulting acoustic noise. This phenomenon is known as jet noise and is a common source of acoustic noise. Some industrial examples of this might include: pneumatic tool exhaust, blow-offs from relief valves, gas turbine exhaust, and air ejectors. Jet noise can be categorized into supersonic or subsonic noise generation.[15] The blowers in this research deal with subsonic noise generation.

The basic mechanism of jet noise is due to turbulent mixing between the ambient air and the jet boundary layer. Mixing results from the viscosity at the boundary layer. This mixing causes an entrainment of air between the high-velocity jet and the ambient air. The resulting turbulence causes sheer stresses that result in the generation of quadrupole-type sound sources. These sources then radiate the noise into the surrounding medium. This section describes the directional and spectral characteristics of jet noise and describes the methods used to identify these characteristics in a Murata blower.

2.2.1 Directionality of Jet Noise and Comparison to Blower Measurements

Since jet noise derives from a quadrupole-type sound source, the directional characteristics are similar to that of an ideal quadrupole source.[16] The maximum noise level for a quadrupole source is emitted at 90º angles. This means a lobe of amplified sound forms at 90º angles. Subsonic jets have a similar characteristic, with the center of the quadrupole located at the nozzle exit and lobes of maximum sound levels occurring at 45º angles off the centerline axis of the flow path.[17] As the jet Mach number increases, convection effects along with shifts due to a doppler shift factor can change the directionality of the lobe. Generally, as the jet Mach number approaches 1.0, the lobes move closer to the centerline of the jet axis. The jet Mach number of the blower is near 0.1 so the lobes would be expect near 45º off center axis. Figure 2-10 demonstrates what a typical directivity plot might look like for such a low-speed jet.[18] The diagram shows the directional characteristics that would be expected if jet noise is a significant acoustic source in Murata blowers.

22
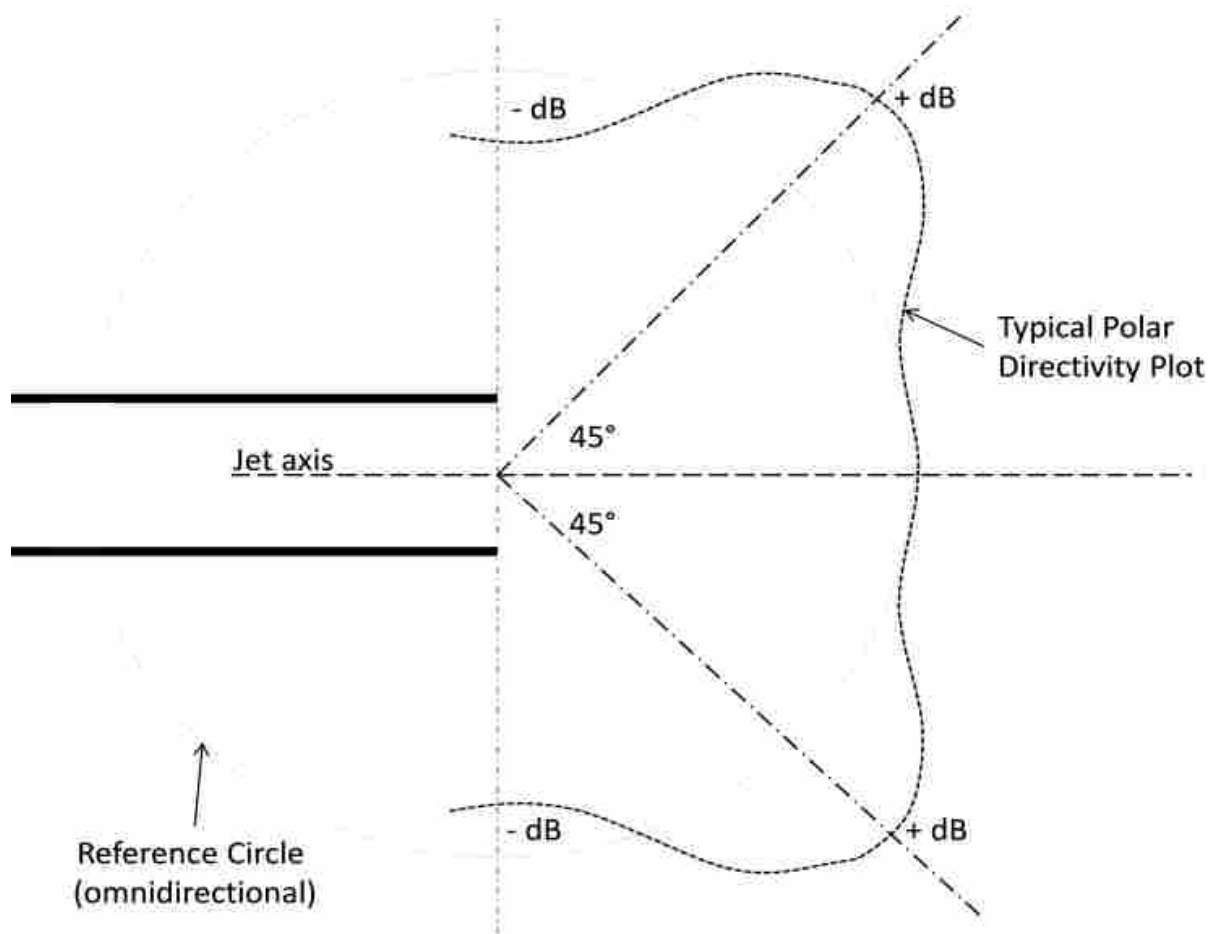
Figure 2-10: Directivity plot for a low-velocity jet.

Directionality testing performed on a Murata blower does show this characteristic but not for audible frequencies. While powered on, a free-field microphone was used to measure the acoustic noise emitted by a Murata blower from 0.5 m. Measurements were taken at 10° intervals to capture the directional characteristics of the jet. The overall sound pressure level for each measurement was computed for three different frequency bandwidths. The resulting directivity plots are shown in a polar plot in Figure 2-11. The profile of the blower during the test setup is shown in the center of the figure.
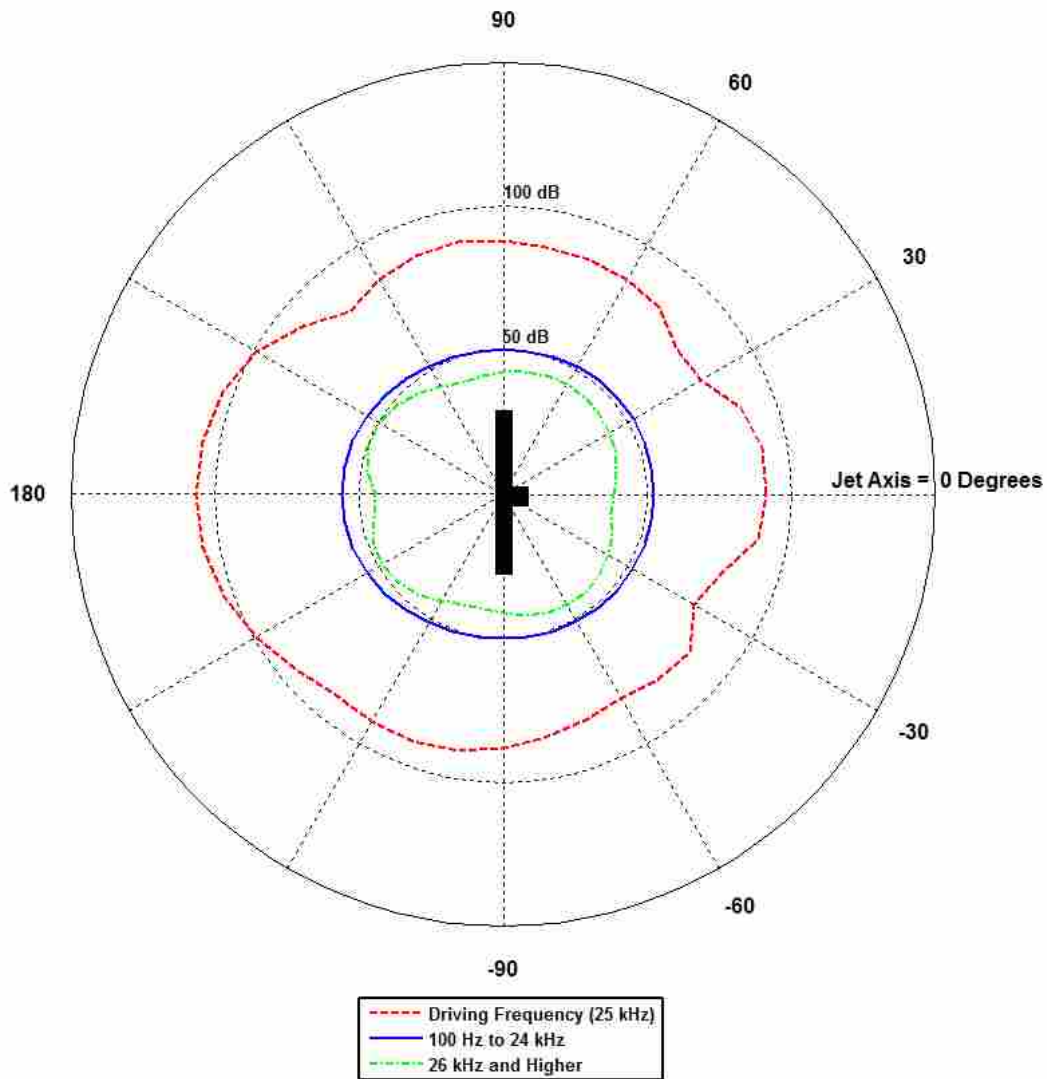
Figure 2-11: Directivity plots of a piezoelectric blower.

The frequencies below the 25 kHz driving frequency are plotted in blue. The directionality appears mostly omnidirectional with the exception of a slight amplitude increase in the rear of the blower. The red polar plot is the driving frequency. It is plotted alone because it dominates the sound pressure level when included with other band widths. The frequencies

above the driving frequency demonstrate the directional lobes characteristic of jet noise and are seen in the green plot. Directional lobes are located at 45º off the jet axis and are also present near 30º off axis from the back side. The measurements show how higher frequencies have directional characteristics of a low-speed jet. The power spectrum density was computed[e] and the result is plotted in Figure 2-12 to better demonstrate the amplitude increases due to the jet. The lobes off the jet axis are located along the 45º and 315 º axes in the plot. This testing suggests the possibility of a jet at frequencies greater than the driving frequency.
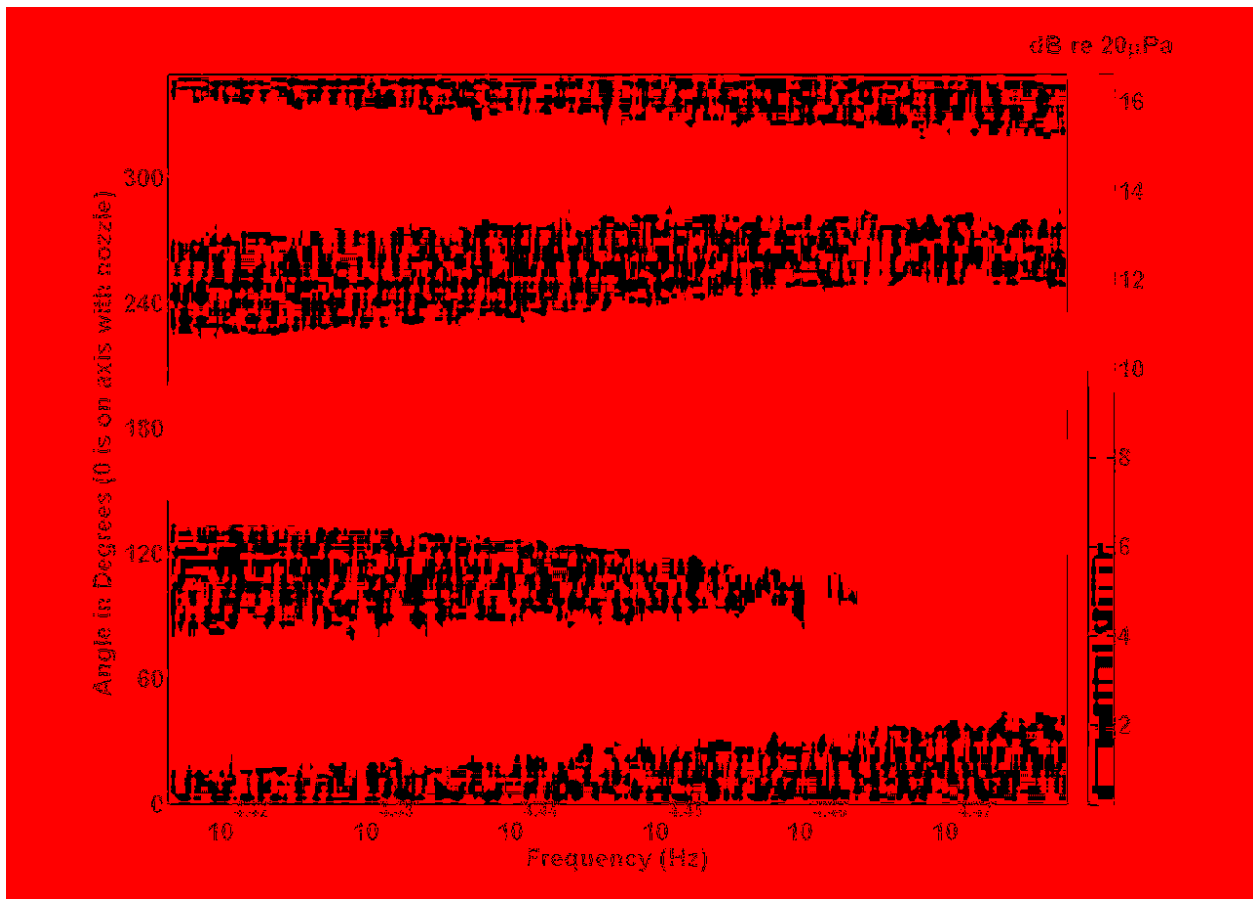
Figure 2-12: Power spectrum density plot for frequencies above the driving frequency.

---

[e] Refer to Appendix A.4 for directivity plotting and power spectral density code used

2.2.2 Spectral Characteristics of Jet Noise and Comparison to Blower Measurements

Subsonic jet noise contains frequency characteristics that can be identified through experimental means.[19] The source of subsonic jet noise is turbulent mixing at the jet boundary layer with the ambient medium. The viscosity of the medium forces mixing to occur and causes a continual entrainment along the jet axis. The thickness of the turbulent mixing region increases as the jet travels farther from the nozzle exit plane. The center portion of the jet has a flat velocity profile as it exits the nozzle. This flat profile is called the potential core of the jet and dissipates while progressing downstream. Figure 2-13 shows what the shear boundary layer and velocity profile might be for a typical subsonic jet as it exits a nozzle.[20, 21]
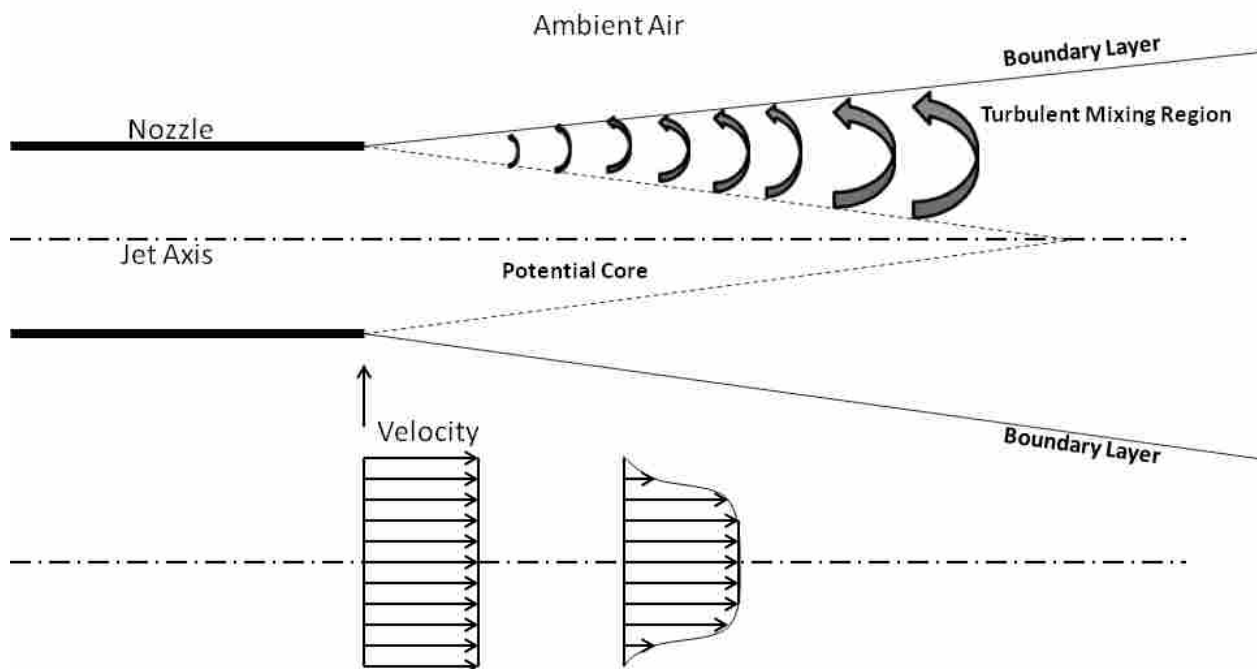
Figure 2-13: Components and velocity profile of a subsonic jet at a nozzle exit.

26

The turbulence upon exiting the nozzle is small-scale and very volatile. As the mixing region for the small scale turbulence progressively increases downstream, the intensity of the turbulence conversely decreases. This concept is important in understanding the characteristics of jet noise. High intensity turbulence at the nozzle exit will produce a high frequency component that dominates at that location. But this dominant frequency will decrease along the jet axis because the intensity of turbulent mixing also decreases. It then stands to reason that a frequency spectrum collected at one location downstream the jet axis will contain contributions from every region; each region having a lower dominant frequency as the distance from the nozzle exit increases. The result is a frequency spectrum that resembles a haystack. The peak amplitude and corresponding frequency of the measured spectrum will depend on the jet Mach number and nozzle properties.[21, 22]

To discover if spectral characteristics of jet noise exist in a Murata blower, a 0.25 in free-field microphone was suspended from the roof of an anechoic chamber and used to measure the acoustic frequency content near the jet boundary layer of the blower. The body of the blower was baffled to expose only the exit nozzle. The baffle was designed to limit the measurement to only noise generated by the air as it leaves the nozzle. The measurements were taken in three locations near the jet boundary layer. This was done assuming one of the tests will end up close enough to the jet boundary layer. Figure 2-14 shows the test setup in the anechoic chamber. All three test locations are 3.35 mm (±0.5 mm) above the exit plane. Measurements 1, 2, and 3 are 1.5, 2.5 and 4.5 mm (±0.5 mm) from the jet axis respectively. Due to the small nature of the blower and the effort to isolate the microphone from the baffle, it was difficult to determine the exact location for each test in relation to the boundary layer. Figure 2-15 best depicts where the measurements were taken in each test.
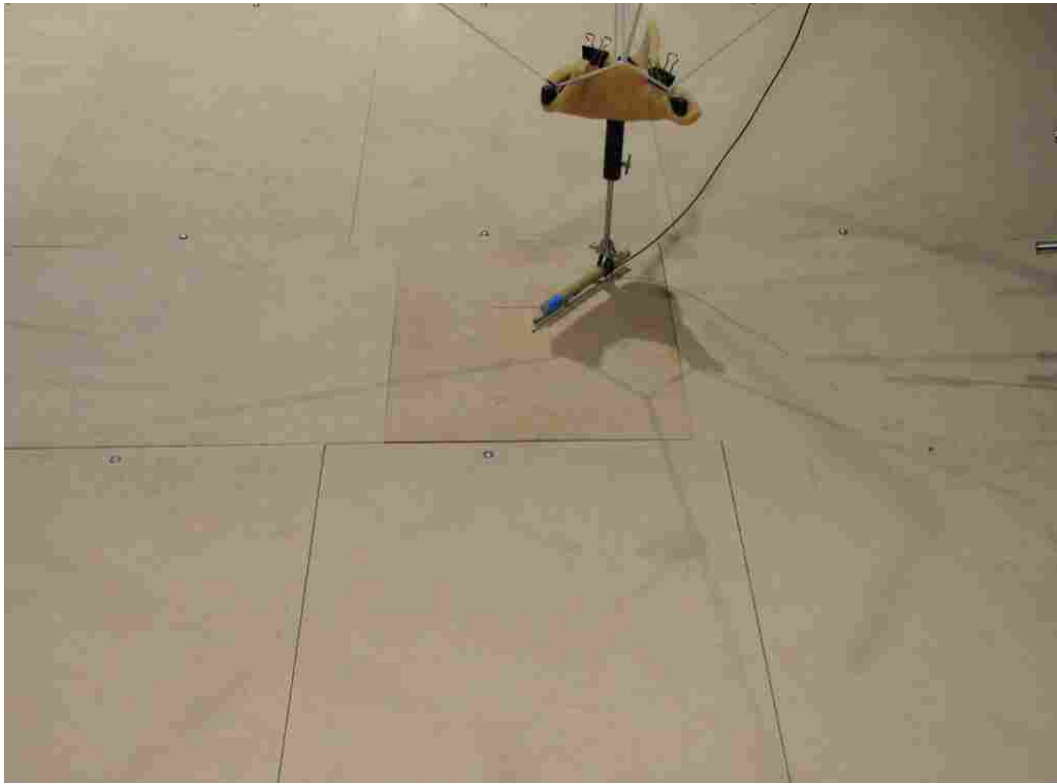
Figure 2-14: Baffle and microphone setup for near field spectral testing.
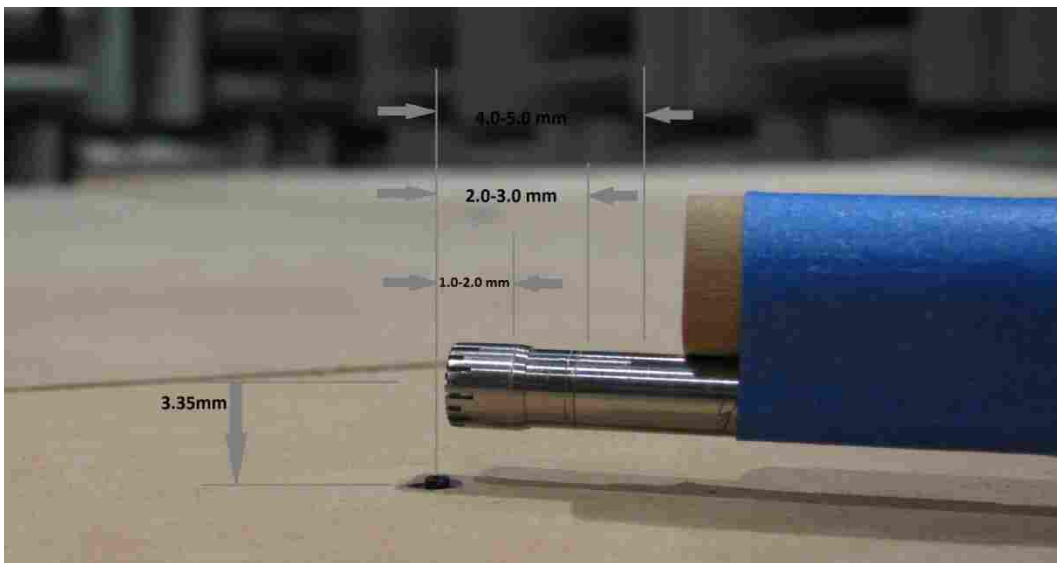
Figure 2-15: Positions of microphone relative to nozzle for spectral testing.

All three test results are plotted[f] with respect to frequency and labeled in Figure 2-16. The blue results were taken closest to the jet axis at 1.5 mm. The haystack like profile is typical of jet noise. However, the profile changes while moving farther away from the jet axis as seen in the red and green plots. The blue profile suggests that jet noise is in fact present and contributes to the sound pressure levels of a Murata blower. The dominant peak frequency is near 1000 Hz. However, the broadband sound power measured from the front face in Figure 2-9 is not audible for frequencies less than 4 kHz.
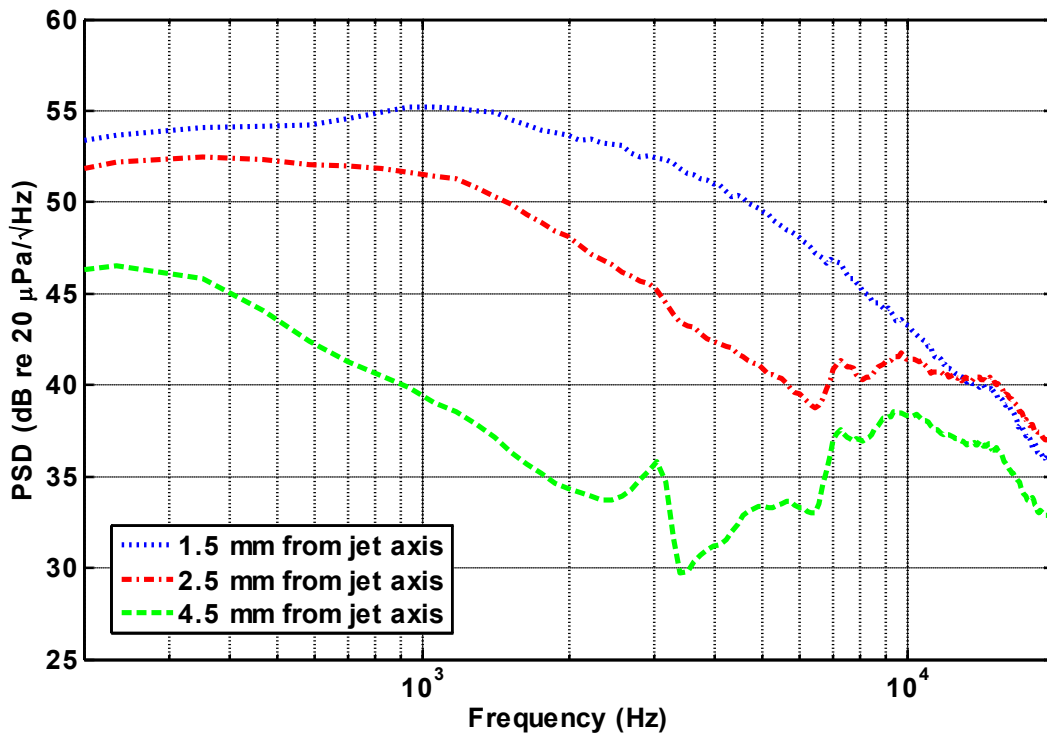
Figure 2-16: Frequency spectra of a blower's air stream.

---

[f] Refer to Appendix A.5 to see how the frequency spectrum was calculated

2.3 Testing Conclusions

The tests conducted in this chapter help explain the measured sound power spectrum of the front face of the Murata blower. The structural vibration contribution to the spectrum was identified and quantified. The operation modes on the blowers face correspond directly to spikes in the measured sound power. The other portions of the spectrum are broadband noise, a common characteristic of jet noise. The audible directivity plot in Figure 2-11 actually indicates that the back side of the blower is slightly louder than the front. It should be noted that the focus of the research is on the sound power emitted by the frontal region of the blower only. The next chapter will discuss the method of modeling the flow through a blower and predicting the SPL of the jet noise when changes are made to the flow path. Modeling the flow in this manner will help understand which geometrical components in the flow path contribute most to the noise emitted by the nozzle.

3   DNS FLOW MODELING

Chapter 3 describes the required steps to model similar flow dynamics of a Murata piezoelectric blower in ADINA using a DNS approach. The sub-sections in this chapter describe the meticulous develop of a reliable, accurate, and dependable DNS model that will simulate air flow through a blower and predict a corresponding flow rate and SPL near the nozzle exit.

Experimental testing from Chapter 2 suggests that the origin of some of the audible broadband noise in piezoelectric blowers is jet noise and/or turbulent mixing. To better understand the structures that contribute to this broadband noise, a flow model can be developed. If a model can be developed that reliably resolves the SPL near the nozzle exit, then geometric parameters of the model can be modified and the corresponding SPL resolved and computed. Any notable changes in the SPL can then be noted. Using the relationship between geometric parameters and SPL, the most significant contributors to the SPL of the broadband noise can be identified.

The reasons for using ADINA and a DNS approach are presented in this chapter. How to construct and solve the DNS model are detailed. The validity of the model is discussed and the model limitations are listed. To support the claim that a DNS approach is reasonable and valid, an informative yet brief background on turbulence theory and flow modeling is given here.

3.1   Turbulence Theory and DNS Modeling

The purpose of this section is to provide the reader with theory on the Reynolds number of a flow field, turbulence in a flow field, the estimation of a Murata blower's Reynolds number, and the numerical modeling of flow dynamics using Direct Numerical Simulation (DNS). Only a basic presentation on these concepts is made here. For greater insight on these subjects, please review the published literature referenced in this section. The discussion here is intended to defend the use of DNS as a means to resolve both the flow and acoustic near field of a Murata piezoelectric blower.

Two chief obstacles that complicate fluid dynamics are geometry and viscosity effects.[23] Most analytical fluid theory applies to idealized and oversimplified geometries such as an infinite plate or flow past a smooth cylinder. Likewise, most real situations have inherent and influential viscous forces that cannot be solved analytically.   These types of problems are generally solved through experimentation or numerically through computational fluid dynamics (CFD).

Viscosity, the first complication, has a destabilizing effect on fluids and gives rise to a disorderly and fluctuating disturbance in flow called turbulence. Because of this complication, turbulence theory is backed up in large part by experimental results and many areas of the field remain underdeveloped.   A primary parameter for correlating this viscous behavior in any Newtonian fluid is the Reynolds number and is defined by Eq. (3.1). Geometric and velocity scales along with kinematic viscosity define the dimensionless number. Its value is used to determine the turbulent character of flow in a system.[24]

$$R_e = \frac{UL}{\nu} \tag{3.1}$$

High Reynolds numbers indicate turbulent fluid flow while lower numbers indicate laminar fluid flow. Despite the chaotic nature of turbulence, it does contain coherent structures that develop and decompose with time. These structures are called eddies. Both the dynamic and geometrical properties of the largest eddies influence the overall mean flow of a fluid system. Large eddies break into smaller eddies as time progresses. This transfers the kinetic energy of the mean flow into smaller and smaller scales of motion until finally it is converted into heat through viscous dissipation. This energy dissipation rate depends largely on the dynamic and geometric properties of the biggest eddies within the flow system. An expression for this rate of dissipation is defined by Eq. (3.2).[25]

$$\varepsilon \sim \frac{u^3}{l} \tag{3.2}$$

The l in the dissipation rate is defined as a fractional[g] section of a characteristic length of the system. The u in the equation is generally a percentage[h] of the mean velocity flow. The characteristic length for a Murata blower would be the nozzle diameter. The mean flow velocity is considered the exit velocity. An application of dimensional analysis with the dissipation rate equation defined in Eq. (3.2) and kinematic viscosity yields an expression for the Kolmogorov scales.[25] These scales are presented in dimensional form in Eq. (3.3) through (3.5). These are the smallest scales of length (η), velocity (U), and time (τ) that can still impact and influence the dynamics of a flow field. Any numerical simulation of a turbulent flow field must account for the dynamic activity happening at these scales which are defined by the dissipation rate of the system.

---

[g] 0.1 is generally multiplied by the geometry's characteristic length
[h] 0.05 is generally multiplied by the mean flow velocity

$$\eta = \sqrt[4]{\frac{\nu^3}{\varepsilon}} \qquad (3.3)$$

$$U = \sqrt[4]{\nu\varepsilon} \qquad (3.4)$$

$$\tau = \sqrt{\frac{\nu}{\varepsilon}} \qquad (3.5)$$

All numerical fluid models are based on the Navier-Stokes equations of Newtonian fluid motion. When coupled with the equation of continuity they describe the motion of a fluid at any unique instant in time. Any attempt to resolve a flow field must be based on the governing equations of fluid motion defined in index notation by Eq. (3.6) and the equation of continuity defined by Eq. (3.7).[26]

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = \mu \frac{\partial^2 u_i}{\partial x_l^2} - \frac{\partial}{\partial x_i}P + \rho g_i \qquad (3.6)$$

$$\frac{\partial u_i}{\partial x_i} = 0 \qquad (3.7)$$

An idealistic approach to resolving a flow field is the discretization of the above equations while applying sufficient and proper boundary conditions. The discretization continues at a smaller and smaller scale until a convergence is reached where smaller discretization no longer affects the flow dynamics. This process is known as a Direct Numerical Simulation (DNS). It may require many iterations before convergence is reached but a DNS approach will provide results equivalent to experimental data assuming the physics are correct.

Achieving a converged solution using DNS will require a time step and element length to be on the order of the Kolmogorov scales in Eq. (3.3) and (3.5). To capture all the dynamic motion in a system, a DNS model must have sufficient spatial and time step scales. If turbulence is present it must be resolved in the computational mesh and the time step must be small enough to capture the motion of the smallest eddies. If the time step is too large or the mesh is not refined sufficiently the model can predict laminar flow because the dissipative forces will be too strong. DNS as a method to resolve a flow field generally becomes too computationally expensive because the required magnitude of discretization is too great. This factor limits the use of DNS to low Reynolds numbers and simple geometries. Because of this dilemma, many averaging models have been developed and used for resolving flow fields such as the RANS model, LES model, RNG k-ε model, RSM model and a variety of others.[27] Each model has its pros and cons over the DNS method.[i] In modeling a Murata blower, the acoustic frequency spectrum is of great interest, and thus averaging the velocity and pressure fields to resolve the flow field is not ideal.[28]

Despite the computational expense of DNS, it is the chosen method of modeling a flow field of a Murata blower in this research. The hope is it will provide the most accurate results. In analyzing the flow of the blowers, the geometry is relatively simple and the Reynolds number is relatively low. Maximum Reynolds numbers of 1000 have been measured having laminar flows in experimental data of axisymmetric jets.[29] As the flow begins to exceed 1000 the flow transitions into turbulent jet flow. The jet of the Murata blower is assumed axisymmetric. By letting L and U in Reynolds Eq. (3.1) be the nozzle diameter[j] and exit velocity[k] of a blower, a

---

[i] For more knowledge on turbulence models refer to D.C. Wilcox, *Turbulence Modeling for CFD,* Third Edition
[j] Nozzle diameter on the Murata blowers is 0.82 mm
[k] Exit velocity was measured using PIV techniques to be 28 m/s

Reynolds number of 1530 can be estimated. Thus the air at the nozzle exit is just past the transition stage of turbulent flow. Furthermore, the Kolmogorov scales of the blower require time and length scales that are reasonable and achievable with the supercomputing facilities at BYU. These scales and calculations are shown in Section 3.4 of this chapter. In summary, the Murata blower has a relatively simple geometry and low Reynolds number, therefore a DNS approach to modeling the flow field is a valid and viable method.

3.2   Model Build Procedure (ADINA)

This section will provide a step by step path for building and solving a DNS model of a piezoelectric blower in ADINA. The software acronym stands for **A**utomatic **D**ynamic **I**ncremental **N**onlinear **A**nalysis. ADINA was initially chosen for this research because of its capacity to model Fluid and Structure Interactions (FSI). At the onset of this research, it was thought that any blower model would need to incorporate the structural interaction of the blower with the fluid flow to accurately predict a sound pressure level. As Chapter 2 demonstrated, such is the case. However, as modeling progressed, FSI components of the model were left out to reduce computation times. When included, the computation time became a barrier to the research. The model in ADINA only accounts for fluid flow. This is not ideal, but still gives good insight into noise generation caused by the air flow.

A need in the model design is variable parameters. One key to the success of the model will be defining parameters that can self adapt when changed by the user. This will facilitate the larger goal of discovering which components in the flow path affect the sound pressure levels the most. The following sections describe the model's geometry, boundary conditions, loading, solving, and data extraction method.

3.2.1 Constructing Geometry (ADINA)

This section steps through requirements for building and meshing the geometry and outlines the geometrical assumptions for modeling a Murata blower. The flow path of air as it passes through a blower and out the nozzle can be assumed axisymmetric about the nozzle axis. Figure 3-1 shows a two-dimensional cross sectional plane that passes through the nozzle axis of the blower. The blower walls on the cross sectional plane are shaded with blue.
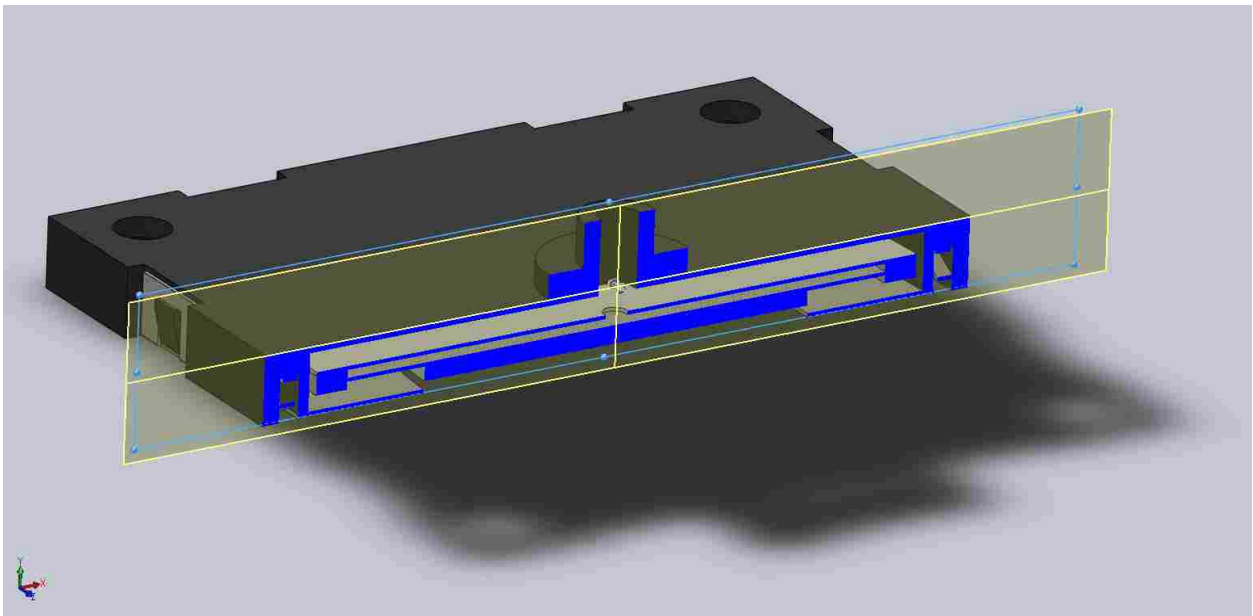
Figure 3-1: Cross sectional plane passing through the nozzle axis.

Figure 3-2 displays the normal view of the cross section. This view is the basis of the geometry used in the model and serves as a benchmark for future results of the model.
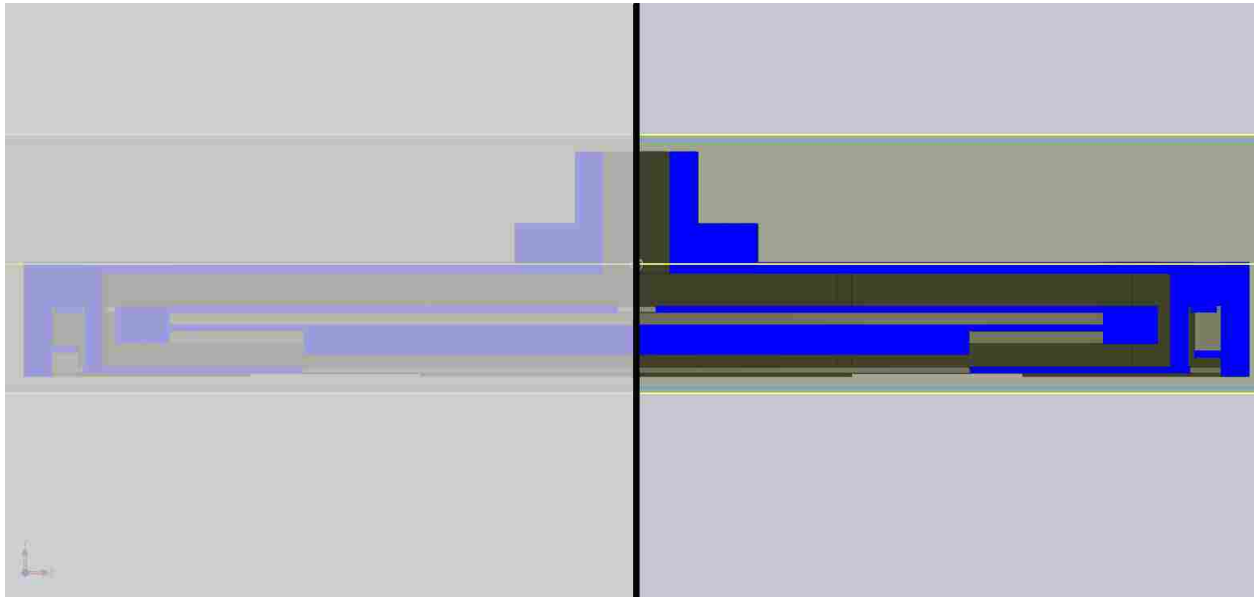
Figure 3-2: Normal view of cross section, basis of model geometry.

Due to symmetry, only one half of the cross section needs to be physically modeled. The flow path on the right side of Figure 3-2 can be drawn, meshed and revolved about the nozzle axis. The walls in the flow path are assumed fixed and the free space between them is modeled and meshed. It is also necessary to model the air space above the nozzle exit as well. This area is where a jet profile develops. Figure 3-3 shows the meshed representation of the flow path and the area above the nozzle. The mesh is more refined in the areas of more volatility or fluid activity. The mesh is less refined in areas of less fluid activity. The actual mesh for the model is more refined than what is shown in the figure.
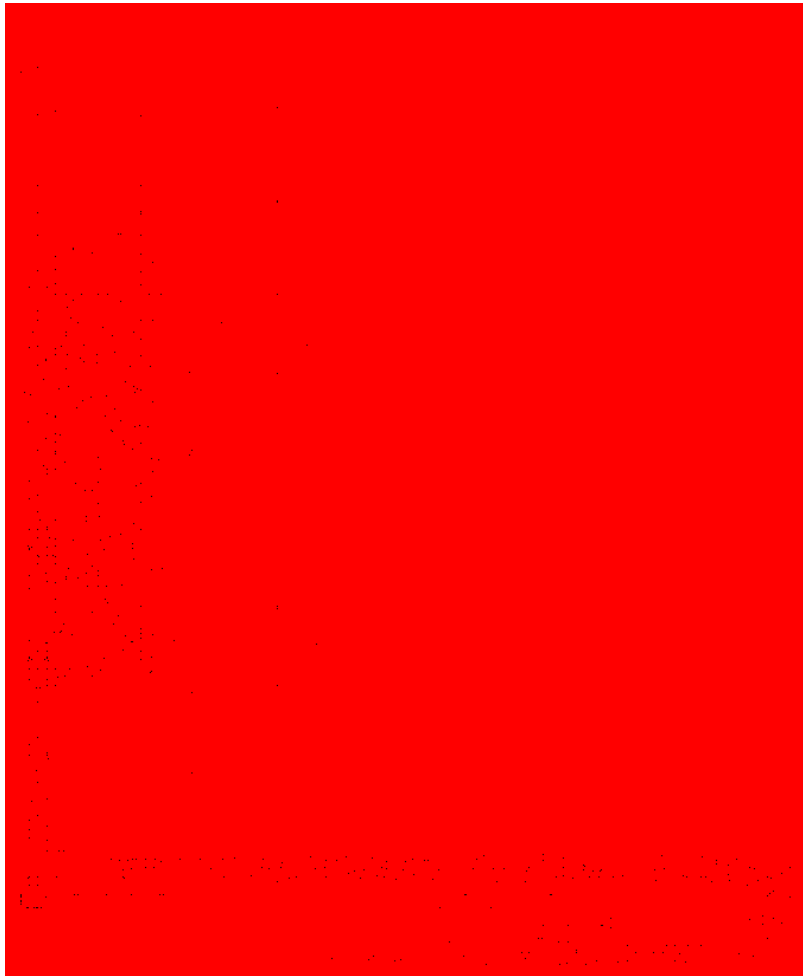
Figure 3-3: Course mesh of the air flow path.

The following steps outline how to build this mesh and geometry and set the correct fluid properties for an accurate DNS solution to occur:

A. Start ADINA from a Unix command window using the prompt: adina

B. Choose "ADINA CFD" from the Program Module drop-down list.

C. Choose "Transient" in the Analysis Type drop-down list.

D. Choose "No FSI" in the FSI/No FSI drop-down list.

E. Choose "Slightly Compressible" in the Flow Assumptions drop-down list. Figure 3-4 shows what the AUI icons bar should look like at this point. Refer to this figure as a guide for the icons that are used in the next steps.
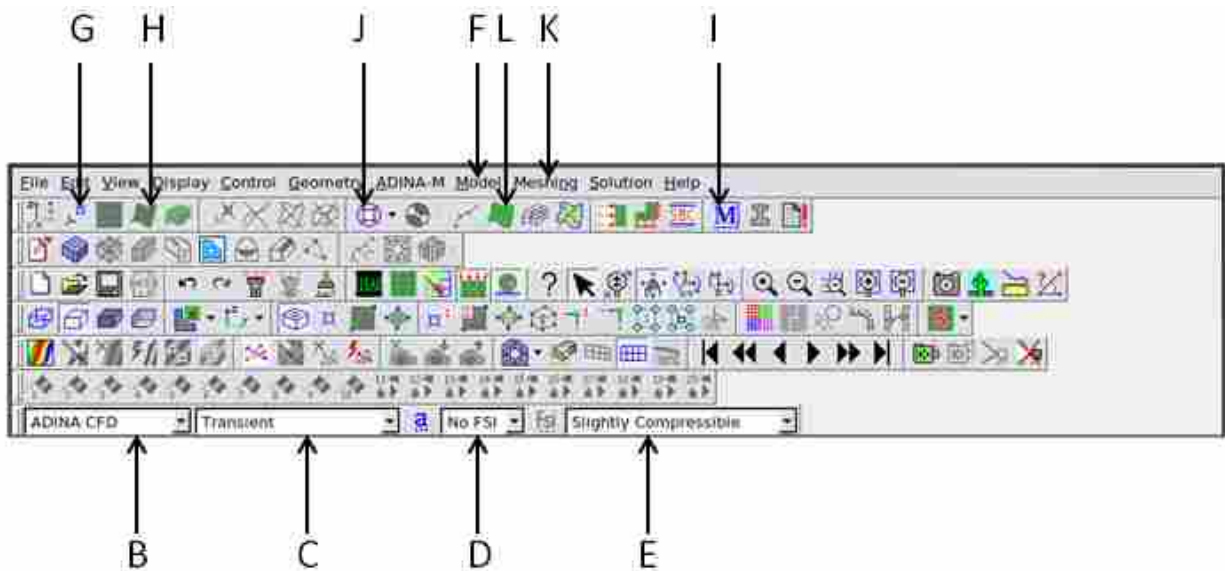
Figure 3-4: Labeled ADINA AUI Icon bar and drop-down lists.

F. Set the flow assumptions by choosing Model → Flow Assumptions, set the Flow Dimension to 2D (in YZ Plane), uncheck the "Include Heat Transfer" button, set the "Flow Type" to Slightly Compressible, select Laminar in the Flow Model drop-down list and click OK. Use the window in Figure 3-5 to verify this step.

G. Next click the define points Icon and enter the locations for each point making up the desired geometry[1]. Leave the X1 column blank. There are 65 points defined in the blower model. An example of how this window should look is given in Figure 3-5.

---

[1] This process is made easier by hand drawing and labeling each point on a sheet of graph paper for future reference. This also makes future steps of defining surfaces and applying load and boundaries easier.

H. Now use the points defined in step G to define new surfaces. This automatically builds corresponding lines. Click the define surfaces Icon and begin defining groups of 4 points as a surface as seen in Figure 3-6. Select Vertex in the Type drop-down list, enter the 4 points that will define the surface, click Add and repeat for each surface required. Click OK. There are 34 surfaces defined in the blower model.

I. Next define the material type. Click the Manage Materials Icon and click the Constant button. In the Define Material with Constant Properties dialog box, add material 1, set the viscosity, density, and bulk modulus, click OK. See Figure 3-6.

J. An element group must be defined at this point. Click the Define Element Groups icon. Add element group number 1, select Axisymmetric for the Element Sub-type, and choose 2-D Fluid in the Type drop-down list. The correct selections are displayed in Figure 3-7.

K. Click the Subdivide Surfaces icon in the Meshing → Surfaces menu and set a uniform division length for all surfaces of 25E-6 m. Later parameters like this one can be made into user defined input values that can be changed before solving the model.

L. Finally, click the Mesh Surfaces icon. Select Free-Form as the Meshing Type and set Nodes per element to 4. Mesh all surfaces by entering the surface numbers in the green rows seen in Figure 3-7. Click Apply. Now the model should display geometry similar to what is seen in Figure 3-3.
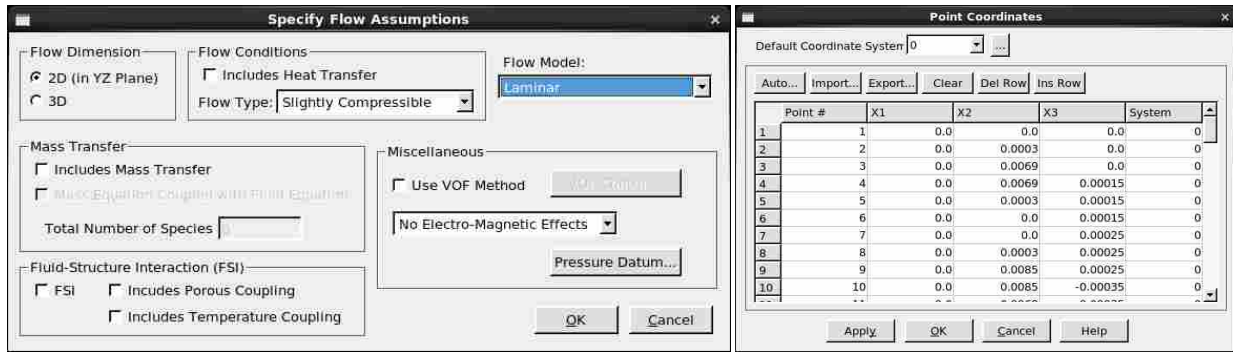
Figure 3-5: Flow assumptions and define coordinates windows.
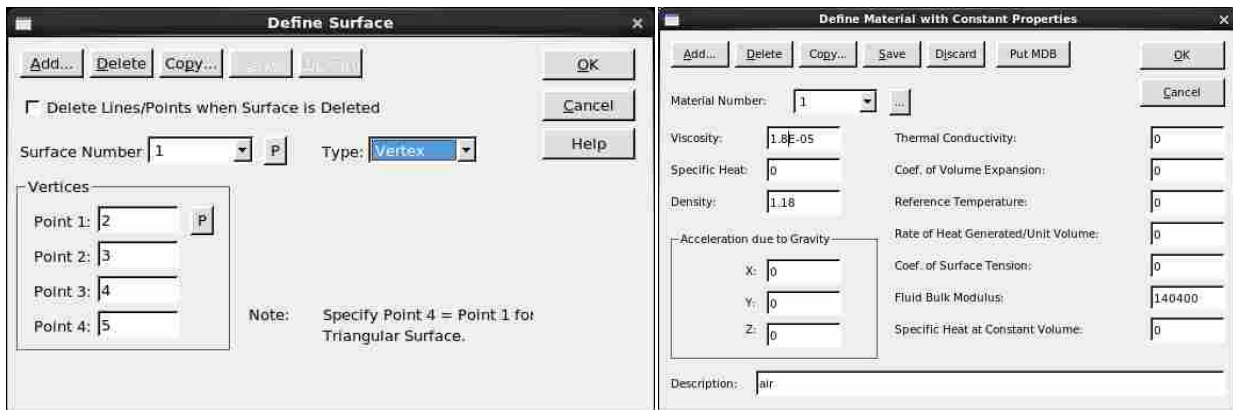
Figure 3-6: Define surfaces and define material with constant properties windows.
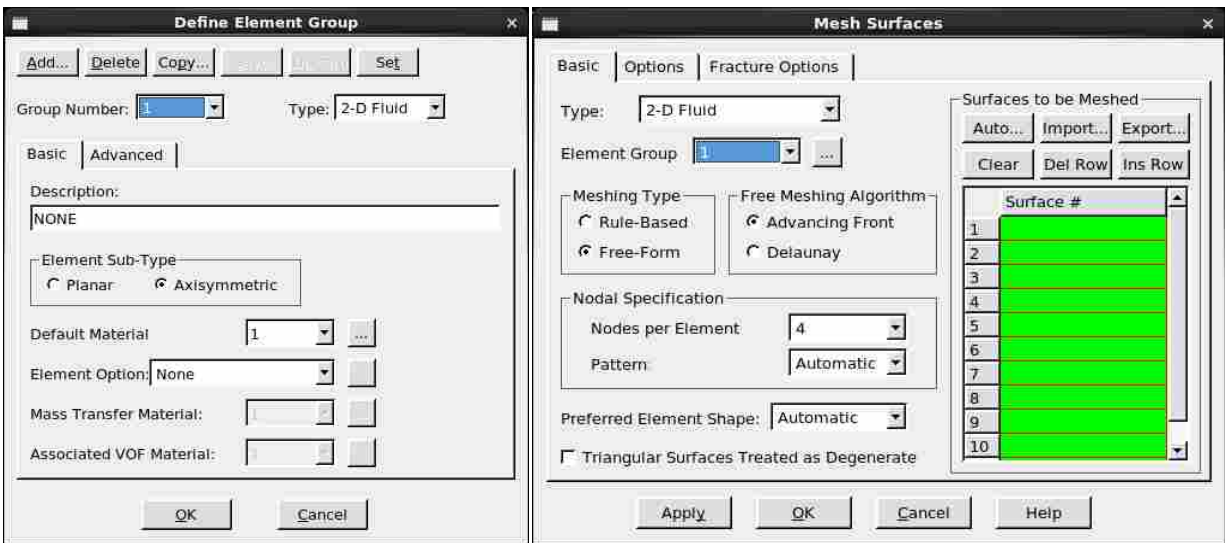
Figure 3-7: Define element group and mesh surfaces windows.

3.2.2 Boundary Conditions and Loading

Defining the boundary conditions and load is presented in this section. Up to this point only the mesh has been generated. The diagram in Figure 3-8 demonstrates how boundary conditions and a load are applied to the blower model. The gray space in the figure represents the blower body. Boundaries are not needed for the entire body space since some areas are not in contact with the meshed flow path. The black lines are defined as no-slip walls everywhere the flow path makes contact with the blower body. The green line is defined as a rotational axis of symmetry. Finally, a dynamic load is applied in the red area. This space is where the piezoelectric ceramic is situated in the Murata blower. It vertically oscillates. In the model, the ceramic is represented by negative space and a sinusoidal velocity load is applied to the outside walls of the ceramic that are in contact with, and therefore influencing, the air flow path.
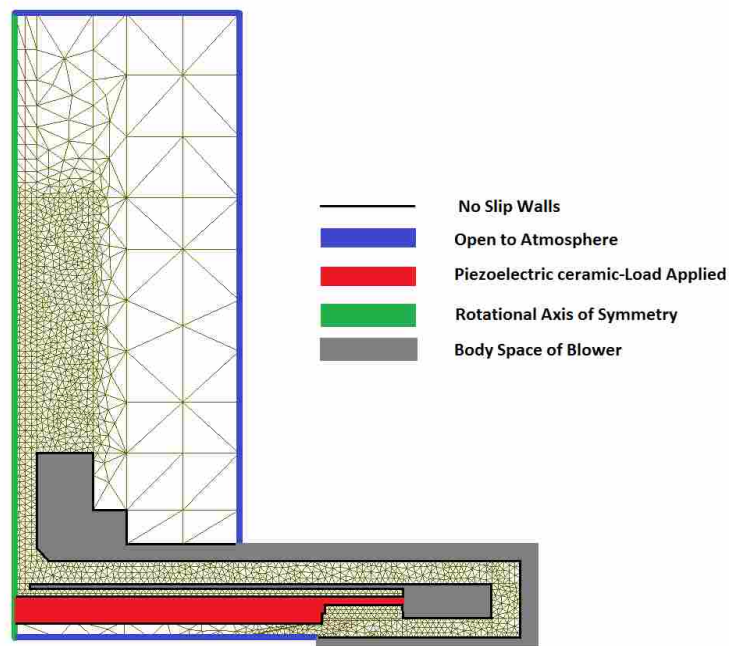
Figure 3-8: Diagram of model boundary conditions and applied loading.

43

The process of applying both the boundary conditions and the dynamic load is outlined below. For more detailed information on how the load is approximated refer to Appendix A.6. Refer to Figure 3-9 for the icon locations mentioned in the following steps:
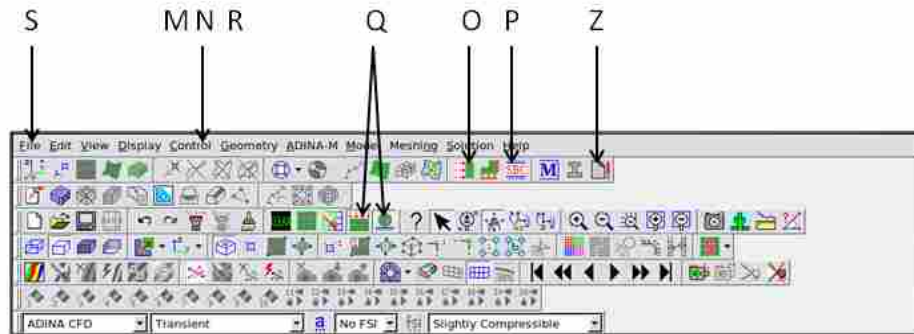
Figure 3-9: Labeled ADINA AUI Icon bar for bounding a model and applying a load.

M. The time step and an estimated time function need defining before the loading can be applied. The time step value and function length will later be made user defined input parameters. For now a generic step and function can be used. Choose Control → Time Step, edit the first row of the table to be the number of steps and stepping value respectively and click OK.

N. Now choose Control → Time Function, add Time Function Number 1 and import or fill in an oscillating time function in the Time and Value table provided. Figure 3-11 gives an example of both the Time Step window and the Time Function window. Select the Graph button to verify the correct function is generated. Click OK twice.

O. To apply the load, identify the line numbers where the load is to be applied by selecting the actual lines in the model. Click the Apply Load icon. Define a new load by choosing Velocity as the Load Type and Line in the Apply To drop-down list. Enter the line numbers where the load is to be applied in the Site # column. Make

sure the Time function is set to 1 for each line entered in the table. Figure 3-11 shows the blower model's Load 1 window. Click Apply.

P. Finally the boundary conditions can be set. Identify which line numbers in the model will need no-slip walls and which will be set as the rotational axis of symmetry by selecting the individual lines in the model and noting the corresponding line number. Select the Define Special Boundary Conditions icon. Add Condition 1, set the Type to Wall, choose Lines in the Apply To drop-down list, select the No-slip condition, and input the line numbers that will need the no-slip wall condition in the Apply to Following Entities table.   Next, add condition 2. This condition is the same as the first but the Slip condition is set to Yes. This boundary condition serves as the rotational axis of symmetry. Input the line numbers as needed in the table and click OK. An example of condition 2 is shown in Figure 3-11.

Q. Finally, click the Show load and Show Boundaries icons on the AUI tool bar to verify the loads and boundary conditions are as desired. The window should appear similar to what is seen in Figure 3-12.
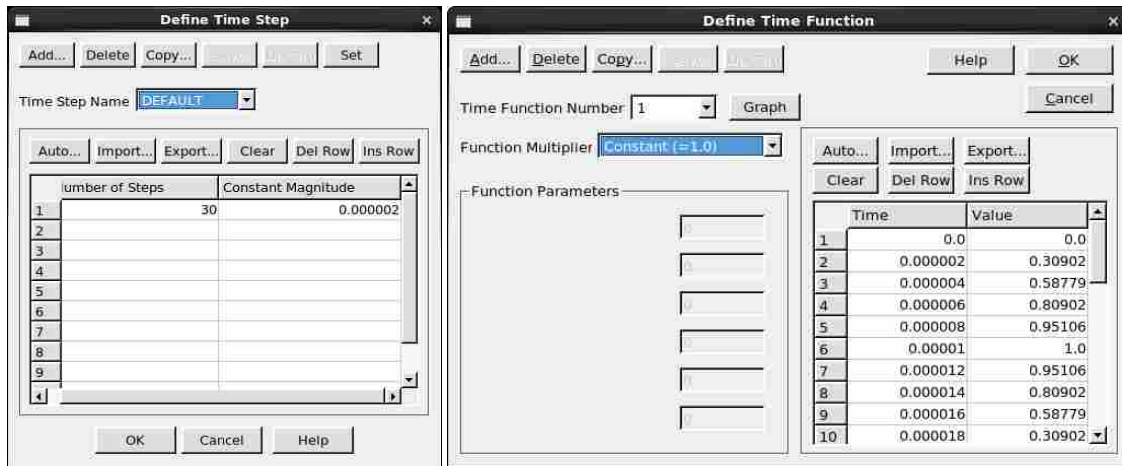
Figure 3-10: Define time step and define time function windows.

Figure 3-11: Apply load window and define special boundary conditions window.

46
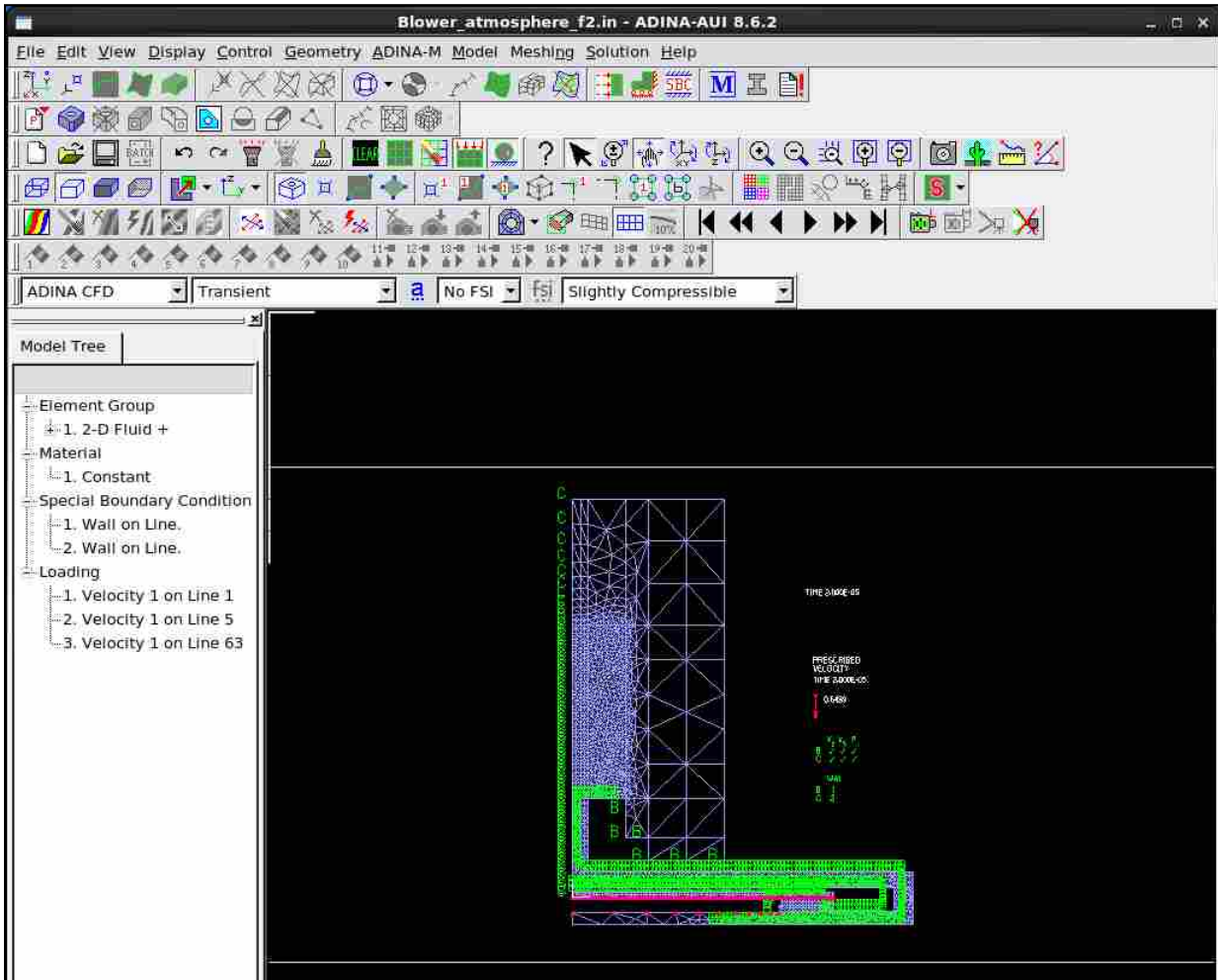
Figure 3-12: Meshed model showing applied boundary conditions and loading.

3.2.3   Parameter Inputs and Solving Process

Resolving the model and defining input parameters is outlined in this section. One of the goals of this research is the development of a model that includes parameters that can quickly and easily be changed by the user without negatively affecting the rest of the model. If the model above is left as is, changing one dimension in the model requires making modifications throughout the entire geometry. For example, if the user desires to change the size of the nozzle radius, the user will have to change the point located at the base of the nozzle as well as the top of the nozzle and all the associated lines that connect to the nozzle walls. Although this can be done, errors have a higher chance of occurrence and it is a long process. It therefore becomes desirous to generate a master parameter list that dictates which dimensions are dependent on others. To continue with the example above, the parameters list could include a variable such as Nozzle Radius that when changed, also changes all associated dimensions. This method eliminates the risk of errors and simplifies changing the dimensions of model. However, if smaller and smaller dimensional scales are input, the model may become unstable. To prevent instability when micro scale inputs are used, the model must be set to analysis in Non-Dimensional Form. The steps required to generate a parameters list, along with how to solve the model, are outlined and explained below (refer to Figure 3-9 for icon location on AUI tool bar):

> R. From the menu bar choose Control → Solution Process and check the Analysis in Non-Dimensional Form box. Fill in the values as seen in Figure 3-13. This step helps the solutions process remain stable while processing large order of magnitude differences in the velocity and length scales.
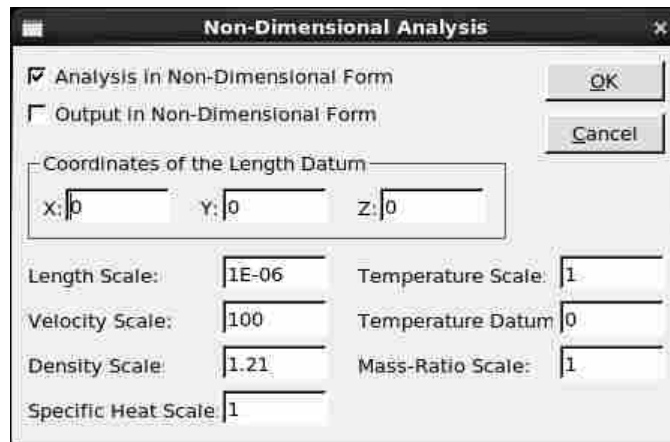
Figure 3-13: Non-Dimensional analysis values needed for the blower model.

S. Choose File → Save As from the menu. In the file extension drop-down list select the .IN file extension, name the file and save. Refer to Figure 3-14 for clarity. As the user builds and makes changes in ADINA, all the commands are saved in a text format. These text commands can be saved as a .IN file and can be viewed and edited by any basic file viewer, then re-opened in ADINA.
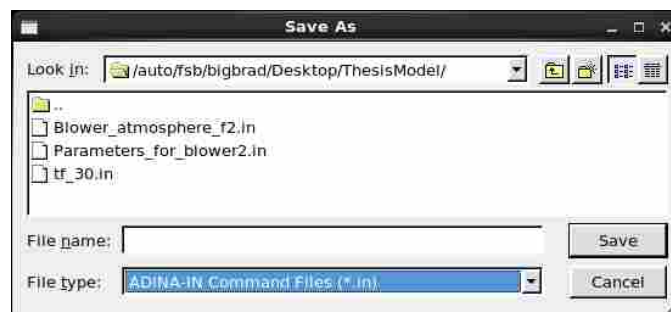
Figure 3-14: Saving the text commands as a .IN file.

T. Open the new .IN file for viewing. Every click while building the model has been recorded here. Clean the file of repeated and redundant commands and organize the file. The most recent change to any part of the file is always at the end of the file.

49

When finished the file should be simplified to the point where steps A through R are easily identifiable. See Appendix A.7 - A.9 for the .IN file used in this research.

U. Create a blank text file with a .IN file extension. This file will serve as the parameters list for the model. Insert the line "READ ParameterFileName.IN" just after the model's program declarations contained in the .IN file. See example below:

```
DATABASE NEW SAVE=NO PROMPT=NO
FEPROGRAM ADINA
CONTROL FILEVERSION=V86
FEPROGRAM PROGRAM=ADINA-F

READ ParameterFileNameHere.in
```

V. In the newly created Parameters file, declare new parameters as the example below demonstrates. Follow the given format. Notice that Var1X will always depend on the values given to Var1 and X.

```
PARAMETER Var1 '10'
PARAMETER Var2 '5'
PARAMETER X '1'
PARAMETER Var1X '$Var1+$X'
```

W. Now, the number values in the model's .IN file can be replaced with the parameter names declared in the Parameter file. This is done by using a dollar sign ($) followed by the parameter's given name. The example below defines three geometrical coordinates all in a horizontal line with point 3 only 1 unit away from point 2:

```
COORDINATES POINT SYSTEM=0
@CLEAR
1 0.00000000000000 0.00000000000000 0.00000000000000 0
2 0.00000000000000 $Var1 0.00000000000000 0
3 0.00000000000000 $Var1X 0.00000000000000 0
```

X. For simplification, the time function can be read in as separate .IN file as well. Use the following format to call a time function and time step parameter:

```
TIMEFUNCTION NAME=1
@CLEAR
READ TimeFunctionNameHere.in
@

TIMESTEP NAME=DEFAULT
@CLEAR
$NumberOfTimeStepsHere $TimeStepValueHere
@
```

Y. Place all three files in the same directory (Example: Model.in Parameters,in and Time.in). To re-open the model in a different work session, simply start ADINA and open the model's .IN file.

Z. Finally, to solve the model click the Data File/Solution icon. Verify that the Run Solutions box is checked and click Save. This will generate a .POR file for post processing. Because of the large number of elements used in this research, this step is saved to the .IN file and the actual solving is submitted to a supercomputer. The supercomputer script files are included in the Appendix A.10.

3.2.4   Post Processing Results (Ensight)

This section describes the technique used to post process and extract desired data from the blower model by running Ensight in batch mode. The computational power required to run the blower model is far beyond what standard desktop computers provide and the output files of the model are on the order of gigabytes. Thus a resolved model solution becomes increasingly hard to post process on any standard desktop machine. Many attempts were made by the author to transfer the gigabytes of data back into ADINA for post processing but none succeeded. It became necessary to post process with batch files on the supercomputer. This means the model is both resolved and post processed while running on the supercomputer. This simplifies the output to only a few hundred text files that need transferring from the supercomputer. The text

files can then be analyzed by programs such as MATLAB. This method of post processing results in only a few megabytes, compared to gigabytes, and makes the post processing systematic, uniform and much faster. Ensight is the program used to post process the data on the supercomputer. But the Ensight commands need to be converted into a more programmable language for batch file submission. Ensight uses Python as its Application Programming Interface (API). A detailed outline for converting Ensight commands into python code is not included in this thesis. However, the python batch files for all the post processing tools are included in Appendix A.11. The following are the four main tools used and designed by the author to extract the resolved velocity and pressure data from the blower model:

1. **Velocity line Tool** - each time step outputs a text file that includes a specified number of velocity values across a given line in space, giving instantaneous velocity along a linear distance.

2. **Pressure Line Tool** - similar to the Velocity Line Tool but outputs the instantaneous pressure along a linear distance.

3. **Picture Tool** – Snaps a formatted picture of color scaled velocity magnitudes at a given instant in time.

4. **Movie Tool** – Produces a video clip of the velocity magnitudes and how they change with time.

3.3   Model Verification

The purpose of this section is to provide evidence that a reliable and dynamic model has been developed that is capable of predicting the effects that geometrical dimensions have on the SPL near the nozzle exit and the flow rate of a piezoelectric blower. There are numerous ways to provide confidence in a flow model solution. Methods for doing so depend greatly on the

character of the problem and the method of solution. For a DNS solution, grid independence must be shown. The results outlined in this section show that the assumptions made in developing the geometry, load, and boundary conditions do in fact produce the sought after flow characteristics of a real blower. Convergence of the model is demonstrated. The required elemental length and time step to achieve grid independence are given and validated. The predicted velocity profile from the model is compared to that of a Murata blower for reference. The acoustic frequency spectra of both the model and a Murata blower are also compared for reference.

3.3.1   Jet Formation

An overview of the resulting flow dynamics of the model is presented in this section. The most difficult part of modeling the Murata blower was making the correct assumptions that would result in the formation of a stream of air. Many builds of the model did not produce a stream of air. The model assumptions were reconsidered until a stream of air was indeed predicted by the model. One assumption is the loading of the ceramic. A sinusoidal velocity load on the walls of the piezoelectric ceramic and pumping chamber is assumed. A correct loading assumption should result in air sucked through the back side of the blower, pushed around the pumping chamber, and finally ejected it out the nozzle. That is the flow path for a Murata blower and a similar flow dynamic was sought after for this DNS model.

Numerous particle trace lines in the flow path of the model are shown in Figure 3-15. They are tracing the spatial path of a particle while the model is running. Notice all the particles travel towards the nozzle exit as the model runs. The particle trace lines demonstrate how air is pulled in through the back side, pushed around the pumping chamber, and ejected out the nozzle. This result demonstrates how the flow dynamics of the Murata blower are in fact being

53

mimicked by the blower model. The assumptions made in the model result in the formation of an air stream out the nozzle. Specific characteristics of the resulting air stream are discussed later in the chapter.
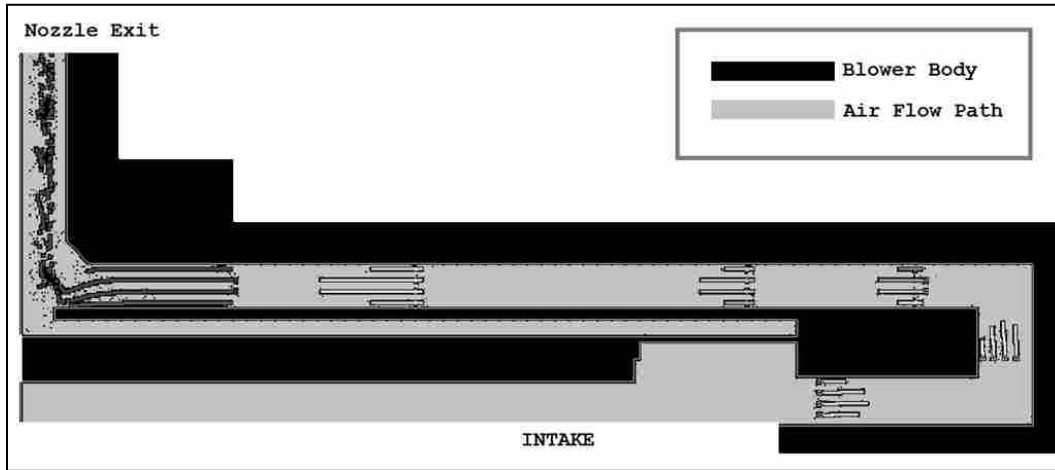
Figure 3-15: Particle trace lines from blower model with intake and nozzle labeled.

3.3.2   Model Convergence

A key component in CFD modeling is model convergence. The required transient time, elemental length, and time step for model convergence are presented in this section. Convergence plots are presented with corresponding elemental lengths. Also, the required Kolmogorov scales are computed and validated for use in the model.

The model prediction for the change in velocity over time at a single point 4.0 mm above the nozzle exit is plotted in Figure 3-16. A transient portion is noted upon start up and the velocity soon reaches a steady state. This steady state flow is a direct current of air with a velocity of 21 m/s. The steady state flow begins to form around 1 ms after start up. There is a slight shift in the transient time depending on the location of the point of interest. For locations closer to the nozzle, the transient time is obviously slightly shorter because there is less required

54

distance of travel. The opposite is true for a measurement farther from the nozzle exit. The delay time for the air to start moving at 4.0 mm above the nozzle can be observed in the very beginning of the plot near 0 ms. The time required to achieve steady state flow in the model proved independent of time step or element length. Therefore, the model run time was generally specified to be 1.5 ms to ensure steady state flow in the solution and to provide enough time data in the steady state region to calculate a frequency spectrum from 1 to 20 kHz.
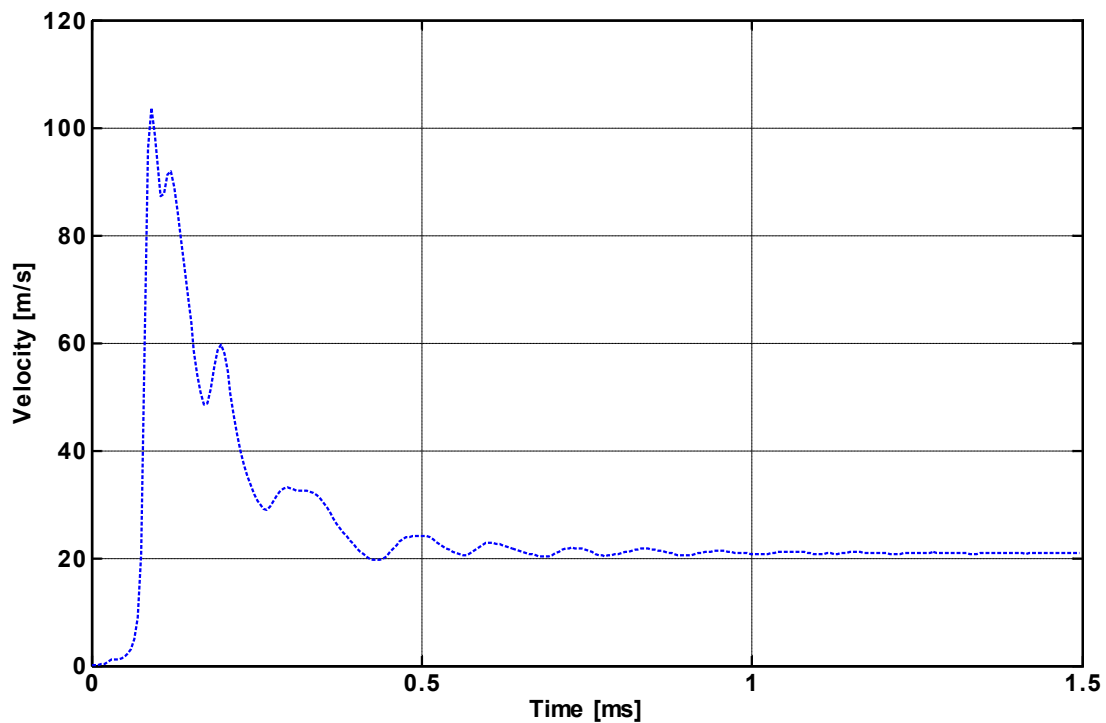
Figure 3-16: Velocity as a function of time 4.0mm above nozzle exit on jet axis.

To achieve grid independence, the elemental length of the mesh was discretized until it no longer had a major effect on the flow parameters of interest. An elemental length and/or time step that cannot capture the desired scales of motion will result in a more laminar flow because the dissipative forces in the equations of motion will be too large. The model was resolved multiple times with smaller and smaller element lengths. The velocity profile at the nozzle exit

for three different element lengths are plotted in Figure 3-17.   The plot clearly demonstrates how

a course element length dissipates the velocity profile of the air stream.
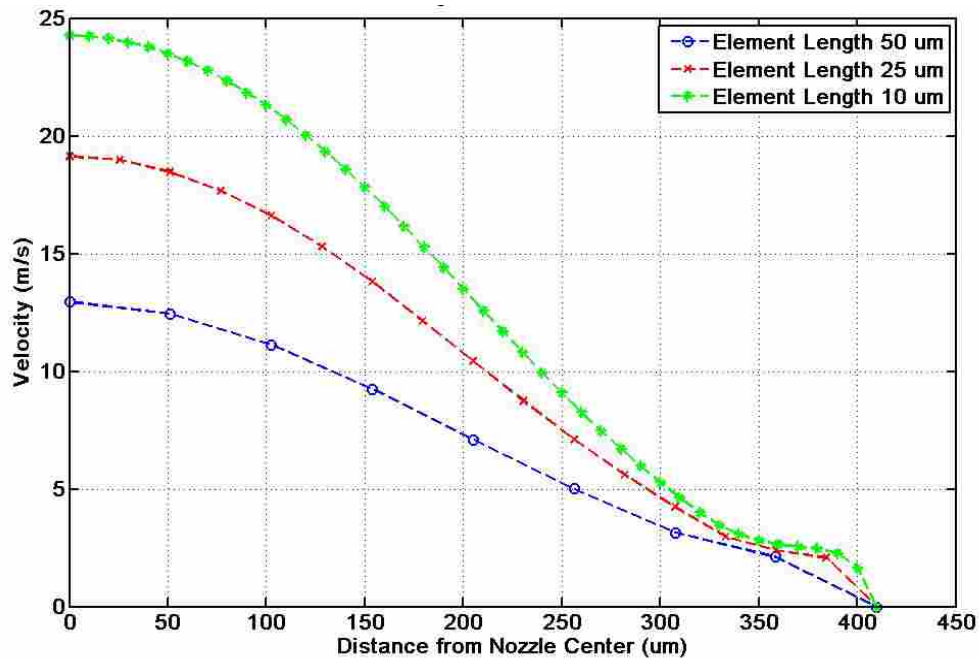
Figure 3-17: Velocity profile change with element length.

The process of refining the element length was continued until convergence was reached.

Table 3-1 shows the percentage change in velocity amplitude as the grid size continued to shrink.

The data in the table were taken from the transient portion[m] of the solution because processing

time became too great beyond 2 μm.   The results show only a 0.32% change in velocity

amplitude when solving with a 5 μm and a 2 μm element length. The differences in the steady

state region would be much less than in the transient region. Thus, with a 5 μm element length,

the model convergences and becomes grid independent.

---

[m] Velocity amplitude values were taken from the model at 0.12 ms along the jet axis at the nozzle exit.

Table 3-1: Grid independence validation chart.

Element Length (μm)	50	25	10	5	2
Velocity Amplitude (m/s)	40.47	62	68.46	72.61	72.38
% Change $\dfrac{(Velocity - PreviousVelocity)}{PreviousVelocity}$		53.20%	10.42%	6.06%	0.32%

The time step used in the model was 1µs. This value was based on the application of the Kolmogorov scales derived in Chapter 2. Both the elemental length scale and time scale can be compared against the scales. Solving Eq. (3.3) and Eq. (3.5) for η and τ yields 17µm and 21µs respectively. This suggests that theoretically the elemental length should be on the order of η and the time step should be on the order of τ to achieve grid independence and model convergence. The elemental length of 5µm used in the model is less than the theoretical 17 µm suggested by the Kolmogorov scale equation. Likewise, the 1 µs time step used by the model is less than the theoretical time scale of 21 µs. Both of these calculations provide further evidence that using DNS to solve the blower model is valid.

The formation of the air stream, the grid independence, and the Kolmogorov scale equations all provide confidence in the solution of the model. To further provide confidence in the model solution, an attempt to compare velocity profiles and frequency spectra between a Murata blower and the model is given in the next two sections.

3.3.3 Velocity Profile Reference

This section discusses the process of comparing the velocity profile predicted by the DNS model to the measured profile of an actual Murata blower. A CFD model is generally an attempt to model a real situation with hopes of using the model to better understand the system in

question. A model is generally idealized and simplified so a solution can be found in a feasible amount of time. However, it should not be simplified to the point where the model no longer is affective at representing its system. For the DNS model, the design and assumptions are based on a Murata blower. However, to be an effective tool, the model does not need to perfectly reproduce every kinematic motion of a Murata blower. Rather it needs to be a good estimate or indicator. The purpose of the blower model is to better understand how changing the geometry of components in the flow path will affect flow rates and sound pressure levels for piezoelectric blowers of similar design. A comparison to a Murata blower is provided in this section as a reference. The comparison should provide more confidence in the ability of the model to predict trends in sound pressure levels and flow rates.

Jet profile theory is a good first step of analyzing the model solution. Table 3-2 defines equations related to jet profiles that will be used to validate the model solution. These formulas are used to predict the velocity profiles for both a laminar and turbulent axisymmetric jet.[29] The DNS model's velocity profile can be compared to the theoretical profile of a jet. By substituting the centerline velocity predicted by the model u_m at a given location x along the jet axis into Eq. (3.9) and Eq. (3.13), the theoretical velocity profile u for a laminar and turbulent jet can calculated respectively. The jet profile equations indicate that the centerline velocity in a jet depends mostly on the nozzle exit velocity and the kinematic viscosity of the fluid. But by using a given centerline velocity from the model at a specific distance from the exit plane, the profile will depend mostly on what that specified distance is and whether the jet is laminar or turbulent.

Table 3-2: Theoretical equations for jet velocity profiles

Laminar Jets (Max Reynolds Number for laminar flow ≈ 1000)		
Center Line Velocity	$u_m = \dfrac{3}{8\pi}\dfrac{M_o}{\nu x}$	(3.8)
Velocity Profile	$u = u_m \dfrac{1}{\left(1 + \frac{1}{4}\xi^2\right)^2}$	(3.9)
Momentum Flux	$M_o = 2\pi \displaystyle\int_0^\infty u^2 r\, dr = constant$	(3.10)
Non-dimensional frame Coordinate	$\xi = \left(\dfrac{3}{16\pi}\right)^{1/2} \dfrac{M_o^{1/2}}{\nu}\dfrac{r}{x}$	(3.11)
Turbulent Jets		
▢Center Line Velocity	$u_m = 12\dfrac{r_o}{x}U_o$	(3.12)
Velocity Profile	$u = u_m e^{-94(r/x)^2}$	(3.13)
x = coordinate from origin of jet, in line with flow direction r = radius from centerline of jet axis r_o = radius of nozzle U_o = nozzle exit velocity ν = kinematic viscosity		

A theoretical laminar and turbulent jet profile are computed by using the centerline velocity of the model at a distance of $x = 2.0$ mm above the nozzle exit. The profiles are plotted[n] for comparison in Figure 3-18.

---

[n] Refer to Appendix A.12 to see the code used to calculate the velocity profile
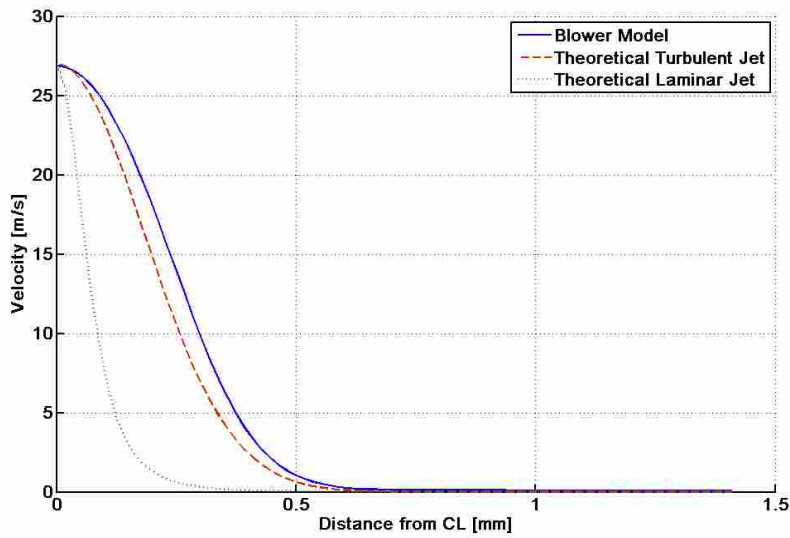
Figure 3-18: Velocity profile of the blower model compared to theoretical profiles.

Notice how a turbulent jet has a broader profile than a laminar jet when given the same centerline velocity. The model however, predicts a slightly broader profile than what is expected from a turbulent jet when given the same centerline velocity. The percent error between the model profile and the turbulent jet profile is plotted in Figure 3-19.
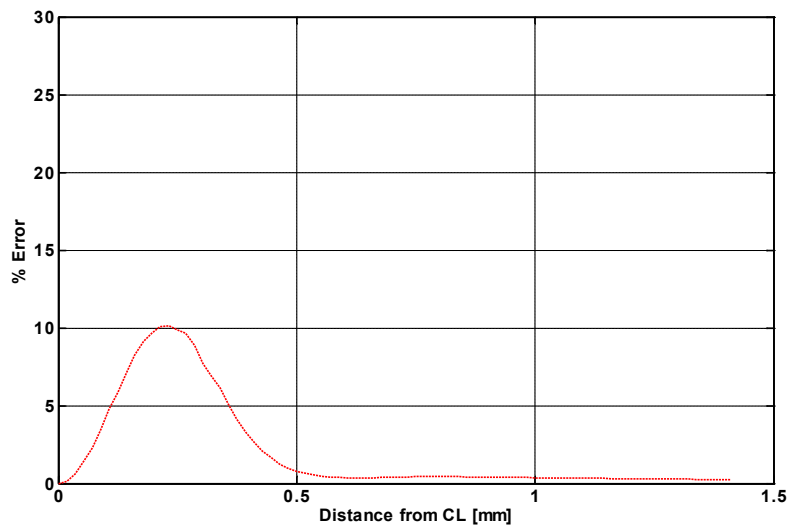
Figure 3-19: Percent error between the model's jet profile and theory.

The percent error between the two profiles is less than 10% along the entire radial axis of the jet. One hypothesis for the broader profile might be the disturbances near the source of the jet. The pumping chamber is directly below the nozzle exit and contains a small hole. Air rapidly passes in and out of this hole and can be influencing the jet stream above. This disturbance could be one reason for the deviation from the theoretical profile. When looking at a Murata blower, an even broader jet profile was measured. Many attempts to experimentally measure the velocity profile of a real blower have been made. But the small size of the blower and corresponding jet stream make obtaining consistent results difficult. The most reliable results came from a Particle Image Velocimetry (PIV) test.[o] The profile of a Murata blower was measured with PIV and is compared to the model result in Figure 3-20.
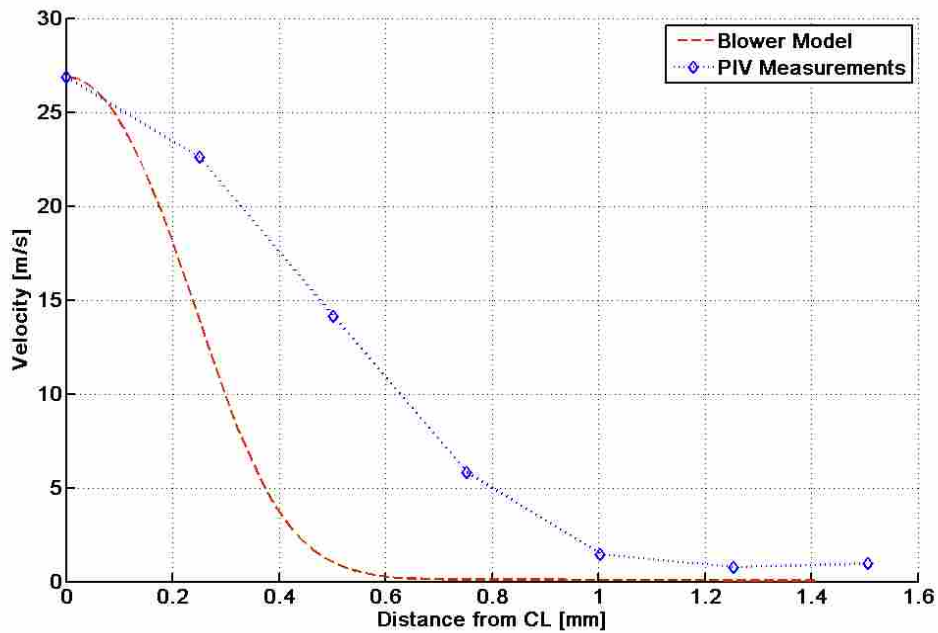
Figure 3-20: Velocity profile of model and PIV test compared at x = 2.0mm.

---

[o] Refer to Appendix A.14 for details of the PIV testing

The markers in the measured profile indicate the resolution of the PIV test. The centerline velocity of the model has less than a 1% difference from the measured center line velocity of the Murata blower. But the profile of the blower is notably wider than what the DNS model predicts. However, matching the profile of a Murata blower is not a necessary function of the model. To determine which components in the flow path influence the flow rate and SPL only relative changes in these values are needed. Benchmark dimensions in the model can be defined and resulting changes in flow rate and SPL can be computed as those dimensions are changed. Since the predicted profile matches a theoretical jet profile within 10%, the model can be used as a tool to identify which components in the flow path impact the flow rate. However, it should be noted that the measured profile of the Murata blower did not resemble an ideal jet profile. The base of the profile was much wider as Figure 3-20 shows.

3.3.4   Frequency Spectrum Reference

In this section, a reference frequency spectrum from a Murata blower is compared to the solution of the DNS model. A frequency spectrum that was measured at 1.5 mm from the nozzle axis and computed in Chapter 2 will be used as the reference data for the comparison. The plots from those measurements[p] are repeated below in Figure 3-21. Recall that the blower was baffled during testing to limit the measurement to the noise exiting from the nozzle area.

---

[p] Refer to Figure 2-15 to see the location of the measurement with respect to the nozzle exit plane.
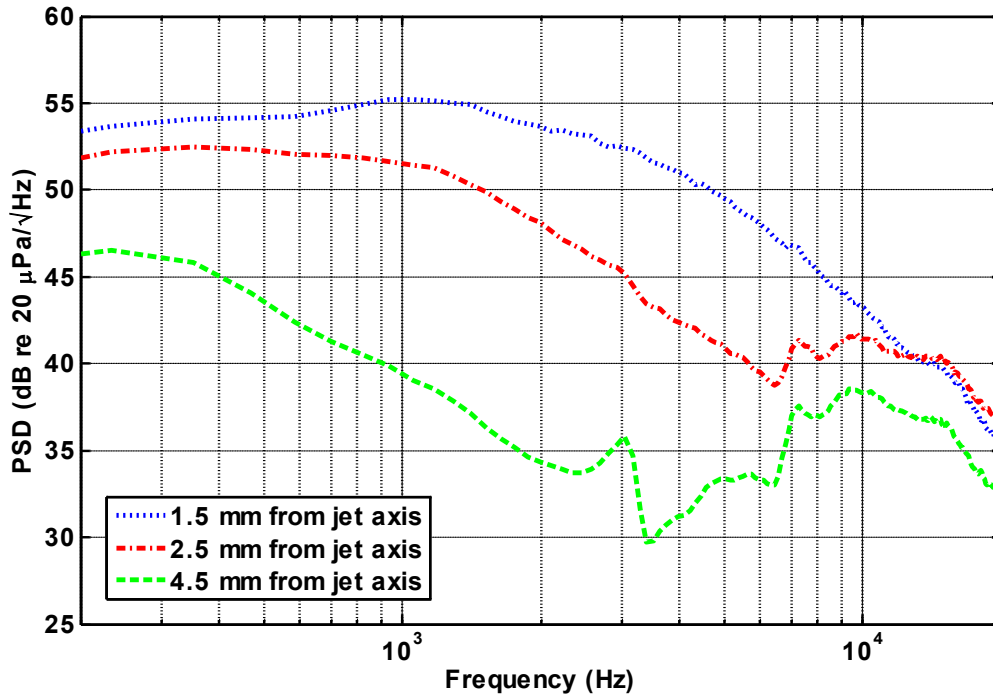
Figure 3-21: Frequency spectra of a blower's air stream.

The DNS model can predict a frequency spectrum that generally falls within the standard deviation of the 1.5 mm measurement in Figure 3-21 . A virtual microphone location is set in the model to capture the changes in acoustic pressure over time and the resulting frequency spectrum is computed. The method used to compare the measured Murata blower spectrums and the DNS model spectrum is described in the remainder of this section.

It is important to understand how the measured data from Figure 3-21 are computed so a proper comparison to the model can be made. When pressure waves are sampled by a microphone, a discrete pressure-vs.-time signal is created. The time domain of the signal can be converted into the frequency domain using a Fourier Transform. To begin this process, the signal is divided into N number of blocks with each block containing ns number of samples per block. The frequency spectrum for each block is individually calculated and then an average is

taken. Thus, the final result is an average spectrum of N time signals each containing ns samples. Knowing that the measured frequency data is in reality an average, leads to the assumption that a standard deviation from that average must exist. The model result could be validated by predicting a solution that falls within the bounds of the standard deviation of the measurements.

In computing the frequency spectrum from the model, it cannot be run past 2.0 ms due to computational time. Thus, the time signal of the model is only long enough for one average to be taken. Meaning, $N = 1$ and ns is the entire length of the steady state portion of the time signal. A good comparison between a Murata blower and the model solution would be to compare the single average of model to the standard deviation of the many averages from the measurements.

Though the comparison seems straight forward, it is physically complicated. The blower model can only predict the pressure values in a location that is meshed. Extending the mesh requires greater computational time. So the model itself limits the solvable area to no more than a few millimeters from the nozzle area. So any measured data for comparison needs to be sampled in this small confined area in real space. This is a difficult task. The small nozzle size, when compared to the microphone, makes precise locations difficult to achieve. The blue spectrum measured at 1.5 mm in Figure 3-21 gives a good reference for comparing the model solution to. Figure 3-22 shows the velocity amplitudes in the model and gives the location where the changes in acoustic pressure were sampled.
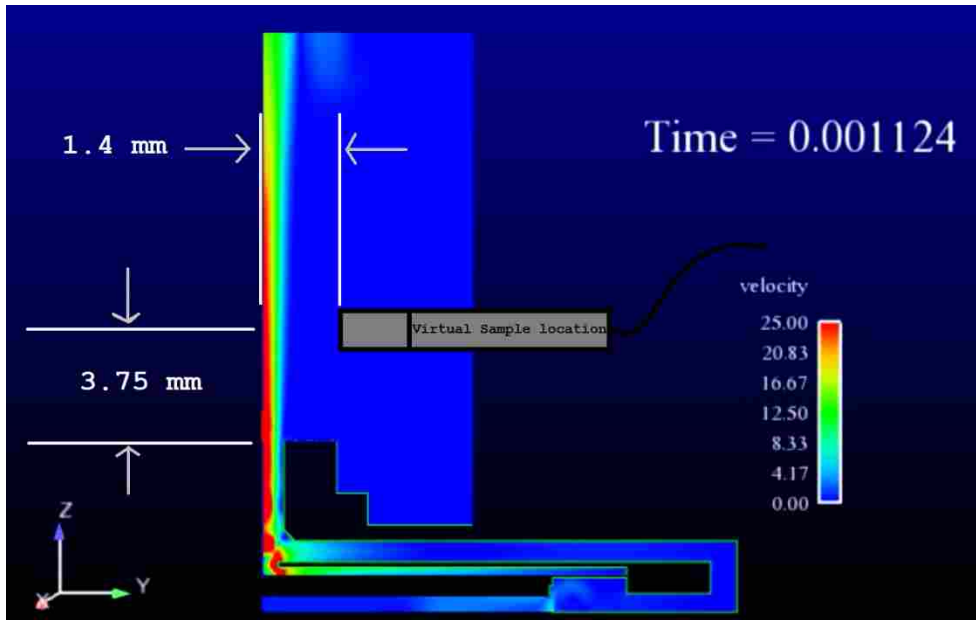
Figure 3-22: Location where acoustic pressure changes are sampled in the model.

Figure 3-23 plots the resulting pressure vs. time signal. The red portion of the signal represents the transient region and is not included in the signal processing. The sinusoidal driving frequency of 25.39 kHz stands out.
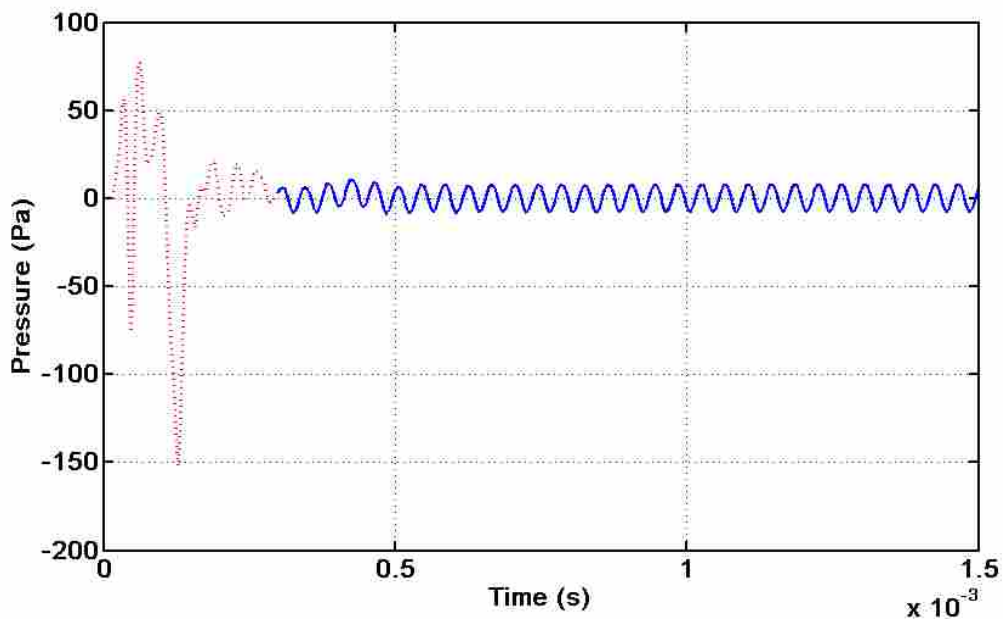
Figure 3-23: Time signal for acoustic pressure changes in blower model.

The resulting frequency spectrum of the signal can be plotted and compared to the measurement from Figure 3-21. The standard deviation of the measurement can also be calculated and plotted. The formula used for computing the standard deviation σ is defined by Eq. (3.14) where x represents frequency and N is the number of blocks in the time signal. Figure 3-24 compares the model result to the measured result.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad (3.14)$$
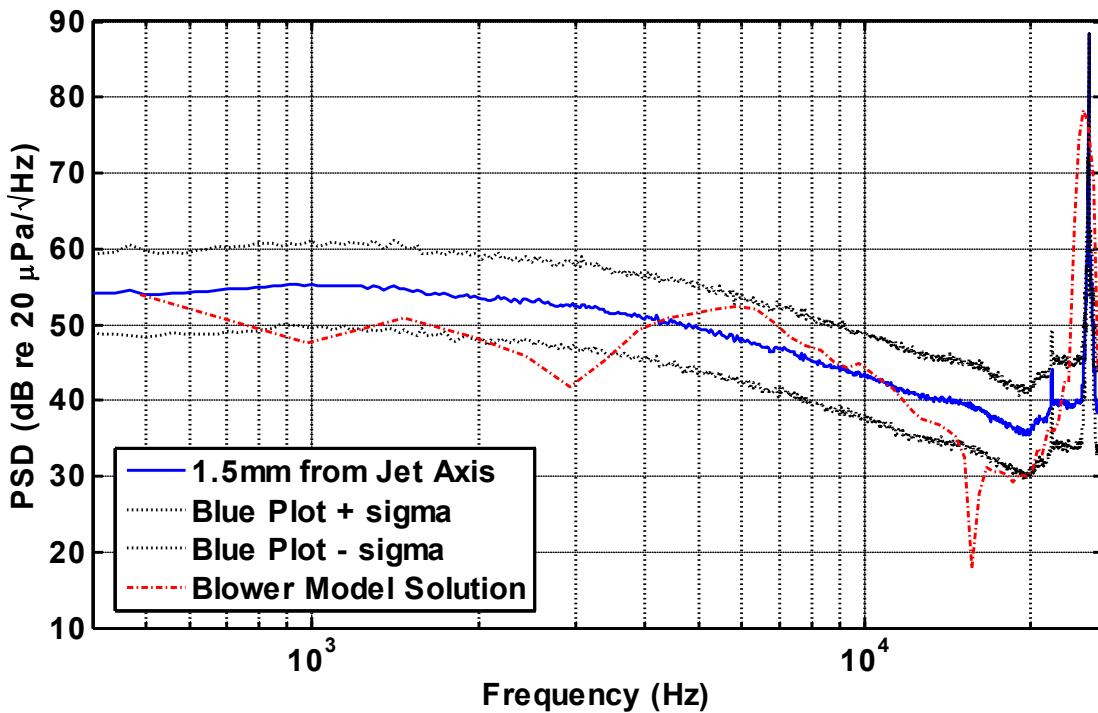
Figure 3-24: Frequency spectrum of model compared to experimental data.

The DNS model solution generally follows the trend of the measured data. The 25.39 kHz driving frequency is clearly seen in both the model and measured spectrums. The majority

of the model solution falls within the standard deviation of the measurement. However, two noticeable exceptions that fall outside the standard deviation are 2.93 kHz and 14.4 kHz. These are both frequencies where operational modes exist as discussed in Chapter 2.

The overall sound pressure level of the model spectrum can be computed[q] and relative changes in SPL can be observed as dimensional changes in the flow path are made. This comparison shows that the model does not perfectly resolve the frequency spectrum of a Murata blower. However, the relative changes to SPL are the important values and can be computed as changes in the flow path are made. This makes the model useful for identifying any critical components or parameters that contribute to jet noise of a Murata blower.

3.4   Model Benefits and Limitations

This section describes the benefit of the DNS model and how it can be used as a tool. The limitations of the model are also important to understand and are detailed here as well.

Benefits: The purpose and benefit of building a DNS blower model was to determine the impact geometrical parameters have on the SPL near the nozzle region and the flow rate in a piezoelectric blower. The model incorporates variable inputs for the nozzle radius, nozzle height, pump-chamber dimensions, inlet gap distance, inlet radius, and chamfer dimensions. The user can simply set a parameter such as nozzle radius to what he desires and the model will produce the solution. Such changes can be repeated and taken to extremes and the model will adapt and rebuild the mesh by itself. This saves the user countless hours of re-meshing. Other features were built into the model to make it robust. A meshed area above the nozzle was built to prevent unwanted formation of reflections during the solution process. Another benefit is only

---

[q] Refer to Appendix A.13 to see how the overall sound pressure level is calculated

67

the desired results can be extracted from the solution. This saves data transfer time and storage space.

Limitations: Relative increases and decreases in flow rates and can be found with the model. But, model solutions do not directly compare to a Murata blower. The predicted SPL are only considering the air flow noise generated at the nozzle exit. Difficulties in isolating the blower's acoustics through baffling methods make direct comparisons to the model difficult. Many issues in this regard could be resolved by simply extending the meshed area further into the far field, thereby eliminating the near field measurement difficulties. But this is not computationally feasible. The model is thus limited to solving only the near field acoustics and velocity profiles 6 mm above the nozzle exit plane.

The biggest limitation of the DNS model is the computation time required for each solution. The model has over 1 million elements resulting in over 1.5 million equations to solve for each time step of the solution. The model solves using 12 processors on a Quad-core Intel Nehalem 2.8 GHz CPU in the BYU Fulton Super Computer Lab. Even with this kind of processing power, the required run time is generally over 200 hrs or 8 to 9 days. Any change to the model geometry will require this lead time. Also, each unique solution generates gigabytes of data, requiring tens of terabytes of data to be stored. The computational expense produces other limitations, such as the resolution of the solution's frequency spectrum. The model only runs for 1.5 to 2.0 ms with a time step of .001 ms. This limits the resolution of the frequency spectrum to about 500 Hz. Thus frequencies less than this resolution cannot be predicted by the model. This is somewhat of a drawback, because audible noise ranges as low as 20 Hz. Running a longer time solution would resolve this problem but leads to computation time issues

discussed above. The storage space required to store the solutions and the computational time to resolve the model are the biggest limitations to using the DNS model.

There are also limits to how large or small the input parameters can be assigned. If an input is too large it may begin to impact other dimensions unintentionally. For the most part, this is not a problem but each solution must be checked for unintended consequences of the input parameters. Another limit to the model worth noting is the directionality of the acoustics. The model can only resolve the pressure field produced from the nozzle area. No diffraction from the back portion of the blower is included in the solution due to the computational time that would be required. Neither is any structural vibration noise solved for in the model. Therefore, the model does not provide SPL data for the entire blower system, only a localized area near the nozzle exit. So the results cannot be taken as conclusive and all encompassing. The model results simply provide understanding of the noise produce by the air stream at the nozzle exit. Results from the model are presented in the next chapter.

4 RESULTS

This chapter presents and discusses the results found from varying geometrical inputs of the DNS model. The flow rates and SPLs predicted by the model are presented and plotted for the following cases: nozzle radius, nozzle height, inlet radius, inlet gap distance, pump-chamber nozzle radius, chamfer dimension, and pump-chamber volume. The SPL values mentioned in this chapter refer to the air stream noise only. Thus improvements in SPL are improvements in the jet noise contribution to the overall SPL of the blower. The geometric parameters with the largest impact on the SPL of the air stream noise and the flow rate will be identified and some optimum dimensions are noted. Figure 4-1 should be used as a reference to understand what dimensions are referred to throughout the section.
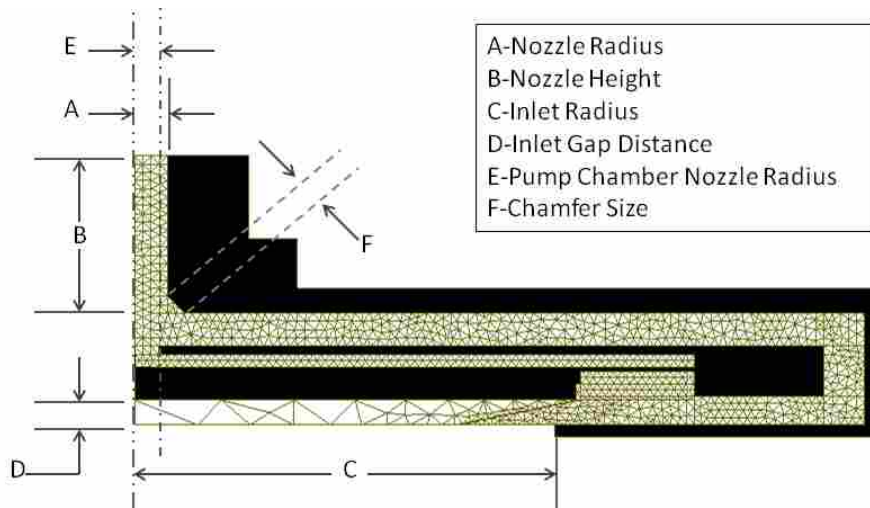
Figure 4-1: Geometrical dimensions reference diagram.

Generally, decreasing flow rate decreases SPL because the Reynolds number moves towards a laminar flow and results in less turbulent mixing. A solution that increases flow rate and decreases SPL is considered ideal for the Murata blower.

Each parameter listed in Figure 4-1 has been measured from a Murata blower with precision calipers. These baseline dimensions are input into the model and solved. The resulting solution predicted a flow rate of 0.57 L/min and an overall SPL from 1 to 20 kHz of 88.9 dB. These values are considered the baseline for comparing later results. Next, an increased and decreased value of each baseline measurement were input into the model and solved. More values were input and solved for to reach a reasonable conclusion on how each parameter affects the performance of a blower.

To better visualize the results in this section, a non-dimensional plot is created.   It plots a non-dimensional value of the flow rate vs. a non-dimensional value of overall sound pressure. This is done by taking the resulting flow rate for every change in parameter dimension and dividing that value by the baseline flow rate. Thus, if a solution is greater than 1, the flow rate is greater than the baseline. Likewise if the solution is less than one, the flow rate has decreased with respect to the baseline. This conversion is also performed on each corresponding SPL value. Now, each solution of the model is plotted with the flow rate value on the x-axis and the SPL value on the y-axis. The baseline model solution then becomes the center of the plot located at (1, 1) on the graph. If a parameter change has little effect, it will lie close to the benchmark position in the center of the plot. A parameter with a large influence on flow rate, SPL, or both, will appear farther away from the center of the graph. This plot is shown in Figure 4-2. All the model results are plotted for comparison to the benchmark blower measurements.
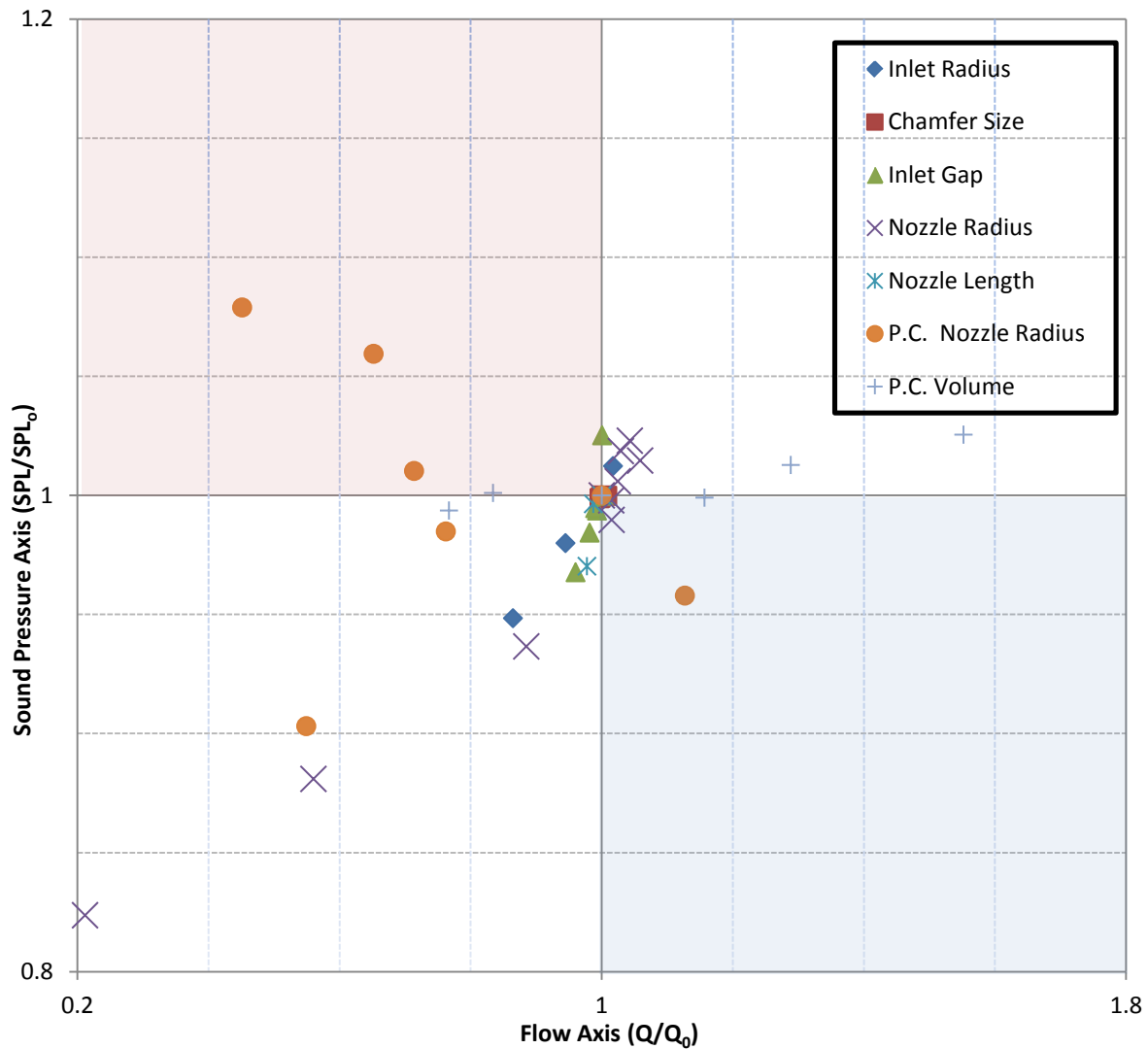
Figure 4-2: Non-dimensional plot of blower model results for a variety of parameters.

The effects on both flow rate and SPL are easier to see with the non-dimensional plot. Any result that falls in the blue quadrant has an increased flow rate accompanied by a decreased SPL. The red quadrant is exactly the opposite with a decreased flow rate and increased SPL. The blue quadrant is desired for a piezoelectric blower that is intended to be quiet and have a high flow rate, while the red quadrant should be avoided. The white quadrants compromise one desired characteristic for the other. Notice that some parameter changes had little effect on

performance by staying close to the center of the graph while others had rather large effects. Each parameter and corresponding results will be further analyzed in the remainder of this chapter.

4.1 Nozzle Radius

The tested nozzle radii ranged from 0.11mm to 0.91mm. The benchmark radius was measured to be 0.41 mm. Most of the results in the non-dimensional plot fall in the white quadrants or very near. This suggests that the nozzle radius will influence the flow rate and SPL but generally at the expense of the other. Both the flow rate and SPL are plotted as a function of radius in Figure 4-3.
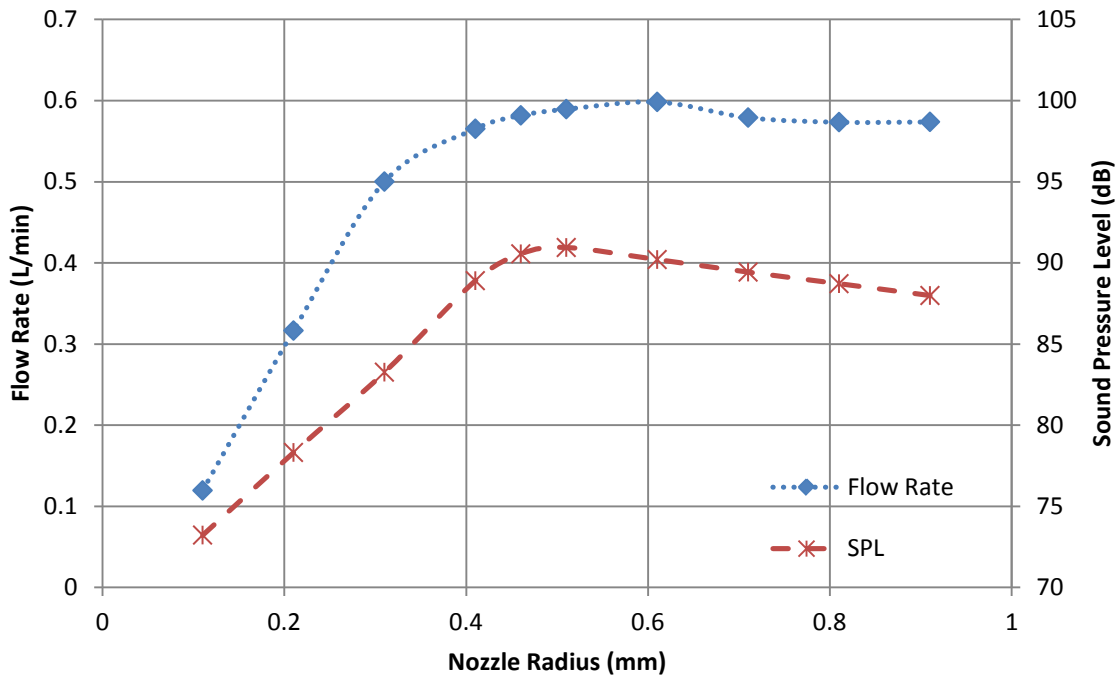
Figure 4-3: Flow rate and SPL as a function of nozzle radius.

The plot demonstrates how the SPL and flow rate depend on the nozzle radius. A peak flow rate is seen at 0.61 mm. For radii smaller than 0.61 mm, the flow rate decreases because the area of the nozzle decreases. The diminishing flow rate also results in a decreasing SPL. Such a result should be expected because a lower flow rate results in a lower Reynolds number making the flow more laminar as it exits the nozzle. The flow rate for radii greater than 0.71 mm remains relatively constant at 0.57 L/min. The radius of the exit nozzle is so large at this point that the controlling mechanism of the flow rate has become the hole in the pump-chamber. In comparing cross sectional areas at this data point, the pump-chamber hole is only 18% of the exit nozzle area. The exit nozzle radius appears to have little effect on the flow rate past this point. It should also be noted that the flow rate begins to drop sharply when the exit nozzle radius is less than 0.3 mm. This is the exact size of the pump-chamber hole radius.

The model results in this section can be compared to the flow rate of a theoretical turbulent jet. The volumetric flow rate of a turbulent jet is defined by Eq. (4.1) with x_o defined as the axial distance from the nozzle exit, r defined as the radius of the nozzle exit, and Q_i defined as the volume flow rate through the nozzle exit.[29]

$$Q = 0.16 \frac{x_o}{r} Q_i \qquad (4.1)$$

The volume flow rate through the nozzle also depends on r and is defined by Eq. (4.2) where U_o is the nozzle exit velocity and can be solved for from Eq. (3.12) when given the center line velocity U_m.

$$Q_i = U_o \pi r^2 \qquad (4.2)$$

Using the centerline velocity predicted by the model, the theoretical flow rate of a turbulent jet can be computed that depends on the nozzle radius. Now, the theoretical flow rate of a turbulent jet as a function of nozzle radius is plotted and compared to the corresponding model solutions in Figure 4-4.
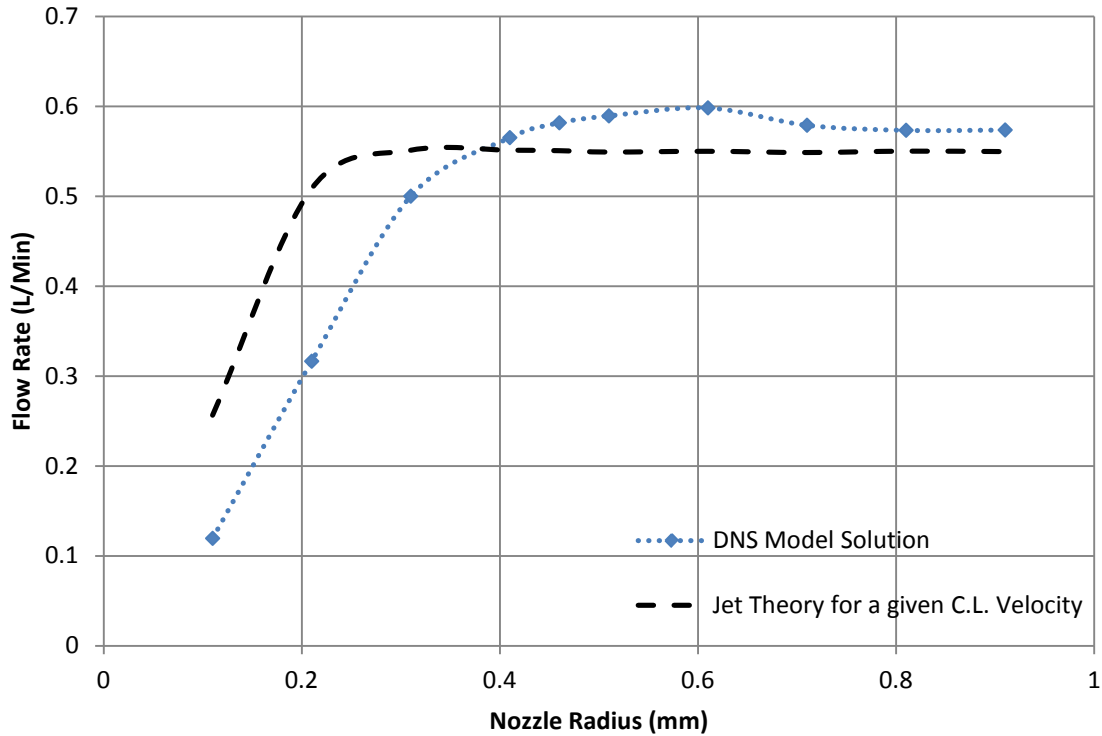
Figure 4-4: Modeled flow rate as a function of nozzle radius compared to jet theory.

The figure shows that the model solutions are all within 8% of what turbulent jet theory predicts with the exception of radii that are less than 0.3 mm. The model departs from theory by 20 to 30% for radii smaller than the pump-chamber hole radius. The model predicts a much faster drop in flow rate that is likely due to the exit nozzle being smaller than the pump-chamber hole.

The results from the model suggest that a reduced nozzle radius will reduce flow rate and simultaneously lower SPL. Other sub-sonic jet research concludes similar findings. A constant jet of air passing through a nozzle is louder for larger diameters.[30] It is a common practice in acoustic noise control to use many smaller nozzles in place of a larger one. The smaller nozzles result in a lower SPL and the use of multiple nozzles compensates for any loss of flow.

In conclusion, a larger nozzle radius generally yields greater flow and higher SPL near the nozzle. The nozzle radius no longer affects the flow rate after 0.61 mm but the pump-chamber continues to produce a jet of air. The model predicts flow rates that are within 8% of what jet theory suggests for nozzle radii greater than the pump-chamber hole radius of 0.3 mm. The nozzle radius impacts the SPL emitted by the blower, but cannot be used to minimize the noise without compromising the flow rate. Replacing the nozzle with many smaller nozzles might be a solution for the Murata blower.

4.2   Nozzle Height

The tested nozzle heights range from 1.9 mm to 2.9 mm. The benchmark nozzle height measured 1.9 mm. Multiple attempts were made at acquiring results for values less than 1.9 mm but the model solution became unstable in each case. The boundary conditions could not adapt to lower heights. A special case would need to be built to run heights less than 1.9 mm. The results for higher nozzles all landed in the white quadrants in Figure 4-2. Again this signifies that no real improvement can occur without compromising either the flow rate or the SPL. However, all the tests remained close to the flow rate axis meaning the change in flow rate was minimal. If a small amount of flow rate can be compromised, the level of sound reduction is comparatively large. Figure 4-5 plots flow rate and SPL as a function of the nozzle height. Minimal changes occur until a nozzle length of 2.4 is reached. After this point, the sound level

77

begins a steep drop off while the flow rate declines at a negligible rate only losing less than 1% of the benchmark flow rate.
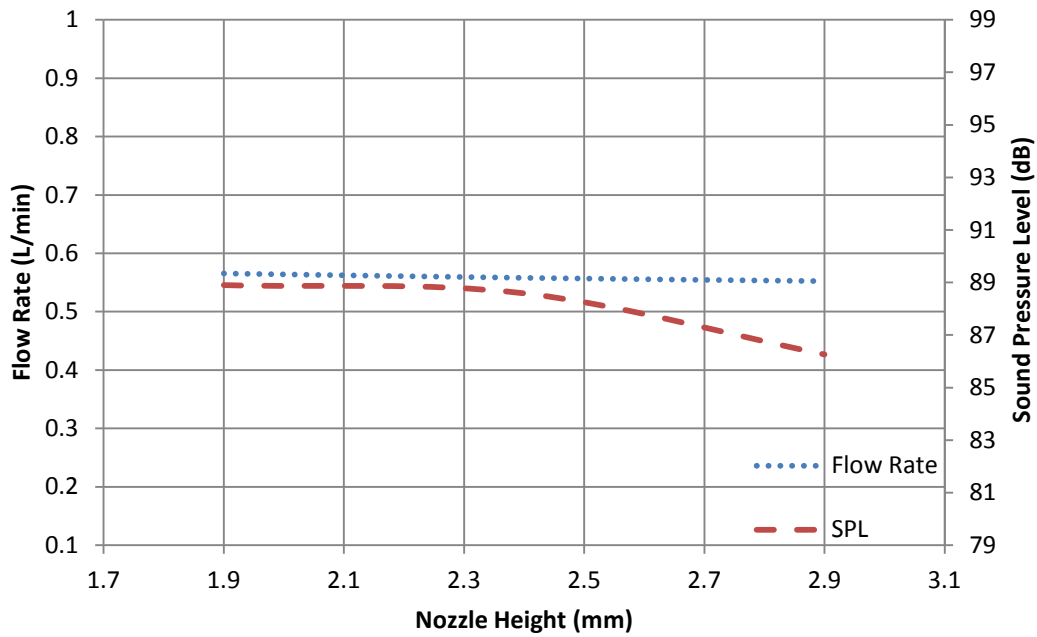
Figure 4-5: Flow rate and SPL as a function of nozzle height.

Although the payoff seems big (a reduction of 3 dB), a nozzle extension of this size nearly doubles the thin profile of the blower. If larger profiles are expectable for the given application of a blower, then mufflers would be a better solution than nozzle extension. Thin mufflers can provide equivalent or better results than a nozzle extension. Testing results have shown that the application of thin muffler reduces the SPL of a Murata blower by up to 13dB and only reduces flow rate by 2%.[r]   In conclusion, an increase in nozzle height can reduce the SPL by at least 3 dB and have a negligible impact on the flow rate for heights up to 2.9 mm but the profile of the blower should be considered before making such a design change.

---

[r] Refer to Appendix A.15 for testing procedure and results.

4.3   Inlet Radius

The tested inlet radii ranged from 4 mm to 8.5 mm.   The benchmark inlet radius measured 5.5 mm.   These results also fall in the white quadrants of Figure 4-2.   As the inlet radius becomes smaller than the 5.5 mm benchmark, the flow rate begins to level off.   This is shown in Figure 4-6.   Attempts to acquire data at smaller radii resulted in unwanted reflections in the model solution.   But, the results presented in the plot led to the assumption that the flow rate and sound level are dependent on the inlet radius and therefore the inlet area.   Both flow rate and SPL decrease as the inlet radius increases.   For radial values below 5.5 mm the decrease in flow rate is less than 1%.   This is the point where increasing the inlet radius begins to impact the flow rate.
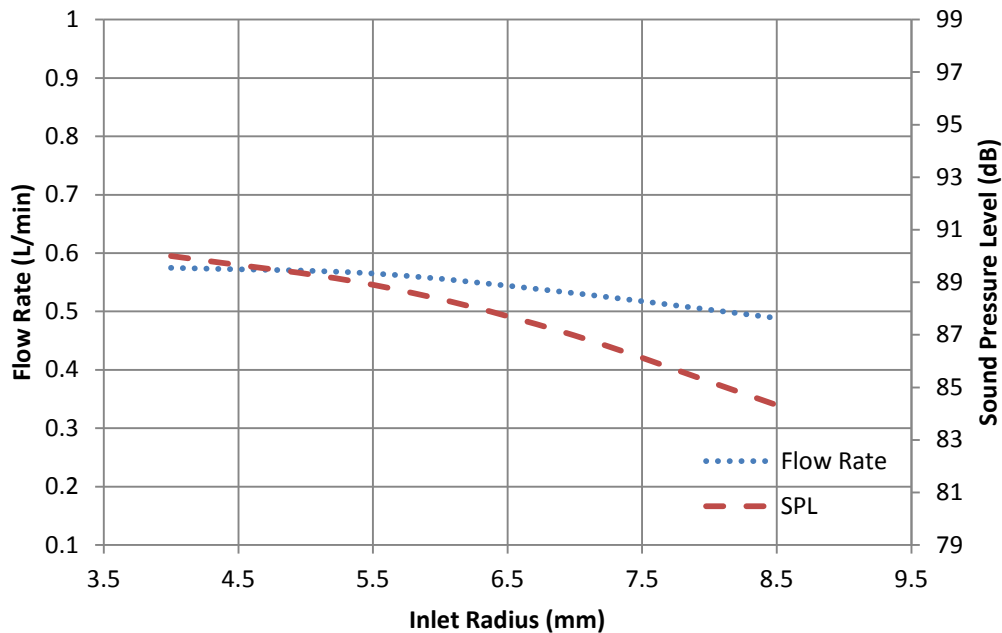
Figure 4-6: Flow rate and SPL as a function of inlet radius.

In conclusion, the inlet radius is another parameter that can increase or decrease the flow rate and SPL simultaneously. However, the decrease in flow rate in less than 1% for radial values less than 5.5 mm while the SPL values in this region experience a greater change. Thus 5.5 mm is the dimension where flow rate remains lest affected and SPL is minimized. The radii larger than 5.5 mm decrease in SPL but also experience larger losses in flow rate. The benchmark dimension is ideally located at this value. Any increase in the dimension will result in flow rate and SPL drops. Decreasing the dimension will result in higher SPL emitted by the nozzle.

4.4 Inlet Gap Distance

The tested inlet gap distances ranged from 0.15 mm to 0.60 mm. The benchmark inlet radius measured 0.30 mm. A review of Figure 4-2 shows that despite all the makers being in the white quadrants, most stay close to the flow rate axis. There are decreases in flow rate for many of the results but they are relatively small compared to the changes in SPL. This is better seen in Figure 4-7 which plots flow rate and SPL as a function of inlet gap distance. The maximum spread among the flow rate results is less than 5%. But a spread of nearly 5 dB in SPL occurs. Because of this, the inlet gap distance can be considered significant. The benchmark at 0.30 mm produces 88.9 dB. By doubling this gap, the model predicts a drop of nearly 3 dB to 86 dB and only compromises the flow rate by 5%. It is also noted that a spike in SPL results from a gap distance of 0.375 mm but the change in flow rate is negligible. This particular length matches some of the cavity dimensions in the flow path of the model.
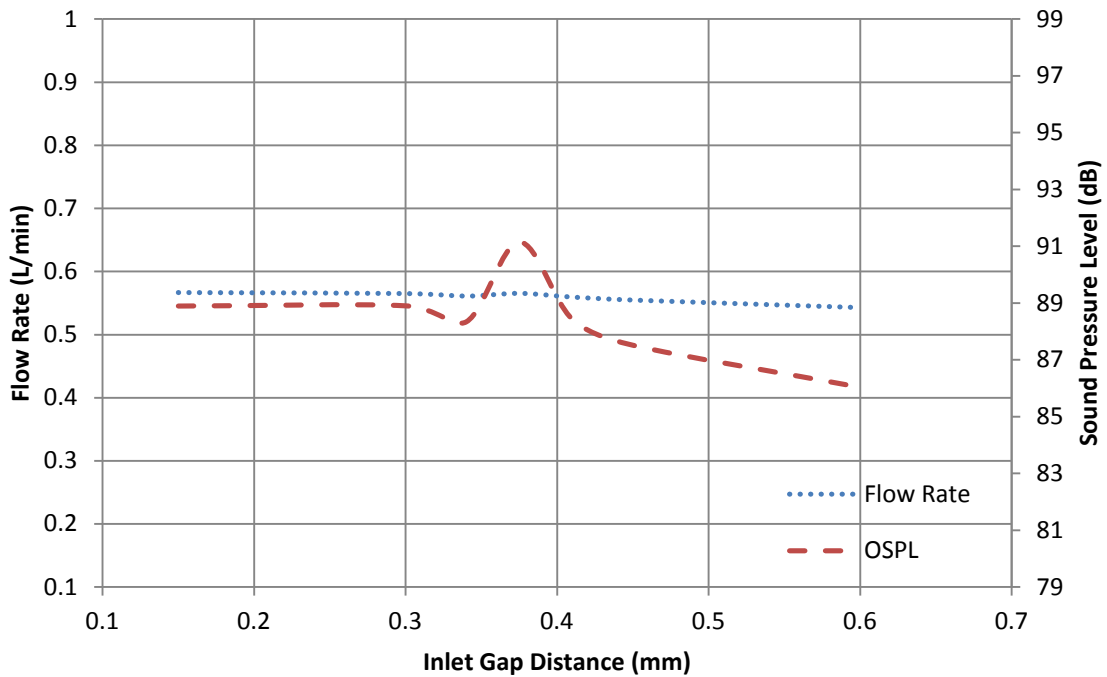
Figure 4-7: Flow rate and SPL as a function of inlet gap distance.

In conclusion, the inlet gap distance has a greater influence on the SPL than on the flow rate. Gap distances greater than 0.4 mm can reduce the SPL by 3 dB and can change the flow rate by less than 5%. Decreasing the benchmark gap size of 0.3 mm to 0.15 mm will not impact SPL or flow rate. It should also be noted that increasing the gap distance will increase the overall thickness of the blower.

4.5   Pump-Chamber Nozzle Radius

The tested pump-chamber nozzle radii ranged from 0.05 mm to 1.0 mm. The benchmark measured 0.30 mm. The orange dots in Figure 4-2 represent the changes made to the pump-chamber nozzle radius. The diverse locations and large spread of the data suggest the pump-chamber dimension is one of the most influential parameters yet tested. Most results are plotted in the red and white quadrants. However, this parameter is the only one thus far to plot in the

blue quadrant. This was achieved by shrinking the nozzle radius to 0.2 mm. The result is an increased flow rate and decreased SPL. Both SPL and flow rate are plotted as a function of the pump-chamber nozzle radius in Figure 4-8.
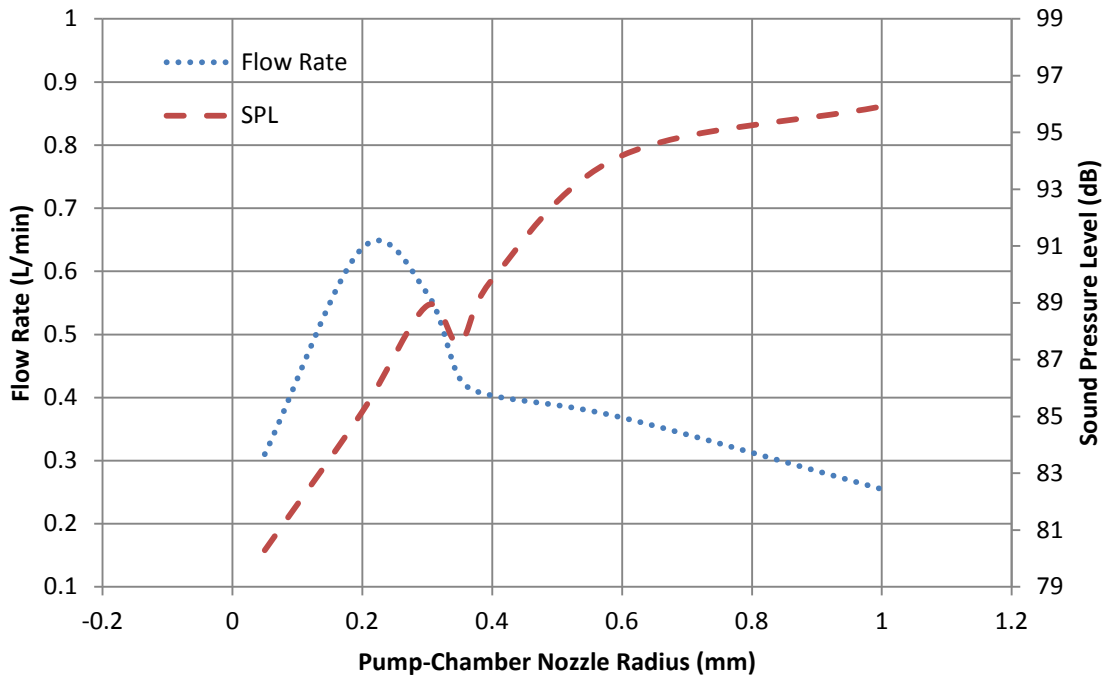
Figure 4-8: Flow rate and SPL as a function of pump-chamber nozzle radius.

The benchmark dimension of 0.3 mm is located where the two plots cross each other. Moving from 0.3 mm to 0.22 mm maximizes the flow rate increasing it by 16%. At the same time this change in radius reduces the SPL by about 3dB. These results show that an optimal radius will exist for the pump-chamber nozzle that produces a maximum flow rate and minimal SPL. Moving above this point will generally cause less flow and more noise. Moving below the optimal radius reduces both flow rate and SPL because the source of flow begins to close off. It is difficult to validate these model solutions experimentally or with theory. The hole in the

pump-chamber in a Murata blower is extremely difficult to modify without damaging the device. Neither can the pump-chamber radius be considered a constant stream of air. Air is rapidly being sucked in and pushed out of the hole.

In conclusion, the pump-chamber nozzle radius can be optimized. Furthermore, if not optimized, the resulting flow dynamics can cause a higher SPL and a lower flow rate. The solution the model gives as the optimal radius is approximately 0.22 mm. Comparing this result to the benchmark, there is a 16% increase in flow rate and a 3 dB drop in SPL. It should be noted that for all these solutions the exit nozzle radius was set to 0.41mm which is nearly double the optimal pump-chamber radius of 0.22 mm. This suggests that the optimal pump-chamber nozzle radius should be near 50% of the exit nozzle radius of a Murata blower.

4.6   Chamfer Dimension

Another dimension of interest is the chamfer size. The tested sizes ranged from 0.10 mm to 0.30 mm. The benchmark size was 0.2 mm. A review of the non-dimensional plot in Figure 4-2 shows all the results placed essentially on top of the benchmark in the center of the plot. This indicates that reducing the sharp corner when air enters the nozzle region has little effect on the blower performance. The SPL and flow rate results are plotted as a function of chamfer size in Figure 4-9 and clearly show that the chamfer does not affect either of the performance indicators.
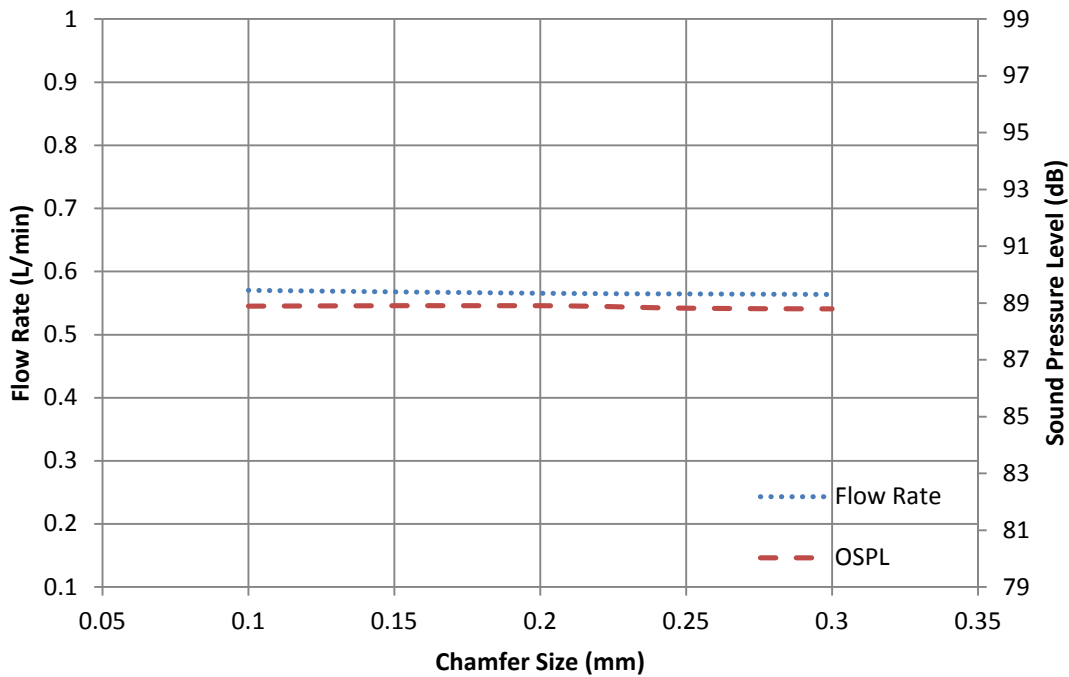
Figure 4-9: Flow rate and SPL as a function of chamfer size.

It was previously assumed by some that rounding the edges in the flow path might have an impact on the SPL by instigating a more laminar flow. The results from the model clearly show that any such effort would need to be more aggressive than simply adding larger fillets or chamfers to corners and/or edges in the flow path.

4.7   Pump-Chamber Volume

Since the only significant improvement to both SPL and flow rate was a result of changing the pump-chamber hole radius, the entire mechanism could deserves more investigation. The volume of the pump-chamber was changed by increasing and decreasing the radius of the chamber. The height of the chamber was not changed. The radii tested ranged

from 3.9 mm to 7.9 mm and resulted in respective volumes ranging from 7.2 mm^3 to 29.4 mm^3.

The benchmark volume was 22.4 mm^3. The flow rate and SPL are both plotted as a function of

the pump-chamber volume in Figure 4-10.
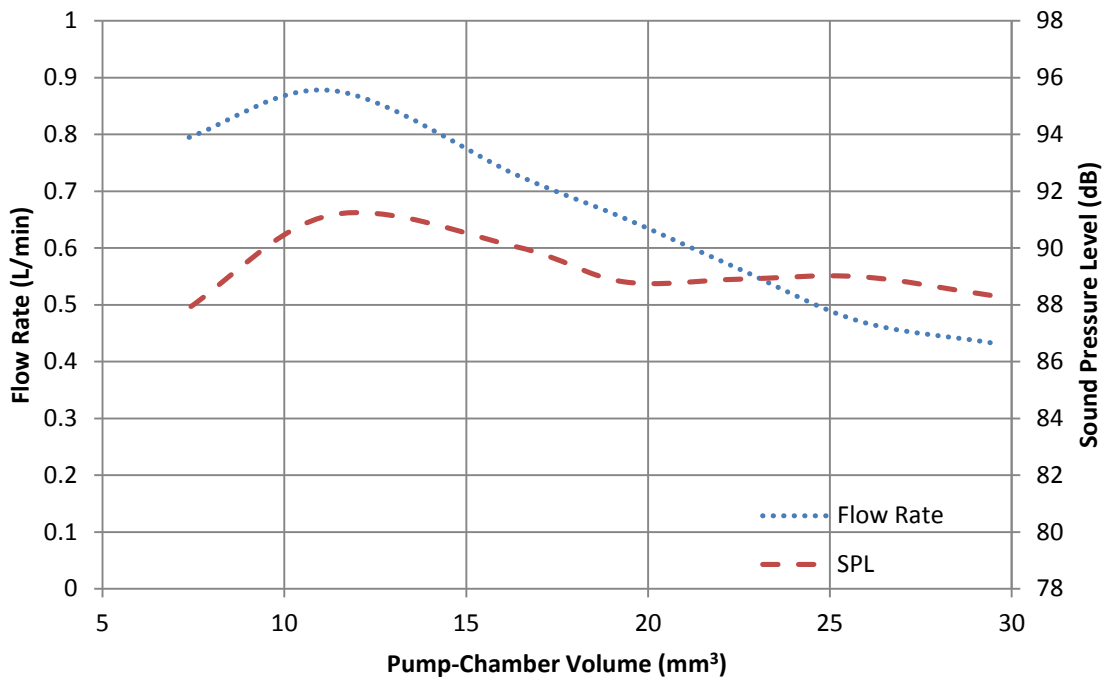
Figure 4-10: Flow rate and SPL as a function of pump-chamber volume.

The plot shows that the volume of the pump-chamber affects the flow rate more than any

previous dimension. The flow rate peaks at 0.87 L/min at a volume of 11.5 mm^3. This is a 55%

increase from the benchmark flow rate. Although the SPL increases with the flow rate, it is a

modest 2 dB gain. At volumes greater than 23 mm^3 the flow decreases while the SPL remains

relatively steady. The flow rate doubles by reducing the volume from 29 mm^3 to 11.5 mm^3 and

in the same range the SPL has a max spread of 2 dB.

These results suggest that the pump-chamber plays an important role in the flow rate of a Murata blower. By reducing the volume of the chamber by 60%, the flow rate can be increased by 50% and the SPL gain out the nozzle will be near 2 dB. These increases were the result of decreasing the radius of the pump-chamber as a whole while the height of the chamber itself remained at 0.15 mm. This suggests that decreasing the radius to height ratio of the pump-chamber can improve the flow rate of a Murata blower. Increasing the height of the chamber will decrease this ratio as well and could produce similar results, but such a change will increase the overall profile of the blower.

4.8   Results Summary

The results from this chapter are summarized and compiled in Table 4-1. The parameters of interest are listed in the first column, followed by a brief summary of key points from the model results. The level of influence each physical parameter had on the SPL of the nozzle and the flow rate is ranked in the last two columns. The table can be used a guideline in future designs of Murata blowers. The decreases in SPL referred to in the table only refer to the noise emitted by the nozzle not the overall noise emitted by the blower. Since the air stream out the nozzle is only one contributor to the total acoustic noise, the dB values cited in the table should not be translated directly to total dB reduction.

Table 4-1: Summary of model data and rank of influence.

FLOW MODEL RESULTS SUMMARY FOR A MURATA BLOWER			
Parameter Name	Summary of Impact on Nozzle Noise and Flow Rate (conclusions based on DNS model data)	Influence on SPL near Nozzle	Influence on Flow Rate
Nozzle Radius	Both flow rate and SPL are a function of the nozzle radius. The DNS model predicts flow rates within 8% of jet theory. Nozzle radii greater than 0.61 mm have little impact on the flow rate. Below this value, reducing the radius reduces flow rate and SPL. Using multiple smaller nozzles could result in less noise and not compromise the flow rate.	High	High
Nozzle Height	SPL is a function of the nozzle height. The SPL was decreased approximately 3dB by adding 1.0 mm to the height of the nozzle (a 33% increase in length). The resulting flow rate also diminished but at a negligible rate (less than a 1% change overall). Since an increase in nozzle height adds to the profile of the blower, the size requirements in the blower's application should be considered. Other options exist that increase the blower profile and provide greater attenuation such as applying a muffler.	Medium	Low
Inlet Radius	The inlet radius affects both the flow rate and SPL. For a radius less than 5.5 mm the flow rate remains unchanged. For radii larger than 5.5 mm the flow rate begins to drop. The SPL decreases with increasing inlet radius. A 5.5 mm inlet radius will provide the maximum flow rate with minimum SPL.	Medium	Medium
Inlet Gap Distance	Gap distances greater than 0.4 mm can reduce the SPL by 3 dB and can change the flow rate by less than 5%. The SPL peaks with a gap distance near 0.375mm and then begins to drop as the gap is increased. Distances less than the 0.375mm avoid the peak SPL and affect the flow rate less than 1%. However, increasing the gap distance will increase the thin profile of the blower.	Medium	Low
Pump-Chamber Nozzle Radius	The hole size in the pump-chamber is a critical parameter and can be optimized. There is an optimum hole radius that produces a high flow rate and low SPL. The optimal radius for the model was nearly equal to 50% of the exit nozzle radius. The benchmark model had 16% increase in flow rate and 3 dB loss in SPL by using a 0.22 mm pump-chamber nozzle radius.	High	High
Chamfer Size	Results show no signs of consequential influence to flow rate or SPL. The chamfer leading into the nozzle was reduce by 50% as well as increase by 50% and the noise levels and flow rates experienced changes of less than 1%.	Low	Low
Pump-Chamber Volume	The volume of the pump-chamber impacts the flow rate more than any other parameter tested. By reducing the radius of the chamber the flow rate saw a 50% increase and the SPL increased by 2 dB.	Medium	High

5 CONCLUSIONS

5.1 Summary

This thesis characterized the acoustic emissions of the front face of a Murata piezoelectric blower. Noise generating mechanisms and components were identified through the development of a DNS flow model of a blower.

5.1.1 Acoustic Noise Characterization

Structural vibrations were proven to contribute to the measured sound power of a Murata blower. Measurements from an SLDV document the presence of three significant operational mode shapes. Plate radiation theory suggested that the modes contribute to spikes of measured sound power on the front surface of a blower. Other experimental testing in an anechoic chamber provided some evidence of jet noise and/or turbulence induced noise in the frequency spectrum of a Murata blower.

5.1.2 DNS Blower Model

The purpose of building a CFD model of a blower was to better understand how physical mechanisms in the blower flow path influence the nozzle noise and the blower flow rate. The components of interest included: the nozzle radius, nozzle height, inlet radius, inlet gap distance, pump-chamber nozzle radius, and chamfer size. The model was built in ADINA and used DNS

to solve the flow field. The model contains over 1 million elements and requires 8 to 9 days to solve on a multi processor supercomputer. It is parameter based; meaning dimensions in the model can be changed without re-programming the entire mesh. The model solution from 1 kHz to 20 kHz is within the standard deviation of comparable measurements of the frequency spectrum of a Murata blower with the exception of two frequencies where operational modes are present. The model was used to predict relative changes in the blower flow rate and the SPL near the nozzle exit while geometrical dimensions in the flow path were changed.

5.1.3   Results

The model results suggest that the most critical mechanism in a Murata blower that influences the flow rate and nozzle noise level is the pumping chamber. The hole size of the pumping chamber influences these properties and can be optimized to reduce and increase them as desired. The other dimensions that were tested and a summary of their influence on SPL and flow rate are briefly listed and described in Table 4-1.

5.2   Recommendations

This research found that sound power is influenced by the structural vibrations of a piezoelectric blower in operation. It is recommended that future research recognize the influence these structural vibrations have on the sound power radiating from the front surface. This research also developed and validated a DNS model in ADINA purposed in identifying influential parameters that contribute to nozzle noise and influence volumetric flow rate. It is recommended that the model solutions and recommendations listed in Table 4-1 be considered in future blower design to minimize acoustic noise levels of the nozzle and maximize flow rates. Significant results involving the pump-chamber of the Murata blower deserve further study. The

90

pump-chamber nozzle radius maximized flow and minimized nozzle noise when equal to 50% of the radius of the exit nozzle. By testing more data sets while changing these two dimensions together, a relationship might be defined that can improve the SPL and flow rate even further. Furthermore, suggestions have been made in this research that the pump-chamber radius to height ratio can be decreased to improve flow rate significantly. More testing in this area is recommended. If possible, models and/or prototype designs might be built to test and document results from implementing multiple small diameter nozzles in place of a single exit nozzle. The model results suggest that a smaller nozzle radius decreases nozzle noise. However, the model in this research was not capable of predicting the affect of multiple smaller nozzles.

A more effective software package for modeling is recommended if further research is to be done. The DNS flow model was developed in ADINA. A more user friendly software program is recommended if similar turbulent models of air streams are to be developed. A more capable CFD package with custom acoustic features specific to this kind of research, such as FLUENT might be recommended for future work in this area.

The time and effort required to develop a DNS model of a turbulent jet is extensive. In addition, the computational time required for solving and post processing the model is undesirable. Other turbulent fluid models might be easier to solve and still provide adequate solutions.[22, 28, 31, 32] A recommendation of turbulent flow models may include Large Eddy Simulation (LES) or Detached Eddy Simulation (DES) based models. If these models could be validated then the solution process would be faster and more efficient.

REFERENCES

1. Murata Manufacturing Co.,
    http://www.murata.com/products/micromechatronics/feature/microblower/index.html,
    11 2012

2. L. E. Kinsler, in *Fundamentals of acoustics* (Wiley, New York, 2000), pp. 1, 312.

3. F. Fahy and P. Gardonio, in *Sound and Structural Vibration : Radiation, Transmission and Response* (Elsevier/Academic, Amsterdam ; Boston, 2007), pp. 135-138.

4. P. J. Morris, *Jet noise from large-scale turbulent structures*, The Journal of the Acoustical Society of America **123** (5), 3020-3040 (2008).

5. D. J. Gorman, in *Vibration analysis of plates by the superposition method* (World Scientific, Singapore ; River Edge, N.J., 1999), pp. 299-316.

6. M. Lematre, G. Feuillard, E. L. Clezio and M. Lethiecq, *Modeling of the influence of a prestress gradient on guided wave propagation in piezoelectric structures*, The Journal of the Acoustical Society of America **120** (4), 1964-1975 (2006).

7. L. Shuyu, *Optimization of the performance of the sandwich piezoelectric ultrasonic transducer*, The Journal of the Acoustical Society of America **115** (1), 182-186 (2004).

8. H. C. Geng, Z. S. Rao and Z. S. Han, *New modeling method and mechanism analyses for active control of interior noise in an irregular enclosure using piezoelectric actuators*, The Journal of the Acoustical Society of America **113** (3), 1439-1447 (2003).

9. J. M. B. P. Sen-Gupta, *Flow through Porous Media in the Presence of Sound*, The Canadian Journal of Chemical Engineering **49** (1971).

10.   O. Lacour, M. Lagier and D. Sornette, *Effect of dynamic fluid compressibility and permeability on porous piezoelectric ceramics*, The Journal of the Acoustical Society of America **96** (6), 3548-3557 (1994).

11.   C. K. W. Tam, *High-speed jet noise: A two source model*, The Journal of the Acoustical Society of America **127** (3), 1743-1743 (2010).

12.   P. A. Lush, *Measurements of subsonic jet noise and comparison with theory*, J Fluid Mech **Vol.46** (3), 477-500 (1971).

13.   F. Fahy and P. Gardonio, in *Sound and Structural Vibration : Radiation, Transmission and Response* (Elsevier/Academic, Amsterdam ; Boston, 2007), pp. 159-175.

14.   L. E. Kinsler, in *Fundamentals of acoustics* (Wiley, New York, 2000), pp. 95-98.

15.   P. J. M. Jark, C. Lau, and Michael J. Fisher, *Measurements in Subsonic and Supersonic Free Jets Using a Laser Velocimeter*, J Fluid Mech **93**, 1-27 (1972).

16.   M. J. Lighthill, *Jet Noise*, Aiaa J **1** (7), 1507-1517 (1963).

17.   H. H. Hubbard, in *Aeroacoustics of Flight Vehicles* (The Acoustical Society of America, 1995), Vol. 1, pp. 259-262.

18.   M. J. Fisher, *Jet Engine Noise Source Location: The Polar Correlation Technique*, Journal of Sound and Vibration **51** (1), 23-54 (1977).

19.   E. W. Graham and B. B. Graham, *Transient Acoustical Sources in an Idealized Jet*, The Journal of the Acoustical Society of America **52** (1B), 221-226 (1972).

20.   A. B. L Maestrello, *Flowfield and Far Field Acoustic Amplification Properties of Heated and Unheated Jets*, Aiaa J **20** (11), 1539-1546 (1982).

21.   M. Lee and J. S. Bolton, *Source characterization of a subsonic jet by using near-field acoustical holography*, The Journal of the Acoustical Society of America **121** (2), 967-977 (2007).

22.   P.-T. Lew, L. Mongeau and A. Lyrintzis, *Noise prediction of a subsonic turbulent round jet using the lattice-Boltzmann method*, The Journal of the Acoustical Society of America **128** (3), 1118-1127 (2010).

23.   F. M. White, in *Fluid mechanics* (McGraw-Hill, New York, 2008), pp. 3-5.

24.   P. K. Kundu and I. M. Cohen, in *Fluid Mechanics* (Academic Press, Amsterdam ; Boston, 2008), pp. 53-55.

25.   H. Tennekes and J. L. Lumley, in *A first course in turbulence* (MIT Press, Cambridge, Mass.,, 1972), pp. 1-24,67.

26.   B. R. Munson, D. F. Young and T. H. Okiishi, in *Fundamentals of fluid mechanics* (J. Wiley, New York, 1998), pp. 279-280.

27.   D. C. Wilcox, in *Turbulence modeling for CFD* (DCW Industries, La Cãnada, Calif., 2006), pp. 1-37.

28.   M. T. Pittard, *Large Eddy Simulation Based Turbulent Flow-Induced Vibration of Fully Developed Pipe Flow*, Brigham Young University, 2003.

29.   R. D. Blevins, *Applied fluid dynamics handbook*. (Van Nostrand Reinhold Co., New York, N.Y., 1984).

30.   H. H. Hubbard, in *Aeroacoustics of Flight Vehicles* (The Acoustical Society of America, 1995), Vol. 1, pp. 275-277.

31.   L. M. Alvin Bayliss, *Simulation of Instabilities and Sound Radiation in a Jet*, Aiaa J **19** (7), 835-841 (1981).

32.   G. Bodard and C. Bailly, *A hybrid method for jet noise predictions based on Large Eddy Simulation and Reynolds-Averaged Navier-Stokes simulations*, The Journal of the Acoustical Society of America **123** (5), 3125-3125 (2008).

33.   M. Raffel, C. E. Willert and J. Kompenhans, in *Particle image velocimetry : a practical guide* (Springer, Berlin ; New York, 1998), pp. 3-8.

APPENDIX A. TEST PROCEDURES AND SETTINGS

A.1 SLDV Transverse Velocity Measurement Settings

The following file describes the setting used by the Scanning Laser Doppler Vibrometer in finding the structural modes and transverse velocity magnitudes of a piezoelectric blower. See section 2.1 for more information on how the measurements were taken and processed.

```
Name:          C:\Documents and Settings\SLDV\Desktop\Brad June
               2012\With Mic as Ref.svd
User:          SLDV
Created:       Wednesday, June 13, 2012
               6:48:26 PM
File version: 8.70
Application version:8.7.2.0
-----------------------------------------------------------------------
Comment

-----------------------------------------------------------------------
Scan Points

Total:         2401
Not Measured: 0                 0.0 %
Valid:         69               2.9 %
Optimal:       2332             97.1 %
Overrange:     0                0.0 %
Invalidated:  0                 0.0 %
Disabled:      0                0.0 %
Not Reachable:0                 0.0 %
Hidden:        0                0.0 %
-----------------------------------------------------------------------
Hardware

Scanning Head:PSV-I-400 LR  (OFV-505)
Junction Box: PSV-E-400-H4
Acquisition Board:National Instruments PCI-4452
-----------------------------------------------------------------------
General

Acquisition  Mode:FFT
Averaging:    Complex
Averaging count:10
PCA (MIMO):   Not active
Cosine correction X:Active
Cosine correction Y:Active
```

97

```
------------------------------------------------------------------------
Frequency

Bandwidth:     20 kHz
Bandwidth from:100 Hz
Bandwidth to: 20 kHz
------------------------------------------------------------------------
Sampling

FFT  Lines:    1600
Overlap:       0 %
Sample frequency:51.2 kHz
Sample time:   80 ms
Resolution:    12.5 Hz
------------------------------------------------------------------------
Trigger

Source:        Off
Phase from reference:On
------------------------------------------------------------------------
Channel Vibrometer (connected to Vibrometer 1)

Direction:     +Z
Range:         10 V
Coupling:      DC
Impedance:     1 MOhm
Differential Input:Off
Quantity:      Velocity
Calibration factor:5e-3  (m/s)/V
Signal Delay: 94e-6   s
Filter Type:  No Filter
Int/Diff Quantity:Velocity
Window:        Rectangle
Signal Enhancement:Active

Channel Reference 1

Reference:     Active
Reference point index:0
Direction:     +Z
Range:         100 mV
Coupling:      DC
Impedance:     1 MOhm
ICP:           On
Differential Input:Off
Quantity:      Voltage
Calibration factor:1
Signal Delay: 0   s
Filter Type:  No Filter
Int/Diff Quantity:Voltage
Window:        Rectangle
Signal Enhancement:Not active
------------------------------------------------------------------------
Signal Enhancement

Speckle Tracking:Not active
Mode:          Standard
------------------------------------------------------------------------
Vibrometer 1

Controller:    OFV-5000
Firmware version:2.0
Sensor head:  OFV-505
Firmware version:1.03
```

```
        Tracking filter:Off

        Velocity output

        Range:         VD-08 5 mm/s/V
        Low pass filter:1.5 MHz
        High pass filter:Off
        -----------------------------------------------------------------
```

A.2 Sound Power Calculation Code (MatLab)

The following code is used to calculate the measured sound power results of Chapter 2.

The code requires the output files of 13 microphones on a revolving boom. The files should

contain the frequency vs time data for each microphone at each location.

```
%-----------------------------------------------------------------
% Boom plots for use with the Labview Data Acquisition
%(based on Labview_Boom_Plot_v2 by Matthew Green)
% Modified to only produce a directional plot and spectrum by Brad Solomon
%-----------------------------------------------------------------

clear all;
set(0,'DefaultAxesFontName','Arial');
set(0,'DefaultAxesFontSize',22);
set(0,'DefaultAxesFontWeight','bold')
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8);
set(0,'DefaultFigurePosition',[50  50  1400/1.5  1000/1.5])

% user defined variables

targetFreq =13500;
numMics = 13;

    % integrates this far in each direction
numHzIntegrate = 12500;

    % full path if not in same folder as .m file
    % on and off used previously for comparing noise control measurements
folderNameANCoff = ('C:\Users\bigbrad\Documents\My Dropbox\Murata\Boom Data\BaffledPower1');
folderNameANCon = ('C:\Users\bigbrad\Documents\My Dropbox\Murata\Boom Data\BaffledPower1');

 % this is hard coded and will change if the file format from Labview changes
rowOffset = 9;

    %radius for small boom with extentions on it.  Average distance =24.625in
r=0.625475;
    %radius for small boom with extentions and baffle setup, Aproximation.
% r=.74295;

    %angles of microphones
thetaO=15*pi/180;


% read in data for control off
cd(folderNameANCoff);
a2 = dir;                 %Find out how big the directory is
sizea2 = size(a2);
numfiles2 = sizea2(1)-2;    %-2 is used to account for the two files '.' and '..'

for k = 1:numfiles2
```

99

```
    if k ==1
        infilnames = cell(k,1);
    end
    infile = getfield(a2,{k+2,1},'name');
    infilenamesANCoff{k,1} = infile;
end                         %Create a cell array of all of the usable files
    %Strings of the needed input file names
infilesANCoff = char(infilenamesANCoff);

%The degree increment placed into the Labiew code
%This may be changed to include the degree signature on the second file
%rather than this way
deginc = 180/numfiles2;

prefreq = dlmread(infile(1,:),'\t',rowOffset,0);
freq = prefreq(:,1);        % Frequency array, can be used to plot the spectrum
df = freq(2);


% read in data for control on
cd('..');
cd(folderNameANCon);
a2 = dir;                   %Find out how big the directory is
sizea2 = size(a2);
numfiles2 = sizea2(1)-2;    %-2 is used to account for the two files '.' and '..'

for k = 1:numfiles2
    if k ==1
        infilnames = cell(k,1);
    end
    infile = getfield(a2,{k+2,1},'name');
    infilenamesANCon{k,1} = infile;
end                         %Create a cell array of all of the usable files
    %Strings of the needed input file names
infilesANCon = char(infilenamesANCon);


%----------- Choosing Bins to Integrate ---------------
targbin = round(targetFreq/df);
numBinsIntegrate = round(numHzIntegrate/df);
if (rem(numBinsIntegrate,2) > 0)
    numBinsIntegrate = numBinsIntegrate;
end

% set how many bins to integrate over
low_bin = targbin-numBinsIntegrate;
high_bin = targbin+numBinsIntegrate;
freq2 = freq(low_bin:high_bin);
% dataANCoff = zeros(1,length(freq2));
% dataANCon = zeros(1,length(freq2));

% load data array for control off
cd('..');
cd(folderNameANCoff);
for i=1:13
    thetai(i)=thetaO*(i-7)-thetaO/2;
end
% for i=1:5;
%     A26(i)=r^2*(cos(thetai(i+2))-cos(thetai(i+1)))*thetaO;
% end
%
% for i=1:5;
%     A812(i)=r^2*(cos(thetai(i+7))-cos(thetai(i+8)))*thetaO;
% end
%
% A=[r^2*thetaO*cos(thetai(2)) A26 2*pi*r^2*(1-cos(thetaO/2)) A812 r^2*thetaO*cos(thetai(13))];
A(1) = r^2*cos(thetai(2))*thetaO;
for n=2:6
    A(n) = r^2*(cos(thetai(n+1))-cos(thetai(n)))*thetaO;
end
A(7) = 2*pi*r^2*(1-cos(thetaO/2))/numfiles2;
```

```
for n = 8:12
    A(n) = r^2*(-cos(thetai(n+1))+cos(thetai(n)))*thetaO;
end
A(13) = r^2*cos(thetai(13))*thetaO;

for k = 1:numfiles2
    infile = infilesANCoff(k,:);
    data = dlmread(infile,'\t',rowOffset,1);
    %Loads the data for the specific file
    %into temp file
    %data(:,13) = 1.4*data(:,13);
        pdata = 10.^(data./10);
    for i=1:13
        pdata(:,i)=A(i)*pdata(:,i);
    end
    %data2 = sum(pdata');
    %dataANCoff = dataANCoff + data2(low_bin:high_bin);

%---------------- Integrate Bins ----------------------------------------
    for j = 1:numMics
        % Get the bins for each
        tempint = 10.^(data(low_bin:high_bin,j)./10);
        %microphone that will be integrated and convert to pressure
        dataANCoff1(j,:)=pdata(low_bin:high_bin,j);
        % Sum the bins of interest
        tempmic = sum(tempint);
        micANCoff(j) = tempmic;
        % Convert back to dB
        tempmic = 10.*log10(tempmic);
        % Array of the integrated mic values
        dBmicANCoff(j) = tempmic;
    end
    % Load all of the Arrays of mics into single array
    dbmicarrayANCoff(k,:) = dBmicANCoff;
    micarrayANCoff(k,:) = micANCoff;
    dataANCoff(k,:,:)=dataANCoff1(:,:);

end
 DAOff=sum(dataANCoff);
 %DAOff(:,7,:)=DAOff(:,7,:)/numfiles2;
 DAOff=sum(DAOff);
 DAOff=reshape(DAOff,1,length(DAOff));

%load data array for control on
cd('..');
cd(folderNameANCon);

for k = 1:numfiles2
    infile = infilesANCon(k,:);
    data = dlmread(infile,'\t',rowOffset,1);


    pdata = 10.^(data./10);
    for i=1:13
        pdata(:,i)=A(i)*pdata(:,i);
    end

%---------------- Integrate Bins ----------------------------------------
    for j = 1:numMics
        tempint = 10.^(data(low_bin:high_bin,j)./10);
        dataANCon1(j,:)=pdata(low_bin:high_bin,j);
        tempmic = sum(tempint);
        micANCon(j) = tempmic;
        tempmic = 10.*log10(tempmic);
        dBmicANCon(j) = tempmic;
    end

    dbmicarrayANCon(k,:) = dBmicANCon;
    micarrayANCon(k,:) = micANCon;
    dataANCon(k,:,:)=dataANCon1(:,:);
```

```
end
 DAOn=sum(dataANCon);
 DAOn=sum(DAOn);
 DAOn=reshape(DAOn,1,length(DAOn));


%The value at mic 7 is averaged since it is being plot at the same spot
%for each boom location
mic7ANCoff = dbmicarrayANCoff(:,7);
mic7ANCon = dbmicarrayANCon(:,7);
avemic7ANCoff = sum(mic7ANCoff)/(length(mic7ANCoff));
avemic7ANCon = sum(mic7ANCon)/(length(mic7ANCon));
avemic7arANCoff = ones(numfiles2,1).*avemic7ANCoff;
avemic7arANCon = ones(numfiles2,1).*avemic7ANCon;
dbmicarrayANCoff(:,7) = avemic7arANCoff;
dbmicarrayANCon(:,7) = avemic7arANCon;

%To add continuity to the plot the first measurement is repeated and added
%at the end
%The measurements from the first boom position need to be inverted to give
%continuity
for t = 1:numMics
    inverANCoff(t) = dbmicarrayANCoff(1,(14-t));
    inverANCon(t) = dbmicarrayANCon(1,(14-t));
end

dbmicarrayANCoff((numfiles2+1),:) = inverANCoff;
dbmicarrayANCon((numfiles2+1),:) = inverANCon;

% Make array to use for the meshplot
% maxmic = max(max(dbmicarray));
endsize = size(dbmicarrayANCoff);
% maxmics = ones(endsize).*maxmic;

%---------------- Calculate Average Reduction ----------------------------
AvgreductiondB = 10*log10(sum(DAOff))-10*log10(sum(DAOn));
AvdBOff=10*log10(sum(DAOff));
AvdBOn=10*log10(sum(DAOn));

%---------------- Plotting -----------------------------------------------
%angles for directivity:
%this is my azimuthal angle of the boom rotation
theta=(0:deginc:180).*pi./180;
%this is my polar angle of the mic
phi=[0.076463376183437 0.245514631172427 0.422149530688674...
    0.614302752557062 0.837624639034357 1.142614577188583...
    1.570796326794897 1.998978076401211 2.303968014555437...
    2.527289901032732 2.719443122901120 2.896078022417367...
    3.065129277406356];

%create mesh of theta and phi
[Phi,Theta]=meshgrid(phi,theta);


%Mic 1 starts at the position of positive x and zero y
%define as x,y,z in spherical coords.
x=dbmicarrayANCon.*cos(Theta).*cos(Phi);
y=dbmicarrayANCon.*sin(Theta).*cos(Phi);
z=dbmicarrayANCon.*sin(Phi);

%coords of hemisphere
xbl=dbmicarrayANCoff.*cos(Theta).*cos(Phi);
ybl=dbmicarrayANCoff.*sin(Theta).*cos(Phi);
zbl=dbmicarrayANCoff.*sin(Phi);
%need to define a "black" colormap for the meshplot
blackmap=zeros(endsize);

%% Plots the dome

figure;
mesh(xbl,ybl,zbl,blackmap,'LineWidth',2);
```

102

```
hidden off;
hold on;
set(gcf,'Color',[1 1 1]);
surf(x,y,z,dbmicarrayANCon);
set(gca,'FontName','arial');
%define min and max values of colormap
% caxis([20,32]);
view(3);
h=colorbar ('vert','FontSize',20);  %displays colorbar
set(get(h,'ylabel'),'String', 'SPL (re 20
\muPa)','FontSize',22,'FontName','arial','FontWeight','Bold')
shading interp
axis equal
xlabel('X')%,'FontSize',20,'fontweight','bold');
ylabel('Y')%,'FontSize',20,'fontweight','bold');
zlabel('Z')%'FontSize',20,'fontweight','bold');
set(gca,'FontSize',20);
title([num2str(targetFreq) ' Hz '],'FontSize',12,'FontWeight','bold');
if (numHzIntegrate > 10)
    title(sprintf('Average Reduction = %3.1f
dB',AvgreductiondB),'FontSize',22,'FontWeight','bold');
else
    title(sprintf('Average Reduction = %3.1f
dB',AvgreductiondB),'FontSize',22,'FontWeight','bold');
end
hold off;
%% Makes the results able to plot in the next section
dataANCoff = dataANCoff;
dataANCon = dataANCon;

DAOff = 10.*log10(DAOff);
DAOn = 10.*log10(DAOn);
%%  Plot Sound Power Spectra of Both Measurements
figure;
% set(gcf,'Color',[1 1 1]);
semilogx(freq2,DAOff,'k-',freq2,DAOn,'r--')
set(gca,'FontName','arial');
% plot(freq2,dataANCoff)
xlabel('Frequency (Hz)')%,'FontSize',18,'fontweight','bold');
ylabel('Sound Power (re 10^-^1^2 W)')%,'FontSize',18,'fontweight','bold');
legend(sprintf('Power1,Overal Sound Power = %3.1f dB',AvdBOff),sprintf('Baffled Power1,Overal
Sound Power = %3.1f dB',AvdBOn));
%set(gca,'FontSize',18);
xlim([targetFreq-numHzIntegrate targetFreq+numHzIntegrate])
grid on
%% Plot Sound Power Spectra of First Measurements (Includes Overall SPL)
figure;
% set(gcf,'Color',[1 1 1]);
semilogx(freq2,DAOff,'b-')
set(gca,'FontName','arial');
% plot(freq2,dataANCoff)
% title(sprintf('Reduction vs. Frequency Integrated over space\nAverage Reduction = %3.1f
dB',AvgreductiondB),'FontSize',18,'FontWeight','bold');
xlabel('Frequency (Hz)')%,'FontSize',18,'fontweight','bold');
ylabel('Sound Power (re 10^-^1^2 W)')%,'FontSize',18,'fontweight','bold');
Title(sprintf('Overall Sound Power = %3.1f dB',AvdBOff));
legend('Measured Sound Power of Front Face','location','NorthWest')
%set(gca,'FontSize',18);
xlim([targetFreq-numHzIntegrate targetFreq+numHzIntegrate])
grid on
%%
cd('..');
```

A.3 Structural Mode Sound Power Calculation Code (MatLab)

The following code reads in the transverse velocity measurements from an SLDV and calculates the radiated sound power at a given frequency. The code generates radiation mode shapes, lambda values, and the grid spacing used during the SLDV test.

```
%------------------------------------------------------------------
%%Power of Radiation modes for Murata Piezoblower by Brad Solomon
%------------------------------------------------------------------

clear all;
set(0,'DefaultAxesFontName','Arial');
set(0,'DefaultAxesFontSize',16);
set(0,'DefaultAxesFontWeight','demi')
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8);
scrsz = get(0,'ScreenSize');
set(0,'DefaultFigurePosition',[scrsz(3)/5 scrsz(4)/7 scrsz(3)/1.5 scrsz(4)/1.5]);
%-----------------------------------------
% function to find distances
%-----------------------------------------
distance=@(x1,y1,x2,y2) sqrt((y2-y1)^2+(x2-x1)^2);
%-----------------------------------------
% function to find distances
%-----------------------------------------
sink=@(k,d) sin(k*d)/(k*d);

%-----------------------------------------
%read in all values (x,y) IF grid is rectagular
%-----------------------------------------
% filename='2_95.txt';
% filename='6_79.txt';
% filename='10_36.txt';
% filename='14_36.txt';
filename='25_39.txt';


range=[11 1 4105 1];
xx=dlmread(filename,'\t',range);
range=[11 2 4105 2];
yy=dlmread(filename,'\t',range);
range=[11 0 4105 0];
i=dlmread(filename,'\t',range);
range=[11 4 4105 4];
real=dlmread(filename,'\t',range);
range=[11 5 4105 5];
im=dlmread(filename,'\t',range);
Ve=real + im;
a=65;
b=63;
xx=xx+.004141;
yy=yy+0.009611;
x=reshape(xx,a*b,1);
y=reshape(yy,a*b,1);
figure
plot(xx,yy,'g.');hold on;
plot(xx(1),yy(1),'r.');hold on;
plot(xx(2),yy(2),'b.');hold on;
plot(xx(1+a),yy(1+a),'k.');hold on;
axis equal
title ('Grid used by SLDV')
%% Define Values
%-------------------------
tic
```

```matlab
r=zeros(length(x),length(x));
R=r;
% f=2.95e3;
% f=6.787e3;
% f=10.36e3;
% f=14.36e3;
f=25.39e3;
w=2*pi*f;
c=343;
k=w/c;
    %kb uses plate properties
kb=(1000*w^2*12*(1-.3^2)/(2e9*.000254))^.25;
rho=1.21;
    %area of each element [m^2]
Ae= distance(x(1),y(1),x(2),y(2))*distance(x(1),y(1),x(a+1),y(a+1));


%-------------------------------------------
% define the r (distance between points) matirx (160 seconds required)
%-------------------------------------------
toc
for i=1:length(x)
    for j=1:length(x)
        r(i,j)=distance(x(i),y(i),x(j),y(j));
    end
end
toc
%-----------------------
%distance check if needed
%-----------------------
% figure
% plot(xx,yy,'g.');hold on;
% plot(xx(1),yy(1),'r.');hold on;
% plot(xx(end),yy(end),'b.');hold on;
% title(['check distance ',num2str(r(1,length(x)))])
%-------------------------------------------
% define the R matrix (radiation resistance matrix) %16 seconds
%-------------------------------------------

for i=1:length(x)
    for j=1:length(x)
        R(i,j)=sink(k,r(i,j));
    end
    toc
end

for i=1:length(x)       % make the diagonal 'ones'
    R(i,i)=1;
end
toc
R=R*w^2*rho*Ae^2/(4*pi*c);
toc
%-------------------------------------------
% Pull out eigen vectors from R
%-------------------------------------------
[Q A]=eig(R);
toc
%-------------------------------------------
% calculate z matrixm (y in fahy equ 3.70)
%-------------------------------------------
Y=Q'*Ve;
toc
%-------------------------------------------
% Calculate the Power total and P(w)
%-------------------------------------------
Pt=Ve'*R*Ve;
for i=1:length(x)
P(i)=A(i,i)*abs(Y(i))^2;
end
toc
% % figure('Position',[100 100 800 800])
% % plot(x,y,'r.')
```

```
% % grid on
% % % % axis ([-.015 .015 -.015 .015])
% % % % figure('Position',[100 100 800 800])
% % hold on
% % plot(x(51),y(51),'b.')
% % % grid on
% % % axis ([-.015 .015 -.015 .015])

%----------------------------------
% mode shape plots
%----------------------------------
figure(1);
modes=[6 3];
for i=1:length(modes)
Qshape=reshape(Q(:,a*b+1-modes(i)),a,b);
m=x(1:a:a*b)*1000;
n=y(1:a)*1000;
subplot(2,1,i);surf(m,n,Qshape);axis tight;
title(['Radiation Mode Shape ',num2str(modes(i)),' (f = ',num2str(f),'Hz)'])
ylabel('[mm]');
xlabel('[mm]');
end
%----------------------------------
%lamda plot
%----------------------------------
for i=1:length(x)
    lam(i)=A(i,i);
end
Pm=lam'.*Y.*conj(Y);     %power radiated by each mode
Pm=Pm./max(Pm);
d=length(x);
for i=1:length(x)
    Pmode(i)=Pm(d);
    d=d-1;
end
lam=lam./lam(end);
    d=length(x);
for i=1:length(x)
    l(i)=lam(d);
    d=d-1;
end

figure(2)
num=.005;
plot(l(1:num*length(x)),'*:')
axis ([1 length(x)*num 0 1])
grid on
xlabel('Mode number')
ylabel('Normalized Contribution')
title('Normalized Lambda Values for Contribution Modes')

figure(3)
num=.005;% how many modes do i want to plot [0-1]
plot(abs(Pmode(1:num*length(x))),'o--')
axis ([1 length(x)*num 0 1])
grid on
xlabel('Mode number')
ylabel('Power Radiated [Normalized]')
title(['Total Radiated Power at ',num2str(f/1000),' kHz = ',num2str(10*log10(Pt/10^-12)),' dB
(ref 10^-^1^2 W/m^2)'])
toc
```

A.4 Directivity Plotting and Power Spectral Density Code (Matlab)

The method used to plot the directivity tests of Chapter 2 is shown below. The code can

plot the power spectral density of individual tests or plot various directivity tests for comparison.

106

```
%-------------------------------------------------------------------
% Notes: all files must have an included "Data" file with file label
% Defalut ID or code must be modified to match and names
% Works as is for ploting only the Polar graph
% Individual files should be names ID000_000.bin and so forth if you
% Can plot multiple polar graphs at once
%-------------------------------------------------------------------
clear all;
clc;
set(0,'DefaultFigurePosition',[-3          39          1280          916]);
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultAxesFontSize',20);
%-----------------------------------------------------------
%set number of tests that need to be ploted     NEEDS CHANGE
%-----------------------------------------------------------
numbertests=1;
%-------------------
%label file names here
%-------------------
testfile1='Normal Blower High Freq';
testfile2='No Nozzle';
testfile3='No Nozzle Side Holes';
testfile4='6 Side Holes';
testfile5='Open Face';

for test=1:numbertests

if test==1;testfile=testfile1;end;
if test==2;testfile=testfile2;end;
if test==3;testfile=testfile3;end;
if test==4;testfile=testfile4;end;
if test==5;testfile=testfile5;end;

pathname=['C:\Users\bigbrad\Desktop\Murata\Tests\Directivity Tests\',testfile,'\Data\'];
testname='ID';
ID=000:035;
fs=204800;                  %sample frequency
q=1;                        %index for each angle of OSPL

for m=1:length(ID);

        if ID(m)<10
            IDstring='00';
        elseif ID(m)<100
            IDstring='0';
        else
            IDstring='';
        end

        filename=[pathname,testname,IDstring,int2str(ID(m)),'_','000.bin'];

        fid=fopen(filename,'r');
        x=fread(fid,inf,'single');
        fclose(fid);

        x=x-mean(x);                %filter out any DC content

        N = 2^floor(log2(length(x)));   %number of samples to nearest lower power of 2
        x = x(1:N);                 % makes length a power of 2
        t = linspace(0,(N-1)/fs,N);
        ns = 2^14;                  % number of samples per block

        f = fs*(0:ns/2-1)/ns;       % freq. scale for ss fft.;
        df = f(2) - f(1);           % width of frequency bins

        ww = hann(ns);
                % generates the window.  .' is used to get the right orientation.
        %ww = ones(1,ns);           % used to mimic a rectangular window.
        W = mean(ww.*conj(ww));     % used to scale the psd
```

```
        blocks=zeros(2*N/ns-1,ns);
        blocks(1,:) = ww.*x(1:ns);
        for k = 2:2*N/ns-1
            blocks(k,:) = ww.*x((k-1)*ns/2:(k+1)*ns/2-1);
        end

        X = fft(blocks,ns,2);
        Xss = X(:,1:ns/2);                % Takes first ns/2 points to make it single sided
        Xss(:,2:length(Xss)) = 2*Xss(:,2:length(Xss));
                % Factor of 2 for frequencies > 0.
        mean_Xss_sq = mean(Xss.*conj(Xss),1)/2;
                % Finds the mean of the magnitude squared.
                                        % the /2 gives the correct rms value for a sine wave.
        Xssrms = sqrt(mean_Xss_sq);      % Finds the rms average of the single sided ffts.
        psd(q,:) = mean_Xss_sq/ns/fs/W;
                % See MATLAB help for periodogram algorithm. Units are (rms amp)^2/Hz

        indlow=min(find(f>26000));       % sets low and high frequencys desired
        indhigh=min(find(f>30000));

        OSPL(q) = 20*log10(sqrt(sum(psd(q,indlow:indhigh)*df))/2e-5);
                %Overall Sound Pressure Level OSPL

        SPL(q)=10*log10(sum(psd(q,indlow:indhigh)*df)/(20e-6)^2);
        d(q) = (q-1)*10;
        q=q+1;
end
A=sum(SPL)/length(SPL);
P=A+10*log10((20e-6)^2*4*pi*.5^2/(1e-12*415))
%--------------------------
%Power Spectrum Density Plot
%--------------------------
figure(1)
pcolor(f(indlow:indhigh),d,10*log10(psd(:,indlow:indhigh)/2e-5^2));
shading interp
set(gca,'xscale','log','Tickdir','out');
h=colorbar;
title(h,'dB re 20\muPa')
xlim([f(indlow),f(indhigh)])
set(gca,'YTick',0:60:360)
xlabel('Frequency')
ylabel('Angle (0 = on axis with nozzle)')
title('Power Spectrum Density Plot - 0.5 m ')
%----------------------------------
%Power Spectrum plot for one angle
%----------------------------------
figure(2)
semilogx(f(indlow:indhigh),10*log10(psd(1,indlow:indhigh)/2e-5^2));
xlabel('Frequency')
ylabel('Magnitude (dB)')
title('Power Spectrum')
xlim([f(indlow),f(indhigh)])
grid on
%--------------------
%3D Surf plot
%--------------------
figure(3)
surf(f(indlow:indhigh),d,10*log10(psd(:,indlow:indhigh)/2e-5^2));
shading interp
set(gca,'xscale','log','Tickdir','out');
colorbar
xlim([f(indlow),f(indhigh)])
set(gca,'YTick',0:60:360)
title('0.5 m (w/Large Muffler)')
%--------------------
%Polar Plot of OSPL
%--------------------
if test==1
figure
polar([d*2*pi/360 d(1)*2*pi/360],[OSPL OSPL(1)],'k')
title('OSPL from 0.5[m] in dB')
```

108

```
end
if test==2
hold on
polar([d*2*pi/360 d(1)*2*pi/360],[OSPL OSPL(1)],'r')
end
if test==3
hold on
polar([d*2*pi/360 d(1)*2*pi/360],[OSPL OSPL(1)],'b')
end
if test==4
hold on
polar([d*2*pi/360 d(1)*2*pi/360],[OSPL OSPL(1)],'g')
end
if test==5
hold on
polar([d*2*pi/360 d(1)*2*pi/360],[OSPL OSPL(1)],'m')
end
end
hold on
legend (testfile1,testfile2,testfile3,testfile4,testfile5,'location','BestOutside')
```

A.5 Frequency Spectrum Code (MatLab)

The code below is used to plot a frequency spectrum of a single microphone measurement. The standard deviation of the measure is also calculated.

```
%-------------------------------------------------------------------------
%------Plot a frequency spectrum and find the STD of the measured data----
%----------------------------by Brad Solomon--------------------------
clear all;
set(0,'DefaultFigurePosition',[3          150         1120          700]);
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',1.5);
set(0,'DefaultLineMarkerSize',8);
set(0,'DefaultAxesFontName','arial');
set(0,'DefaultAxesFontWeight','demi');
set(0,'DefaultAxesFontSize',16);
colvect=[0,0,0; 1,0,0; 0,0,1; 0,.8,0; .7,.3,.3;];  % black, red, blue, dark green, brown
set(0,'DefaultAxesColorOrder',colvect);
set(0,'DefaultAxesLineStyleOrder',{'-','--'});  %plots all solid lines, and then dashed

pathname='C:\Users\bigbrad\Documents\My Dropbox\Murata\Baffled Jet noise April 28\';
testname='ID';

% ID=[000,001,002,003,004];
ID=[000];
CH=[0];
fs = 60000;
q=1;  %index for combining all the files if needed

for m=1:length(ID);

    filename=[pathname,testname,sprintf('%03.0f',ID(m)),'_',sprintf('%03.0f',CH),'.bin'];

    fid=fopen(filename,'r');
    x=fread(fid,inf,'single');
    fclose(fid);

    x=x-mean(x);                        %filter out any DC content

        N = 2^floor(log2(length(x)));   %number of samples to nearest lower power of 2
        x = x(1:N);                     % makes length a power of 2
        t = linspace(0,(N-1)/fs,N);
        ns = 2^11;                      % number of samples per block
```

```
        f = fs*(0:ns/2-1)/ns;            % freq. scale for ss fft.;
        df = f(2) - f(1);                % width of frequency bins

        ww = hann(ns);                   % generates the window.   .' is used to get the right
orientation.
        %ww = ones(1,ns);                % used to mimic a rectangular window.
        W = mean(ww.*conj(ww));          % used to scale the psd

        blocks=zeros(2*N/ns-1,ns);
        blocks(1,:) = ww.*x(1:ns);
        for k = 2:2*N/ns-1
            blocks(k,:) = ww.*x((k-1)*ns/2:(k+1)*ns/2-1);
        end

        X = fft(blocks,ns,2);
        Xss = X(:,1:ns/2);               % Takes first ns/2 points to make it single sided
        Xss(:,2:length(Xss)) = 2*Xss(:,2:length(Xss));  % Factor of 2 for frequencies > 0.
        mean_Xss_sq = mean(Xss.*conj(Xss),1)/2;
                % Finds the mean of the magnitude squared.
                                         % the /2 gives the correct rms value for a sine wave.
        Xssrms = sqrt(mean_Xss_sq);      % Finds the rms average of the single sided ffts.
        psd(q,:) = mean_Xss_sq/ns/fs/W;
                % See MATLAB help for periodogram algorithm. Units are (rms amp)^2/Hz

        % Standard Deviation from measurement
        for i=1:length(Xss(:,1))
        Xss_sq=Xss(i,:).*conj(Xss(i,:))/2;
        XssrmsSTD = sqrt(Xss_sq);
        psdII(i,:)= Xss_sq/ns/fs/W;
        end
        for i=1:length(psdII(1,:))
        ans=std(10*log10(psdII(:,i)/2e-5^2));
        s(q,i)=ans;                      %s = standard deviation
        end

    indlow=min(find(f>1000));
    indhigh=min(find(f>20000));
    %indhigh=min(find(f==max(f)));

    OSPL(q) = 20*log10(sqrt(sum(psd(q,indlow:indhigh)*df))/2e-5);
    OASPLtd(q)=10*log10(mean(x.^2)/2e-5^2);

    d(q) = (q-1)*10;
    q=q+1;
end

%% STD plot
figure
    semilogx(f,s,'b')
    xlim([1000 20000])
    grid on
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    title ('STD')

%%
%%% Plot PSD vs Frequency
% aproximately 4.5mm from center line
%plots a single band with the std showing as well
figure
% hold on
    semilogx(f,10*log10(psd(1,:)/2e-5^2),'b')
    xlim([80 28000])
    ylim([0 100])
    grid on
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    title ('PSD at location 4.5mm from center line')
    hold on
    semilogx(f,10*log10(psd(1,:)/2e-5^2)+s(1,:),'k:')
    hold on
```

```
    semilogx(f,10*log10(psd(1,:)/2e-5^2)-s(1,:),'k:')

        % aproximately 1-2mm from center line
figure
    semilogx(f,10*log10(psd(2,:)/2e-5^2),':b')
    hold on
    semilogx(f,10*log10(psd(3,:)/2e-5^2),'-.r')
    xlim([80 28000])
    ylim([0 100])
    grid on
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    title ('PSD at location 1-2mm from center line')
    legend('Take 1','Take 2')

        % aproximately 2-3mm from center line
figure
    semilogx(f,10*log10(psd(4,:)/2e-5^2),':b')
    hold on
    semilogx(f,10*log10(psd(5,:)/2e-5^2),'-.r')
    xlim([80 28000])
    ylim([0 100])
    grid on
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    title ('PSD at location 2-3mm from center line')
    legend('Take 1','Take 2')

        % compare all three locations
figure
    semilogx(f,10*log10(psd(2,:)/2e-5^2),':b')
    hold on
    semilogx(f,10*log10(psd(4,:)/2e-5^2),'-.r')
    hold on
    semilogx(f,10*log10(psd(1,:)/2e-5^2),'--g')
    xlim([1000 20000])
    ylim([20 70])
    grid on
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    title ('PSD at specified locations [Jet Radiation](Center Axis of Mic located 3.35mm above
nozzle exit)')
%   legend('1-2mm from Center Line','2-3mm from Center Line','4-5mm from Center
Line','Location','NorthWest')
    legend('1-2mm from Center Line','2-3mm from Center Line','4-5mm from Center Line')
 hold on
    semilogx(f,10*log10(psd(2,:)/2e-5^2)+s(2,:),'k:')
    hold on
    semilogx(f,10*log10(psd(1,:)/2e-5^2)-s(1,:),'k:')

%%
figure
semilogx(f,10*log10(psd(1,:)/2e-5^2),'-k',...
    f,10*log10(psd(6,:)/2e-5^2),'--k',...
    f,10*log10(psd(10,:)/2e-5^2),'-b',...
    f,10*log10(psd(22,:)/2e-5^2),'--b')
xlabel('Frequency (Hz)')
ylabel('PSD (dB re 20 \muPa/\surdHz)')
xlim([1000 80000])
ylim([-20 100])
title('Generation 3')
legend('Nearfield Blower 1','Nearfield Blower 2',...
    'Farfield Blower 1','Farfield Blower 2')
grid on

%%
figure
semilogx(f,10*log10(psd(1,:)/2e-5^2),'-k',...
    f,10*log10(psd(6,:)/2e-5^2),'--k',...
    f,10*log10(psd(10,:)/2e-5^2),':k',...
    f,10*log10(psd(14,:)/2e-5^2),'-b',...
```

111

```
    f,10*log10(psd(18,:)/2e-5^2),'--b',...
    f,10*log10(psd(22,:)/2e-5^2),':b')%,...
%    f,10*log10(psd(8,:)/2e-5^2),'-r',...
%    f,10*log10(psd(9,:)/2e-5^2),'--r')
hold on
% semilogx(fa,10*log10(psdA(1,:)/2e-5^2),'-b')%,...
%    fa,10*log10(psdA(2,:)/2e-5^2),'--r',...
%    fa,10*log10(psdA(4,:)/2e-5^2),'-.r')
xlabel('Frequency (Hz)')
ylabel('PSD (dB re 20 \muPa/\surdHz)')
xlim([1000 80000])
ylim([-20 100])
title('Adina Model Comparison')
grid on
```

A.6 Results from Blower Load Measurement and Approximation

This section describes the assumptions and methods used to develop the approximate load applied to the blower model. At the onset of this research, the flow mechanism of piezoelectric blowers was not understood. A portion of this research also involved validating the theory that rapid oscillations can instigate a controlled flow of air. By replicating the dynamic motion of the piezoelectric ceramic in the blower model, the fluid dynamics of the flow generation could be better understood. Thus it is important to make accurate assumptions and measurements in applying the load to the model.

The first step in applying the load consists of understanding the dynamic motion of a blower pumping chamber. The rear side of the pumping chamber has attached the piezoelectric ceramic. This is exposed and can be seen on the back side of a Murata blower. While being driven at the specified frequency of 25.4 kHz, a blower is scanned by an SLDV to capture the displacement amplitude and phase of points on the ceramic.   The result of this measurement is presented in Figure 7-1. The figure shows the back side of a blower and the operating shape of the ceramic while being driven.
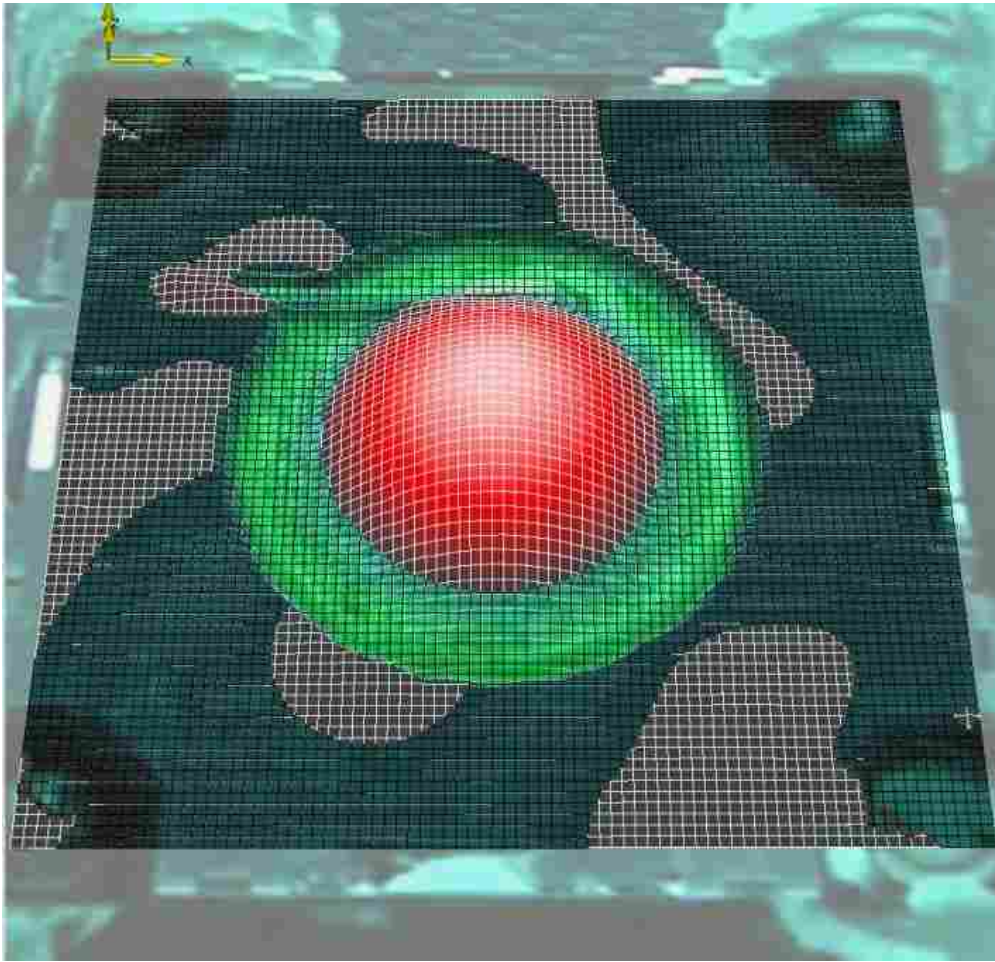
Figure 7-1: Dynamic motion of the piezoelectric ceramic.

The test shows how the ceramic has a simple parabolic operating shape that oscillates back and forth. It is this motion that results in the generation of air flow in a Murata blower. The corresponding velocity of each grid point was extracted from the test measurements. Now, an assumption is made that a piston like motion with a magnitude equal to the RMS value of the actual motion will result in a similar loading on the system. Thus, an ideal piston load is applied to the pumping chamber in the model. The amplitude is defined as the RMS value of the measured operating mode shape from the SLDV. This approximation makes modeling the load relatively simple and straight forward.

As seen in Chapter 3, the approximated load does in fact result in the generation of air flow. This loading assumption not only simplifies the model but provides concrete evidence of the flow mechanism in piezoelectric blowers. It shows that rapid oscillations of the pumping chamber do produce a controlled flow of air.

A.7 ADINA Model Build File

This is the exact .IN file used in this research. The file calls for two additional .IN files for the time function and parameters list. Those .IN files are given later in Appendix A.

```
DATABASE NEW SAVE=NO PROMPT=NO
FEPROGRAM ADINA
CONTROL FILEVERSION=V86

FEPROGRAM PROGRAM=ADINA-F

READ Parameters_for_blower2.in

MASTER ANALYSIS=TRANSIENT MODEX=EXECUTE TSTART=0.00000000000000,
     IDOF=10001 TURBULEN=NO HYDRO=YES STREAM=YES TRACTB=DEFAULT,
     IRINT=DEFAULT AUTOMATI=NO SOLVER=DEFAULT COMPRESS=SLI,
     FSINTERA=NO NMASS=0 MASSCOUP=NO MAP-OUTP=NONE MAP-FORM=NO,
     NONDIMEN=YES MAXSOLME=0 MTOTM=2 RECL=3000 ALE=NO THERMAL-=NO,
     UPWINDIN=CONTROL-VOLUME MESHUPDA=CURRENT MESHADAP=NO,
     COUPLING=ITERATIVE POROUS-C=NO CELL-BCD=YES VOF=NO FCBI=YES,
     TURB-ITE=COUPLED EM-MODEL=NO ALE-CURV=YES ENSIGHT-=FORMATTED

NONDIMENSION INPUT=YES OUTPUT=NO L=0.0000010000000000000,
     XREF=0.00000000000000 YREF=0.00000000000000,
     ZREF=0.00000000000000 V=100.000000000000 D=1.21000000000000,
     CP=1.00000000000000 T=1.00000000000000 TREF=0.00000000000000

MATERIAL CONSTF NAME=1 XMU=$Mu CP=0.00000000000000,
     XKCON=0.00000000000000 BETA=0.00000000000000 QB=0.00000000000000,
     RHO=$Rho TREF=0.00000000000000,
     GRAV-X=0.00000000000000 GRAV-Y=0.00000000000000,
     GRAV-Z=0.00000000000000 SIGMA=0.00000000000000,
     KAPPA=$Kappa CV=0.00000000000000 MDESCRIP='air'

EGROUP TWODFLUID NAME=1 SUBTYPE=AXISYMMETRIC MATERIAL=1 INT=3,
     RESULTS=STRESSES DEGEN=NO DISSP=NO SOLID=NO UPWINDIN=DEFAULT,
     OPTION=NONE FLOWTYPE=DEFAULT VOF-MATE=1 DESCRIP='NONE'

***choose the time function you want to use here-
TIMEFUNCTION NAME=1
@CLEAR
READ tf_30.in
@

TIMESTEP NAME=DEFAULT
@CLEAR
$NumTimeStep $TimeStep
@

COORDINATES POINT SYSTEM=0
@CLEAR
1 0.00000000000000 0.00000000000000 0.00000000000000 0
```

114

```
2 0.00000000000000 $HoleRad 0.00000000000000 0
3 0.00000000000000 $Rad 0.00000000000000 0
4 0.00000000000000 $Rad $Thickness 0
5 0.00000000000000 $HoleRad $Thickness 0
6 0.00000000000000 0.00000000000000 $Thickness 0
7 0.00000000000000 0.00000000000000 $PumpRoof 0
8 0.00000000000000 $HoleRad $PumpRoof 0
9 0.00000000000000 $OutRad $PumpRoof 0
10 0.00000000000000 $OutRad $OutBottom 0
11 0.00000000000000 $Rad $OutBottom 0
12 0.00000000000000 $Rad $PumpfloorTh 0
13 0.00000000000000 $PumpStep1Rad $PumpfloorTh 0
14 0.00000000000000 $PumpStep1Rad $PumpfloorStep1 0
15 0.00000000000000 $PumpStep2Rad $PumpfloorStep1 0
16 0.00000000000000 $PumpStep2Rad $PumpfloorStep2 0
17 0.00000000000000 $ConserveSpace $PumpfloorStep2 0
18 0.00000000000000 0.00000000000000 $NozzleH 0
19 0.00000000000000 $HalfRad $NozzleH 0
20 0.00000000000000 $NozzleRad $NozzleH 0
21 0.00000000000000 $NozzleRad $ChamferY 0
22 0.00000000000000 $NozzleRad $TopSpace 0
23 0.00000000000000 $ChamferX $TopSpace 0
24 0.00000000000000 $NozzleRad $PumpRoof 0
25 0.00000000000000 $PcX $PumpRoof 0
26 0.00000000000000 $OutRad $TopSpace 0
27 0.00000000000000 $SideSpace $TopSpace 0
28 0.00000000000000 $SideSpace $PumpRoof 0
29 0.00000000000000 $SideSpace $OutBottom 0
30 0.00000000000000 $SideSpace $BottomSpace 0
31 0.00000000000000 $OutRad $BottomSpace 0
32 0.00000000000000 $Rad $BottomSpace 0
33 0.00000000000000 $Inlet $BottomSpace 0
34 0.00000000000000 $PumpStep1Rad $PumpfloorStep2 0
35 0.00000000000000 $PumpStep1Rad $OutBottom 0
36 0.00000000000000 $PumpStep2Rad $OutBottom 0
37 0.00000000000000 $Rad $PumpfloorStep1 0
38 0.00000000000000 $Rad $PumpfloorStep2 0
39 0.00000000000000 $Inlet2 $BottomSpace 0
40 0.00000000000000 $HalfRad $ChamferY 0
41 0.00000000000000 $HalfRad $TopSpace 0
42 0.00000000000000 $ConserveSpace $BottomSpace 0
***Atmosphere build on
43 0.00000000000000 $Inlet $Bottom 0
44 0.00000000000000 $MaxRad $Bottom 0
45 0.00000000000000 $MaxRad $BodyStep2 0
46 0.00000000000000 $NozzleRad3 $BodyStep2 0
47 0.00000000000000 $NozzleRad3 $BodyStep1 0
48 0.00000000000000 $NozzleRad2 $BodyStep1 0
49 0.00000000000000 $NozzleRad2 $NozzleH 0
50 0.00000000000000 0.00000000000000 $AtmosphereH 0
51 0.00000000000000 $HalfRad $AtmosphereH 0
52 0.00000000000000 $NozzleRad $AtmosphereH 0
53 0.00000000000000 $NozzleRad2 $AtmosphereH 0
54 0.00000000000000 $NozzleRad3 $AtmosphereH 0
55 0.00000000000000 $MaxRad $AtmosphereH 0
56 0.00000000000000 $MaxRad $BodyStep1 0
57 0.00000000000000 $NozzleRad3 $NozzleH 0
58 0.00000000000000 $MaxRad $NozzleH 0
59 0.00000000000000 0.00000000000000 $ExtendH 0
60 0.00000000000000 $MaxRad $ExtendH 0
61 0.00000000000000 $HalfRad $ExtendH 0
62 0.00000000000000 $NozzleRad $ExtendH 0
63 0.00000000000000 $NozzleRad2 $ExtendH 0
64 0.00000000000000 $NozzleRad3 $ExtendH 0
65 0.00000000000000 $HalfRad $PumpRoof
@

SURFACE VERTEX NAME=1 P1=2 P2=3 P3=4 P4=5
SURFACE VERTEX NAME=2 P1=1 P2=2 P3=5 P4=6
SURFACE VERTEX NAME=3 P1=6 P2=5 P3=8 P4=7
SURFACE VERTEX NAME=4 P1=7 P2=65 P3=19 P4=18
```

```
SURFACE VERTEX NAME=5 P1=19 P2=20 P3=21 P4=40
SURFACE VERTEX NAME=6 P1=40 P2=21 P3=22 P4=41
SURFACE VERTEX NAME=7 P1=41 P2=22 P3=8 P4=65
SURFACE VERTEX NAME=8 P1=22 P2=23 P3=25 P4=8
SURFACE VERTEX NAME=9 P1=21 P2=22 P3=23 P4=21
SURFACE VERTEX NAME=10 P1=23 P2=25 P3=9 P4=26
SURFACE VERTEX NAME=11 P1=9 P2=28 P3=27 P4=26
SURFACE VERTEX NAME=12 P1=9 P2=10 P3=29 P4=28
SURFACE VERTEX NAME=13 P1=29 P2=30 P3=31 P4=10
SURFACE VERTEX NAME=14 P1=11 P2=10 P3=31 P4=32
SURFACE VERTEX NAME=15 P1=33 P2=32 P3=38 P4=34
SURFACE VERTEX NAME=16 P1=38 P2=11 P3=35 P4=34
SURFACE VERTEX NAME=17 P1=35 P2=11 P3=37 P4=14
SURFACE VERTEX NAME=18 P1=14 P2=37 P3=12 P4=13
SURFACE VERTEX NAME=19 P1=15 P2=14 P3=34 P4=16
SURFACE VERTEX NAME=20 P1=16 P2=39 P3=33 P4=34
SURFACE VERTEX NAME=21 P1=16 P2=17 P3=42 P4=39
***Atmosphere surfaces
SURFACE VERTEX NAME=22 P1=50 P2=51 P3=19 P4=18
SURFACE VERTEX NAME=23 P1=51 P2=52 P3=20 P4=19
SURFACE VERTEX NAME=24 P1=52 P2=53 P3=49 P4=20
SURFACE VERTEX NAME=25 P1=53 P2=49 P3=57 P4=54
SURFACE VERTEX NAME=26 P1=49 P2=57 P3=47 P4=48
SURFACE VERTEX NAME=27 P1=47 P2=56 P3=45 P4=46
SURFACE VERTEX NAME=28 P1=57 P2=58 P3=56 P4=47
SURFACE VERTEX NAME=29 P1=57 P2=58 P3=55 P4=54
SURFACE VERTEX NAME=30 P1=50 P2=59 P3=61 P4=51
SURFACE VERTEX NAME=31 P1=51 P2=61 P3=62 P4=52
SURFACE VERTEX NAME=32 P1=52 P2=62 P3=63 P4=53
SURFACE VERTEX NAME=33 P1=53 P2=63 P3=64 P4=54
SURFACE VERTEX NAME=34 P1=54 P2=64 P3=60 P4=55

SUBDIVIDE SURFACE NAME=1 MODE=LENGTH SIZE=$ElementLength
@CLEAR
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
22
23
24
@

SUBDIVIDE SURFACE NAME=27 MODE=LENGTH SIZE=$ElementLength2
@CLEAR
28
29
@

SUBDIVIDE LINE NAME=76 MODE=DIVISIONS NDIV=$NumDiv RATIO=0.200,
     PROGRESS=GEOMETRIC CBIAS=NO
@CLEAR
79
@

SUBDIVIDE LINE NAME=74 MODE=DIVISIONS NDIV=$NumDiv RATIO=5,
```

116

```
      PROGRESS=GEOMETRIC CBIAS=NO

SUBDIVIDE LINE NAME=78 MODE=LENGTH SIZE=$ElementLength2
@CLEAR
98
97
95
93
91
89
96
64
@

SUBDIVIDE LINE NAME=88 MODE=DIVISIONS NDIV=$NumDiv2 RATIO=5.000000000,
      PROGRESS=GEOMETRIC CBIAS=NO
@CLEAR
63
@

SUBDIVIDE LINE NAME=90 MODE=DIVISIONS NDIV=$NumDiv2 RATIO=0.20,
      PROGRESS=GEOMETRIC CBIAS=NO
@CLEAR
92
94
65
@

GSURFACE NODES=4 PATTERN=AUTOMATIC NCOINCID=BOUNDARIES NCEDGE=1234,
      NCVERTEX=1234 NCTOLERA=1.00000000000000E-05 SUBSTRUC=0 GROUP=1,
      PREFSHAP=TRIANGULAR MESHING=FREE-FORM SMOOTHIN=NO DEGENERA=NO,
      COLLAPSE=NO MIDNODES=CURVED METHOD=DELAUNAY FLIP=NO
@CLEAR
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
@

***NO SLIP WALL
BOUNDARY-CON WALL NAME=1 GTYPE=LINES SLIPC=0.00000000000000 MOVING=NO,
```

```
        VTYPE=CONVENTIONAL VT=0.00000000000000 NCURVT=0,
        DX=1.00000000000000 DY=0.00000000000000 DZ=0.00000000000000,
        X0=0.00000000000000 Y0=0.00000000000000 Z0=0.00000000000000,
        ALL-EXT=NO THERMAL=HEAT-FLUX TVALUE=0.00000000000000 NCURT=0
@CLEAR
2 0
3 0
8 0
16 0
*23 0
27 0
28 0
31 0
33 0
37 0
38 0
39 0
29 0
34 0
35 0
41 0
42 0
44 0
51 0
54 0
55 0
56 0
57 0
60 0
73 0
79 0
78 0
83 0
82 0
@

***AXIS OF SYMETRY
BOUNDARY-CON WALL NAME=2 GTYPE=LINES SLIPC=1.00000000000000 MOVING=NO,
        VTYPE=CONVENTIONAL VT=0.00000000000000 NCURVT=0,
        DX=1.00000000000000 DY=0.00000000000000 DZ=0.00000000000000,
        X0=0.00000000000000 Y0=0.00000000000000 Z0=0.00000000000000,
        ALL-EXT=NO THERMAL=HEAT-FLUX TVALUE=0.00000000000000 NCURT=0
@CLEAR
7 0
10 0
14 0
68 0
88 0
@

***LOAD PROPERTIES
LOAD VELOCITY NAME=1 VX=FREE VY=FREE VZ=$Velocity

APPLY-LOAD BODY=0
@CLEAR
1  'VELOCITY' 1  'LINE' 1 0 1 0.00000000000000 0 0.00000000000000,
        0.00000000000000 0
2  'VELOCITY' 1  'LINE' 5 0 1 0.00000000000000 0 0.00000000000000,
        0.00000000000000 0
3  'VELOCITY' 1  'LINE' 63 0 1 0.00000000000000 0 0.00000000000000,
        0.00000000000000 0
@

***Write output file -"f" needs to match script on fsl-
ADINA-F OPTIMIZE=SOLVER FILE='f.dat' FIXBOUND=YES,
        MIDNODE=NO OVERWRIT=YES FORMATTE=YES
DATABASE SAVE PERMFILE='f.idb' PROMPT=NO
*END SAVE=NO IMMEDIATE = NO
```

A.8 ADINA Time Function

The time function used for loading the blower model is expressed in Eq. (7.1) as an amplitude of velocity that is dependent on time and frequency. The frequency is defined as 25,000Hz. The function is plotted in Figure 7-2 and a corresponding .IN file of the time function is given at the end of this section.

$$V(t) = \sin(2\pi f t) \qquad (7.1)$$
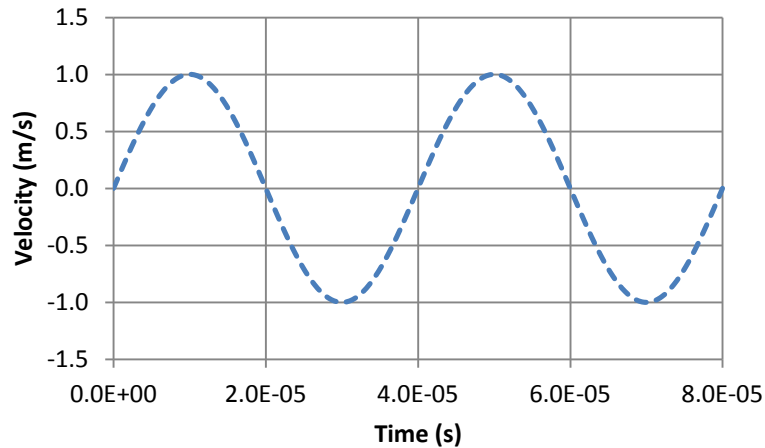
Figure 7-2: Velocity load time function used in the model.

```
***30 steps, Time Step = 2,  Example only
0.00000000000000 0.00000000000000
2.00000000000000E-06 0.309020000000000
4.00000000000000E-06 0.587790000000000
6.00000000000000E-06 0.809020000000000
8.00000000000000E-06 0.951060000000000
1.00000000000000E-05 1.00000000000000
1.20000000000000E-05 0.951060000000000
1.40000000000000E-05 0.809020000000000
1.60000000000000E-05 0.587790000000000
1.80000000000000E-05 0.309020000000000
2.00000000000000E-05 1.22510000000000E-16
2.20000000000000E-05 -0.309020000000000
2.40000000000000E-05 -0.587790000000000
2.60000000000000E-05 -0.809020000000000
2.80000000000000E-05 -0.951060000000000
3.00000000000000E-05 -1.00000000000000
3.20000000000000E-05 -0.951060000000000
3.40000000000000E-05 -0.809020000000000
3.60000000000000E-05 -0.587790000000000
3.80000000000000E-05 -0.309020000000000
4.00000000000000E-05 6.43150000000000E-16
```

```
4.20000000000000E-05 0.309020000000000
4.40000000000000E-05 0.587790000000000
4.60000000000000E-05 0.809020000000000
4.80000000000000E-05 0.951060000000000
5.00000000000000E-05 1.00000000000000
5.20000000000000E-05 0.951060000000000
5.40000000000000E-05 0.809020000000000
5.60000000000000E-05 0.587790000000000
5.80000000000000E-05 0.309020000000000
6.00000000000000E-05 -3.18520000000000E-15
```

A.9 ADINA Parameters List

This is the outline for the parameters list used in this thesis. A value can be changed and any dependant parameters in the model will automatically be changed as needed without user intervention.

```
***PARAMETERS FOR BLOWER DIMEMSIONS

***PUMP CHAMBER
PARAMETER Rad '6.9e-3'
PARAMETER HoleRad '.3e-3'
PARAMETER Thickness '.15e-3'
PARAMETER PcX '$HoleRad+.2e-3'
PARAMETER PcLX '$HoleRad/2'

***PUMP CHAMBER OUTLINE GEO
PARAMETER PumpRoofTh '.1e-3'
PARAMETER PumpRoof '$PumpRoofTh+$Thickness'
PARAMETER OutRad '8.5e-3'
PARAMETER OutBottom '-.35e-3'
PARAMETER PumpfloorTh '-.05e-3'
PARAMETER PumpfloorStep1 '-.2e-3'
PARAMETER PumpfloorStep2 '-.4e-3'
PARAMETER PumpStep1Rad '5.505e-3'
PARAMETER PumpStep2Rad '5.45e-3'

***NOZZLE/BODY GEO
PARAMETER NozzleH '2.55e-3'
PARAMETER NozzleRad '.41e-3'
PARAMETER HalfRad '$NozzleRad/2'
PARAMETER TopSpace '.65e-3'
PARAMETER Chamfer '.2e-3'
PARAMETER ChamferX '$NozzleRad+$Chamfer'
PARAMETER ChamferY '$TopSpace+$Chamfer'
PARAMETER SideSpace '9.0e-3'
PARAMETER BottomSpace '-.7e-3'
PARAMETER ConserveSpace '0'
PARAMETER Inlet '4e-3'
PARAMETER Inlet2 '$Inlet-($PumpStep1Rad-$PumpStep2Rad)'

***FLUID PROPERTIES
PARAMETER Mu '1.5E-5'
PARAMETER Rho '1.18'
PARAMETER Kappa '1.404E5'

***TIME PROPERTIES -How many steps to save-
PARAMETER NumTimeStep '30'
PARAMETER TimeStep '2e-6'

***MESH PROPERTIES
PARAMETER ElementLength '25E-6'
```

120

```
PARAMETER ElementLength2 '$ElementLength*10'
PARAMETER NumDiv '5'
PARAMETER NumDiv2 '26'
**PARAMETER NumDiv 'int(.59e-3/$ElementLength/5)'
**PARAMETER NumDiv2 'int(Extend/$ElementLength/5)'

***LOAD VALUES
PARAMETER Velocity '2.1'

***ATMOSPHERE DIMENSIONS -3/8 in above nozzle-
PARAMETER Bottom '-.9e-3'
PARAMETER MaxRad '4e-3'
PARAMETER BodyStep2 '.95e-3'
PARAMETER BodyStep1 '1.55e-3'
PARAMETER NozzleRad2 '1.41e-3'
PARAMETER NozzleRad3 '2.0e-3'
PARAMETER AtmosphereH '$NozzleH+4.5e-3'
PARAMETER ExtendH '$AtmosphereH+3.25e-3'
**note, 1/8"=3.175mm,1/16"=1.5875mm Highest is 9.525e-3m
```

A.10 FSL Script Files

The section contains two critical scripts used during this research. The following script is used to submit a batch job to the BYU Fulton Super Computer that will run a CFD ADINA .IN file:

```
#!/bin/bash
# -l Batch sysetem details
# -N Name of my output files
# -m Email options:
#       a sends email if job is aborted
#       b sends email when job begins
#       e sends email when job ends
# -M Specify Email address

#PBS -l nodes=1:ppn=12,pmem=2gb,walltime=220:00:00
#PBS -N June1
#PBS -m bae
#PBS -M snapkick.brad@gmail.com

# Execute fsi simulations.  Change 1GB for memory allocation as needed.
# Note that the -t 4 stands for 4 processors ("t" = threads).  Use -t 4
# on the linux machines since they have 2 dual-core processors.

PROG1="/fslapps/adina_8.6.2/tools/aui8.6"
PROGARGS1="-m 2GB -cmd -s Blower_atmosphere_f.in"
OUTFILE1="fmake.out"

PROG2="/fslapps/adina_8.6.2/tools/adina8.6"
PROGARGS3="-m 2GB -t 12 f"
OUTFILE3="frun.out"

# The following line changes to the directory that you submit your job from
cd $PBS_O_WORKDIR

$PROG1 $PROGARGS1 > $OUTFILE1
$PROG2 $PROGARGS3 > $OUTFILE3
```

The following script is used to submit Ensight batch files to the BYU Fulton Super Computer. The outputs of each job depend on the Python script which is presented in the next section of the Appendix.

```
#!/bin/bash
# -l Batch sysetem details
# -N Name of my output files
# -m Email options:
#       a sends email if job is aborted
#       b sends email when job begins
#       e sends email when job ends
# -M Specify Email address

#PBS -l nodes=1:ppn=12,pmem=2gb,walltime=0:30:00
#PBS -N June1-CL2
#PBS -m bae
#PBS -M snapkick.brad@gmail.com

# Execute fsi simulations.  Change 1GB for memory allocation as needed.
# Note that the -t 4 stands for 4 processors ("t" = threads).  Use -t 4
# on the linux machines since they have 2 dual-core processors.

cd $PBS_O_WORKDIR

/fslapps/ensight/bin/ensight90 -batch -X -p PythonCodeNameHere.py
```

A.11 Ensight Batch Script Tools

The follow scripts are written in python code. Each scripts function is described in section 3.2.4 of this thesis. These scripts are batched using the command script of the previous section. They can also, all be ran in one single batch if desired.

1. **The Velocity Line Tool**: (The x, y, x2, y2, and steps values can be changed as desired)

```
ensight.sendmesgoptions(version="9.03 (a)")
ensight.part.select_default()
ensight.part.modify_begin()
ensight.part.elt_representation("3D_feature_2D_full")
ensight.part.modify_end()
ensight.data.binary_files_are("big_endian")
ensight.data.format("case")
ensight.data.shift_time(1.000000,0.000000,0.000000)
ensight.data.replace("/fslhome/bigbrad/compute/thesisWork/june/1/f.case")
ensight.view_transf.view_recall("+X")
ensight.tools.line("ON")
ensight.view_transf.function("line")
x=0
y=2.55e-3
x2=0
y2=10.25e-3
steps=1500
ensight.view_transf.line(1,0.000000e+00,x,y)
ensight.view_transf.line(2,0.000000e+00,x2,y2)
```

```
for i in range(0,steps+1,2):
    ensight.solution_time.current_step(i)
    ensight.solution_time.update_to_current()
    ensight.variables.activate("velocity")
    ensight.part.select_begin(1)
    ensight.query_ent_var.begin()
    ensight.query_ent_var.description("")
    ensight.query_ent_var.query_type("generated")
    ensight.query_ent_var.number_of_sample_pts(80)
    ensight.query_ent_var.constrain("line_tool")
    ensight.query_ent_var.line_loc(1,0.000000,x,y)
    ensight.query_ent_var.line_loc(2,0.000000,x2,y2)
    ensight.query_ent_var.distance("arc_length")
    ensight.query_ent_var.variable_1("velocity")
    ensight.query_ent_var.generate_over("distance")
    ensight.query_ent_var.variable_2("DISTANCE")
    ensight.query_ent_var.end()
    ensight.query_ent_var.query()
    ensight.curve.select_begin(0)
    ensight.curve.save("formatted","CenterVel"+str(i))
    ensight.sendmesgoptions(version=0)
```

2. **The Pressure Line Tool**: (The y1, y2, x, steps, delta, and samples values can be changed as needed)

```
ensight.sendmesgoptions(version="9.03 (a)")
ensight.part.select_default()
ensight.part.modify_begin()
ensight.part.elt_representation("3D_feature_2D_full")
ensight.part.modify_end()
ensight.data.binary_files_are("big_endian")
ensight.data.format("case")
ensight.data.shift_time(1.000000,0.000000,0.000000)
ensight.data.replace("/fslhome/bigbrad/compute/thesisWork/convergence/50/f.case")
ensight.view_transf.view_recall("+X")
ensight.tools.line("ON")
ensight.view_transf.function("line")
y1=2.725e-3
y2=6.075e-3
x=1.2e-3
steps=50
delta=1
samples=20
ensight.view_transf.line(1,0.000000e+00,x,y1)
ensight.view_transf.line(2,0.000000e+00,x,y2)
for i in range(0,steps+1,delta):
    ensight.solution_time.current_step(i)
    ensight.solution_time.update_to_current()
    ensight.variables.activate("pressure")
    ensight.part.select_begin(1)
    ensight.query_ent_var.begin()
    ensight.query_ent_var.description("")
    ensight.query_ent_var.query_type("generated")
    ensight.query_ent_var.number_of_sample_pts(samples)
    ensight.query_ent_var.constrain("line_tool")
    ensight.query_ent_var.line_loc(1,0.000000,x,y1)
    ensight.query_ent_var.line_loc(2,0.000000,x,y2)
    ensight.query_ent_var.distance("arc_length")
    ensight.query_ent_var.variable_1("pressure")
    ensight.query_ent_var.generate_over("distance")
    ensight.query_ent_var.variable_2("DISTANCE")
    ensight.query_ent_var.end()
    ensight.query_ent_var.query()
    ensight.curve.select_begin(0)
    ensight.curve.save("formatted","Press"+str(i))
    ensight.sendmesgoptions(version=0)
```

3. **The Picture Tool**:

```
ensight.text.select_default()
ensight.legend.select_palette_begin("velocity")
ensight.function.palette("velocity")
ensight.function.range(0,25)
ensight.function.modify_begin()
ensight.function.number_of_levels(7)
ensight.function.edit_level(2)
ensight.function.value(4.166667e+00)
ensight.function.edit_level(3)
ensight.function.value(8.333333e+00)
ensight.function.edit_level(4)
ensight.function.value(1.250000e+01)
ensight.function.edit_level(5)
ensight.function.value(1.666667e+01)
ensight.function.edit_level(6)
ensight.function.value(2.083333e+01)
ensight.function.edit_level(7)
ensight.function.value(2.500000e+01)
ensight.function.edit_level(2)
ensight.function.rgb(0.0000e+00,6.6667e-01,1.0000e+00)
ensight.function.edit_level(3)
ensight.function.rgb(0.0000e+00,1.0000e+00,6.6667e-01)
ensight.function.edit_level(4)
ensight.function.rgb(0.0000e+00,1.0000e+00,0.0000e+00)
ensight.function.edit_level(5)
ensight.function.rgb(6.6667e-01,1.0000e+00,0.0000e+00)
ensight.function.edit_level(6)
ensight.function.rgb(1.0000e+00,6.6667e-01,0.0000e+00)
ensight.function.edit_level(7)
ensight.function.rgb(1.0000e+00,0.0000e+00,0.0000e+00)
ensight.function.modify_end()
ensight.sendmesgoptions(version=0)

ensight.file.image_format("jpg")
ensight.file.image_format_options("Quality 75")
ensight.anim_recorders.render_offscreen("ON")
ensight.file.image_numpasses(4)
ensight.file.image_stereo("current")
ensight.file.image_screen_tiling(1,1)
ensight.file.image_file("pictureATM1")
ensight.file.save_image()
ensight.sendmesgoptions(version=0)
```

4. **The Movie Tool**:

```
ensight.sendmesgoptions(version="9.03 (a)")
ensight.part.select_default()
ensight.part.modify_begin()
ensight.part.elt_representation("3D_feature_2D_full")
ensight.part.modify_end()
ensight.data.binary_files_are("big_endian")
ensight.data.format("case")
ensight.data.shift_time(1.000000,0.000000,0.000000)
ensight.data.replace("/fslhome/bigbrad/compute/thesisWork/june/1/f.case")
ensight.view_transf.axis("y")
ensight.view_transf.rotate(0.000000e+00,-9.000000e+01,0.000000e+00)
ensight.frame.select_begin(0)
ensight.view_transf.axis("z")
ensight.view_transf.rotate(0.000000e+00,0.000000e+00,-9.000000e+01)
ensight.view_transf.zoom(8.100000e-01)
ensight.variables.activate("velocity")
ensight.part.select_begin(1)
ensight.part.modify_begin()
ensight.part.colorby_palette("velocity")
ensight.part.modify_end()
ensight.clip.select_default()
ensight.part.modify_begin()
```

```
ensight.clip.mesh_plane("Z")
ensight.clip.tool("xyz")
ensight.part.modify_end()
ensight.clip.select_default()
ensight.part.modify_begin()
ensight.clip.value(0.002550)
ensight.clip.tool("xyz")
ensight.part.modify_end()
ensight.part.select_begin(1)
ensight.clip.begin()
ensight.clip.value(0.002550)
ensight.clip.domain("intersect")
ensight.clip.tool("xyz")
ensight.clip.end()
ensight.clip.create()
ensight.part.select_begin(4)
ensight.part.select_begin(4)
ensight.vctarrow.begin()
ensight.vctarrow.variable("velocity")
ensight.vctarrow.end()
ensight.vctarrow.create()
ensight.part.select_begin(5)
ensight.part.select_begin(5)
ensight.part.modify_begin()
ensight.part.colorby_palette("velocity")
ensight.part.modify_end()
ensight.view_transf.function("global")
ensight.legend.format("%.2f")
ensight.legend.format("%.2f")
ensight.legend.format("%.2f")
ensight.legend.select_palette_begin("velocity")
ensight.legend.format("%.2f")
ensight.legend.width(2.000000e-02)
ensight.legend.height(2.500000e-01)
ensight.legend.title_name("Velocity [m/s]")
ensight.legend.location_x(8.000000e-01)
ensight.legend.location_y(2.500000e-01)
ensight.view_transf.function("global")
ensight.solution_time.time_annotation("ON")
ensight.text.select_begin(0)
ensight.legend.select_default()
ensight.text.location_x(6.350000e-01)
ensight.text.location_y(7.352201e-01)

ensight.text.select_default()
ensight.legend.select_palette_begin("velocity")
ensight.function.palette("velocity")
ensight.function.range(0,25)
ensight.function.modify_begin()
ensight.function.number_of_levels(7)
ensight.function.edit_level(2)
ensight.function.value(4.166667e+00)
ensight.function.edit_level(3)
ensight.function.value(8.333333e+00)
ensight.function.edit_level(4)
ensight.function.value(1.250000e+01)
ensight.function.edit_level(5)
ensight.function.value(1.666667e+01)
ensight.function.edit_level(6)
ensight.function.value(2.083333e+01)
ensight.function.edit_level(7)
ensight.function.value(2.500000e+01)
ensight.function.edit_level(2)
ensight.function.rgb(0.0000e+00,6.6667e-01,1.0000e+00)
ensight.function.edit_level(3)
ensight.function.rgb(0.0000e+00,1.0000e+00,6.6667e-01)
ensight.function.edit_level(4)
ensight.function.rgb(0.0000e+00,1.0000e+00,0.0000e+00)
ensight.function.edit_level(5)
ensight.function.rgb(6.6667e-01,1.0000e+00,0.0000e+00)
ensight.function.edit_level(6)
```

```
        ensight.function.rgb(1.0000e+00,6.6667e-01,0.0000e+00)
        ensight.function.edit_level(7)
        ensight.function.rgb(1.0000e+00,0.0000e+00,0.0000e+00)
        ensight.function.modify_end()
        ensight.sendmesgoptions(version=0)


        ensight.file.image_format("avi")
        ensight.file.image_format_options("Compression MJPEG BitRate 8000")
        ensight.anim_recorders.render_offscreen("ON")
        ensight.file.image_numpasses(4)
        ensight.file.image_stereo("current")
        ensight.file.image_screen_tiling(1,1)
        ensight.file.animation_file("June1")
        ensight.file.animation_window_size("NTSC")
        ensight.file.animation_window_xy(704,480)
        ensight.file.animation_frames(1501)
        ensight.file.animation_multiple_images("OFF")
        ensight.file.animation_play_flipbook("OFF")
        ensight.file.animation_play_time("ON")
        ensight.file.animation_reset_flipbook("OFF")
        ensight.file.animation_reset_traces("OFF")
        ensight.file.animation_reset_time("ON")
        ensight.file.save_animation()
        ensight.sendmesgoptions(version=0)
```

A.12 Velocity Profile of Model Code (MatLab)

This code is used to post process the text files generated by the Velocity Line tool in the previous Appendix. It will plot velocity vs. time of a single point and the average velocity profile. It calculates an exact flow rate and it will generate a theoretical velocity profile based on a given centerline velocity.

```
%%
%-------------------------------------------------------
%---Plot Velocity profile w Line tool.py file outputs---
%------------------Brad Solomon-----------------------
clear all;
start = 6;
numpoints = 80;      %change number of subdivisions on line
steps = 1500;        %change number of time step files
dt=1e-6;             %time step from file
Deltasteps = 5;      %The number of samples taken in ensight
stop=numpoints*5+5;
range=[start,0,stop,0];
%----------------------------
%----Pullout velocity and radial points
%----------------------------
for step=1:Deltasteps:steps
file=zeros;
% file=dlmread(['VelatNoz',num2str(step-1)],'\t',range);
% file=dlmread(['VelatTop',num2str(step-1)],'\t',range);
% file=dlmread(['VelatTime',num2str(step-1)],'\t',range);
file=dlmread(['Vel',num2str(step-1)],'\t',range);
    v=1;
    for i=1:5:length(file)
        vel(step,v)=file(i);
        v=v+1;
    end
end
```

```
    x=1;
    for i=2:5:length(file)
        rad(x)=file(i);
        x=x+1;
    end
    rad=rad*10^3;    %changes to um from mm

%%
%-----------------------------------------
%-----Plot Velocity vs Time of single node-
%-----------------------------------------
set(0,'DefaultAxesFontName','Arial');
set(0,'DefaultAxesFontSize',16);
set(0,'DefaultAxesFontWeight','demi')
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8);
scrsz = get(0,'ScreenSize');
set(0,'DefaultFigurePosition',[scrsz(3)/5 scrsz(4)/7 scrsz(3)/1.5 scrsz(4)/1.5]);

z=1;
velshort=zeros;
for y=1:length(vel)
    if vel(y,1)>0
        velshort(z)=vel(y,1);
        z=z+1;
    end
end
time=0:steps/length(velshort):steps-1;
time=time*dt;
figure
plot(time*1000,velshort,'k--')
xlabel('Time [ms]')
ylabel('Velocity [m/s]')
title('Velocity Vs Time at Center Line and 2.0mm above Nozzle Exit')
grid on
%%
%---average profile calc----
for i=1:length(rad)
    %you need to decide where to start taking it based on the above time
    %plot. check for time comvergence.
avel(i)=mean(vel(1001:Deltasteps:end,i));
% avel(i)=mean(vel(201:Deltasteps:end,i));
end
%%
%---------------------------------
%-----Plot Average profile---------b,k,r,g
%---------------------------------
% figure
hold on
plot(rad,avel,'b--')
xlabel('Distance from CL [mm]')
ylabel('Velocity [m/s]')
title('Title Here')
grid on
legend('legend here')
%-------------------------------
%find Volume flow rate for profile
%-------------------------------
X=rad/1000;
Y=X.*avel;
Vdot=2*pi*trapz(X,Y);
Vdot=Vdot/.001*60   %units now L/min
%%
%-------------------------------------------------------
%-----Find and plottheoretical Velocity Profile (u)-----
%-------------------------------------------------------
X=2.5e-3;       %distance from orgin of jet along line of flow [m](includes Xo=.5mm)
ro=.41e-3;      %radius of nozzle [m]
% Uo=27;        %flow velocity at nozzle [m/s]
% Um=12*ro/X*Uo;%CenterLine Velocity
```

127

```
Um=27;            %CenerLine Velocity at X location if known
u=Um*exp(-94*(rad*10^-3/X).^2);
hold on
plot(rad,u,'k--')


%-------laminar theoretical profile----
Um= 27;            %center line velocity
X=2.5e-3;          %distance from orgin of jet along line of flow [m]
v=1.8e-5/1.2;      %kinematic viscosity
Mo=Um*8*pi*v*X/3;
E=(3/(16*pi))^.5 *Mo^.5*rad*10^-3/(v*X);
u=(3*Mo)/(8*pi*v*X)./(1+(E.^2)/4).^2;
hold on
plot(rad,u,'k:')
```

A.13 Overall Sound Pressure Level of Model Code (MatLab)

The following code takes the pressure vs. time data from the tool shown in Appendix A.11 and calculates the overall sound pressure level across a given bandwidth. The code will also plot a frequency spectrum for said bandwidth.

```
%----------------------------------------
%------Find OSPL from Model results---------
%------------By Brad Solomon---------------
clear all;
numplots=1;                        %change for number of plots wanted
x1 = dlmread('PressureEdit');
t = x1(:,2);
x(:,1) = x1(:,1);
x(:,1)=x(:,1)-mean(x(:,1));

%----number of samples to nearest lower power of 2----
N = 2^floor(log2(length(x)));
% N=1395;

%-----makes length a power of 2 chops off the begining portion of sample---
for a=1:numplots
xc(:,a) = x(length(x)-N+1:length(x),a);
end

%------Set up time array-----
dt = t(2)-t(1);
fs = 1/dt;
% t=0:dt:dt*(length(x)-1);

%----number of samples per block----
% ns = 2^9;
ns = N;

%---PSD calc---
for a=1:numplots
[psd(a,:) F(:,a)]= pwelch(xc(:,a),hann(ns),[],[],fs);
end
df=F(2,1)-F(1,1);
%---OSPL---
    for a=1:numplots
     indlow=min(find(F(:,a)>1000));       %low frequency
     indhigh=min(find(F(:,a)>20000));     %high frequency
     OSPL(a) = 20*log10(sqrt(sum(psd(a,indlow:indhigh)*df))/2e-5);
    end

%% Plot time data
```

128

```
for a=1:numplots
    figure
    plot(t,x(:,a),'r:')
    xlabel('Time (s)')
    ylabel('Pressure (Pa)');
    grid on
    axis manual
    hold on
    plot(t(length(x)-N+1:length(x)),xc(:,a),'b')
end

%% Plot Frequency Spectrum
%--adina data---
for index=1:numplots
    figure
% subplot(2,1,1)
%     hold on
    semilogx(F(:,index),10*log10(psd(index,:)/2e-5^2),'--b')
    xlabel('Frequency (Hz)')
    ylabel('PSD (dB re 20 \muPa/\surdHz)')
    xlim([1000,20000])
    ylim([0 60])
%     title(['Angle: ',int2str((index-1)*10),'   OASPL: ',num2str(OSPL(index),'%3.1f'),' dB'])
    grid on
end
```

A.14 PIV Measurement Setup and Results

This section describes the details of measuring the velocity profile of a blower with PIV.[33] Experimental results are needed to provide a benchmark for the predicted values from the model. With the small size of the blower it was determined that the best method for experimental measurement of the velocity profile would be Particle Image Velocimetry (PIV) testing. PIV testing produced more consistent results than other testing methods and included more information about the velocity field than most other testing methods. See this reference to gain a better understanding of the fundamentals of PIV testing.

Photographs of the PIV test setup are shown in Figure 7-3 and Figure 7-4. The PIV tests are performed using a New Wave Research Solo II laser in conjunction with a cylindrical lens, creating a 1mm thick laser sheet. The laser is a double pulsed Nd:YAG laser. The camera is a high speed LaVision Imager Intense. A pressure source, hooked up to a LaVision VZ particle seeder that atomizes olive oil droplets, feeds particles to the blower. There was an open hole in

the seeding pipe located just before the blower inlet to mitigate back pressure loading on the blower. The data was processed using the DaVis software package.

The image in Figure 7-5 was taken during a PIV test and shows some of the seeding particles in motion. The test resulted in the velocity vector field shown in Figure 7-6. While Figure 7-7 shows the corresponding velocity magnitudes. The noise present in the figures near the top and bottom right hand side of the grid is not part of the experimental data. The results of the test are used as a reference for the blower model.
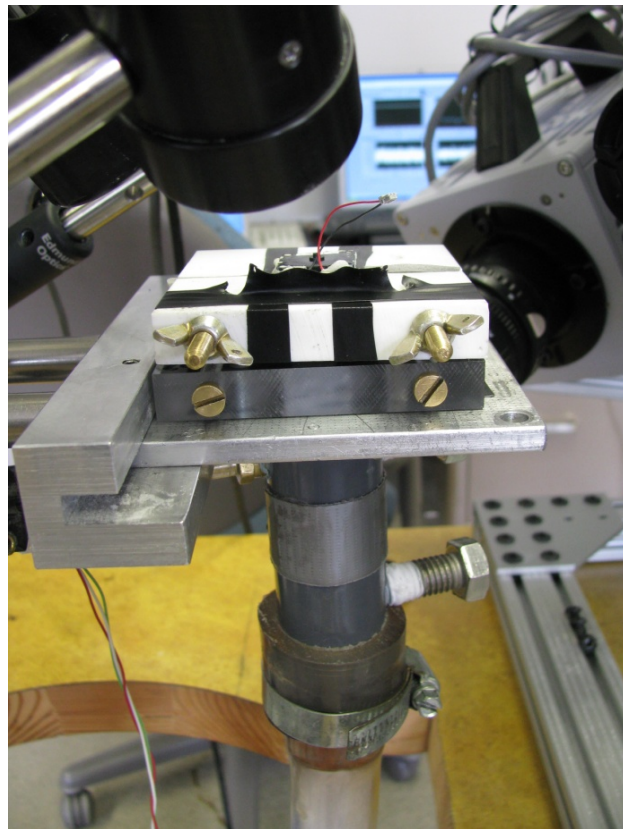
Figure 7-3: Mounted blower with attached seeding tube for PIV testing.
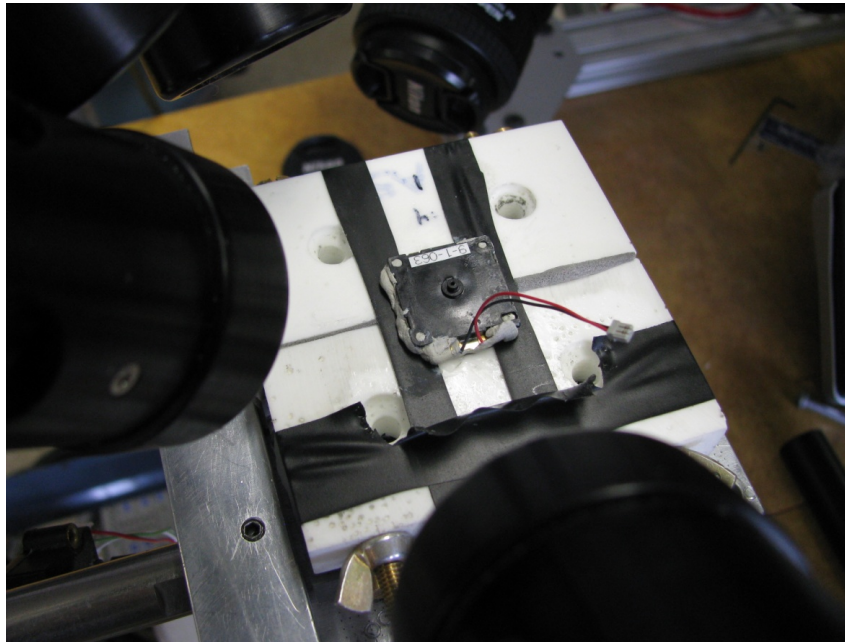
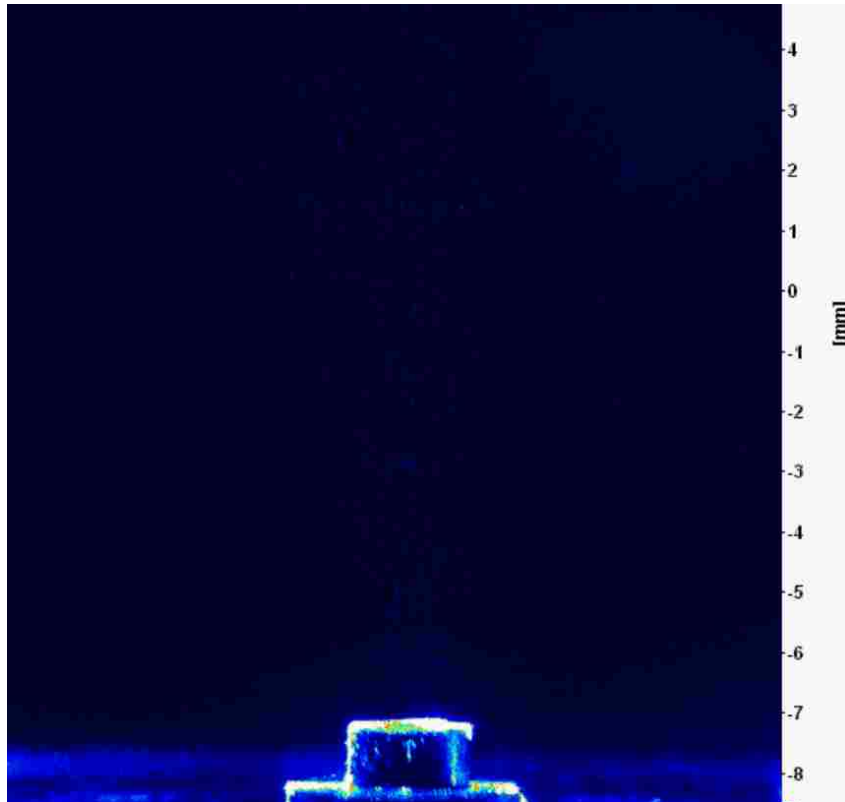**Figure 7-4: The top view of the blower mounted for PIV testing.**

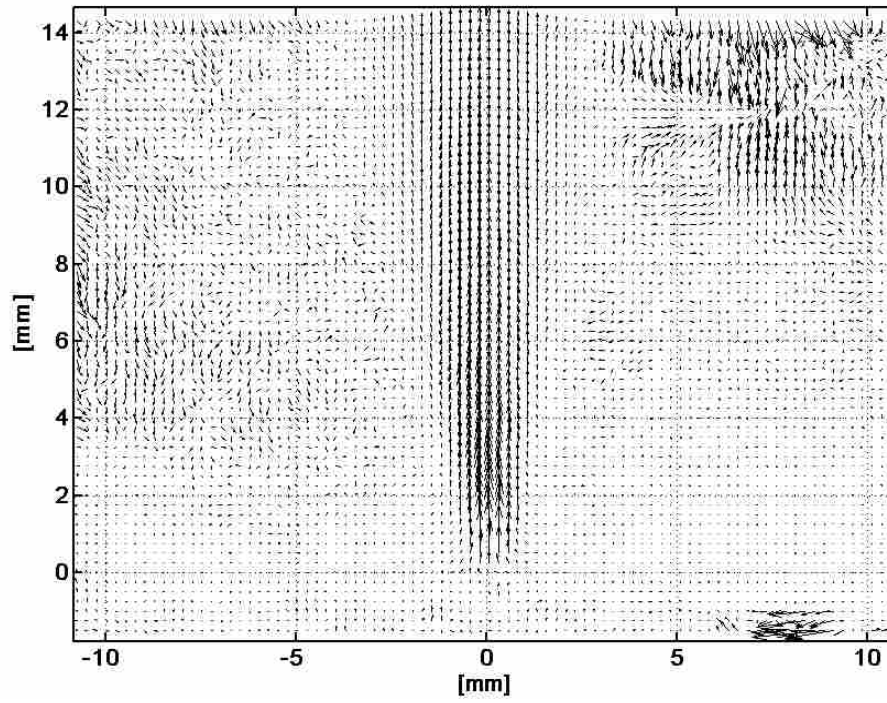**Figure 7-5: Snap shot during PIV test of particle seeds.**

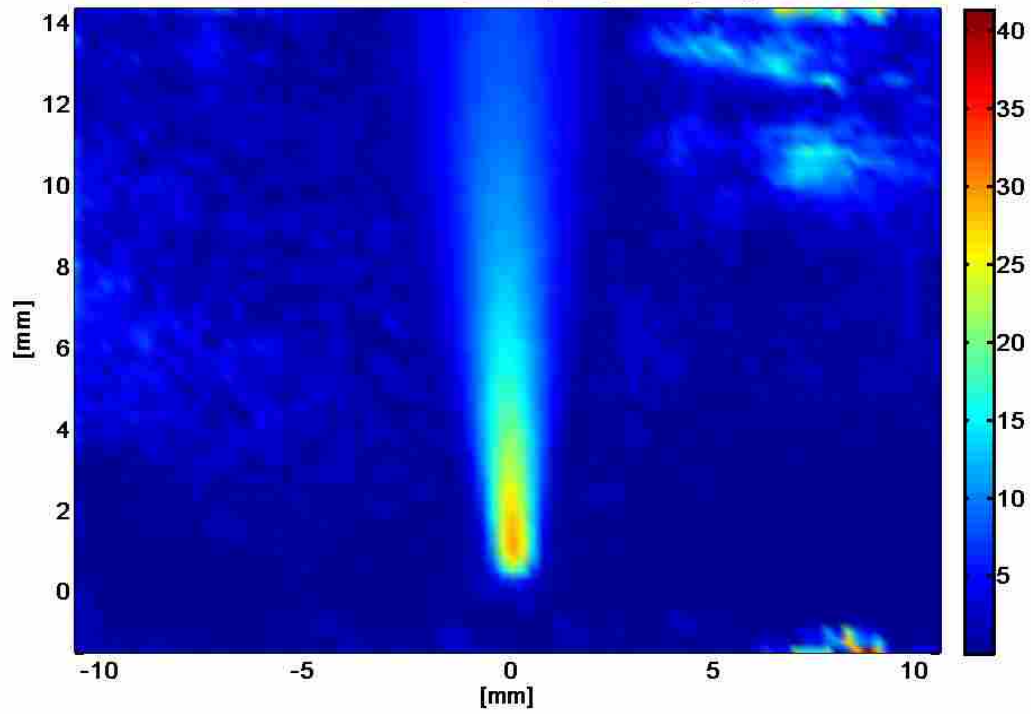**Figure 7-6: Velocity vector field of piezoelectric blower from PIV testing.**

Figure 7-7: Velocity magnitude of the flow field from PIV testing.

A.15 Directivity Testing with Mufflers

This section describes the test procedure used to measure the directivity of a blower with a muffler attached and shows the results in a table. Figure 7-8 shows the testing setup.
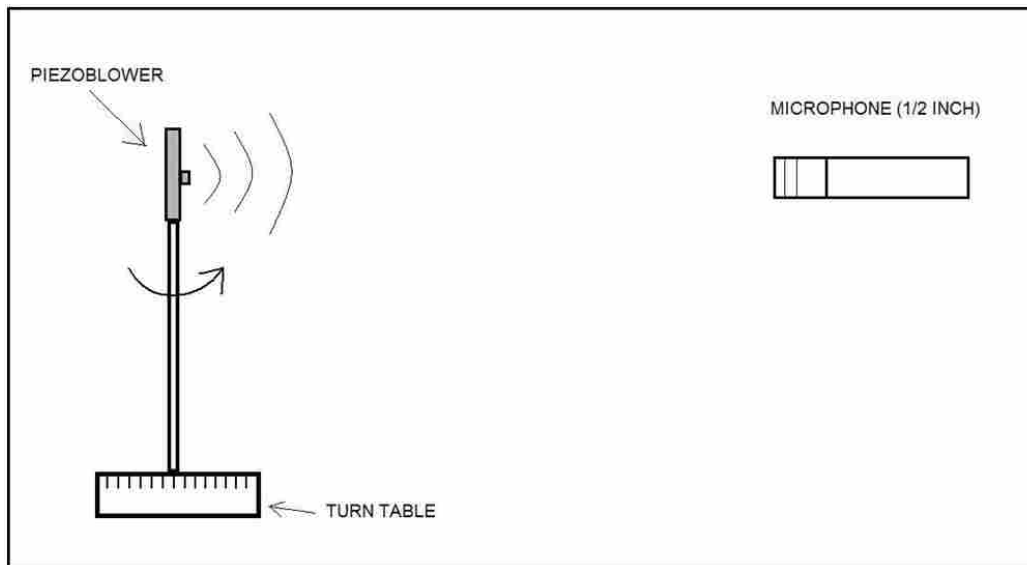

Figure 7-8: Directivity testing setup.

The blower is mounted on a rod connected to a turntable. The direction of air flow is parallel to the ground. Rotating the turntable allowed the blower to rotate also. A stationary ½ inch microphone was placed 1 meter from the blower nozzle in an anechoic chamber. The nozzle of the blower and the microphone are positioned at the same height. The setup allows for measurements to be recorded at the desired angle by turning the table to the desired position.

Data was sampled at 50 kHz for 10 seconds at 10 degree intervals. The overall sound pressure level across the audible bandwidth was calculated for each 10 second recording. A polar plot was generated showing overall sound pressure level changes for three cases: a normal blower, and blower with a thin muffler, and a blower with a thicker muffler. The mufflers were attached with adhesive tape to ensure a proper seal. Figure 7-9 shows how the mufflers were attached and Figure 7-10 shows the resulting polar plots.

133

Figure 7-9: Two separate mufflers attached to two separate piezoelectric blowers.
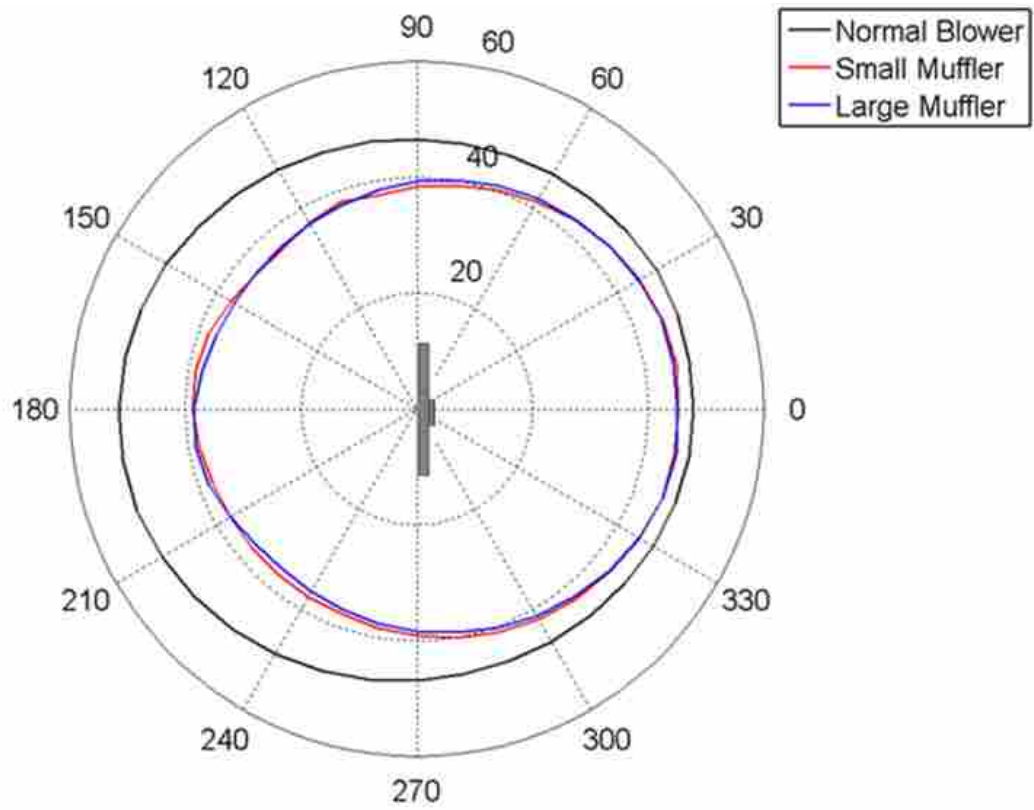

Figure 7-10: Directivity plots of a normal blower compared to two muffler cases.

The radial axis is dB and the 0 angle is normal to the front face of the blower. The percentage of change in flow rate was also measured for the three cases. Some of these results are summarized in Table 7-1.

Table 7-1: Summary of muffler results.

Description	Front (0°) (dB)	Change in dB	Back (180°) (dB)	Change in dB	Change in Flow Rate (%)
Normal	47.85	0	51.55	0	0
Large Muffler	45.06	-2.79	38.48	-13.07	-1.8
Small Muffler	45.2	-2.65	38.74	-12.81	-2

The results show that the mufflers attenuate a significant amount of noise on the back side of the blower and not affect the flow rate significantly.