Experimental Analysis of an Advanced-Design, Quasi-Radial Diffuser Receiving an Oscillating Jet

by

Andrew R. Mathes

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
In partial fulfillment of the requirements for the degree of
Master of Science
in
Engineering Mechanics

Demetri P. Telionis, Chair
Muhammad Hajj
Zafer Gurdal

July, 1997
Blacksburg, Virginia
Experimental Analysis of an Advanced-Design, Quasi-Radial Diffuser Receiving an Oscillating Jet

by

Andrew R. Mathes

Demetri P. Telionis, Chairman

Engineering Mechanics

(ABSTRACT)

Methods of experimental analysis of airflow through an advanced-design, quasi-radial diffuser receiving an oscillating jet induced by a rotating baffle were devised and performed. The initially unsteady flow was found to lose its unsteadiness as it progressed through the diffuser and exited into the atmosphere. Within the diffuser, regions of slow flow in the lee of ventilation holes exist due to entrainment. The addition of a convergence insert across from these holes increased flow speeds, but also deflected the flow farther away from the holes, aiding separation. The predominantly steady flow over a flat plate upon which the exiting flow impinged produced an asymmetric static pressure profile. This was probably due to the mechanism that set up the oscillations in the flow, the rotating baffle.
Acknowledgements

The author would like to express his gratitude to his committee chairman and graduate advisor, Dr. Demetri P. Telionis, for his guidance, support, and patience. His encouragement helped calm many frustrations along the way and provided the author with the resolve to push forward. The author is also particularly grateful to Drs. Muhammad Hajj and Zafer Gürdal for their serving on his graduate committee and for their suggestions and constructive criticism regarding this work.

Very special thanks are extended to Norman Schaeffler, for his computer programming contributions to this work as well as his many helpful suggestions throughout. Also thanks to Bill Shaver and Archie Montgomery for their work with various machining of parts and construction of rigs throughout this work.

The author would finally like to acknowledge the generous financial and scientific support of the Dupont corporation and in particular, Mike Davis and Mike Moore, without whose help this work would not have been possible.
# Table of Contents

1. **Introduction and Literature Review** ................................................................. 1  
   1.1 Simple Diffuser Flow .................................................................................... 2  
   1.2 The Apparatus Studied in this Work .......................................................... 2  
2. **The Experimental Programme** ....................................................................... 5  
   2.1 Measuring Instruments .................................................................................. 5  
   2.1.1 Pressure Transducers ................................................................................ 5  
   2.1.2 The Pitot Tube .......................................................................................... 6  
   2.1.3 The Hot-wire Anemometer ....................................................................... 7  
   2.2 Computer & Peripherals ............................................................................... 7  
   2.3 The Radial Diffuser ...................................................................................... 8  
   2.3.1 The Jet Nozzle .......................................................................................... 10  
   2.3.2 The Rotating Baffle ............................................................................... 11  
   2.3.3 Front and Back Plates ............................................................................. 12  
   2.4 Instrumentation Rigs .................................................................................... 13  
   2.4.1 Intra-Diffuser Measurement Setup .......................................................... 13  
   2.4.2 The Impingement Plate ........................................................................... 14  
   2.4.3 Unsteady Measurement Setup ................................................................. 16  
   2.5 Experimental Procedure ............................................................................. 16  
   2.5.1 Intra-Diffuser Measurements .................................................................. 16  
   2.5.2 Impingement Plate Measurements ......................................................... 16  
3. **Discussion of Results** .................................................................................... 18  
   3.1 Intra-Diffuser Measurements ....................................................................... 18  
   3.1.1 Velocity Profiles ...................................................................................... 18  
   3.1.2 Animation of Intra-Diffuser Velocities .................................................... 32  
   3.2 Impingement Plate Measurements ............................................................... 34  
4. **Conclusions and Recommendations** ............................................................. 39  
5. **References** ..................................................................................................... 41  
6. **Appendices** .................................................................................................... 42  
   6.1 Basic Pressure Transducer Data Acquisition .............................................. 42  
   6.2 Pressure Transducer Data Acquisition with Probe Positioning and Ensemble Delineation 48  
   6.3 Post-Processing of Data .............................................................................. 56  
7. **Vita** ................................................................................................................ 58
List of Multimedia Objects

Figure 1  Schematic diagram of quasi-radial diffuser ............................................................. 3
Figure 2  Pitot tube schematic ................................................................................................ 6
Figure 3  Hardware setup schematic ...................................................................................... 8
Figure 4  Diffuser assembly profile schematic ................................................................. 8
Figure 5  Front view of diffuser and baffle ......................................................................... 9
Figure 6  Definition of diffuser-relative directions ............................................................ 9
Figure 7  Nozzle block position relative to baffle ................................................................. 10
Figure 8  Nozzle dimensions ............................................................................................. 11
Figure 9  Photograph of baffle ........................................................................................... 11
Figure 10  Technical schematic of baffle ........................................................................... 11
Figure 11  Rendered drawing of baffle ............................................................................... 11
Figure 12  Profile schematic of diffuser .............................................................................. 12
Figure 13  Profile of regular front plate assembly .............................................................. 12
Figure 14  Profile of front plate with convergence insert .................................................... 12
Figure 15  3-d view of regular front plate .......................................................................... 13
Figure 16  3-d view of front plate with convergence insert .................................................. 13
Figure 17  Photograph of traversing assembly .................................................................... 13
Figure 18  Locations of Pitot tube access holes ................................................................. 14
Figure 19  Pitot / static tube configuration ......................................................................... 14
Figure 20  Technical schematic for impingement plate ....................................................... 15
Figures 21-33  Phase-averaged velocity profiles (regular front plate) .............................. 19-25
Figures 34-46  Phase-averaged velocity profiles (front plate w/ insert) .......................... 25-31
Figure 47  Animation of intra-diffuser velocities for regular front plate ......................... 33
Figure 48  Animation of intra-diffuser velocities for front plate w/ insert ......................... 33
Figure 49  Frequency spectra for 1500rpm for a) hot-wire and b) pressure transducer ....... 34
Figure 50  Frequency spectra for 1800rpm for a) hot-wire and b) pressure transducer ...... 34
Figure 51  Average pressures on impingement plate ......................................................... 37
Figure 52  Animated pressures on impingement plate ......................................................... 37
1. Introduction and Literature Review

The problem of how to spread a jet laterally so as to cover a given area most efficiently is fundamentally important to many areas of science and industry. This is the case for the convective cooling or heating of a material, as for example the curing of composite materials with hot air and the spray depositing of particles onto a surface, e.g. the coating of an object with enamel or paint. Each application may have different requirements, therefore the method used from case to case to develop the desired fluid flow may vary. Often, these flows can be produced using some type of diffuser.

Simple diffusers have been extensively studied and are well understood. In fact, they are presently a standard topic covered in introductory fluid mechanics textbooks [White, 1986]. One great limitation of simple diffusers is their inability to spread the flow much more than 7-degrees. Beyond this, separation occurs within the diffuser, effecting increased losses and thus decreased efficiency in terms of pressure recovery. If a much higher degree of spreading is desired, an advanced diffuser design is required. In the case of this research, a rotating baffle upon which a round jet impinges causes the flow to sweep back and forth as it enters the diffuser. In this way the flow is spread out much farther, but now other issues must be addressed, including those related to the flow oscillation.

Here we study the flow within the aforementioned diffuser as well as the flow after exiting the diffuser and impinging on a flat plate. A number of fundamental areas of interest regarding this diffuser are addressed, including possible methods of controlling the amount of spreading effected by the diffuser on the flow, the possible presence of vortical structures in the flow, and the persistence of the flow oscillation through the diffuser. This diffuser was borrowed from industry and is identical to devices that are currently being used in the production of a spray-deposited material. It is hoped that the methods used in the present study, as well as the results obtained will be helpful to others in the study of similar devices.
1.1 Simple Diffuser Flow

Diffusers are very loosely defined as any device that effects the spreading out of a fluid flow so as to fill an increased cross-sectional area. In the majority of cases, diffusers consist of gradually expanding ducts. As fluid flows through a diffuser, its kinetic energy is converted to pressure due to the deceleration of the flow. Typical designs of diffuser include those with rectangular cross-sections, called planar diffusers, conical diffusers, and radial diffusers, but much more complicated designs exist. Planar and conical diffusers have been extensively studied and have been a popular topic for thesis work in the past [Kennedy, 1969] [Jeangprajak, 1981] [Haynes, 1981] [Myatt, 1981]. The literature available on radial diffusers is more sparse.

The geometry of the expansion in a diffuser is of great importance, for if the expansion is too sudden, separation occurs, bringing with it extra losses, which may be unwanted. For simple diffuser designs, the relationship between these losses and the expansion geometry, as well as the overall performances of different simple diffuser designs, are very well understood and documented [White, 1986], [Schetz & Fuhs, 1996]. In the latter reference, a dimensionless quantity referred to as an *area ratio* is defined as the ratio of the diffuser exit to inlet areas. This is a measure of the theoretical pressure recovery expected. However, in our case, the diffuser is unconstrained by walls in one dimension, making theoretical predictions of pressure recovery much more complicated and difficult, since the effective inlet and exit areas are dependent on the fluid properties and the geometry of the diffuser between the inlet and exit.

1.2 The Apparatus Studied in this Work

In this research, a type of diffuser was studied that differs from the simple diffusers already discussed in a number of significant ways. This diffuser seems to be very similar to a simple radial diffuser when examined superficially, but the assumption of radial flow within it is not entirely correct.
This diffuser, shown schematically in Figure 1, contains an expansion in which the cross-sectional area actually contracts in the transverse (radial) direction and is open in the circumferential. This raises the question of whether the contraction accelerates the flow or spreads it out in the circumferential direction, effecting a deceleration in the radial direction. Of course, both may happen, but which effect dominates for what conditions? Also, the flow through this diffuser oscillates, sweeping back and forth between the expansion / contraction section periodically. This oscillation, furthermore, is not symmetrical in its two directions of motion.

Due to the complex and unique design of the diffuser studied in this work, published works on the inner workings of it or even similar devices are practically nonexistent. Any prior work that was done on identical diffusers is proprietary and not available for public scrutiny. However, reports on work done on the fundamental aspects of simple diffuser flows as well as the mechanics of jet flows are prevalent. Appropriate models based on such fundamental research may help us to decide what to look for and how to interpret our results.
The flow through the diffuser takes on various characteristics of different jet flows as it progresses. Initially, a round jet enters the diffuser. It is then transformed into some sort of quasi-radially expanding wall jet. The flow exiting the diffuser might be modeled as a planar jet. This flow impinges on a flat plate as indicated in Figure 1.

Fundamental aspects of various jet flows have been well studied and widely published. The mechanics of turbulent jet evolution for various varieties have been quite well documented and understood for some time [Rajaratnam, 1976]. More specific research on the development characteristics of low-speed planar jets has also been performed [Alcouffe, 1994].

Often in turbulent jets, organized vortical structures emerge. The various modes of this organization as well as its dependence on the flow conditions are fairly well understood as well [Crow & Champagne, 1970]. The issues of jet stability and how jets interact with each other and surfaces of impingement are also prurient to this work. These have been investigated for simple cases [Fisher, 1994] [Bokde, 1989]. Also, the possibility that vortical structures may appear in this diffuser flow both in the free jet exiting the diffuser and the impinging jets is significant. Prior work done on these issues is also easily available [Bokde, 1989] [Doligalski, Smith, & Walker, 1994] [Ho & Gutmark, 1987].

In summary, a method of increasing the spreading of a diffuser is explored. By sweeping the flow back and forth between two converging plates, the spreading is definitely increased, but other factors become important such as the effects of the unsteady flow. The general goals of this research are to learn how the flow behaves within and after exiting the diffuser in question, and to study the effects of modifying the diffuser with a convergence insert. Since vortical structures are common in turbulent jet flows, it is expected that they might play a role here as well. Also, it is expected that the direction of baffle rotation might set up asymmetries in the flow that are noteworthy.
2. The Experimental Programme

2.1 Measuring Instruments

2.1.1 Pressure Transducers

The pressure transducers used were Endevco brand, model 8510B-2, silicon diaphragm with a typical range of ±2 psig, requiring +10Vdc excitation. A Wheatstone bridge as well as an amplifier are highly recommended to be used in conjunction with these transducers. The model 2210 Signal Conditioning Amplifier (SCA), manufactured by Measurements Group, Inc., was used for this purpose. This device contains a Wheatstone bridge and a low-pass filter in addition to the amplifier. The output signal from the transducer was first conditioned and amplified by the model 2210 SCA, then converted into pressures by referring to the transducer’s calibration data. The low-pass filter was set at 1kHz and the gain at 500.

The transducers required periodic recalibration. Zero-order calibrations were performed before collecting each set of data. Second-order calibrations were performed less frequently, depending on how much the transducers had been used recently. Second-order calibrations were performed by fitting second-order polynomials through the voltage versus pressure data collected from the transducers.
2.1.2 The Pitot Tube

The Pitot tube was designed in 1732 and exists today in a number of variations. The design used in this research consists of a 0.042” O.D. / 0.032” I.D. stainless steel tube bent into an L-shape. A pressure transducer connected to one end of the tube senses the pressure that is present at the other end.

When the Pitot tube is placed into a fluid flow as shown in Figure 2, the total pressure of the fluid, $P_{\text{tot}}$, is sensed. If the static pressure, $P_s$, is subtracted from $P_{\text{tot}}$, the dynamic pressure remains. This dynamic pressure can be used to find an approximate corresponding velocity from Bernoulli’s equation for incompressible flow. Solving Bernoulli’s equation for $u$ yields the following:

$$ u = \sqrt{\frac{2(P_{\text{tot}} - P_s)}{\rho}} \quad (1) $$

where $\rho$ is the density of the fluid. This equation is known as the Pitot formula.

Disadvantages of the Pitot tube include its decreasing accuracy with increasing flow angularity with respect to the probe axis. A plot of the error versus flow angularity can be seen in [White, 1986, p. 353]. Also, the Pitot tube is not suitable for low-velocity measurement in gases because of the small dynamic pressure developed. The primary advantages of the Pitot tube are the lack of need for probe calibration and the ease of use.

2.1.3 The Hot-wire Anemometer

The hot-wire anemometer is very well suited to measuring unsteady flows, due to its high frequency response. Also, its small size makes it suitable for measuring within small regions of high velocity gradients, including boundary layers. The hot-wire anemometer
measures velocity by heating a fine wire with an electric current, then measuring the heat loss caused by convection from the flow over it. The relationship between the heat loss from and the flow velocity over the wire is known as King’s Law:

\[ q = I^2R \approx a + b(\rho V)^n \] (2)

where \( q \) is the heat loss; \( I \) and \( R \) are the current and resistance through the wire respectively; \( \rho \) and \( V \) are the fluid density and local velocity respectively; and \( a, b, \) and \( n \) are constants found during calibration. Typically, \( n=1/3 \) at very low Reynolds numbers and equals 1/2 at high Reynolds numbers. This device can be operated either by holding the current constant and measuring the resistance change or vice versa. The hot-wire anemometer used in this research was model number 1053B manufactured by TSI Inc.

Disadvantages of the hot-wire anemometer include its fragility, the need for frequent recalibration, and the dependence of its output on the fluid’s thermodynamic properties. Its primary advantages are its high frequency response and small size.

### 2.2 Computer & Peripherals

Probe positioning, data acquisition, and data reduction were all controlled by an i386/66 PC. Fast Fourier transforms were performed on hot-wire signals by a Hewlett Packard Digital Signal Analyzer running in 100 sample averaging mode. Endevco pressure transducer data were collected using an ISC-16 I/O board, manufactured by R.C. Electronics, Inc. and installed in the PC. The data-acquisition and reduction software used is listed in the Appendices (§6). A reflective object proximity sensor, model OPB700AL by Optek Technology, Inc., provided a trigger for use in unsteady data collection.

A schematic diagram of the hardware setup for Endevco transducer and Pitot tube measurements is shown in Figure 3.
2.3 The Radial Diffuser

The radial diffuser studied in this work was provided by a materials engineering company and is similar in design to a device they use in the manufacturing of one of their products. It consists of three main areas of interest to this work: a rotating baffle, ion guns, and two plates referred to as the front plate and the back plate. Schematic diagrams of the diffuser are presented in Figures 4 and 5. As can be seen in Figure 4, a jet of air enters the apparatus and impinges upon a baffle. The baffle directs the jet along a wall past ion guns then between the front and back plates before it exits the diffuser.
In this report we refer to certain directions relative to the diffuser. These are defined as follows:

*Plate direction:* The direction perpendicular to the back plate surface, positive is toward the front plate.

*Cross direction:* The direction perpendicular to the plate direction; positive being toward the right in Figure 5 (direction of baffle rotation). *Cross-directional angle* refers to the angular distance from the top center of the plates; positive being clockwise and the center of the front plate being the vertex.

These directions are shown in Figure 6 as well.
The nozzle block was positioned relative to the baffle hub as shown in Figure 7. Measurements in this figure are given in inches (±0.001”).

2.3.1 The Jet Nozzle

The initial jet entering the diffuser was produced with compressed air blowing through a nozzle block. A pressure regulator on the air supply ensured that the pressure within the chamber was approximately 22.4 psi while the diffuser was running. This corresponded to a Reynold’s number of approximately 200,828 at the nozzle exit. The inlet of the nozzle block was positioned in line with the exit. Regulated, compressed air entered the inlet via an 18 inch long, 1 inch I.D. PVC tube and the nozzle was fitted with a convergence as is shown in Figure 8 in an effort to ensure that sharp turns leading up to the exit did not play a role in creating organized structures in the exiting nozzle jet. Measurements in Figure 8 are given in inches (±0.1”).

Figure 7 - Nozzle block position relative to baffle
2.3.2 The Rotating Baffle

Subsequent to its issuance from the nozzle, the jet impinges upon a three-lobed baffle, shown in Figure 9. As the baffle rotates, the jet is directed in different directions due to the varying geometry presented to the impinging flow. Unless otherwise noted, the baffle rotation speed was 1800rpm. In a very general sense, as the flat areas between the convex lobes present themselves to the jet, the deflected flow changes radial directions following the turning of the baffle. As the convex lobes present themselves, the deflected flow sweeps back in the opposite direction. The precise behavior of this flow is more complex and is discussed later in §3. Technical schematics of the baffle and a rendered drawing are shown in Figures 10 and 11. Measurements presented are in inches (±0.05”).
2.3.3 Front and Back Plates

Subsequent to its encounter with the baffle, the flow passes between the front plate and the back plate. Ahead of the front plate is a hollowed out area called the ion gun corona, which is not studied in this work. The back plate is basically flat, with the addition of ventilation holes as shown in Figure 12.

Two designs of front plates were used: one with an inner surface shaped as a bowl with a slope of 5° and one with a convergence piece added. These two configurations are shown in Figures 13 and 14 and as three-dimensional surfaces in Figures 15 and 16.
2.4 Instrumentation Rigs

A portion of the data presented in this work was collected by traversing a probe across the region of study. All such data collection was automated by mounting a probe on the sliding platform of a traversing scale. This scale was driven by stepper motors, which were controlled by computer. A typical setup for the traversing system is shown in Figure 17.

2.4.1 Intra-Diffuser Measurement Setup

Holes were drilled in both designs of front plates (regular, and with convergence insert) as shown in Figure 18 to allow Pitot tubes to be inserted and traversed across the width between the front and back plate.
Measurements in this figure are given in inches (±0.025”). Each Pitot tube access hole had an accompanying hole upstream of it (towards the center of the arc) from which corresponding static pressures were read. Each pair of holes was 1/4” apart to match the length of the Pitot tube after the bend so the static pressure was measured directly underneath the tip of the Pitot tube (see Figure 19). The inner set of holes was positioned so that the Pitot tube would be just downstream of the convergence piece in the modified front plate to allow monitoring of some of the effects of the convergence piece on the neighboring region.

2.4.2 The Impingement Plate

Data were also collected from multiple pressure transducers simultaneously, positioned at different locations on a flat plate upon which the flow exiting the diffuser impinged, shown in Figure 20. This plate was drilled and tapped as shown to allow Endevco
transducers to be screwed into it from one side. The other side had much smaller holes drilled into it at the same locations to allow the transducers to sense the pressures present on that surface of the plate. This plate was positioned 13 inches downstream from the exit of the diffuser (back plate edge), and centered with respect to the diffuser exit.

Figure 20 - Technical schematic of impingement plate
2.4.3 Unsteady Measurement Setup

In some cases, unsteady measurements were taken and the data ensemble averaged. Ensemble averaging requires a marker to signify the beginning of each ensemble. The frequency of appearance of this marker should coincide with the driving frequency of the periodic phenomenon being studied. In the case of this work, a reflective object proximity sensor was utilized for this purpose. The sensor was positioned next to the rotating baffle so that when a thin piece of reflective tape glued to the baffle passed before it, it produced an electronic pulse that acted as the trigger for unsteady data collection.

2.5 Experimental Procedure

2.5.1 Intra-Diffuser Measurements

Pitot tubes and static pressure tubes were inserted into the access holes as described in §2.4.1. These Pitot tubes were then traversed across the width of the operating diffuser (in the plate direction) one at a time. Unsteady pressure measurements were recorded and ensemble averaged. This was done for both the regular front plate and the front plate with convergence insert. It was hoped that from these measurements, the changes in flow speed and cross-directional spreading caused by the convergence insert could be quantified.
2.5.2 Impingement Plate

The impingement plate was positioned 13” downstream of, normal to, and centered with respect to the diffuser exit. Initial measurements were made with an Endevco transducer that was mounted on the plate at a position along the plate-directional zero line and approximately 3 inches positive from the cross-directional zero line. A hot-wire was positioned 0.02” over the plate at the same plate coordinate as the transducer and simultaneous frequency spectra were obtained from the readings while the diffuser ran. The motivation for this was the suspicion that there might have been vortical structures present on the surface of the plate that had a definite frequency associated with them, which would appear in the frequency spectra.

Following this, Endevco transducers were mounted four-at-a-time in the impingement plate, measurements were made over time and ensemble averaged, then the transducers were moved to a new location so that, after several runs, composite pressure profiles of the entire plate were found. These profiles could then be related to the corresponding baffle position by their phase with respect to the trigger pulse.
3. Discussion of Results

3.1 Intra-Diffuser Measurements

3.1.1 Velocity Profiles

Velocity profiles across the diffuser’s internal width obtained via Pitot tube pressure measurements are presented in Figures 21 through 46 in the form of colored contour plots. In these figures, the vertical axis represents the phase angle, which in this case is equivalent to the baffle position and the horizontal axis represents the location across the width of the inside of the diffuser. Non-dimensional quantities used in these figures were defined as follows:

\[ x^* = \frac{x}{D_{width}} \quad V^* = V \sqrt{\frac{\rho}{2P_0}} \quad r^* = \frac{r}{r_0} \tag{3}, (4), (5) \]

where \( x \) is the location across the width inside the diffuser (\( x=0 \) is on the front plate inner surface), \( D_{width} \) is the total width inside the diffuser at the given measurement location, \( V \) is the measured velocity, \( P_0 \) is the plenum pressure, \( \rho \) is the density of the fluid, \( r \) is the radial distance from the center of the front plate to the measurement location, and \( r_0 \) is the radial distance from the center of the front plate to the leading edge of the front plate / ion corona assembly. The parameters \( r^* \) and \( \theta \) are fixed for each figure and their values are given in the titles. The velocity profiles between the regular front plate and the back plate are shown in Figures 21 through 33, and those between the front plate with the convergence insert and the back plate are shown in Figures 34 through 46.
Figure 21 - Phase-averaged velocity profiles (regular front plate)

Figure 22 - Phase-averaged velocity profiles (regular front plate)
Figure 23 - Phase-averaged velocity profiles (regular front plate)

Figure 24 - Phase-averaged velocity profiles (regular front plate)
Figure 25 - Phase-averaged velocity profiles (regular front plate)

Figure 26 - Phase-averaged velocity profiles (regular front plate)
Figure 27 - Phase-averaged velocity profiles (regular front plate)

Figure 28 - Phase-averaged velocity profiles (regular front plate)
Figure 29 - Phase-averaged velocity profiles (regular front plate)

Figure 30 - Phase-averaged velocity profiles (regular front plate)
Figure 31 - Phase-averaged velocity profiles (regular front plate)

Figure 32 - Phase-averaged velocity profiles (regular front plate)
Figure 33 - Phase-averaged velocity profiles (regular front plate)

Figure 34 - Phase-averaged velocity profiles (front plate w/insert)
Figure 35 - Phase-averaged velocity profiles (front plate w/insert)

Figure 36 - Phase-averaged velocity profiles (front plate w/insert)
Figure 37 - Phase-averaged velocity profiles (front plate w/insert)

Figure 38 - Phase-averaged velocity profiles (front plate w/insert)
Figure 39 - Phase-averaged velocity profiles (front plate w/insert)

Figure 40 - Phase-averaged velocity profiles (front plate w/insert)
Figure 41 - Phase-averaged velocity profiles (front plate w/insert)

Figure 42 - Phase-averaged velocity profiles (front plate w/insert)
Figure 43 - Phase-averaged velocity profiles (front plate w/insert)

Figure 44 - Phase-averaged velocity profiles (front plate w/insert)
Figure 45 - Phase-averaged velocity profiles (front plate w/insert)

Figure 46 - Phase-averaged velocity profiles (front plate w/insert)
Some interesting features worth noting about these figures are the increased velocities and cross-directional spreading associated with the convergence insert, the “fattening” of the high-velocity region associated with the same, and the varying positions of the peaks relative to the phase angle from figure to figure. The results of a cursory examination of these graphs suggests that the convergence insert effects an increased velocity in the region of the inner set of measurements ($r^*=2.73$), yet the velocities recorded radially farther out ($r^*=3.426$) are not as drastically increased. However, the flow spreads out farther in the cross-direction with the insert present than without. More specific analysis of this phenomenon is discussed in §4.

3.1.2 Animation of Intra-Diffuser Velocities

The velocity measurements obtained with Pitot tubes and presented in the previous section were rearranged and collected together in the form of spatial contours. Each frame corresponds to a specific phase. The periodic phenomenon can thus be animated by quick succession of these frames. These animations, shown in Figures 47 and 48, show the evolution of the flow within the diffuser as the baffle rotates through one-third of a revolution. In each of these two figures, the horizontal axis represents the cross-directional angle and the vertical axis represents the non-dimensional variable, $x^*$, defined earlier. The colored contours represent the non-dimensional velocity, $V^*$; and the heading of each figure includes both the radial distance of the measurements presented in the form of the non-dimensional variable, $r^*$, and the phase angle, $f$, all defined earlier.

In these animations, the sweeping of the flow back and forth between the two plates can be seen as the baffle rotates. The animation corresponding to the front plate with convergence insert shows higher velocities concentrated closer to the front plate than in the case of the regular front plate. Also the position of the high velocity region corresponding to the same phase for the two different front plates is different, one seeming to lag behind the other.
Figure 47 – Animation of intra-diffuser velocities for regular front plate (click to animate)

Figure 48 - Animation of intra-diffuser velocities for front plate w/ insert (click to animate)
3.2 Impingement Plate

Data obtained from simultaneous hot-wire anemometer and pressure transducer measurements on the impingement plate are presented in Figures 49 and 50.

In both cases, the graph labeled “a” is the spectrum of frequencies contained in the hot-wire signal when it was positioned 0.02” above the plate outside the impingement region and the graph labeled “b” is the corresponding frequency spectrum obtained from a pressure transducer mounted on the plate at the same location. The data presented in Figure 48 were obtained with the baffle rotating at 1500rpm and the data presented in
Figure 49 were obtained with the baffle rotating at 1800rpm. Data were collected for the different rotation speeds in an effort to help identify those spikes in the frequency spectra which were due to the flow oscillation introduced by the rotation of the baffle and those that indicated other periodic phenomena that might have been present, such as vortical structures.

Spikes in the pressure transducer’s frequency spectrum seem to indicate frequencies associated with the baffle rotation. Recall that the baffle had three symmetric lobes. This suggests that the mean flow downstream of the baffle would oscillate with a frequency three times that associated with the rotation speed. The relationship between \( w \), the baffle rotation speed in rpm; and \( f \), the corresponding flow oscillation frequency in Hz, is as follows:

\[
 f = 3 \left( \frac{w}{60} \right)
\]

For baffle rotation speeds of 1500 and 1800 rpm, the corresponding fundamental frequencies, \( f \), would be 75 and 90 Hz. Thus, the spikes present in the pressure transducer signal coincide with those associated with the mean flow oscillation and its harmonics due to the baffle's rotation at the two different speeds. Also present in the pressure transducer’s frequency spectra are two wide humps around 700Hz and 900Hz which don’t seem to shift with changing baffle rotation speeds. Otherwise, no other spikes are prevalent.

Spikes in the hot-wire anemometer’s frequency spectra are found roughly at 180Hz, 420Hz, 480Hz and continuing at 60Hz intervals. Unlike the pressure transducer’s response, these spikes do not shift with the change in baffle rotation speed. The fact that the spikes found in the pressure transducer’s response were not, in general, reflected in that of the hot-wire seems to indicate that these are not related to the flow. One possibility is that these spikes represent sound waves generated by the air impinging on the rotating baffle, which are detectable by pressure transducer but not by hot-wire anemometer. If these frequencies had been present in the flow, one would expect them to
be detected also by the hot-wire anemometer. Furthermore, although the spikes in the hot-wire frequency spectra seem more likely to indicate some aspect of the flow on the plate, the fact that they don’t appear in the pressure transducer spectra casts some doubt on this as well. Whereas the spikes in the pressure transducer spectra were found at intervals equal to integer multiples of the prescribed mean-flow oscillation frequency, the spikes in the hot-wire spectra were positioned at intervals of 60Hz. This was not a predicted frequency of mean-flow oscillation for either baffle rotation speed.

Following this, the impingement plate was instrumented with pressure transducers four at a time and composite pressure distributions were obtained across the entire plate over a single rotation of the baffle. Since all data sets were phase-averaged with respect to the baffle rotation, the measurement made in one set at a given phase of the ensemble corresponded to the same baffle position as a measurement made in another set at the same phase. Thus the measurements made at all points on the plate for a given phase of the ensembles should correspond to the same phase of the actual oscillation of the flow across the plate. This presumes that the oscillation of the flow across the plate is the same as that associated with the baffle rotation.

A colored contour plot of the average reduced pressures across the plate over several revolutions of the baffle is presented in Figure 51. Phase-averaged pressure profiles across the plate at ten equally spaced phases within one-third of a baffle revolution are presented in Figure 52 as an animation. Non-dimensional values presented in Figures 51 and 52 were defined as follows:

\[
P^* = \frac{P}{P_0} \quad X = \frac{x}{D_{exit}} \quad Y = \frac{y}{D_{exit}} \quad (7), (8), (9)
\]

where \(P_0\) is the plenum pressure of the initial jet, \(P\) is the measured pressure, \(D_{exit}\) is the exit width of the diffuser, \(x\) and \(y\) are coordinates on the plate, \(x\) corresponding to the cross direction, and \(y\) corresponding to the plate direction, as defined in §2.3.
Figure 51 - Average pressures on impingement plate

Figure 52 - Animated pressures on impingement plate
It can be seen from Figure 51 that the average pressures present on the impingement plate are not symmetric either in the X or Y directions. Rather, the core region is offset to the left on the graph. The entire profile seems to follow this trend as well. It is also interesting to note that the pressure gradients in the Y direction are much greater for X close to zero than farther away. This seems to indicate that the impinging flow is spreading out away from the X=0 line. This is likely partially due to the semi-radial nature of the flow. Assuming the air exiting the diffuser is expanding in the plate direction as it flows radially out, by the time it encounters those areas of the plate where $X \gg 0$, it has had more time to expand than for those areas where $X \approx 0$. Also, the pressures found are all rather small compared to the plenum pressure ($P^* \approx 10^{-4}$). The two areas at the very top of the graph where the pressures increase correspond to where support arms were connected to the plate and should probably be ignored here.

Figure 52 shows very little in the way of discernible pattern as far as the high-pressure region’s movement on the plate is concerned. The presence of individual high-pressure regions divided by lower pressure regions observed in this animation was unexpected and explanations for how this could physically happen seemed far-fetched. Coupled with the results of the hot-wire and pressure transducer frequency spectra analyses, the most reasonable explanation for the lack of discernible pattern in Figure 52 is that the flow present on the plate indeed was not in sync with the baffle’s rotation. The possibility still exists that this is due to flow instabilities such as large vortical structures on the plate, but as previous investigation indicated, the appearance of these would have to be at a varying frequency.
4. Conclusions and Recommendations

Within the diffuser, the flow sweeps through the cross direction following the baffle rotation as the flat regions of the baffle present themselves, then more quickly sweeps back in the opposite direction. This is an oscillating pattern which, due to turbulence and viscous diffusion, disappears with increasing distance downstream of the diffuser so that far away on a flat plate, the periodicity cannot be detected in the flow impinging upon it.

As evidenced in Figures 21 through 33 and Figure 47, a low-velocity region exists near the back plate for $r^* = 2.73$. We believe this to be due to the ventilation holes, which allow outside air to be entrained into the flow. The possibility of a region of recirculating flow existing in the area a short distance downstream of the ventilation holes prompted the addition of a convergence insert on the front plate across from the region of these holes. It was hoped that this would both push the flow more against the back plate and speed it up, effecting a reduced region of slower flow in the region of these holes. However, as shown in Figures 34 through 46 and Figure 48, the convergence instead pushed the flow more against the front plate and actually seems to have increased the region of slow flow near the back plate.

Since measurement stations were located at only 15-degree increments, accurate estimation of the range through which the flow sweeps in the cross direction is difficult. With the regular front plate, interpolated results lead to an estimate that the flow within the diffuser covered cross-directional angles from approximately $-45^\circ$ to $48^\circ$. When the convergence insert was in place, this range increased to approximately $-50^\circ$ to $52^\circ$. The increase in velocity associated with the convergence insert seems to be largely a local phenomenon, only existing in the neighborhood of the insert itself. Therefore, if achieving an increased cross-directional spread range is desired in similar diffusers, while maintaining the same exit velocity, the use of a convergence insert such as the one used here might yield positive results.
The search for organized structures in the flow over the impingement plate seems to have yielded negative results. Dominant frequencies in the readings of a hot-wire anemometer and a pressure transducer placed in the flow do not agree. This suggests that those frequencies found in the signals were due to phenomena other than fluctuation in the flow velocity. For instance, the pressure transducer may have picked up sound waves, and the hot wire may have picked up some E.M. waves from the motor. Furthermore, although precautions were taken to protect the electronics from interference from the supply voltage lines, this is always a possible source of noise in the signals. In any case, the likelihood of vortical structures existing in the flow over the plate seems negligible. The conclusion that the flow over the plate is free of a definite periodic fluctuation is corroborated by composite pressure profiles on the plate derived from phase-averaged measurements. This is a significant point, for it suggests that the nature of the flow exiting the diffuser be such that the strong periodicity of the flow near the diffuser’s exit disappears just a short distance downstream.

Recommendations for future work include varying the position of the convergence insert to see how that affects the size of the slow flow region against the back plate, using an insert which is larger for low C.D. angles than for high, increasing the number of C.D. positions for intra-diffuser measurements. Also, it is recommended that a hot-wire anemometer be used in the future for intra-diffuser measurements. Finally, the addition of fins in the ventilation holes might help reduce the slow flow region by increasing turbulence in the entrained flow, thereby shrinking the boundary layer against the back plate.
References

Alcouffe, D. P., Development Characteristics for Low Speed Turbulent Planar Jets, M.S. Thesis, University of Virginia, 1994

Bokde, A.L.W., An Experimental Investigation of Interactions Which Develop Between Adjacent Low Speed, Turbulent, Quasi-Two-Dimensional Free Jets Impinging upon a Wall, M.S. Thesis, University of Virginia, 1989


Fisher, S.S., *Stability and interaction characteristics of jets*, Report - Department of Mechanical, Aerospace, and Nuclear Engineering, University of Virginia, 1994


6. Appendices

6.1 Basic Pressure Transducer Data Acquisition

The following program was written in C to direct measurements on the impingement plate:

```c
// DuPress.C          Endevco Pressure Transducer Measuring System for the
//                     DuPont Project.
//
// Written by Norman W. Schaeffler
// THIS IS THE STRIPPED DOWN NO THRILLS ECONOMY MODEL !!!!!
//
//
// C includes
#include <graph.h>
#include <time.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>

// Project Includes
#include "screen.h"

// Defines
#define CursorHome   _settextposition( 23, 2 )
#define Internal 0
#define External 1
#define NumberOfChannels 4

// Structure Definitions

// Function Prototypes
void  AcquireDataOnTrigger( void );
void  SaveData( void );
void  Delay( long );
void  BuildScreen( void );
void  GetDate( char date_string[40] );
unsigned char  HI( unsigned int );
unsigned char  LOW( unsigned int );
int  ReadCalFile( FILE * );

// Global Variables
int   TriggerFlag;
int   *RC_Data;
unsigned int   SamplingRate, NumberOfSamples, ClockTime, BufferSize;
float   SamplingTime;
long   DelayTime;
char   filename[20], date[40], CalFilename[20];
FILE   *DataFile, *CalFile;
```
float Calibration[4][3];

void main(int argc, char *argv[])
{
    char buffer[10], ch, ans;
    int i, j;
    float one, two, tri;

    _setbkcolor(0);
    _settextcolor(15);
    _clearscreen(_GCLEARSCREEN);

    printf("\n\n  DuPont Research Project -- Pressure Data Acquisition Package\n");
    printf(" v1.1a (C) 1996 Shi*Ware Productions\n\n");

    if(argc != 3) {
        printf("ERROR: A Command Line Parameter Was Omitted!\n\n");
        printf("Syntax: dupress CalFileName DataFileName\n\n");
        exit(0);
    }
    else {
        strcpy(filename, argv[2]);
        strcpy(CalFilename, argv[1]);
    }

    if((CalFile = fopen(CalFilename, "r")) == NULL) {
        printf("fopen failed for Calibration file!\n\n");
        exit(0);
    }

    i = ReadCalFile(CalFile);
    fclose(CalFile);

    printf("The Following calibration information was found:\n\n");
    for(i = 0; i < 2; i++) {
        printf("Endevco #%d:\t	Endevco #%d:\n", i+1, i+3);
        printf("Zero Offset: %9.6f\tZero Offset: %9.6f\n", Calibration[i][0], Calibration[i+2][0]);
        printf("Linear Coeff: %9.6f\tLinear Coeff: %9.6f\n", Calibration[i][1], Calibration[i+2][1]);
        printf("Quadratic Coeff: %9.6f\tQuadratic Coeff: %9.6f\n", Calibration[i][2], Calibration[i+2][2]);
    }
    printf("Is this information valid? ( Y or N )\n");

    while(1) {
        while(!kbhit()) {}
        ans = getch();
        if(ans == 'n') || (ans == 'N') exit(0);
        if(ans == 'y') || (ans == 'Y') break;
        if(ans == 0) ans = getch();
    }

    if((DataFile = fopen(filename, "w")) == NULL) {
        printf("fopen failed for data file! Bad news, buster!\n");
        exit(0);
    }

    NumberOfSamples = 32768;
    SamplingRate = 10000.0;
    SamplingTime = NumberOfSamples/NumberOfChannels/SamplingRate;
RC_Data = (int huge *)malloc( NumberOfSamples, sizeof(int)));
TriggerFlag = External;

DelayTime = (long)( SamplingTime * 1000 + 1000 );
ClockTime = ( 1000000 / SamplingRate );

BuildScreen();

// Put date and sampling information into data file
fprintf( DataFile, "%s\n", date );
fprintf( DataFile, "%u   %u\n", ClockTime,
          NumberOfSamples/NumberOfChannels );
for( i = 0; i < 4; i++ ) {
    one = Calibration[i][0];
    two = Calibration[i][1];
    tri = Calibration[i][2];
    fprintf( DataFile, "%9.6f %9.6f %9.6f\n", one, two, tri);
}

// With all that out of the way, Go get some data!!!!
AcquireDataOnTrigger();
while( ( inp(0x0309) & 1) != 1 ) {} // wait for trigger
_settextposition( 16, 25);
_outtext("Triggered.                           ");
Delay( DelayTime ); // Delay while data is being acquired
SaveData();

_setbkcolor( 0 );
_settextcolor( 15 );
_clearscreen( _GCLEARSCREEN );
}

void AcquireDataOnTrigger( void )
{
    int data, i, hi,lo;
    unsigned char k;
    float voltage, value;
    outp(0x030B,0x0); // Disabling Data Acquisition System
    for ( i=0; i<4; i++ ) {
        // Loading Multiplexer Table
        outp(0x0310,0x0);
        outp(0x0310,0x1);
        outp(0x0310,0x2);
        outp(0x0310,0x3);
    }
    outp(0x031A,0x08); // Trigger channel 1, IRQ3 on post trigger delay
    outp(0x0311,0x00); // Select Internal Clock
    outp(0x0308,0x0B); // Trigger slope, external trigger, positive polarity
    outp(0x0307,0x74); // set clock
    outp(0x0305, LOW(ClockTime) ); // Load LSB
    outp(0x0305, HI(ClockTime) ); // Load MSB
    outp(0x0307,0x32); // Set Number of channels each burst
    outp(0x0304, LOW(NumberOfChannels)); // Load LSB
    outp(0x0304, HI(NumberOfChannels)); // Load MSB
    outp(0x0307,0xB2); // set post trigger delay
outp(0x0306, LOW(NumberOfSamples));  // Load LSB
outp(0x0306, HI(NumberOfSamples));  // Load MSB

outp(0x0315, 0x00);  // Select bank B
outp(0x031B, 0x07);  // reset bank A memory pointer for 32K buffer
BufferSize = 32767;

outp(0x030A, 0x0);  // Enable Data Acquisition
outp(0x030C, 0x0);  // Enable Trigger Logic

_settextposition(16, 25);
_outtext("Ready For Trigger ...");

}

void SaveData(void)
{
    int DataStart, data, i, hi, lo;
    float voltage, delta_time;

    _settextposition(16, 25);
    _outtext("Saving Data ...");

    outp(0x0314, 0x03);  // select bank A

    for (i = 0; i < NumberOfSamples; i++)
    {
        lo = inp(0x031C);
        hi = inp(0x031C);
        data = lo + 256 * hi;
        RC_Data[i] = data;
    }

    _settextposition(16, 25);
    _outtext("Saving Data to file ...");

    for (i = 0; i < NumberOfSamples; i++)
    {
        if( ((i/4*4) == i) && (i != 0)) fprintf(DataFile, "\n");
        fprintf(DataFile, "%d ", RC_Data[i]);
    }

}

int ReadCalFile(FILE *CalFile)
{
    float c1, c2, c3;
    int i, j;

    fscanf(CalFile, "%d", &i);
    fscanf(CalFile, "%f %f %f", &c1, &c2, &c3);
    for( j = 0; j < 8; j++)
        fscanf(CalFile, "%d %f %f %f", &i, &c1, &c2, &c3);
    for( j = 0; j < 4; j++ )
    {
        fscanf(CalFile, "%d %f %f %f", &i, &c1, &c2, &c3);
        Calibration[j][0] = c1;
        Calibration[j][1] = c2;
        Calibration[j][2] = c3;
    }

}

void Delay ( long Milliseconds )
{

Attempts to create a Turbo Pascal-like delay for x milliseconds routine

```c

clock_t time_1, time_2;

    time_1 = clock();  // Returns the amount of time in milliseconds
    time_2 = clock();  // since the program has begun.

while ( (time_2 - time_1) < Milliseconds )
{
    time_2 = clock();
}
}

void BuildScreen( void )
{
    int i;
    char buffer[90];

    _setbkcolor( 1 );
    _settextcolor ( 15 );
    _clearscreen( __GCLEARSCREEN );

    for ( i = 1; i < 80; i++)
    {
        _settextposition( 2, i );
        _outtext( Horizontal );
        _settextposition( 24, i );
        _outtext( Horizontal );
    }

    for ( i = 3; i < 25; i++ )
    {
        _settextposition( i, 0 );
        _outtext( Vertical );
        _settextposition( i, 80 );
        _outtext( Vertical );
    }

    _settextposition(24, 0);
    _outtext( LowerLeftCorner );

    _settextposition( 24, 80 );
    _outtext( LowerRightCorner );

    _settextposition(2, 0);
    _outtext( TopLeftCorner );

    _settextposition( 2, 80 );
    _outtext( TopRightCorner );

    _settextposition( 6, 0);
    _outtext( LeftSideDivider );

    for ( i = 1; i < 79; i++)
    _outtext( Horizontal );

    _settextposition( 3, 16);
    _outtext("D u P o n t  R e s e a r c h   P r o j e c t");
    _settextposition( 5, 13 );
    _outtext("U n s t e a d y   P r e s s u r e   M e a s u r e m e n t");
```
void GetDate( char date_string[40] )
{
    struct dosdate_t date;
    char day_year[10];

    dos_getdate( &date );

    switch ( date.month )
    {
        case 1:
            strcpy( date_string, "January" );
            break;
        case 2:
            strcpy( date_string, "February" );
            break;
        case 3:
            strcpy( date_string, "March" );
            break;
        case 4:
            strcpy( date_string, "April" );
            break;
        case 5:
            strcpy( date_string, "May" );
            break;
        case 6:
            strcpy( date_string, "June" );
            break;
        case 7:
            strcpy( date_string, "July" );
            break;
        case 8:
            strcpy( date_string, "August" );
            break;
        case 9:
            strcpy( date_string, "September" );
            break;
        case 10:
            strcpy( date_string, "October" );
            break;
        case 11:
            strcpy( date_string, "November" );
            break;
        case 12:
            strcpy( date_string, "December" );
            break;
    }
```c
strcpy( date_string, "November" );
break;
case 12:
    strcpy( date_string, "December" );
    break;
}
strcat( date_string, " ");
sprintf( day_year,"%u, %4d", date.day, date.year);
strcat( date_string, day_year);
}

unsigned char HI( unsigned int input)
{
    unsigned char hi_byte;
    hi_byte = (unsigned char)( input >> 8 );
    return( hi_byte );
}

unsigned char LOW( unsigned int input)
{
    unsigned char lo_byte;
    lo_byte = (unsigned char)( (input << 8 ) >> 8 );
    return( lo_byte );
}

6.2 Pressure Transducer Data Acquisition with Probe Positioning and
Ensemble Delineation

The following program was written in C to direct the intra-diffuser measurements:
```

```c
// DuPress.C          Endevco Pressure Transducer Measuring System for the
//                      DuPont Project.
//
// Written by Norman W. Schaeffler
// Modified 6/25/96 by Andrew R. Mathes to incorporate
// traversing.
// Modified 8/6/96 by A.R.M. to delineate ensembles by
// trigger pulses.
// THIS IS THE STRIPPED DOWN NO THRILLS ECONOMY MODEL !!!!!
//
// C includes
#include <graph.h>
#include <time.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
// Project Includes
#include "screen.h"

// Defines
#define CursorHome _settextposition( 23, 2 )
#define Internal 0
#define External 1
#define NumberOfChannels 1

// Structure Definitions

// Function Protoypes
void AcquireDataOnTrigger( void );
void SaveData( float position );
void Delay( long );
void BuildScreen( void );
void GetDate( char date_string[40] );
void MoveMotor( int direction, float distance);
void ZeroPosition( void );
unsigned char HI( unsigned int );
unsigned char LOW( unsigned int );
int ReadCalFile( FILE * );

// Global Variables
int TriggerFlag, npoints;
int *RC_Data, EnsembleSize;
unsigned int SamplingRate, NumberOfSamples, ClockTime, BufferSize;
unsigned int NumberOfEnsembles;
float SamplingTime;
long DelayTime;
char filename[20], date[40], CalFilename[20];
FILE *DataFile, *CalFile;
float Calibration[4][3];
float x_pulse, dx;

void main( int argc, char *argv[] )
{
    char buffer[10], ch, ans;
    int i, j;
    float one, two, tri, distance, position;
    _setbkcolor( 0 );
    _settextcolor( 15 );
    _clearscreen( _GCLEARSCREEN );
    x_pulse=314.96;

    printf(" Enter number of points: ");
    scanf("%d", &npoints);
    printf(" Distance between points (mm): ");
    scanf("%f", &dx);
    printf(" Enter number of ensembles: ");
    scanf("%d", &NumberOfEnsembles);

    distance=npoints*dx;
    _clearscreen( _GCLEARSCREEN );
    ZeroPosition();
    _clearscreen( _GCLEARSCREEN );

    printf("\n\n DuPont Research Project -- Pressure Data Acquisition Package\n");
if( argc != 3 ) {
    printf("ERROR: A Command Line Parameter Was Omitted!\n\n");
    printf("Syntax:  dupress CalFileName DataFileName\n\n");
    exit(0);
} else {
    strcpy( filename, argv[2] );
    strcpy( CalFilename, argv[1] );
}

if ( ( CalFile = fopen( CalFilename, "r")) == NULL ) {
    printf("fopen failed for Calibration file!\n");
    exit(0);
}

i = ReadCalFile( CalFile );
fclose( CalFile );

printf("The Following calibration information was found:\n");
for( i = 0; i < 2; i++ ) {
    printf("Endevco #%1d:\t			Endevco #%1d:\n", i+1, i+3);
    printf("      Zero Offset: %9.6f		      Zero Offset: %9.6f\n",
            Calibration[i][0], Calibration[i+2][0]);
    printf("     Linear Coeff: %9.6f		     Linear Coeff: %9.6f\n",
            Calibration[i][1], Calibration[i+2][1]);
    printf("  Quadratic Coeff: %9.6f		  Quadratic Coeff: %9.6f\n\n",
            Calibration[i][2], Calibration[i+2][2]);
}
printf("Is this information valid? ( Y or N )\n");

while( 1 ) {
    while ( ! kbhit() ) {} 
    ans = getch();
    if( (ans == 'n') || (ans == 'N') ) exit(0);
    if( (ans == 'y') || (ans == 'Y') ) break;
    if ( ans == 0 ) ans = getch();
}

if ( ( DataFile = fopen( filename, "w") ) == NULL ) {
    printf("fopen failed for data file! Bad news, buster\n");
    exit(0);
}

EnsembleSize = 1333;
NumberOfSamples = 2048;
SamplingRate = 40000.0;

SamplingTime = NumberOfSamples/NumberOfChannels/SamplingRate;
RC_Data = (int huge *)( halloc( NumberOfSamples, sizeof(int)));
TriggerFlag = External;

DelayTime = (long) ( SamplingTime * 1000 + 1000 );
ClockTime = ( 1000000 / SamplingRate );

BuildScreen();

// Put date and sampling information into data file

fprintf( DataFile, "%s\n", date );
fprintf( DataFile, "%u  %u\n", ClockTime, 
        NumberOfSamples/NumberOfChannels );
fprintf( DataFile, "%d %f\n", npoints, dx );
fprintf( DataFile, "%d\n", NumberOfEnsembles );

i=0;
    one = Calibration[i][0];
    two = Calibration[i][1];
    tri = Calibration[i][2];
    fprintf( DataFile, "%9.6f %9.6f %9.6f\n", one, two, tri);

// With all that out of the way, Go get some data!!!!

for ( position = 0; position <= distance; position = position + dx) {
for (i = 1; i <= NumberOfEnsembles; i++) {
    AcquireDataOnTrigger();
    while( (inp(0x0309) & 1) != 1 ) {} // wait for trigger
    _settextposition( 16, 25);
    _outtext("Triggered.                           ");
    Delay( DelayTime ); // Delay while data is being acquired
    SaveData(position);
    }
    if (position < distance) MoveMotor(1,dx);
}
MoveMotor(2,distance);

_setbkcolor( 0 );
_settextcolor ( 15 );
_clearscreen( _GCLEARSCREEN );}

void MoveMotor ( int direction, float distance)
{  
    int steps,i,j,val,vlow,vhi,tdel;

    outp(0x37A, inp(0x37A) | 0x01);  // enable motor
    tdel=3000;
    if (direction == 2)
        outp(0x378,( inp(0x378) & 0xfe));  // ensures bit1 = 0
    else if (direction == 1)
        outp(0x378,( inp(0x378) | 0x01));   // ensures bit1 = 1

    steps = (int) (distance * x_pulse);

    for (i=1; i<steps; i++) {
        outp(0x378,inp(0x378) | 0x02);
        for (j=0; j<tdel; j++){}    
        outp(0x378,inp(0x378) & 0xfd);
        for (j=0; j<tdel; j++){}
    }

    outp(0x37A,inp(0x37A) & 0xfe);  //disables motor
}

void ZeroPosition (void)
{
    float zero_dist;
    int dir;

    printf("Enter direction (1 for +, 2 for -): ");
    scanf("%d", &dir);
    printf("Enter distance to move probe (zero to exit): ");
}
scanf("%f", &zero_dist);
while (((int) (zero_dist) != 0) { 
  MoveMotor(dir,fabs(zero_dist));
  printf("Enter direction (1 for +, 2 for -): ");
  scanf("%d", &dir);
  printf("Enter distance to move probe (zero to exit): ");
  scanf("%f", &zero_dist);
})
void AcquireDataOnTrigger( void )
{
  int data, i, hi,lo;
  unsigned char k;
  float voltage, value;
  outp(0x030B,0x0);     // Disabling Data Acquisition System
  for (i=0; i<16; i++)
    outp(0x0310,0x0);
  outp(0x031A,0x08);   // Trigger channel 1, IRQ3 on post trigger delay
  outp(0x0311,0x00);   // Select Internal Clock
  outp(0x0308,0x0B);   // Trigger slope, external trigger, positive polarity
  outp(0x0307,0x74);   // set clock
  outp(0x0305, LOW(ClockTime) ); // Load LSB
  outp(0x0305, HI(ClockTime) ); // Load MSB
  outp(0x0307,0x32);     // Set Number of channels each burst
  outp(0x0304, LOW(NumberOfChannels)); // Load LSB
  outp(0x0304, HI(NumberOfChannels)); // Load MSB
  outp(0x0307,0xB2);     // set post trigger delay
  outp(0x0306,LOW( NumberOfSamples ) ); // Load LSB
  outp(0x0306,HI( NumberOfSamples ) ); // Load MSB
  outp(0x0315,0x00);     // Select bank B
  outp(0x031B,0x03);     // reset bank A memory pointer for 2k buffer
  BufferSize = 2047;
  outp(0x030A,0x0);     // Enable Data Acquisition
  outp(0x030C,0x0);     // Enable Trigger Logic
_settextposition( 16, 25);
_outtext("Ready For Trigger ... ");
}
void SaveData( float position )
{
  int DataStart, data, i, hi,lo;
  float voltage, delta_time;
_settextposition( 16, 25);
_outtext("Saving Data ... ");
  outp(0x0314,0x03);     // select bank A (2k buffer)
  for ( i=0; i< EnsembleSize; i++)
  { 
    lo = inp(0x031C);
hi = inp(0x031C);
data = lo + 256 * hi;
RC_Data[i] = data;
}

_settextposition(16, 25);
_outtext("Saving Data to file ... ");

for ( i = 0; i < EnsembleSize; i++) {
    if ( ((i/4*4) == i) && ( i != 0 )) fprintf(DataFile, "n", position,"n");
    fprintf(DataFile, "%d \n", RC_Data[i]);
}

int ReadCalFile( FILE * CalFile ) {
    float c1, c2, c3;
    int i, j;
    fscanf(CalFile, "%d", &i);
    fscanf(CalFile, "%f %f %f", &c1, &c2, &c3);
    for ( j = 0; j < 8; j++ )
        fscanf(CalFile, "%d %f %f %f", &i, &c1, &c2, &c3);
    for ( j = 0; j < 4; j++ )
        fscanf(CalFile, "%d %f %f %f", &i, &c1, &c2, &c3);
    Calibration[j][0] = c1;
    Calibration[j][1] = c2;
    Calibration[j][2] = c3;
}

void Delay ( long Milliseconds ) {
    // Attempts to create a Turbo Pascal-like delay for x milliseconds routine
    //
    clock_t time_1, time_2;
    
    time_1 = clock(); // Returns the amount of time in milliseconds
    time_2 = clock(); // since the program has begun.

    while ( (time_2 - time_1) < Milliseconds )
    {
        time_2 = clock();
    }

void BuildScreen( void ) {
    int i;
    char buffer[90];
    _setbkcolor(1);
    _settextcolor(15);
    _clearscreen(_GCLEARSCREEN);
    
    for ( i = 1; i < 80; i++)
    {
        _settextposition(2, i);
        _outtext(Horizontal);
        _settextposition(Horizontal);
    }
}
for ( i = 3; i < 25; i++ )
{
    _settextposition( i, 0 );
    _outtext( Vertical );
    _settextposition( i, 80 );
    _outtext( Vertical );
}

_settextposition(24, 0);
_outtext( LowerLeftCorner );

_settextposition( 24, 80 );
_outtext( LowerRightCorner );

_settextposition(2, 0);
_outtext( TopLeftCorner );

_settextposition( 2, 80 );
_outtext( TopRightCorner );

_settextposition( 6, 0);
_outtext( LeftSideDivider );

for ( i = 1; i < 79; i++)
    _outtext( Horizontal );

_outtext( RightSideDivider );

_settextposition(3, 16);
_outtext("D u P o n t R e s e a r c h  P r o j e c t");
_settextposition(5, 13);
_outtext("U n s t e a d y  P r e s s u r e  M e a s u r e m e n t");

_settextposition(24, 24);
_outtext(" ESM Fluid Mechanics Laboratory ");

_settextposition(8, 31);
GetDate( date );
_outtext( date );

_settextposition(11, 26);
sprintf(buffer, "Data File Name: ");
strcat( buffer, filename);
_outtext( buffer );

_settextposition(13, 27);
if( TriggerFlag == Internal )
    _outtext("Internal Trigger Selected");
else
    _outtext("External Trigger Selected");

_settextposition(16, 10);
_outtext("RC Status:");

}

void GetDate( char date_string[40] )
{
    struct dosdate_t date;
    char day_year[10];
_dos_getdate( &date );

switch ( date.month )
{
    case 1:
        strcpy( date_string, "January" );
        break;
    case 2:
        strcpy( date_string, "February" );
        break;
    case 3:
        strcpy( date_string, "March" );
        break;
    case 4:
        strcpy( date_string, "April" );
        break;
    case 5:
        strcpy( date_string, "May" );
        break;
    case 6:
        strcpy( date_string, "June" );
        break;
    case 7:
        strcpy( date_string, "July" );
        break;
    case 8:
        strcpy( date_string, "August" );
        break;
    case 9:
        strcpy( date_string, "September" );
        break;
    case 10:
        strcpy( date_string, "October" );
        break;
    case 11:
        strcpy( date_string, "November" );
        break;
    case 12:
        strcpy( date_string, "December" );
        break;
}

strcat( date_string, " ");
sprintf( day_year, "%u, %4d", date.day, date.year);
strcat( date_string, day_year);

}

unsigned char HI( unsigned int input )
{
    unsigned char hi_byte;
    hi_byte = (unsigned char)( input >> 8 );
    return( hi_byte );
}

unsigned char LOW( unsigned int input )
{
    unsigned char lo_byte;
    lo_byte = (unsigned char)( (input << 8 ) >> 8 );
    return( lo_byte );
}
6.3 Post-Processing of Data

The following program was written in Pascal to both reduce the measured voltages into pressures / velocities, and to ensemble average the data:

```pascal
program ens_avg5(infile,outfile);

const
  baffreq=30;

var
  infile,outfile:text;
  junkstring:string[20];
  junkint,i,j,index,samples_per_rev,samples_per_ens:integer;
  pulses,npoints,dx,divisor,numb,vel,press:real;
  c: array [1..3] of real;
  indata: array [1..1350] of real;
  junkchr:char;

begin
  assign(infile,'c:\dupont\pressure\tunnel.dat');
  reset(infile);
  assign(outfile,'c:\dupont\pressure\tunnel.out');
  rewrite(outfile);

  readln(infile,junkstring);
  readln(infile,pulses,samples_per_ens);
  readln(infile,npoints,dx);
  readln(infile,numb_ens);
  readln(infile,c[1],c[2],c[3]);

  samples_per_ens:=1333;
  for index:=1 to 1350 do
    indata[index]:=0;
  write(outfile,'0
');
  for index :=1 to samples_per_ens  do
    write( outfile,((index-1)*(360/1333.333)):6:2,' ');
  writeln( outfile);
  for i:= 1 to npoints do
    begin
      for j:=1 to ( numb_ens) do
        begin
          for index:=1 to samples_per_ens do
            begin
              readln(infile,numb);
              v:=((numb*20)/4095)-10;
              vel:=3.048*sqrt(2*abs(press)*133.322/1.203);
              indata[index]:= indata[index]+ vel;
            end;
        end;
      write( outfile,(dx*i+0.05):5:3,' ');  
      for index:=1 to samples_per_ens do
        begin
          {index*pulses/1000000}
          write(outfile,(indata[index])/( numb_ens):10:6,' ');
          indata[index]:=0;
        end;
    end;
writeln(outfile);
```

```
end;
writeln(outfile);
end;
close(infile);
close(outfile);
end.
7. Vita

Name: Andrew Robert Mathes
Birthplace: Charleston, WV.
Birthdate: October 27, 1973
Undergraduate: Virginia Polytechnic Institute and State University, Blacksburg, VA. (graduated 1995)
Undergraduate Major: Engineering Science and Mechanics