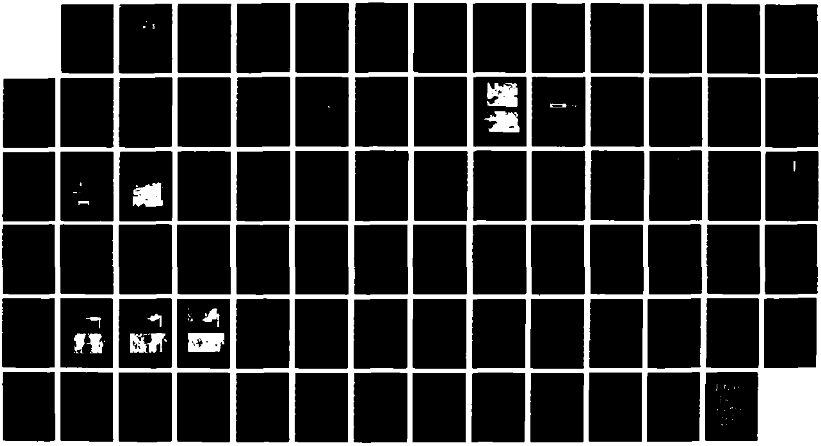
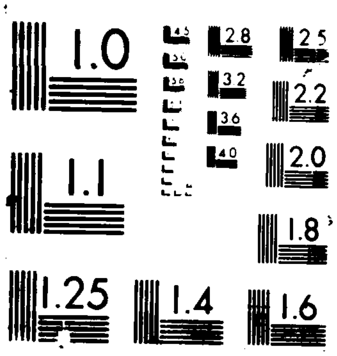
AD-A187 629 INVESTIGATION OF DYNAMIC STALL USING LDV (LASER DOPPLER VELOCIMETRY): MEAN FLOW STUDIES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA R R RYLES SEP 87 1/1

UNCLASSIFIED F/G 20/4 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

②

AD-A187 629

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

INVESTIGATION OF DYNAMIC STALL USING LDV:
MEAN FLOW STUDIES

by

Richard Randolph Ryles

September 1987

Thesis Advisor: S. Bodapati

Approved for public release; distribution unlimited.

87 12 [illegible]

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 67
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
INVESTIGATION OF DYNAMIC STALL USING LDV: MEAN FLOW STUDIES

12 PERSONAL AUTHOR(S)
Ryles, Richard R.

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM ____ TO ____	14 DATE OF REPORT (Year, Month, Day) 1987 September	15 PAGE COUNT 80

16 SUPPLEMENTARY NOTATION

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
LDV, Laser Doppler Velocimetry, Dynamic Stall, Schlieren

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis lays the foundation for the dynamic stall investigation being conducted at the Fluid Mechanics Laboratory at NASA-Ames Research Center. Using existing optical and electrical equipment, a new dedicated Micro-VAX computer and Labstar software, an indraft transonic wind tunnel and able technicians to make the proper interface hardware, the project came together in a new test facility at the Fluid Mechanics Laboratory. The goal of the thesis was to obtain both qualitative and quantitative information about the wake profiles of an airfoil in steady state operations at varying angles of attack and tunnel conditions. To accomplish this task, schlieren photography was used to obtain a qualitative picture of the flow field. With this information, a two component Laser Doppler Velocimeter was set up to accurately measure the velocity profiles that correspond to the schlieren photographs.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Professor Satyanarayana Bodapati	22b TELEPHONE (Include Area Code) (415) 694-4144
22c OFFICE SYMBOL 67Bu	

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

19. (Continued)
Once this preliminary work is completed, the same apparatus will be used to further investigate the unsteady dynamic stall phenomenon.

Investigation of Dynamic Stall
Using LDV: Mean Flow Studies

by

Richard Randolph Ryles
Major, United States Army
B.S., United States Military Academy, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

Author: ____________________
Richard Randolph Ryles

Approved by: Satya Bodapati
Satyanarayana Bodapati, Thesis Advisor

____
for M. F. Platzer, Chairman,
Department of Aeronautical Engineering

G Schacher
G. E. Schacher
Dean of Science and Engineering

ABSTRACT

-This thesis lays the foundation for the dynamic stall investigation being conducted at the Fluid Mechanics Laboratory at NASA-Ames Research Center. Using existing optical and electrical equipment, a new dedicated Micro-VAX computer and Labstar software, an indraft transonic wind tunnel and able technicians to make the proper interface hardware, the project came together in a new test facility at the Fluid Mechanics Laboratory. The goal of the thesis was to obtain both qualitative and quantitative information about the wake profiles of an airfoil in steady state operations at varying angles of attack and tunnel conditions. To accomplish this task, schlieren photography was used to obtain a qualitative picture of the flow field. With this information, a two component Laser Doppler Velocimeter was set up to accurately measure the velocity profiles that correspond to the schlieren photographs. Once this preliminary work is completed, the same apparatus will be used to further investigate the unsteady dynamic stall phenomenon.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The research for this thesis was conducted at the Fluid Mechanics Laboratory of NASA-Ames Research Center under the Navy-NASA Joint Institute of Aeronautics. This research program was supported by the Army Research Office and the Air Force Office of Scientific Research. Research was conducted under the guidance of Professor Satyanarayana Bodapati, Thesis Advisor. Dr. M. S. Chandrasekhara of the Joint Institute was always available for consultations. I would like to offer my sincere gratitude to both of them for providing me with an opportunity to learn the techniques of Laser Doppler Velocimetry, schlieren Photography and wind tunnel experimental techniques. Their patience and encouragement throughout the many months of research provided me with the impetus to persevere during the many difficult times and continue to pursue my objectives in this thesis.

My sincere appreciation also goes to the support personnel of the Fluid Mechanics Laboratory who cheerfully assisted me throughout the research project. A special thanks goes to Michael Fidrich and Jose Espina, laboratory technicians. Without their extraordinary talents and hard work, the experiment could not have been accomplished. Each problem was quickly and efficiently solved by this crew.

I would also like to thank Sterling Software's two programmers in the Fluid Mechanics Laboratory, Charley Hooper and Gordon Goodfellow. Their device drivers provided an inordinate amount of assistance in interfacing my computer program with the different hardware devices. Charley was especially helpful in solving each and every computer problem that I encountered.

Finally, I wish to formally thank my wife, Melissa, whose patient understanding, moral support and loving encouragement kept me going when the wheels of research seemed to go slower than I desired. Also, a special thanks to Margaret, my daughter, for cheerfully putting up with the numerous separations encountered during my travels to NASA-Ames and for allowing me to use "her" computer.

I. INTRODUCTION

A. DESCRIPTION OF THE OVERALL DYNAMIC STALL PROJECT

Static stall is a well understood and documented phenomenon. A wing or rotor blade that is operating at a high α, angle of attack, will experience a point where separated flow will cause an accelerated decrease in lift and a large rapid increase in drag. This α at which the wing will produce the maximum lift, $C_{l_{max}}$, is a very important aspect of airfoil performance. This stall α determines the stall speed of the aircraft. Considerable effort has been expended to find ways to decrease the stall speed by increasing the $C_{l_{max}}$. The majority of this effort has been directed toward a gradual approach to the stall α, in other words, static stall research.

The wing can be made to continue to produce lift past the static stall α if the wing angle of attack is increased rapidly past the stall α. The phenomenon of dynamic stall is manifested in the rapid decrease of the lift at the higher stall α. There are four basic stages of dynamic stall that have been documented [Ref. 1]:

1. Trailing-edge flow reversal, progressing forward along the airfoil.
2. Formation, growth and shedding of the vortex.
3. Full stall.
4. Boundary layer reattachment.

The understanding of this phenomenon will be beneficial to the helicopter, where constantly oscillating rotor blades are consistently in the dynamic stall region throughout their revolution, and jet fighters, where agile and rapid maneuvering through the stall α is required for survivability. By quantifying the velocity field around and behind the airfoil, a better understanding of the dynamics of this phenomenon will be gained. This new analytical information will provide valuable information to the design engineer to expand the flight envelope of future fighter aircraft and improve the performance of the helicopter rotor system.

B. THESIS GOALS

This thesis lays the foundation for the dynamic stall investigation project being conducted at the Fluid Mechanics Laboratory at NASA-Ames Research Center. Using existing optical and electrical equipment, a new dedicated Micro-VAX computer, new Labstar software, an existing transonic indraft wind tunnel and able technicians to make the proper interfaces, the project came together in a new test cell of the Fluid Mechanics Laboratory. The goal of this thesis was to first obtain stroboscopic schlieren photographs of a stationary airfoil at different α's to obtain qualitative data on the flow past the airfoil and the size, shape and extent of the wake and then set up a Laser Doppler Velocimeter to obtain quantitative velocity data in order to construct an accurate velocity profile of the flow field.

C. SCHLIEREN AND LASER DOPPLER VELOCIMETRY BASICS

1. Schlieren

Many optical flow visualization techniques have been developed for gas flows. Each of these techniques are sensitive to variations in density and derivatives of density in the flow. Interferometry is sensitive to the density of the gas flow; ρ. Schlieren is sensitive to the density gradient or the change in density with respect to distance; $\delta\rho/\delta y$. Shadowgraphy is sensitive to the second derivative of the density; $\delta^2\rho/\delta y^2$. All the techniques rely on the fact that the speed of light is a function of the local index of refraction in that medium. In gases, the index of refraction is a function of density.

The schlieren, German for streak, technique allows one to view, qualitatively, the density gradient in a given flow field. This visualization system is considerably simpler and cheaper to use than other techniques. Therefore, it was chosen as the preliminary technique to obtain qualitative information about the air flow around and in the wake of the airfoil. The schlieren setup involves using a line source of light and projecting it through the test section where the flow field alters the path of the light. The line source is usually a monochromatic gas discharge tube. The light is projected through a narrow slit to provide a line of light of uniform illumination. The dimensions of the slit control the definition of the image seen on the screen. A high quality concave mirror is usually used to direct the light from the source through the test section. On the other side of the test section, another similar mirror is used to reflect the light to a viewing screen or camera. Each point of the line source will illuminate each point in the test section if the source is collimated and of equal optical distance on both sides of the test section. The light from the second mirror is focused onto a knife edge. The placement and adjustment of the knife edge is very critical as the sensitivity of the entire system depends on it. The knife edge is placed in the plane of the source image and interferes with a portion of this image. A portion of the light from each point in the test section is blocked which decreases the light impinging on the viewing screen. The density gradients encountered in the test section normal to the knife edge cause a refraction of light allowing more light from that portion of the test section to pass over the knife edge and illuminate the viewing screen. By rotating both the line source and the knife edge both directional derivatives of density can be viewed. Also, through the lateral adjustment of this knife edge at the source image, a fine control over the degree of detail resulting from this technique can be exercised.

The viewing screen can be any image recording device. Polaroid cameras can be set up for instantaneous results or 35 mm cameras can be mounted at the focal length of the lens to obtain numerous photographs during the testing. Video cameras can also be used.

2. Laser Doppler Velocimetry

The techniques for measuring velocity in the past used sensors, which need to be introduced into the flow. This necessarily meant intruding into and affecting the flow. During the last two decades, a new technique, Laser Doppler Velocimetry or LDV, has been available. LDV's are indeed a better technique. It is nonintrusive and is capable of directly measuring velocities of particles in the flow. Laser Doppler Velocimetry has been very successful in measuring flow velocities in turbulent and reverse flows. It has been successfully used for the past two decades to obtain velocity measurements accurately.

Laser Doppler Velocimetry uses two coherent[1] light beams from the same source, which intersect to form the measurement volume. The two light beams form an interference fringe system that is crossed by particles moving with the flow. The light scattered by the particles is Doppler shifted[2]. The frequency of this scattered light is directly proportional to the particle velocity. With

---

[1] Coherent Light = light in which the electromagnetic waves maintain a fixed phase relationship. (Random House College Dictionary)

[2] Doppler Shift = the magnitude of the apparent change in frequency resulting from the relative motion of the light source and the receiver. (Random House College Dict)

suitable selection of flow particles, known also as seeds, the fluid velocities can be measured from the particle velocities. Using photodetectors, the light scattered by the particles is converted into electrical signals. This signal, after processing, provides frequency information from which the velocity can be determined. The measurement of two components of the velocity can be obtained by setting up two orthogonal systems with laser beams of two colors, in the same optics train, thus identifying each component with a particular color.

II. EXPERIMENTAL APPARATUS AND PROCEDURE

A. SCHLIEREN SETUP

The schlieren system for this experiment was set up using both a continuous light source and a stroboscopic source. The continuous source was used to provide a time averaged schlieren photograph and also to allow use of a video tape recorder to record the time history of the flow. To provide instantaneous schlieren photographs of the flow field, the stroboscope was used. See Figure 2.1 for the schematic of the setup.

To obtain the 35 mm photographs of the flow field, a US Scientific Strobrite Stroboscope was used. The strobe had five settings, 20 Hz to 1 KHz, to adjust the duration of the strobe. The 70 Hz setting proved to be the optimal setting. A mechanically adjustable slit in front of the strobe was used to produce a line source of light needed for the technique to operate effectively. A large concave mirror with a 305 cm focal length reflects this light sheet towards the test section. The test section had optical access through both sides of the tunnel. The test section's dimensions were 13 cm by 25 cm and was constructed out of 2.5 cm thick plexiglas. The light passing through the test section was then reflected by an identical mirror located opposite the first mirror on the other side of the test section as shown

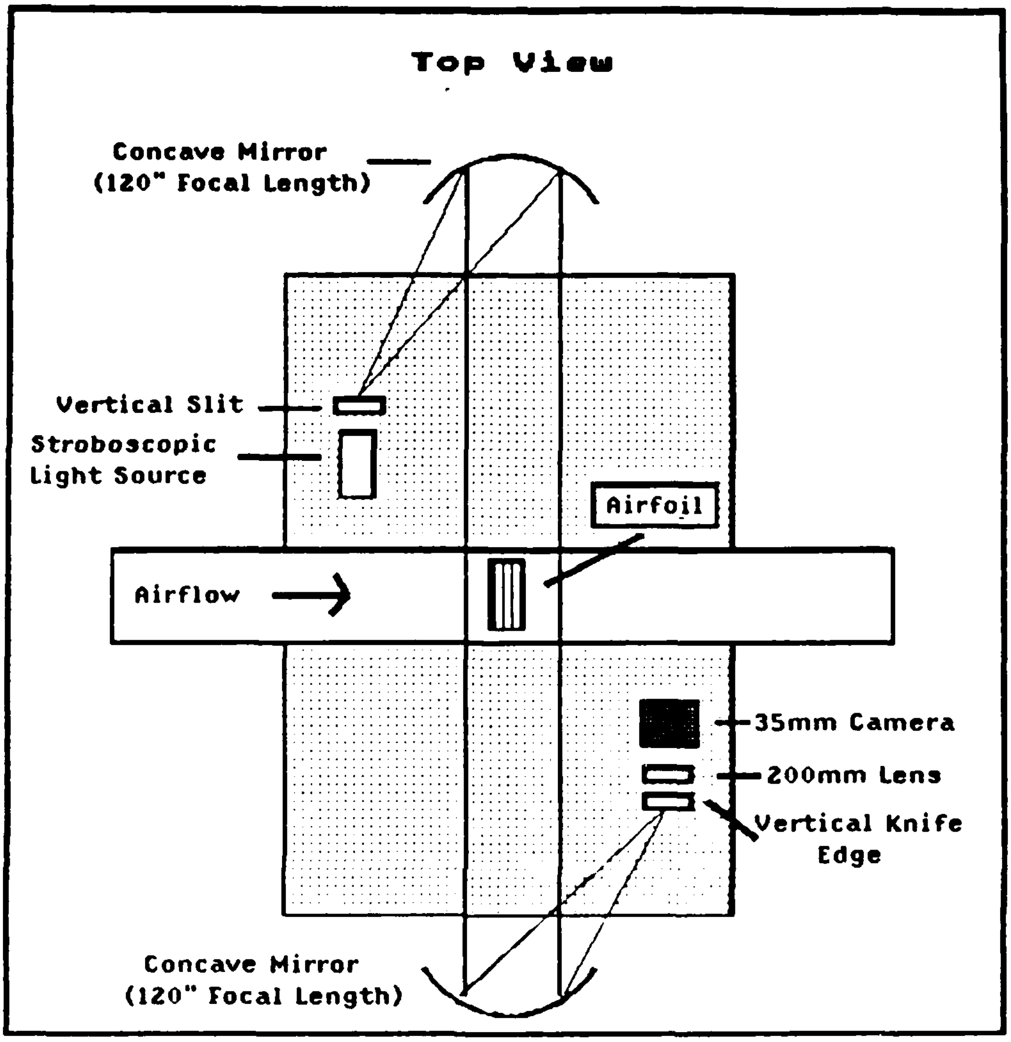


Figure 2.1 Schlieren Schematic

In Figure 2.1. Light from the second mirror was focussed on a vertical knife edge whose position could be adjusted to get a highly sensitive system. The knife edge was oriented vertically to obtain the visual signature of the density gradient in the vertical plane of the test section. After the knife edge, a 200 mm lens was used to focus the image onto the photographic device.

Since the test section is 25 cm wide, the photographs obtained from the system actually represent density gradient effects averaged over the test section width, as is always the case with the schlieren technique.

A Nikon 35 mm camera body with motordrive (but no lens) was used to record the schlieren images. The camera was placed at the focal point of the 200 mm lens. Fine adjustments to focus the image in the camera were made by a micrometer slide on the camera mounting mechanism. Pictures were taken by opening the shutter with a remote shutter release cable, before the strobe was fired remotely and then closing the shutter. With a little practice, this procedure was quite rapid. Photographs were taken at two different Mach numbers, 0.30 and 0.45, and at three different α's; 0°, 7.7°, 15.5°. See Figures A.1 through A.6, Appendix A, for examples of schlieren photographs. This series of photographs was studied to analyze the flow qualitatively and to determine the locations in the wake for obtaining the velocity data for the two different Mach numbers.

B. LASER DOPPLER VELOCIMETER SETUP

This section will discuss the specific LDV system setup in test cell number four at the Fluid Mechanics Laboratory. The following sections will be discussed: transmitting optics, receiving optics and photodetectors, test section, signal processing and data processing. Figure 2.2 shows a photograph of the transmitting optics and Figure 2.3 shows a photograph of the receiving side. Individual schematics will be discussed in detail within the following sections.

The laser used is a Spectra Physics, Model 164, 5 Watt Argon-Ion Laser capable of outputting light from 0.351 to 0.528 microns and up to twelve discrete wavelengths simultaneously [Ref. 2:p. 23]. By choosing the blue beam, 0.4880 microns, and the green beam, 0.5145 microns, for the two component Laser Doppler Velocimetry System, nearly 75% of the initial output power could be used. This standard practice was adapted in this case.

1. Transmitting Optics

Owing to traversing requirements, the laser was higher on the optics table than the 10.8 cm optical height of the transmitting optics. A set of two mirrors at 45° angles to the direction of the incident light were used to lower the beam to the require height. Figure 2.4 shows a detailed schematic of the entire transmitting optics system.

The vertically polarized beam exiting the multiline laser, after being directed by mirrors, is passed through a beam collimator, part of the TSI Model 9106 set. Incident light is collimated if the beam diameter does not change as it passes through space. In reality, however, it is sufficient if the collimator can insure this quality through the

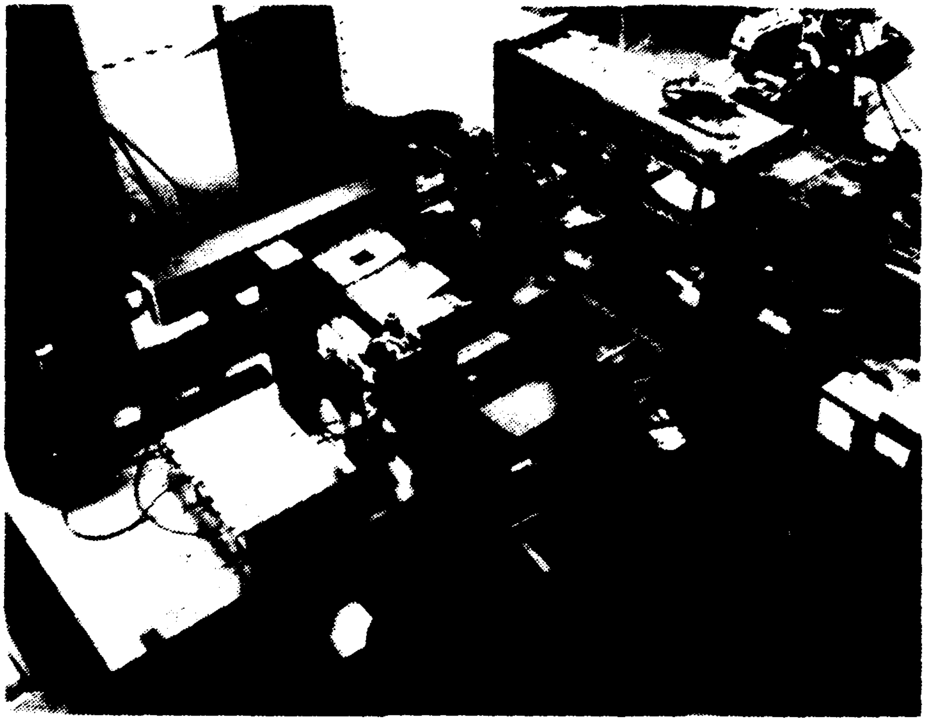


Figure 2.2 Transmitting Optics Photograph

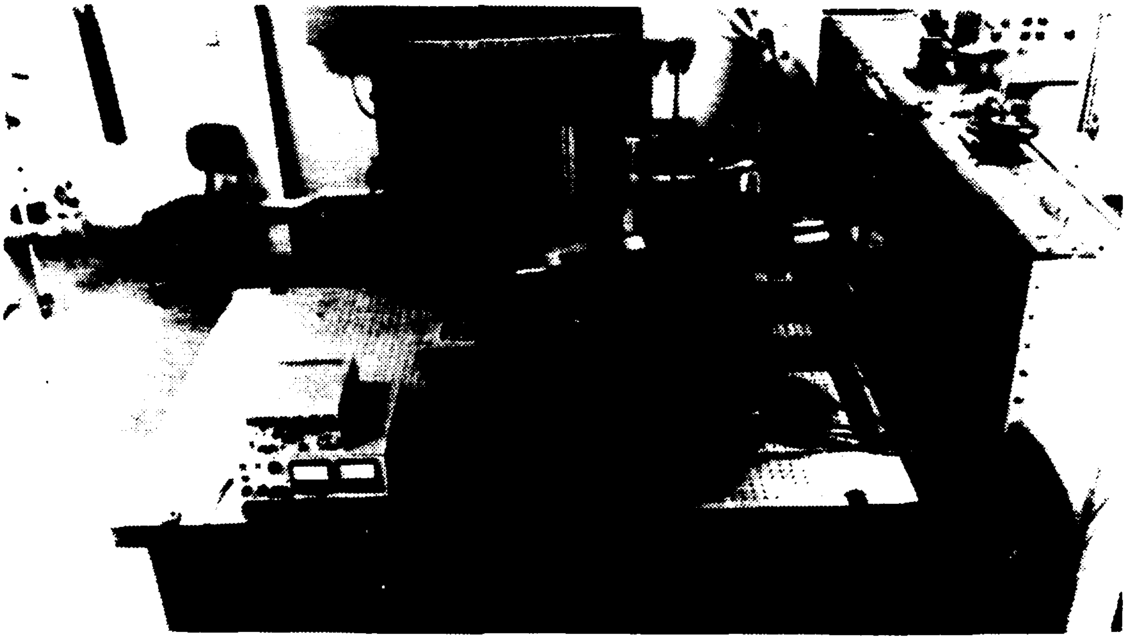


Figure 2.3 Receiving Optics Photograph

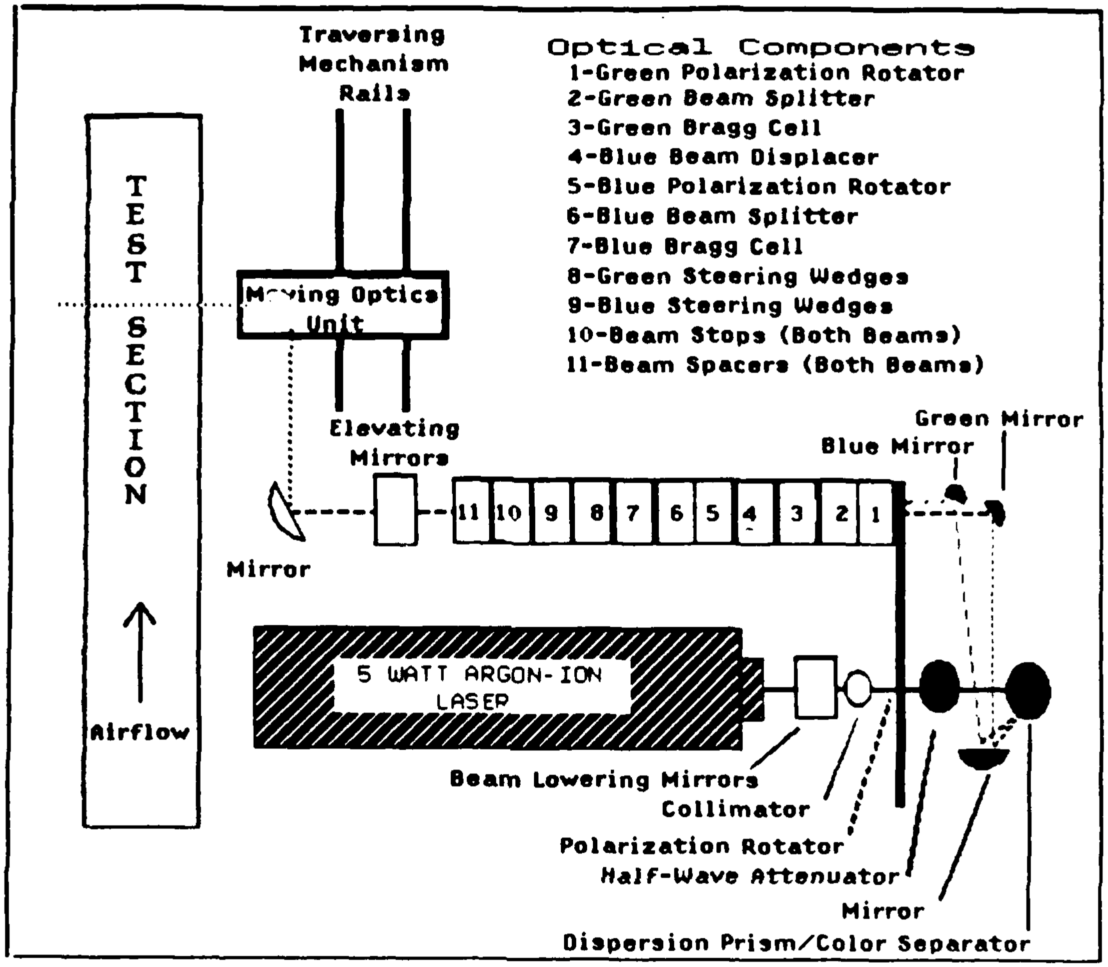


Figure 2.4 Transmitting Optics Schematic

relatively short optical path of the system. Proper collimation is said to be achieved when the waist[1] of the beam coincides with the focal point of the transmitting lens. By sliding the collimator sleeve in and out and observing the beam "spots" on a distant wall, the collimator can be set.

---

[1] Waist = the point on a beam where the diameter is the smallest.

From the collimator, the beam passes through a polarization rotator and an attenuator. The polarization rotator changes the polarization of the incoming laser beam to conform to the polarization requirements of the next optical component, a dispersion prism. The attenuator works in tandem with the rotator to insure passage of only the horizontally polarized light through these components. The dispersion prism separates the wavelengths, or colors, of the light. The two primary colors of this laser, green (0.5145 microns) and blue (0.4880 microns), are used to form the two component system. A set of mirrors, TSI Model 9107, is used to direct each of the beams to the appropriate axis for the transmitting optics as shown in Figure 2.4. As can be seen from the figure, the green beam is directed down the centerline of the optics and the blue is directed 25 mm to the right of the green.

The centerline, green, beam passes through a polarization rotator, TSI Model 9102, to rotate the beam polarization 90°, then a beam splitter, TSI Model 9115. The polarization change is necessary to insure proper splitting into two beams from the incident light. The beam splitter divides the incident beam into two beams of equal intensity and parallel to the optical axis, spaced 50 mm apart.

Since the flow field to be investigated will have a turbulent wake with possible separated flow regions, a frequency shift system is used for each of the two colored beams. Upshifting of the frequency of the laser light prior to entering the test section will allow detection of the direction the particle traverses through the probe volume. The shift results in a moving fringe pattern that will generate a photodetector signal equal to the shift frequency when measuring a particle with zero velocity. This system allows a greater range of measurements of negative velocities and, thus, is required for this system to obtain valid measurements.

The pair of green beams now enter a frequency shifting device called the Bragg cell. This device is an acousto-optic unit that upshifts the frequency of an incident beam. The shift is set at 40 MHz. This device is placed in the optical path of one of the two beams forming the probe volume for that color. The outgoing beam will be shifted by a 40 MHz shift relative to the other beam. The Bragg cell operates by deflecting one beam in a fan-like pattern in the plane intersecting both incident beams. Each separate deflected beam has a frequency that is a multiple of 40 MHz from the incident beam. The Bragg cell should be adjusted to shift the light intensity to the first order beam. This is accomplished by tilting the Bragg cell in its mount. This first order beam should have an intensity greater than

90% of the incident light beam [Ref. 3:p. 148]. An optical wedge is included in this device at the output end of the Bragg Cell to steer the deflected, first order beam parallel to the unshifted beam. The result of this Bragg shifting is an interference pattern in the probe volume that will move relative to a fixed particle. This moving fringe pattern can be used to differentiate the direction of motion through the probe volume. Particles moving with the direction of motion of the fringes gives a lower frequency and against the fringe movement gives a higher frequency as shown in the following equation. The particle moving through the probe volume will produce a signal whose frequency is

$$f_{signal} = |f_{shift} +/- f_{Doppler}|$$

where the sign is + for particles moving in the opposite direction to the fringe movement [Ref. 2:p. 67].

For optic tube support purposes a TSI, Model 9176, Support is used. This device also serves to increase the effectiveness of the beam stops of the blue beam by increasing the deflected beam separation distance.

The off-axis blue beam is now brought to the centerline using a beam displacer, TSI Model 9174. This model employs a rhomboid prism. It displaces the beam a fixed, 25 mm, distance to the centerline of the optics while keeping the beam parallel to the axis. The blue beam then passes through the same series of optical components as the green beams: polarization rotator, beam splitter and Bragg cell.

The next two elements in the optics train are beam steering modules, TSI Model 9175, one for each of the colors. These modules allow for steering of the shifted beam to insure proper beams crossing.

The Bragg cell produces several beams, of which only the first harmonic, the beam that is shifted by 40 MHz, is important. All the other unwanted beams are blocked off by means of a beam stop. In the present case, a beam stop was used on each of the two shifted beams.

Following the beam stops, beam spacers, TSI Model 9113-13, are used to for each set of beams to reduce the distance between the beam pairs forming the probe volume. The spacing was reduced to 13 mm by using these prisms; while maintaining polarity, coherence, beam diameter and parallel optical paths. The primary use of these elements is to control the fringe spacing in the probe volume by changing the angle of intersection of the beam. Additionally, however, they permit transmission of the beams into the test section by requiring only small mirrors and optic components.

The four beams from the optic tube are directed to the focusing lens with five mirrors. The first two move the beams from 10.8 cm to 23 cm above the optical table and the next two reflect the beams back 180° toward the last mirror in this series. This mirror directs the four beams vertically toward the transmitting optics lens. The four beams then enter the transmitting optics lens to focus the four beams to a point. A final mirror is used to direct this focused beam to the test section. The focusing lens and the last three mirrors are mounted on a traversing mechanism to enable traversing of the focused beams in any of the three orthogonal directions as shown in Figure 2.6.

2. Receiving Optics and Photodetector

All the receiving optics are mounted on a traversing mechanism which is slaved to the traversing mechanism on the transmitting side. The two sides were aligned to within 0.01 mm and could be moved together. A Taskmaster controller is used to move the two traverses both manually and through the use of computer programs. See Figure 2.5.

The receiving optics are used to capture the light scattered by the particles crossing the probe volume and focus this light onto photodetectors. It consists of several components; that are mounted on a hardware arm connected to the traversing mechanism. The scattered light is collected by a series of three lenses that collimate the light and direct it toward the polarization color separator. The last lens acts as a spatial filter in addition to being a beam collimator. Its function is to allow light focused on the lens to pass through while blocking all the rest of the light. It is used to reduce the stray light scattered from around the probe volume, as is the case in measurements close to a solid surface.

The polarization beam separator separates the two wavelengths present in the scattered light into its constituent colors; green and blue. This splitter reflects light with horizontal polarization upward while transmitting the vertically polarized light through it. This insures separation of the two primary colors in the scattered light.

Two notch filters are used at this stage, on each of the colors to eliminate residual color contamination. The

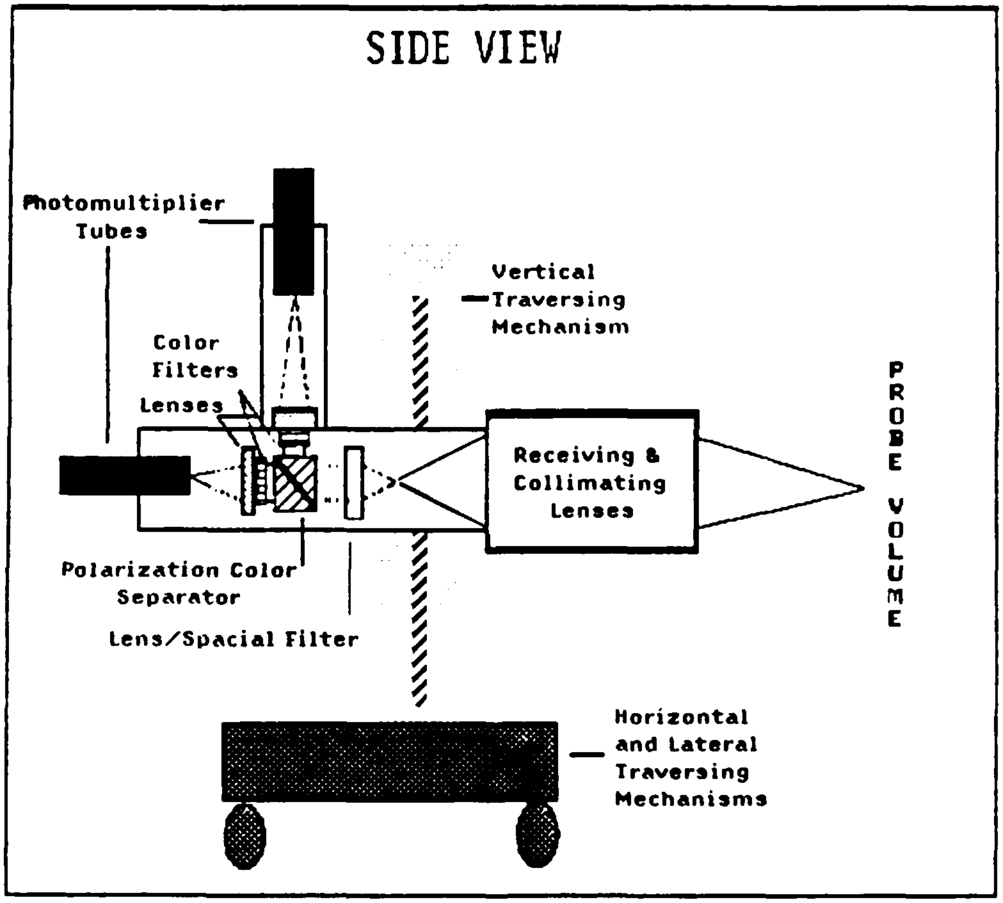


Figure 2.5 Receiving Optics Schematic

separated scattered light is now focused on to the photodetector aperture with a lens.

The photodetector converts the light signal into an electric signal. The photodetector used is a photomultiplier tube or PMT. The advantages of the PMT over the other types are low noise, high gain and broad bandwidth while its fragility and the need for a high voltage source are its disadvantages [Ref. 2:p. 45].

3. Test Section

The transonic test section is formed by inserting upper and lower ventilated walls in a subsonic indraft wind tunnel. The test section's dimensions are 13 cm by 25 cm.

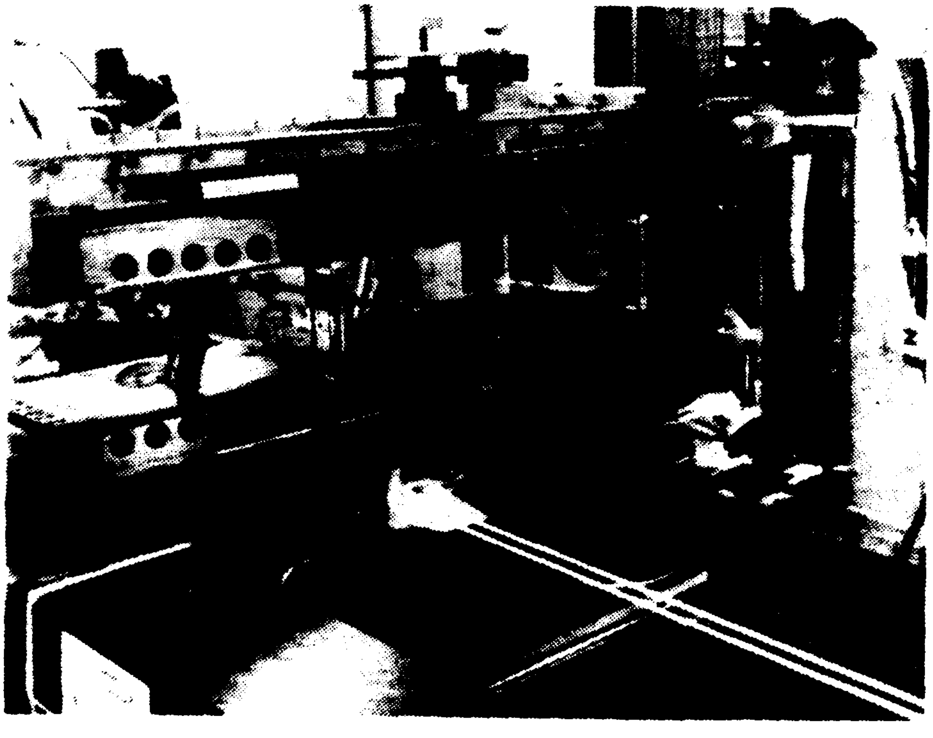


Figure 2.6 Transmitting Side of Test Section

A 7.5 cm chord NACA 0012 airfoil was mounted between the side walls at the centerline. This airfoil is attached at quarter chord to a external device to oscillate the airfoil through various angles of attack. The side walls of the test section are made of 2.5 cm thick plexiglas to allow complete and unobstructed optical access to the airfoil. See Figure 2.6. The mechanism to mount the model partially blocks the views of the leading edge of the model. However, this was not a limitation for the present study.

4. Signal Processing

The signal from the photodetector must be processed to obtain the frequency information in the scattered light. Figure 2.7 shows the block diagram of the signal processing components. The signal from the PMT is amplified by an EIN Model 403LA, linear amplifier to boost signal strength by 37 dB to transmit the signal over a long distance and to be of appropriate strength for signal processing.

A downmixing circuit is used next to decrease the effective frequency shift from 40 MHz to around 10 Mhz to improve the counter resolution. The two elements of the downmixer are a Tektronix SG503 Levelled Sine Wave Generator and a Hewlett-Packard, Model 10534A, Passive Diode Mixer. The expected Doppler frequency should be estimated to properly set the mixer frequency. Using the range of Mach numbers expected as -0.2 to +0.5, the free stream velocity range can be calculated using the set of equations in section III A and 20° C as the ambient total temperature. The velocity range resulting from this calculation is -67 to +167 meters per second. With the fringe spacing of 19.52 microns and the second equation on page 35, the estimated Doppler frequencies are -3.4 to +8.6 MHz. Using the equation on page 34, the signal and mixing frequency should be expected to fall within the range of 36.6 to 48.6 MHz. By using a starting mixing frequency of 35 MHz, the calculated signal frequency reaching the counters will

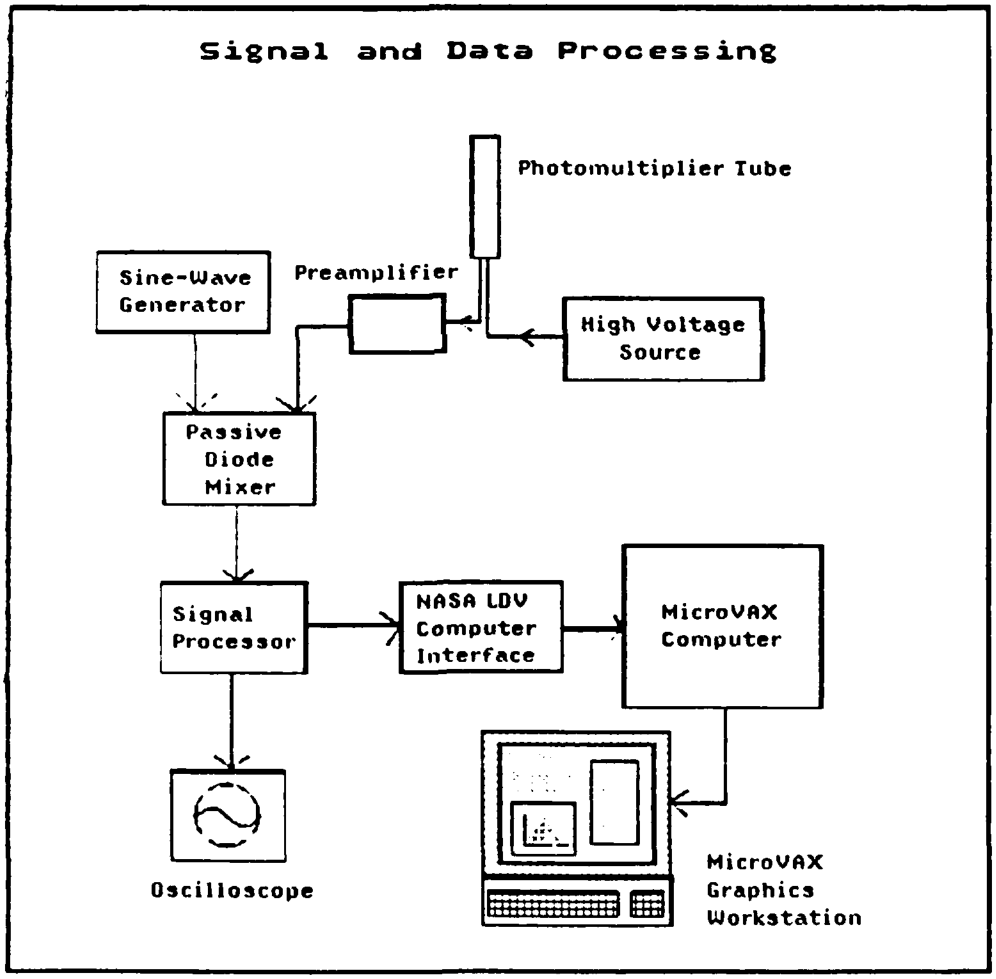


Figure 2.7 Signal and Data Processors

be 1.6 to 13.6 MHz. The mixing frequency should be adjusted whenever the tunnel conditions change or in attempts to obtain better counter resolution.

The next signal processing unit encountered is the Series 3000 Macrodyne LDV Processor. This device has several functions: variable high and low pass filters, input and output ports, threshold and overload adjustments and frequency counter with fringe passing selection switch. Both the low pass filters, to remove the high frequency noise, and the high pass filters, to reduce the low frequency pedestal from the signal, are adjustable to provide the optimal filtering of the signal. Once the signal has been thus conditioned, a frequency counter is used to measure the signal frequency and convert it to digital time data that can be sent to the data processor. The counter operates on two different systems to validate the Doppler signal: a three level detection scheme and a 5/8 or 10/16 comparison. The three level detection scheme shown in Figure 2.8 must be met before a signal is validated. A threshold level is set on the counter to be just above the level of the steady state noise. The signal must pass the following gates to be accepted: (1) exceeds the threshold setting (2) crosses zero and (3) exceeds a negative threshold. To pass this check, the selected number of fringe crossings must sequence through each check without error. This detection scheme is notably effective in noise discrimination. Once a signal is received and passes the first check, the counter starts to measure the Doppler period by opening a time gate and stopping at the switch set number of cycles. This setting of cycles is able to be selected from a switch on the counter control panel. Validated data from the switch being in the 10/16 cycles position tend to be better for strong Doppler bursts because the greatest number of cycles measured tends to reduce the effects of noise in the signal. However, this setting also restricts the motion of the particle through the probe volume and thus introduces artificial biases of

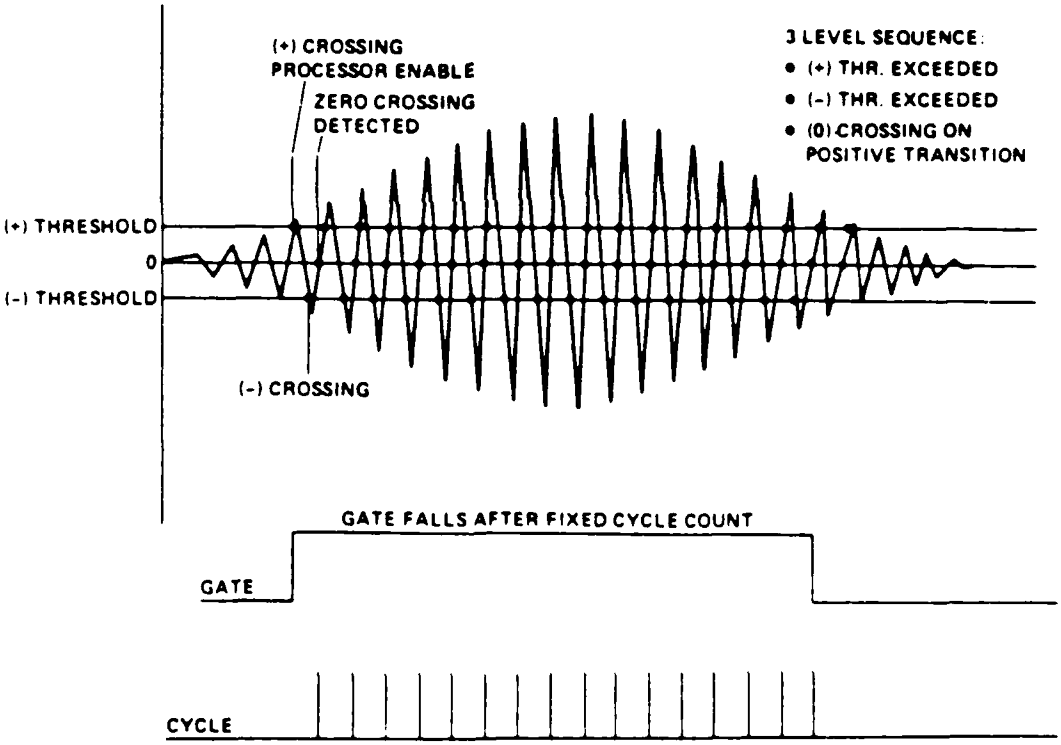


Figure 2.8 Three Level Counter Detection Scheme

the data. The particle must traverse the probe volume at its largest length to insure validated data for 16 fringe crossings. For this reason, present experiments were carried out with the switch set to the 5/8 position.

5. Data Processing Hardware

Once the data is received from the Macrodyne signal processors, it must be multiplexed and transferred to the computer for velocity calculations. The NASA LDV Computer Interface is now used to control the multiplexing of data, sequencing of data, provide a digital display to track successful counts of the Doppler frequency and, generally, control the flow of data between the signal processor and the Micro-VAX computer. Since a two component Laser Doppler Velocimeter is used in this experiment, the multiplexer sends four channels of data for each valid Doppler burst. The first data word contains the time information, the second contains any status information from the signal and the last two words contain the frequency data from each component. Due to the limited nature of this experiment, the first two words of data are not useful in further velocity calculations. Therefore, just the frequencies from each particle caused Doppler burst are used to calculate the respective particle velocities.

The Micro-VAX computer stores all acquired data in a buffer set up by acquisition programs. The next step after data acquisition is to calculate the velocities. These computer programs reduce the massive amounts of raw data obtained into meaningful velocity data that can be output in tabular form and graphical form.

The following is a simplified path followed by the numerous computer programs to reduce the data. At least 1000 samples from each component are required to obtain a statistically steady sample frequency. This frequency data must then be converted to the true Doppler frequency by the following equation:

$$f_d = f_s - f_B + f_M$$

where f_d is the true frequency resulting from the particle traversing the probe volume, f_s is the signal frequency measured by the counter, f_B is the Bragg shift frequency of 40 MHz and f_M is the frequency of the downmixer. The downmixer circuit permits decreasing of the effective frequency shift to values within the 1 to 10 MHz range. The true frequency data is then averaged over the samples taken to obtain a mean value and a sample standard deviation. Using this mean value, for each channel, all samples that are within 2 σ, two standard deviations, of the mean are saved and the rest are discarded. This procedure accounts for about 5% reduction of data and will insure samples that are too low or high will not bias the final results. A new mean and standard deviation are calculated for each channel. Next the probe volume dimensions and parameters are calculated with primary emphasis on the fringe spacing equation:

$$\delta = \lambda / [2 * \sin(\theta/2)]$$

where δ is the fringe spacing in microns, λ is the beam wavelength in microns and θ is the angle of intersection of the beams. Knowing the fringe spacing, the velocity can be determined using the relation:

$$u = f_d * \delta$$

where u is the velocity of the particle in meters per second and f_d is frequency of the particle measured by the counters in MHz.

Once the velocity components of the particle are determined streamwise and in the transverse directions, the computer stores the results in a data table for further analysis. The probe position is also saved in this data file. The entire process is now repeated at different locations of interest in the flow.

C. LABORATORY APPARATUS

Besides the Laser Doppler velocimeter and schlieren systems described earlier, additional instrumentation was used to obtain the wind tunnel pressure and temperature data. A 24 port scanivalve pressure transducer system was set up to measure the total and static pressures at various points inside the operating tunnel. This scanivalve system was driven by a digital to analog interface, built on the premises, which allowed computer control of the "stepping" and "homing" of the scanivalve.

Using the Digital to Analog interface on the MicroVAX computer, these signals to step the scanivalve system were transferred from a computer program to the physical unit for action. Once the stepper motor was sent the appropriate signal to move, the Analog to Digital interface board on the computer sampled the pressure transducer signals and loaded the data into the computer for conversion into pressure data and later analysis.

The atmospheric pressure was recorded from a mercury barometer with micrometer scale mounted on the laboratory wall near the wind tunnel. The total temperature was recorded on a mercury thermometer mounted on the barometer. Both of these measurements will soon be replaced with Paroscientific pressure sensors and thermocouples that will have a direct link with the computer.

A TSI Six-Jet Atomizer, Model 9306, was used to provide particle seeding for the LDV system.

Figure 2.9 shows the general laboratory setup and location of the data processing and ancillary equipment.

D. DATA PROCESSING

1. Introduction

The Laser Doppler velocimeter yields a large amount of raw data during its operation. This data must be stored

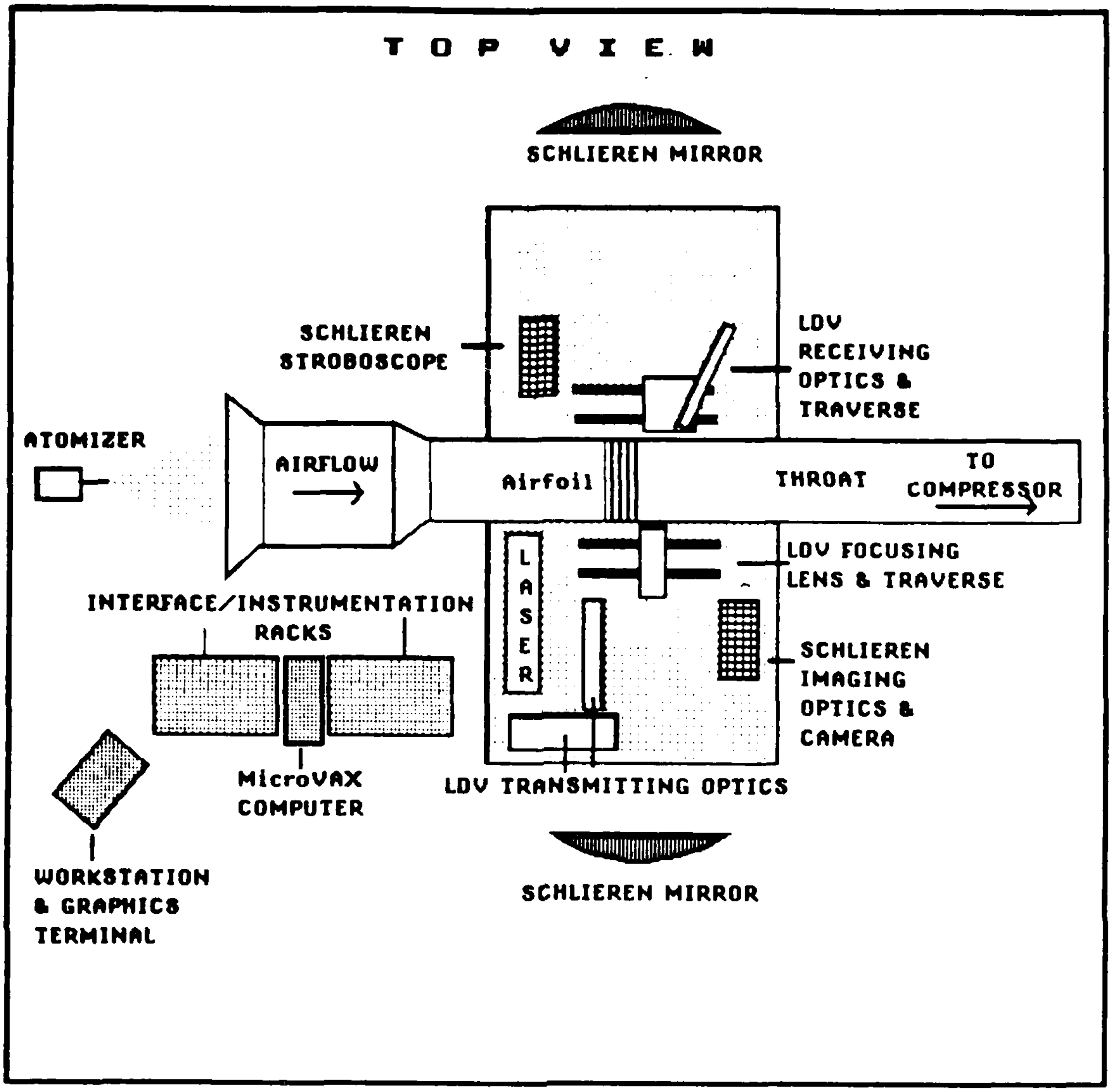


Figure 2.9 Laboratory Setup

in the computer in a logical sequence in order to properly calculate the velocities and all other associated data.

2. Computer Programs

The following discussion pertaining to the functioning of each of the computer programs that was written to accomplish the task of managing the vast amount of data in the most efficient and timely manner.

a. Master Program

MASTER controls the overall flow of data during each of the subroutine calls, acquires all the data and reduces it to a usable form.

First, the wind tunnel conditions are sampled by the use of various subroutines: DA11J1LAB, SETUP, GETDATA, DETACH and OUTPUTD. The traversing mechanism is then positioned with subroutine HWTRVT to move the probe volume of the LDV to the proper location in the flow. Next, the LDV data is acquired with subroutine HWWA. The samples are taken, velocities calculated using subroutine VELOCITY and a statistical mean and standard deviation are calculated. The data is then reduced with subroutine REDUCE_DATA to preclude the use of any velocities greater or less than two standard deviations away from the mean. Each of the two velocity components are then recorded along with the position in the flow. The velocity profiles are obtained from the data using post-acquisition processing programs. An option is given during the acquisition process to view a histogram of the resulting velocities to check on the distribution of data samples with subroutine HISTOGRAM. A sample histogram from a test run is shown in Figure 2.10.

b. Scanivalve Programs

First a signal is sent to the scanivalve to "home" to port 0 for the first pressure reading This is accomplished by using the subroutine DA11J1LAB. Next, a call

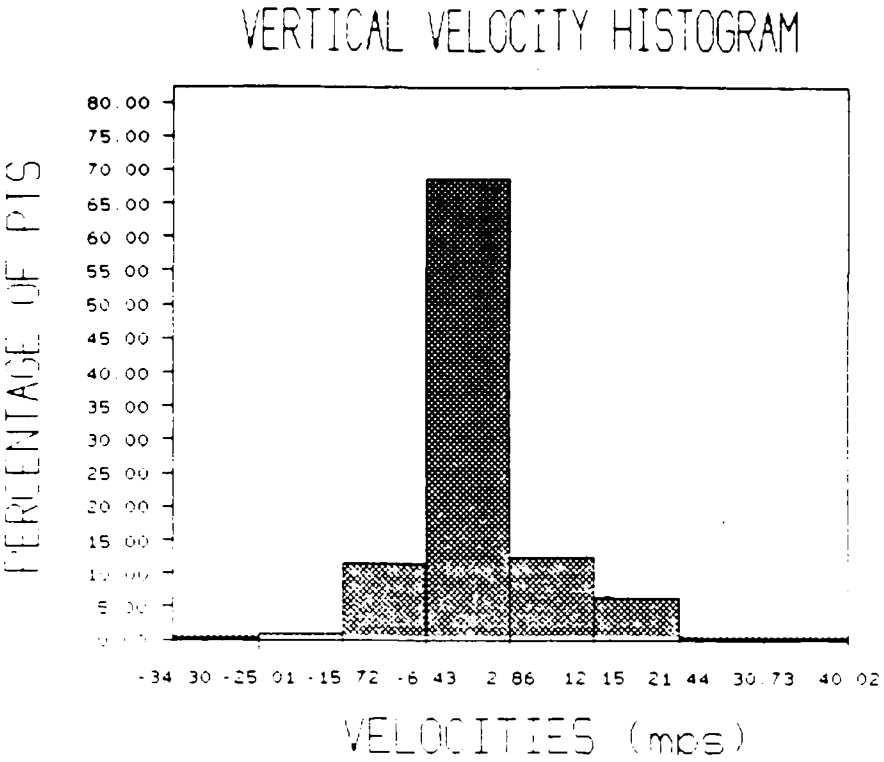


Figure 2.10 Sample Histogram

to subroutine SETUP is made to connect the computer's internal timing clock and Analog to Digital interface board to program control. Also, the sample acquisition rate and number of samples are interactively entered.

Now, the computer is ready to acquire data from the scanivalve system. Since there are 24 ports to sample on the scanivalve, a loop is entered in which samples are taken, scanivalve stepped and data reduced during each loop. Subroutines GETDATA and DA11J1LAB acquire the data and step the scanivalve respectively. DA11J1LAB has an adjustable time delay to give time for the pressure in each port to equalize after scanivalve stepping.

After all 24 ports are sampled, the mean and standard deviation voltages are calculated for each port. The calibration pressure attached to port 0 and the atmospheric pressure at port 1 are now entered interactively in consistent units; inches of mercury in this case. With these two pressures entered, a calibration line is calculated and the remaining voltages from the scanivalve system are converted to pressures. Using port 2, settling chamber stagnation pressure, and port 7, test section static pressure, a tunnel Mach number is calculated. See section III A for the equations used in these calculations. Next, the total temperature is entered and the free stream velocity is calculated.

For use in labelling a program generated plotting file for post-acquisition analysis, the angle of attack of the test section airfoil and current time are entered.

The last part of the scanivalve data acquisition programs close the loop on the analog to digital interface by detaching the clock and driver. A diagnostic output file is also generated for post-acquisition analysis.

c. LDV Data Acquisition Program

Both the transmitting and receiving optics of the Laser Doppler Velocimeter are mounted on three axis motorized traverses that are slaved together to move the probe volume to any point of interest in the flow. Using a pre-existing software driver for the Taskmaster traverse control unit, the optics could be moved at will from the computer workstation.

Although a pre-existing software driver was available for the NASA-LDV Computer Interface, significant changes had to be made to the code for it to work on the present system. The main Laser Doppler Velocimetry acquisition subroutine "DAMUX" was responsible for loading the incoming data from the multiplexer into data arrays. Once the raw frequency data is in arrays, it has to be decoded to correspond to usable numbers.

The time word output, as 14 bits, by the counter has a mantissa and exponent format: with a 10 bit mantissa and a 4 bit exponent. The computer reads it as a 16 bit word and adds two bits that are not used for data handling. These two bits are only used during direct computer links. This 16 bit word is output as a complementary word that is reversed; that is, the zeros are output as ones and the ones as zeros then the entire word is reversed (i.e. the least significant bit which is on the right most side bit is moved to the left most bit in the word). First, the word is made into its complement then both the mantissa and the exponent are reversed prior to decoding. The exponent is preset from the counter front panel switch. The following is the standard 16 bit word composition and what actually results from the output:

Correct output

0111010001 0111 01 = 465x27 = 59,520

Mantissa Exp NU

Actual output

01 0001 0111010001

NU Exp Mantissa

Translated by the computer

0100010111 0100 01 = 279x24 = 4,464

First step: Take its complement

10 1110 1000101110

NU Exp Mantissa

Next: reverse the order

0111010001 0111 01 = 59,520

Mantissa Exp NU

where NU stands for Not Used. In order to fill the sixteen bit word, the 14th and 15th bit are added to control data flow but have no significance to the value of the data. From this last 16 bit word is calculated the Doppler shift frequency. The output from the NASA-LDV Computer Interface is actually representative of the time elapsed for the particle to traverse 8 fringes of the probe volume. The Doppler period is expressed by the following equation:

$$t_d = 1/32\ (D_9D_8 \ldots D_0) \times 2^{EXP}\ \text{ns}$$

where $D_9 .. D_0$ represents the binary mantissa and EXP is the preselected exponent set on the counter. From the period, the Doppler frequency is calculated using the following equation:

$$f_d = 1000/t_d\ \text{MHz}$$

This Doppler frequency is then be used to calculate the velocity of the traversing particle. Since frequency shifting is used to discriminate the actual direction of the particle through the probe volume, the Doppler frequency must be adjusted to account for the Bragg and mixing frequencies. See equation on page 34 for this relationship. Once the actual frequency of the particle is determined, the fringe spacing is calculated by the first equation on page 35 and the velocity is found using the second equation on the same page.

d. LDV Support Programs

Three subroutines are used to provide the supporting functions to the LDV data acquisition. These are VELOCITY, HISTOGRAM and REDUCE_DATA.

VELOCITY is responsible for calculating the fringe spacing, extracting the particle frequency from the signal and then calculating the two velocities.

HISTOGRAM takes the velocity data and plots a histogram of the percentage of velocities that fall within a given range. The x axis of the histogram is divided equally into 8 segments from the minimum to maximum velocity values. The option to plot a velocity histogram is given after each LDV acquisition to spot check the distribution of the data. A nearly Gaussian or normal distribution is expected if the system is operating properly. This will also show the effects of noise as evidenced by the broad band of the histogram values.

REDUCE_DATA reduces the velocity data to only those values falling within two standard deviations of the mean. This results in a 4.5% rejection of data which will reduce the effect of high frequency noise on the counters and tend to reduce the effect of very large particles moving slowly through the probe volume.

III. DATA ACQUISITION

A. TUNNEL CONDITIONS

In order to obtain good correlation between the schlieren photographs and the Laser Doppler Velocimeter obtained data, the operating tunnel conditions would have to be nearly identical. Two sets of conditions were chosen: one corresponding to an approximate Mach number of 0.3 and the other 0.5. Once the tunnel was running at operating conditions, the computer program MASTER calculates the test section Mach number and free stream velocity according to the following relationships. Assuming isentropic flow in the test section, the section Mach number, M, is calculated from the stagnation pressure, P_o, and the static pressure, P, using the following relationship:

$$M = (([P_o/P]^{(\gamma-1/\gamma)}-1)x(2/[\gamma-1]))^{1/2}$$

where $\gamma = 1.4$. The resulting calculated Mach numbers for the two conditions are as follows: 0.295 and 0.450. From these Mach numbers, the corresponding free stream velocities are calculated.

$$V = M * a$$

where a is the sonic velocity.

$$a = (\gamma RT)^{1/2}$$

where R is the universal gas constant and T is the static temperature in the test section. The static temperature can be calculated from the total temperature, T_O, measured on the wall near the tunnel by the next relationship:

$$T = T_O/(1 + [(\gamma-1)/2]*M^2)$$

B. SCHLIEREN PHOTOGRAPHS

1. Introduction

Once the wind tunnel was running at the first of two operating conditions, manometer board readings were recorded and Mach number calculated using the first equation on the preious page. The schlieren photographs were then taken at four different angles of attack and three different strobe intensities to obtain the optimal photographs.

2. Test Conditions and Procedure

Using the procedures detailed in section II A, the schlieren photographs were taken at the settings and conditions listed in Table I, below.

Upon completion of the wind tunnel runs, the 35mm film was developed and printed. From these schlieren photographs, the conditions resulting in the best depiction of the wake profiles were chosen to represent each Mach number and angle of attack. These are annotated with an asterisk, *, in Table I, below.

Using these photographs, a short HP-41 CX calculator program was written to relate photographic dimensions to

TABLE I. SCHLIEREN PHOTOGRAPH PARAMETERS

	M	α (deg)	Strobe Intensity (Hz)
*	0.30	0.0	20
	0.30	0.0	70
	0.30	0.0	150
	0.30	7.7	20
	0.30	7.7	70
*	0.30	7.7	150
*	0.30	15.5	20
	0.30	15.5	70
	0.30	15.5	150
	0.30	23.6	20
	0.30	23.6	70
	0.30	23.6	150
*	0.45	0.0	20
	0.45	0.0	70
	0.45	0.0	150
*	0.45	7.7	20
	0.45	7.7	70
	0.45	7.7	150
	0.45	15.5	20
*	0.45	15.5	70
	0.45	15.5	150

real test section dimensions. Once the real dimensions of the wake were known, the Taskmaster traverse mechanism was calibrated. This yielded a 1:1 relationship in the chordwise direction and a 1:4 relationship in the vertical direction. This means that the Taskmaster digital control panel reads true values for the X direction, chordwise, and four times the real value in the Z direction, vertical. Arbitrarily choosing the trailing edge of the airfoil at 0° α as an initialization point for the Taskmaster, a table of LDV sampling points was made for the first three angles of attack. See Table II, below, for a sample of the Taskmaster table at $\alpha = 7.7^{\circ}$

Table II. TASKMASTER LDV SAMPLING LOCATIONS

Alpha = 7.7 deg

Points Sampled	X	Z -->																	
20	2.00	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5	-1	-1.5	-2	-2.5
20	1.50	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5	-1	-1.5	-2	-2.5
20	1.00	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5	-1	-1.5	-2	-2.5
20	.50	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5	-1	-1.5	-2	-2.5
20	.00	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5	-1	-1.5	-2	-2.5
19	-.50	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5	0	-.5		-1.5	-2	-2.5
17	-1.00	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5				-1.5	-2	-2.5
17	-1.50	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2	1.5	1	.5				-1.5	-2	-2.5
11	-2.25	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5										-2.5
12	-3.00	7	6.5	6	5.5	5	4.5	4	3.5	3	2.5	2									-2.5
===																					
176	(Total locations sampled)																				

C. LASER DOPPLER VELOCIMETRY DATA ACQUISITION

1. Calibration of the LDV System

The relationship between the Doppler frequency and the velocity of the particle traversing the probe volume is given by the second equation on page 35. The particle velocity is directly proportional to the Doppler frequency by a factor of the fringe spacing. By the first equation on page 35, fringe spacing is only a function of the transmitting lens angle and wavelength of the light. Since the wavelength of the light is accurately known to within

0.05 microns, or 0.01%, an accurate lens angle is all that is needed for calibration [Ref. 4].

The following relationship was used to calculate the lens angle:

$$\theta = 2 * \tan^{-1} (E * d / (2 * f))$$

where E is the beam expansion ratio, 1.0 in this case: d is the incident beam spacing, preset at 13 mm with TSI Model 9113-13; and f is the wavelength of the light.

The lens angle in this case was not accurate because the focal length of the focusing lens was not known. Therefore, calibration was necessary to insure accurate velocity measurements were obtained.

The calibration method chosen was to operate the wind tunnel at known conditions, calculate the free stream velocity from the scanivalve pressure data and then move the probe volume via the Taskmaster to coincide with a position in which the free stream velocity could be measured. From this procedure, a constant could be applied to the velocity equation as follows:

$$u = f_d * \delta * (\text{Calibration Constant})$$

This calibration constant will compensate for the estimated focusing lens focal length.

2. Test Conditions and Procedure

Once calibration is completed, the tunnel is brought to the test conditions, the Taskmaster is moved to the first sample point and the MASTER computer program is run with the adjusted fringe spacing to reflect the calibration.

At each sampling point, 1000 samples will be taken to insure statistical accuracy of the mean flow calculations. The scheduled sampling points can be seen in Table II, page 48. After each Taskmaster traverse, the oscilloscope is checked to insure no drastic loss of the signal occurred during the movement. If the Doppler signal decreases, slight adjustments to the receiving optics traverse may be needed to regain the signal strength. To optimize the data received, slight adjustments may have to be made, also, to the selected mixing frequency, mixing frequency amplitude, counter threshold setting or high and low pass filter settings.

After Table II data points are sampled, the airfoil angle of attack or test Mach number is changed and the above procedure is repeated.

IV. RESULTS AND DISCUSSION

A. SCHLIEREN

The first thesis goal was attained early enough to provide qualitative information concerning the airflow around and in the wake of the airfoil. Schlieren 35mm photographs were produced for both selected mach numbers, 0.3 and 0.45, and for four different angles of attack; 0°, 7.7°, 15.5° and 23.6°. The best photographs for each Mach number and the first three angles of attack are displayed in Appendix A, pages 55-57. From these photographs, LDV sampling points were chosen as seen from Table II on page 48.

Three turbulent areas are evident on all the schlieren photographs. As seen in Figure A.4 on page 56, the airflow is slightly disturbed upstream of the airfoil due to the presence of a pitot-static tube extending into the airflow just forward of the airfoil. This tube has been subsequently removed to clean up the flow. At the top of the same photograph, a wedge shaped disturbance can be seen. This and the lower floor turbulent layer are results of the transonic wall slottings found on both the roof and the floor of the test section. Although a concern for later unsteady flow studies, these two areas do not interfere with the wake profiles resulting from a steady flow analysis.

B. LASER DOPPLER VELOCIMETER

Due to equipment problems and time constraints, the LDV data acquisition part of this thesis was not able to be accomplished. All systems are now operational, to include computer programs, however, there is inadequate time to acquire data and analyze the velocity profiles obtained.

Calibration runs were made with a limited degree of success when the wind tunnel was run under Mach 0.3 and the Laser was not powered over 2 watts. The problem with the Laser is easily repaired, however, time consuming. The upper limit on the wind tunnel Mach number is caused by the evaporation of the seeding particles in high speed flow. This, too, can be easily overcome by the proper selection of seeding particles and related atomizer.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The thesis provided invaluable assistance to the dynamic stall project at the Fluid Mechanics Laboratory. The schlieren photography was able to provide the qualitative data on the steady state flow field. With this accomplished, the schlieren system in this laboratory was able to be easily tailored to the next step: unsteady flow analysis.

The LDV work provided the laboratory with the essential computer programs, hardware interfaces and drivers, optics systems and auxiliary required equipment to enable the project to proceed with very little additional work to achieve the steady state velocity profiles. The ground work has been accomplished to enable the project to continue forward with the systems on hand to study the effects of dynamic stall in both the steady and unsteady environments.

B. RECOMMENDATIONS

The schlieren system is fully functional, however, the Laser Doppler Velocimeter system still requires a little additional effort to make it productive. The Laser doughnut condition needs to be addressed to enable higher powers to be utilized. The types of seeding available to the system needs to be addressed to insure operability of the tunnel and data acquisition systems above Mach 0.3. With these minor fixes and a little fine adjustments of the signal processing equipment, the LDV is ready for action.

APPENDIX A: SCHLIEREN PHOTOGRAPHS

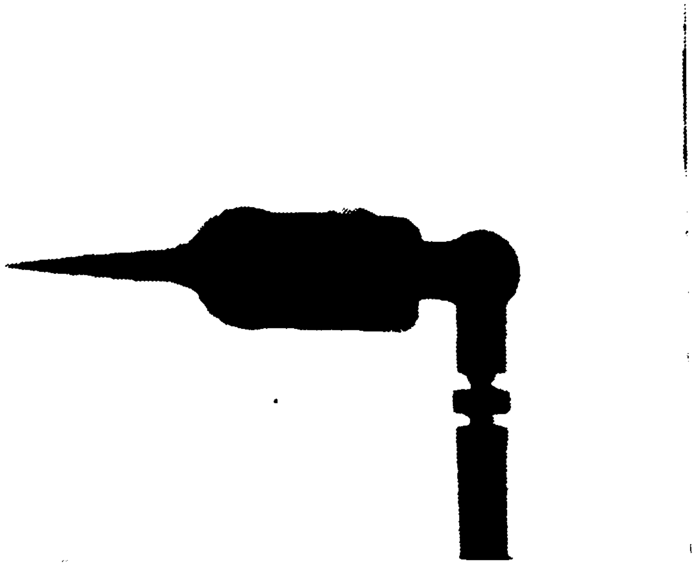


Figure A.1 Mach Number = 0.3, α = 0°

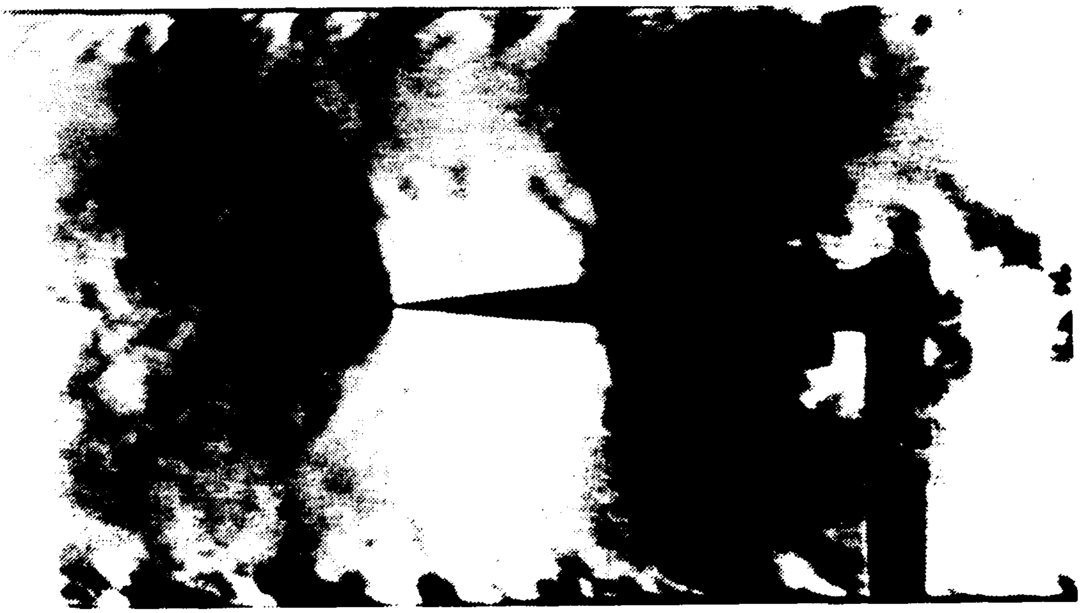


Figure A.2 Mach Number = 0.45, α = 0°

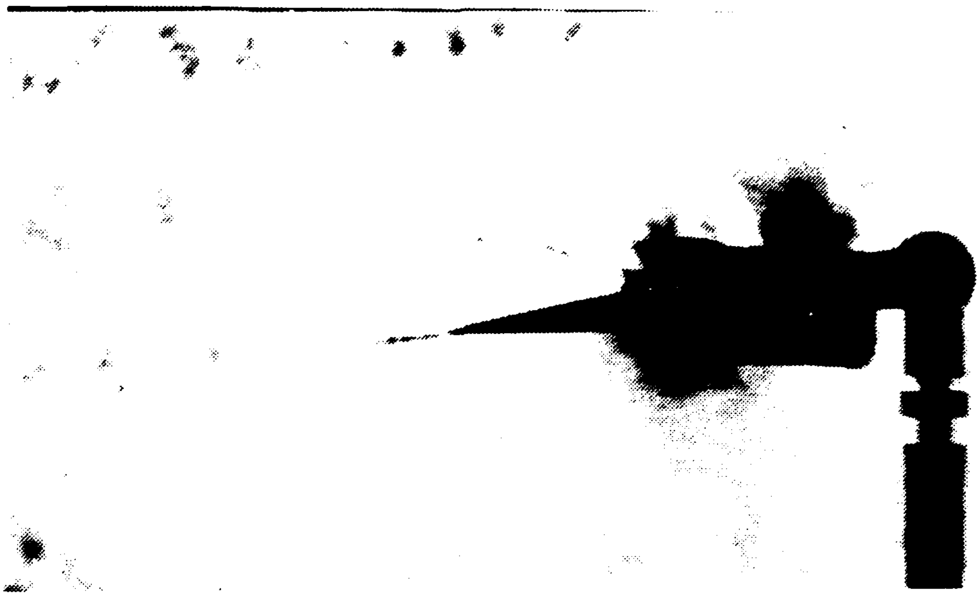


Figure A.3 Mach Number = 0.3, α = 7.7°

Figure A.4 Mach Number = 0.45, α = 7.7°

Figure A.5 Mach Number = 0.3, α = 15.5°

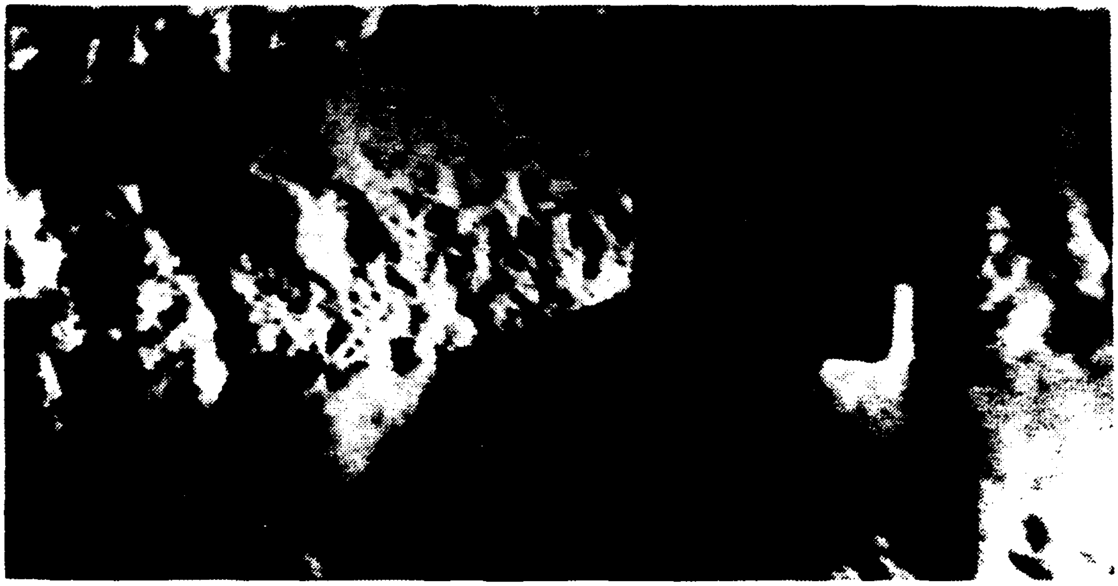


Figure A.6 Mach Number = 0.45, α = 15.5°

APPENDIX B: COMPUTER PROGRAMS

```
      PROGRAM MASTER
C-----------------------------------------------------------------------
C  Master Program written by Maj Richard R Ryles, NPS Student 28 Apr 87 |
C                                             Last Update  6 Sep 87   |
C                                                                     |
C  This program will gather data from the scanivalve system, move the |
C  Taskmaster and acquire LDV velocity data.                          |
C  This program should be used with subroutine SETUP, GETDATA, DETACH, OUTPUTD
C  HWTRVT, HWWA, HISTOGRAM and VELOCITY.                              |
C-----------------------------------------------------------------------
      INCLUDE 'SYS$LIBRARY:LIOSET.FOR'
C-----------------------------------------------------------------------
C       Declare variables                                             |
C-----------------------------------------------------------------------
      INTEGER DEV_ID,CLK_ID
      INTEGER NCHANL,HISTO,BLUE_GREEN
      INTEGER NPT,NPTS,NSAMPLES,COUNT
      INTEGER I,J,NSTEPS,IER,ANSWER,END,TALLY,TIME
      REAL    VOLTAGES(5000),U_U(200),V_V(200),U_V(200),DELX,DELZ
      REAL*8  SUM,AMEAN,X,XX,Y,YY,STD_DEV,STD_DEV_V,STD_DEV_U,SUM_V,SUM_U
      REAL*8  SUM_VERT,SUM_HORIZ
      REAL    RATE,SCANI_VOLT(0:23,2),SCANI_PRESS(0:23),VERT_VEL(8200)
      REAL    CALIB_PRESS,SLOPE,INTERCEPT,HORIZ_VEL(8200),FINAL_V,FINAL_U
      REAL    X_POSITION,Z_POSITION,MEAN(2),M,FREE_STREAM,TO,ATMOS_PRESS
      REAL    X_LOC(200),Z_LOC(200),U(200),V(200),ALPHA,ZERO,CORR_FACTOR
      REAL*8  UU,VV,UV
      CHARACTER*1 ANS,VOLT_ANS
      LOGICAL HOME
C-----------------------------------------------------------------------
C       Scanivalve data acquisition                                   |
C-----------------------------------------------------------------------
      OPEN(UNIT=99,FILE='RESULTS.DAT',STATUS='NEW')
      TALLY = 0
      HOME = .TRUE.
      NSTEPS = 0
      CALL DA11J1LAB (HOME,NSTEPS,IER)
      IF (IER.NE.1) GOTO 170
      ANSWER = 0
      WRITE (*,*) 'Skip to Taskmaster? (y/n) [DEF - No]'
      READ (*,1) ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') GOTO 90
      WRITE (*,*) 'Print the scanivalve diagnostic data to a file? (Y/N)',
     &            ' [DEF - No]'
      READ (*,1) VOLT_ANS
      CALL SETUP (CLK_ID,DEV_ID,RATE,NCHANL,NPT,NPTS)
10    CONTINUE
      HOME = .FALSE.
      NSTEPS = 1
      DO 50 J = 0, 23
         WRITE (*,2) J
         IF (J.NE.0) CALL DA11J1LAB (HOME,NSTEPS,IER)
         IF (IER.NE.1) GOTO 150
20       CONTINUE
         CALL GETDATA (DEV_ID,NPT,NPTS,VOLTAGES)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Calculate mean and standard deviation
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SUM = 0.0
         DO 30 I = 2, NPT
            SUM = SUM + VOLTAGES(I)
```

```
30       CONTINUE
         AMEAN = SUM / FLOAT(NPT-1)
         SUM = 0.D0
         DO 40 I = 2, NPT
                X = VOLTAGES(I) - AMEAN
                Y = X * X * (1.0/FLOAT(NPT-1))
40              SUM = Y + SUM
         STD_DEV = SQRT(SUM)
         IF (VOLT_ANS.EQ.'y'.OR.VOLT_ANS.EQ.'Y') CALL OUTPUTD (RATE,NPT,
     &                                  NCHANL,AMEAN,STD_DEV,J,VOLTAGES)
         SCANI_VOLT(J,1) = AMEAN
50       SCANI_VOLT(J,2) = STD_DEV
      CALL DETACH (CLK_ID,DEV_ID)
      IF(VOLT_ANS.EQ.'Y'.OR.VOLT_ANS.EQ.'y') THEN
          OPEN(UNIT=15,FILE='SCANI_VOLT.DAT',STATUS='NEW')
          DO 60 I = 0, 23
60           WRITE (15,3) I, SCANI_VOLT(I,1), SCANI_VOLT(I,2)
      ENDIF
C * * * * * * * * * * * * * * * * * * * * * * *
C     Calculate scanivalve calibration line
C     and convert all voltages to pressure
C * * * * * * * * * * * * * * * * * * * * * * *
      WRITE (*,*) 'Enter Scanivalve Calibration Pressure:'
      READ (*,*) CALIB_PRESS
      WRITE(*,*)'Enter Calibration Pressure Correction Factor:'
      READ(*,*)CORR_FACTOR
      CALIB_PRESS = CALIB_PRESS + CORR_FACTOR
      WRITE(*,*)'Enter atmospheric pressure:'
      READ(*,*) ATMOS_PRESS
      SLOPE = (ATMOS_PRESS-CALIB_PRESS)/(SCANI_VOLT(1,1)-SCANI_VOLT(0,1))
      INTERCEPT = CALIB_PRESS - SCANI_VOLT(0,1)*SLOPE
      WRITE(99,*)'CALIB_PRESS',CALIB_PRESS
      WRITE(99,*)'CORR_FACTOR',CORR_FACTOR
      WRITE(99,*)'ATMOS_PRESS',ATMOS_PRESS
      OPEN(UNIT=16,FILE='SCANIPORT.DAT',STATUS='NEW')
      DO 70 I = 0, 23
             SCANI_PRESS (I) = SCANI_VOLT(I,1) * SLOPE + INTERCEPT
70       WRITE(16,*)I, SCANI_PRESS(I)
      CLOSE(16)
      IF (VOLT_ANS.EQ.'Y'.OR.VOLT_ANS.EQ.'y') THEN
         DO 80 I = 0, 23
80                  WRITE (15,4) I,SCANI_PRESS(I),SCANI_VOLT(I,2)*SLOPE+INTERCEPT
      ENDIF
      CLOSE(15)
      WRITE(*,*)' '
      WRITE(*,*)'Equation of the calibration line ='
      WRITE(*,*)'Y =',slope,' X +',intercept
      WRITE(99,*)'Y =',slope,' X +',intercept
      WRITE(*,*)' '
C * * * * * * * * * * * * * * * * * * * * * * *
C     Calculate tunnel Mach and velocity
C * * * * * * * * * * * * * * * * * * * * * * *
      IF(SCANI_PRESS(7).EQ.0.0) SCANI_PRESS(7) = 1.0
      IF(SCANI_PRESS(2).LT.SCANI_PRESS(7)) GOTO 85
      M = SQRT(((SCANI_PRESS(2)/SCANI_PRESS(7))**(2./7.)-1.)*5.)
85    WRITE(*,*)'Tunnel Mach Number =',M
      WRITE(99,*)'Tunnel Mach Number =',M
      WRITE(*,*)' '
      WRITE(*,*)'Enter Total Temperature (C)'
      READ(*,*) TO
      WRITE(99,*)'TOTAL TEMP', TO
      WRITE(*,*)' '
      TO = TO + 273.15
      FREE_STREAM = M * SQRT(401.52*TO/(1.+(M**2/5.)))
      WRITE(*,*)'Free Stream Velocity (mps) =',FREE_STREAM
      WRITE(99,*)'Free Stream Velocity (mps) =',FREE_STREAM
      IF(FREE_STREAM.EQ.0.0) FREE_STREAM = 1.E-10
      WRITE(*,*)' '
```

```
      WRITE(*,*)'Enter angle of attack (deg):'
      READ(*,*) ALPHA
      WRITE(99,*)'ALPHA', ALPHA
      WRITE(*,*)'Enter Current Time: (4 digit integer)'
      Read(*,*) TIME
      WRITE(99,*)'TIME',TIME
      WRITE (*,*) 'Skip to end? (y/n) [DEF - No]'
      READ (*,1) ANS
      IF (ANS.EQ.'y'.OR.ANS.EQ.'Y') GOTO 160
C-----------------------------------------------------------------------
C        Taskmaster mover and LDV data acquisition                     |
C-----------------------------------------------------------------------
90    WRITE (*,*) 'Which color is connected to the left (#3) channel?',
     &            ' (Blue=1,Green=0)'
      READ (*,9) BLUE_GREEN
      WRITE(99,*)'BLUE_GREEN',BLUE_GREEN
      IF(FREE_STREAM.EQ.0.0) FREE_STREAM = 1.E-6
      OPEN (UNIT=25,FILE='FINAL.DAT',STATUS='NEW')
      WRITE (25,*) '    Taskmaster                  (Final.Dat)'
      WRITE (25,*) '     pos (in)     velocity/std dev  velocity/std dev'
      WRITE (25,16)
      WRITE (*,*) 'Skip to LDV? (y/n) [Def - NO]'
      READ (*,1) ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') GOTO 110
      IF(TALLY.EQ.0) THEN
         WRITE(*,*)'Enter X-Position for calibration'
         READ(*,*) X_POSITION
         WRITE(*,*)'Enter Z-Position for calibration'
         READ(*,*) Z_POSITION
      ENDIF
100   CALL HWTRVT (DELX,DELZ)
      WRITE (*,*) 'Move the Taskmaster again prior to LDV Acquisition? (Y/N)'
      READ (*,1) ANS
C * * * * * * * * * * * * * * * * * * * * *
C     Record position of LDV at all times
C * * * * * * * * * * * * * * * * * * * * *
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
         X_POSITION = X_POSITION + DELX
              Z_POSITION = Z_POSITION + DELZ
              GOTO 100
      ENDIF
C * * * * * * * * * * * * * * * * * * * * *
C     Acquire LDV data
C * * * * * * * * * * * * * * * * * * * * *
110   CALL HWWA (NSAMPLES,ANSWER,BLUE_GREEN)
C * * * * * * * * * * * * * * * * * * * * *
C     Increment position and record it
C     Read velocity data, sum it and find
C        standard deviation
C * * * * * * * * * * * * * * * * * * * * *
      X_POSITION = X_POSITION + DELX
      Z_POSITION = Z_POSITION + DELZ
      SUM_VERT = 0.D0
      SUM_HORIZ = 0.D0
      OPEN (UNIT=35,FILE='VELOCITY.DAT',STATUS='OLD')
      DO 120 I = 1, NSAMPLES
         IF (BLUE_GREEN.EQ.0) READ(35,7) VERT_VEL(I),HORIZ_VEL(I)
         IF (BLUE_GREEN.EQ.1) READ(35,7) HORIZ_VEL(I),VERT_VEL(I)
         SUM_VERT = SUM_VERT + VERT_VEL(I)
120      SUM_HORIZ = SUM_HORIZ + HORIZ_VEL(I)
      CLOSE(35)
      FINAL_V = SUM_VERT / FLOAT(NSAMPLES)
      FINAL_U = SUM_HORIZ / FLOAT(NSAMPLES)
      SUM_V = 0.D0
      SUM_U = 0.D0
      UU = 0.D0
      VV = 0.D0
      UV = 0.D0
```

```
      DO 130 I = 1, NSAMPLES
         X = VERT_VEL(I) - FINAL_V
         XX = HORIZ_VEL(I) - FINAL_U
         Y = X * X * (1.0/FLOAT(NSAMPLES-1))
         YY = XX * XX * (1.0/FLOAT(NSAMPLES-1))
         UU = UU + ((HORIZ_VEL(I)**2)-(FINAL_U**2))
         VV = VV + ((VERT_VEL(I)**2)-(FINAL_V**2))
         UV = UV + ((HORIZ_VEL(I)*VERT_VEL(I))-(FINAL_U*FINAL_V))
         SUM_V = Y + SUM_V
130      SUM_U = YY + SUM_U
      UU = UU / FLOAT(NSAMPLES)
      VV = VV / FLOAT(NSAMPLES)
      UV = UV / FLOAT(NSAMPLES)
      STD_DEV_V = SQRT (SUM_V)
      STD_DEV_U = SQRT (SUM_U)
      WRITE(*,*) ' '
      WRITE (*,*)'Velocities and std devs ='
      WRITE (*,*)'U dir',FINAL_U,STD_DEV_U
      WRITE (*,*)'V dir',FINAL_V,STD_DEV_V
      WRITE (99,*)'Velocities and std devs ='
      WRITE (99,*)'U dir',FINAL_U,STD_DEV_U
      WRITE (99,*)'V dir',FINAL_V,STD_DEV_V
      WRITE (*,*)' '
      WRITE(*,*)'First cut RMS values'
      WRITE(*,*)'U-RMS,V-RMS',UU,VV
      WRITE(*,*)'UV-RMS',UV
      WRITE(99,*)'First cut RMS values'
      WRITE(99,*)'U-RMS,V-RMS',UU,VV
      WRITE(99,*)'UV-RMS',UV
C * * * * * * * * * * * * * * * * * * * * * * * *
C     Eliminate data outside two sigma from mean
C * * * * * * * * * * * * * * * * * * * * * * * *
      IF (BLUE_GREEN.EQ.1) THEN
             MEAN(1) = FINAL_U
             MEAN(2) = FINAL_V
             CALL REDUCE_DATA (NSAMPLES,COUNT,END,MEAN,STD_DEV_U,STD_DEV_V,
     &                  BLUE_GREEN)
             FINAL_U = MEAN(1)
             FINAL_V = MEAN(2)
             WRITE (99,*)'Reduced velocities and std devs ='
             WRITE (99,*)'U dir',FINAL_U,STD_DEV_U
             WRITE (99,*)'V dir',FINAL_V,STD_DEV_V
      ELSE
             MEAN(2) = FINAL_U
             MEAN(1) = FINAL_V
             CALL REDUCE_DATA (NSAMPLES,COUNT,END,MEAN,STD_DEV_V,STD_DEV_U,
     &                   BLUE_GREEN)
             FINAL_U = MEAN(2)
             FINAL_V = MEAN(1)
             WRITE (99,*)'Reduced velocities and std devs ='
             WRITE (99,*)'U dir',FINAL_U,STD_DEV_U
             WRITE (99,*)'V dir',FINAL_V,STD_DEV_V
      ENDIF
C * * * * * * * * * * * * * * *
C     Calculate RMS values
C * * * * * * * * * * * * * * *
      OPEN(UNIT=43,FILE='VELOCITY.DAT',STATUS='OLD')
      UU = 0.D0
      VV = 0.D0
      UV = 0.D0
      DO 210 I = 1, END
         IF (BLUE_GREEN.EQ.0) READ(43,7) VERT_VEL(I),HORIZ_VEL(I)
         IF (BLUE_GREEN.EQ.1) READ(43,7) HORIZ_VEL(I),VERT_VEL(I)
         UU = UU + ((HORIZ_VEL(I)**2)-(FINAL_U**2))
         VV = VV + ((VERT_VEL(I)**2)-(FINAL_V**2))
 210     UV = UV + ((HORIZ_VEL(I)*VERT_VEL(I))-(FINAL_U*FINAL_V))
      CLOSE(43)
      UU = UU / FLOAT(END)
```

```fortran
      VV = VV / FLOAT(END)
      UV = UV / FLOAT(END)
      WRITE(*,*)' '
      WRITE(*,*)'Reduced RMS values'
      WRITE(*,*)'U-RMS,V-RMS',UU,VV
      WRITE(*,*)'UV-RMS',UV
      WRITE(99,*)'U-RMS,V-RMS',UU,VV
      WRITE(99,*)'UV-RMS',UV
      WRITE (25,8) X_POSITION,Z_POSITION,FINAL_U,STD_DEV_U,FINAL_V,STD_DEV_V,
     &             UU,VV,UV
      TALLY = TALLY + 1
C * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Options to view histogram and take data again
C * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(*,*)' '
      WRITE (*,*) 'Do you want a Histogram? (1/0)'
      READ (*,*) HISTO
      IF (HISTO.EQ.1) CALL HISTOGRAM(COUNT,END,BLUE_GREEN)
      WRITE (*,6)
      READ (*,1) ANS
      IF (ANS.EQ.'n'.OR.ANS.EQ.'N') GOTO 140
      GOTO 100
140   CLOSE (25)
      CLOSE(99)
C * * * * * * * * * * * * * * * * * * * * * * * * *
C     Convert output data into proper format for
C     "PLOTCV" Vector Plot of velocity profiles
C * * * * * * * * * * * * * * * * * * * * * * * * *
      OPEN(UNIT=45,FILE='FINAL.DAT',STATUS='OLD')
      READ(45,14)
      OPEN(UNIT=46,FILE='GRA.DAT',STATUS='NEW')
      WRITE(46,13) M,FREE_STREAM,ALPHA,TIME,11,16,10
      ZERO = 0.0
      DO 150 I=1,TALLY
             READ(45,11)X_LOC(I),Z_LOC(I),U(I),V(I),U_U(I),V_V(I),U_V(I)
             X_LOC(I)=X_LOC(I)+3.0
             Z_LOC(I)=(Z_LOC(I)+2.5)*.24
             U(I) = U(I) / FREE_STREAM
             V(I) = V(I) / FREE_STREAM
             U_U(I) = U_U(I) / FREE_STREAM
             V_V(I) = V_V(I) / FREE_STREAM
             U_V(I) = U_V(I) / FREE_STREAM
150          WRITE(46,12)I,X_LOC(I),Z_LOC(I),U(I),V(I),ZERO,U_U(I),V_V(I),ZERO,
     &             U_V(I),ZERO
      CLOSE(45)
      CLOSE(46)
160   WRITE (*,5)
C--------------------------------------------------------------------------
C        Format statements                                                |
C--------------------------------------------------------------------------
1     FORMAT (A1)
2     FORMAT (1X,'Scanivalve # ',I2)
3     FORMAT (1X,'Port #',I2,3X,'Average Voltage =',F12.4,3X,
     &           'Standard Deviation =',F12.4)
4     FORMAT (1X,'Port #',I2,3X,'Pressure =',F12.4,' (psi)',3X,
     &           'Standard Deviation =',F12.4,' (psi)')
5     FORMAT (//1X,'This is the END!'//)
6     FORMAT (/1X,'Move taskmaster again and take data? (y/n) [DEF - Y]'/)
7     FORMAT (1X,2F11.4)
8     FORMAT (1X,2F7.2,F10.2,F8.2,F10.2,F8.2,1P3E12.4)
9     FORMAT (I1)
11    FORMAT(1X,2F7.2,F10.2,8X,F10.2,8X,1P3E12.4)
12    FORMAT(1X,I3,1P10E12.4)
13    FORMAT(1X,'GRA.DAT, MACH NO = ',F4.2,', FREE STREAM VELOCITY = ',F5.1,
     &           ', ALPHA = ',F4.1,/1X,'September 9, 1987',4x,i4,' hrs.'
     &        //4X,'NPROF  NSAMPL   NCOEF'/7X,I2,6X,I2,6X,I2/1X,' ID    X-POS',
     &        '        Z-POS          U          V          W          U↑2',
     &        '          V↑2        W↑2        UV          UW')
```

```
14      FORMAT(//)
16      FORMAT(6X,'x',6X,'z',12X,'u',17X,'v',11X,'u↑2',9X,'v↑2',10X,'uv')
        STOP
C------------------------------------------------------------------------
C         Scanivalve Error Statements                                   |
C------------------------------------------------------------------------
170     IF(HOME) THEN
          WRITE (*,*)'Problem in homing Scanivalves'
        ELSE
                  WRITE (*,*)'Problem in stepping Scanivalves'
        ENDIF
        IF (IER.EQ.2) WRITE (*,*)'Data transfer failed in J-Board'
        IF (IER.EQ.3) WRITE (*,*)'Failure in detaching J-Board'
        IF (HOME) GOTO 10
        GOTO 20
        END
```

```
      SUBROUTINE SETUP(CLK_ID,DEV_ID,RATE,NCHANL,NPT,NPTS)
C-----------------------------------------------------------------------
C  Subroutine written by Maj Richard R Ryles, NPS Student, 13 May 87   |
C                                              Updated    15 Jul 87   |
C                                                                      |
C  This subroutine is intended to attach the internal clock and A-D board |
C  to the main program and allow them to be used for data acquisition.  |
C                                                                      |
C  This subroutine should be used with subroutines GETDATA, DETACH     |
C-----------------------------------------------------------------------
      INCLUDE 'SYS$LIBRARY:LIOSET.FOR'
C-----------------------------------------------------------------------
C     Declare local variables                                          |
C-----------------------------------------------------------------------
      CHARACTER*4 CLK_DEV
      CHARACTER*1 ANS
      INTEGER SYS_STAT                   !Status ret by LIO calls
      INTEGER DEV_ID,CLK_ID              !LIO device ID
      INTEGER NCHAN                      !Channel #
      INTEGER NPT,NPTS
      REAL    RATE,TIME
C-----------------------------------------------------------------------
C     Input desired clock and sample parameters                        |
C-----------------------------------------------------------------------
      WRITE(*,1)
      READ(*,2)NCHANL
10    WRITE(*,3)
      READ(*,*)RATE
      IF(RATE.GT.600.00)THEN
              WRITE(*,4)
              GOTO 10
      ELSE
              IF(RATE.LT.0.01)WRITE(*,5)
              IF(RATE.LT.0.01)GOTO 10
      ENDIF
20    WRITE(*,6)
      READ(*,7) NPT
      IF(NPT.GT.5000)THEN
              WRITE(*,8)
              GOTO 20
      ELSE
              IF(NPT.LE.0)WRITE(*,9)
              IF(NPT.LE.0)GOTO 20
      ENDIF
      NPT = NPT + 1
      TIME = FLOAT(NPT-1)/RATE
      WRITE(*,11)TIME,TIME/60.
      NPTS = 2 * NPT
      CLK_DEV = 'KZA0'
      WRITE(*,12)
      READ(*,13)ANS
      IF (ANS.EQ.'y'.OR.ANS.EQ.'Y') THEN
              WRITE(*,14)
              READ(*,15)CLK_DEV
      ENDIF
C-----------------------------------------------------------------------
C     Attach the KWV11-C Clock and set up for mapped I/O               |
C-----------------------------------------------------------------------
      SYS_STAT = LIO$ATTACH (CLK_ID, CLK_DEV, LIO$K_MAP)
              IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
      SYS_STAT = LIO$SET_R (CLK_ID, LIO$K_CLK_RATE, 1,RATE)
              IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
      SYS_STAT = LIO$SET_I (CLK_ID, LIO$K_TRIG, 1, LIO$K_IMMEDIATE)
              IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
C-----------------------------------------------------------------------
C     Attach the ADV11-D and set up for mapped I/O                     |
C-----------------------------------------------------------------------
      SYS_STAT = LIO$ATTACH (DEV_ID, 'AZA0', LIO$K_MAP)
```

```fortran
                IF(.NOT. SYS_STAT) CALL LIBSSIGNAL(%VAL(SYS_STAT))
C-----------------------------------------------------------------------
C Set up the ADV for synchronous transfer                              |
C       Set channel (nchanl)                                           |
C       Gain of 1                                                      |
C       Trigger on LIO$READ and fill buffer as fast as possible        |
C-----------------------------------------------------------------------
        SYS_STAT = LIO$SET_I (DEV_ID, LIO$K_AD_CHAN, 1,NCHANL,)
                IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
        SYS_STAT = LIO$SET_I (DEV_ID, LIO$K_AD_GAIN, 1, 1,)
                 IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
        SYS_STAT = LIO$SET_I (DEV_ID, LIO$K_TRIG,2,LIO$K_CLK_POINT,CLK_ID)
                IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
C-----------------------------------------------------------------------
C       Format Statements                                              |
C-----------------------------------------------------------------------
 1      FORMAT(1x,'Enter channel number to sample: (0,1,8,9)')
 2      FORMAT(I2)
 3      FORMAT(/1x,'Input desired Clock Rate (600 Hz - 0.01 Hz):'/)
 4      FORMAT(1x,'Following directions! Input too high.'//20x,'Try again'/)
 5      FORMAT(1x,'Too low! Lots of time on your hands?'//20x,'Try again'/)
 6      FORMAT(/1x,'No. of data points: (any number less than 5,001)'/)
 7      FORMAT(I4)
 8      FORMAT(/1x,'Hey! Follow directions! Too high. Try again!'/)
 9      FORMAT(/1x,'Hey! Last input is nonsense! Try again!'/)
11      FORMAT(/1x,'The approximate run time for each port will be ',f6.1,
     &          ' seconds (or ',f5.2,' minutes)'/)
12      FORMAT(/1x,'Do you want to change the Clock Device Name from [KZA0]?',
     &          ' (y/n) [DEF - no]'/)
13      FORMAT(A1)
14      FORMAT(1x,'Enter Clock Device Name: (KZA0,KZB0)')
15      FORMAT(A4)
C-----------------------------------------------------------------------
        RETURN
        END
```

```
        SUBROUTINE GETDATA(DEV_ID,NPT,NPTS,VOLTAGES)
C----------------------------------------------------------------------
C       Subroutine written by Maj Richard R Ryles, NPS Student, 28 Apr 87  |
C                                                   Update 15 Jul 87  |
C                                                                     |
C       This subroutine is intended to acquire the data from the A-D board |
C       with the specified parameters from subroutine SETUP. It also will  |
C       translate the A-D data into voltages to be converted to physical   |
C       quantities in the main program.                                    |
C                                                                     |
C       This subroutine should be used with subroutines SETUP,DETACH,OUTPUTD |
C----------------------------------------------------------------------
        INCLUDE 'SYS$LIBRARY:LIOSET.FOR'
C----------------------------------------------------------------------
C       Declare local variables                                       |
C----------------------------------------------------------------------
        INTEGER SYS_STAT                      !Status ret by LIO calls
        INTEGER DEV_ID                        !LIO device ID
        INTEGER DATA_LENGTH                   !Number of bytes of data read
        INTEGER NPT,NPTS                      !Number of points to sample
C----------------------------------------------------------------------
C       Declare data buffers                                          |
C----------------------------------------------------------------------
        INTEGER*2 RAW_DATA(5000)              !5000 point raw data buffer
        REAL    VOLTAGES(5000)                !5000 voltages output
C----------------------------------------------------------------------
C Get a raw_data buffer of 'npt' values                               |
C       Uses LIO$READ to read the A/D                                 |
C       Note length of buffer is in bytes                             |
C       As is the returned data length                                |
C----------------------------------------------------------------------
        SYS_STAT = LIO$READ(DEV_ID, RAW_DATA, NPTS, DATA_LENGTH, )
                IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
C----------------------------------------------------------------------
C       Convert the raw data to voltages using LSP$FORMAT_TRANSLATE_ADC   |
C----------------------------------------------------------------------
        CALL LSP$FORMAT_TRANSLATE_ADC(RAW_DATA, VOLTAGES, NPT, , , )
C----------------------------------------------------------------------
        RETURN
        END
```

```
      SUBROUTINE DETACH(CLK_ID.DEV_ID)
C----------------------------------------------------------------------
C    Subroutine written by Maj Richard R Ryles, NPS Student 28 Apr 87  |
C                                              Update 16 jun 87        |
C                                                                      |
C    This subroutine is intended to detach the internal clock and A-D board  |
C    from the main program and allow them to be used for others application.  |
C                                                                      |
C       This subroutine should be used with subroutine SETUP           |
C----------------------------------------------------------------------
      INCLUDE 'SYS$LIBRARY LIOSET FOR'
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
C     Declare local variables
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
      INTEGER SYS_STAT        !Status ret by LIO calls
      INTEGER DEV_ID.CLK_ID   !LIO device ID
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
C     Detach from the Clock
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
      SYS_STAT = LIO$DETACH (CLK_ID.)
            IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
C     Detach from the A/D
C••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
      SYS_STAT = LIO$DETACH(DEV_ID, )
            IF(.NOT. SYS_STAT) CALL LIB$SIGNAL(%VAL(SYS_STAT))
      WRITE(*,1)
1     FORMAT(/1x,'Both clock and A-D device are detached'/)
      RETURN
      END
```

```
      SUBROUTINE OUTPUTD(RATE,NPT,NCHANL,AMEAN,STD_DEV,J,VOLTAGES)
C-----------------------------------------------------------------------
C     Subroutine written by Maj Richard R Ryles, NPS Student 28 Apr 87 |
C                                                   Update 24 Jul 87 |
C                                                                    |
C     This subroutine is intended to output diagnostic results to a file. |
C                                                                    |
C     This subroutine should be used with subroutine SETUP, DETACH, GETDATA |
C-----------------------------------------------------------------------
C     Declare local variables                                        |
C-----------------------------------------------------------------------
      INTEGER   NCHANL                          !Channel #
      INTEGER   J                               !General loop indexes
      INTEGER   NPT                             !Number of points
      REAL*8    AMEAN,STD_DEV
      REAL      RATE,VOLTAGES(5000)
C-----------------------------------------------------------------------
C     Output data to a file                                          |
C-----------------------------------------------------------------------
      OPEN(UNIT = 10, FILE = 'SCANIDIAG.DAT', STATUS = 'NEW')
      WRITE(10,1) J, RATE, NPT-1, NCHANI, AMEAN, STD_DEV
1     FORMAT(/1x,'Scanivalve #',i3/1x,'Clock Rate =',f6.1,5x,'Data Points',
     &        'Sampled =',i4/1x,'Channel',5x,'Mean',4x,'Standard Deviation'/
     &        3x,i3,2f12.4/)
      WRITE(10,*)'Scanivalve #',J
      DO 20 I=1,NPT
20         WRITE(10,*)'I=',I,VOLTAGES(I)
      CLOSE(10)
      RETURN
      END
```

```
C************************************************************
C
      SUBROUTINE HWWA(NSAMPLES,ANSWER,BLUE_GREEN)
C
C PURPOSE:
C        Drives the acquisition of the LDV data.
C
C ERROR HANDLING
C    IER = 1 means no "fatal" error.
C
C EXTERNAL REFERENCES
C    NAME       DESCRIPTION AND SOURCE
C    DVWA       Calls DRV11-WA device driver to acquire A/D data.
C
C DEVELOPMENT HISTORY:
C        DATE    INITIALS   DESCRIPTION
C      07/26/83    TML      Adapted from DAPNT
C      12/12/85    CLH      Modified to print results in either
C                           octal or integer.
C      02/25/87    GBG      Modified for use with DRV11-WA and VMS
C      05/06/87    CLH      Modified to accept "new" DAMUX1.
C      06/08/87    RRR      Modified to subroutine + deleted CLH 5/87 Mod
C      07/15/87    RRR      Modified to calculate velocities + optimized
C                           for speed of execution
C      07/24/87    RRR      Modified to view Histogram
C      08/29/87    RRR      Modified to accept calibration constant
C
C AUTHOR(S): Ted Lichtenstein, Informatics General Corp.
C-***********************************************************
C
      IMPLICIT NONE
C
      PARAMETER          NRAWMX=80000
      PARAMETER          LUNTI=5
      PARAMETER          LUNOUT=6
C
      INTEGER*2          IRAWDT(NRAWMX)
      INTEGER*2          I1RAWDT(NRAWMX),I2RAWDT(NRAWMX)
      INTEGER            IER,HISTO,Nfr,ONE,TWO
      INTEGER            IYESNO, NRAWIN, I, BLUE_GREEN
      INTEGER            NCHANS, NSAMPLES, INDEX, JNDEX
      INTEGER            NS,ANSWER,COUNT,END
      REAL               TIME(NRAWMX),FREQ(NRAWMX),VELOC(NRAWMX),I3RAWDT(NRAWMX)
      REAL               MIX(2),GAM(2),DEL(2),KAP(2),D(2),I(2),MEAN(2),FREQUENCY(2,8001)
      REAL               FREQONE,FREQTWO,FRE(NRAWMX),FREQ1_AVE,FREQ2_AVE,CAL_CONST(2)
      LOGICAL            DATSELCT (3)
      CHARACTER*19       DIRECTION
      CHARACTER*1        ANS
C
      DATA MIX(1),MIX(2) / -1.E-6,-1.E-6/
C
      INCLUDE '($SYSSRVNAM)'
C
 100  CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Fill the raw data buffer with this bit pattern:
C     "1010101010101010"
C     This equals 125252(8), -21846(10).
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 200 I = 1, NRAWMX
         IRAWDT(I) = '125252'O
         I1RAWDT(I) = '000000'o
         I2RAWDT(I) = '000000'o
         I3RAWDT(I) = 0.0
 200  CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Input number of samples to be acquired
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      NCHANS = 4
300   WRITE (LUNTI, 10)
10    FORMAT (' Enter number of samples:')
      READ (LUNTI, *) NSAMPLES
      NRAWIN = NSAMPLES * NCHANS
      IF (NRAWIN .GT. 32767 .OR. NRAWIN .LE. 0) THEN
         WRITE (LUNTI, 20)
20       FORMAT (' The number of samples * number of channels must be 32,767'
     &              ' or less.')
         GO TO 300
      END IF
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Set up the LDV parameters prior to acquisition
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IF(BLUE_GREEN.EQ.1)THEN
         GAM(1) = 0.488
         GAM(2) = 0.5145
         IF(CAL_CONST(1).EQ.1.) THEN
            WRITE(*,*)'Enter Blue calibration constant'
            READ(*,*) CAL_CONST(1)
         ENDIF
         IF(CAL_CONST(2).EQ.1.) THEN
            WRITE(*,*)'Enter Green calibration constant'
            READ(*,*) CAL_CONST(2)
         ENDIF
      ELSE
         GAM(1) = 0.5145
         GAM(2) = 0.488
         IF(CAL_CONST(1).EQ.1.) THEN
            WRITE(*,*)'Enter Green calibration constant'
            READ(*,*) CAL_CONST(1)
         ENDIF
         IF(CAL_CONST(2).EQ.1.) THEN
            WRITE(*,*)'Enter Blue calibration constant'
            READ(*,*) CAL_CONST(2)
         ENDIF
      ENDIF
      IF(MIX(1).LT.0.0) THEN
         IF(BLUE_GREEN.EQ.1)WRITE(*,*)'Input BLUE mixing frequency (MHz)'
         IF(BLUE_GREEN.EQ.0)WRITE(*,*)'Input GREEN mixing frequency (MHz)'
         READ(*,*) MIX(1)
      ENDIF
      IF(MIX(2).LT.0.0) THEN
         IF(BLUE_GREEN.EQ.1)WRITE(*,*)'Input GREEN mixing frequency (MHz)'
         IF(BLUE_GREEN.EQ.0)WRITE(*,*)'Input BLUE mixing frequency (MHz)'
         READ(*,*) MIX(2)
      ENDIF
      IF(ANSWER.EQ.0) THEN
         WRITE(*,*)'Do you want probe volume specifications? (y/n)'
         READ(*,'(a)') ANS
         IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') ANSWER = 1
      ENDIF
      WRITE(*,*)' '
C
C       Select the data to be returned.
C
      DATSELCT (1) = .FALSE.
      DATSELCT (2) = .FALSE.
      DATSELCT (3) = .TRUE.
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     LDV Data Acquisition
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ONE = 1
      TWO = 2
      OPEN(UNIT=5,FILE='VELOCITY.DAT',STATUS='NEW')
      DO INDEX = 1, NSAMPLES
         CALL DAMUX (NCHANS, ONE, DATSELCT, , , IRAWDT, IER)
         IF(IER .NE. 1) WRITE(*,*)' IER in DAMUX =',IER
```

```
C • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Conversion to Frequency
C • • • • • • • • • • • • • • • • • • • • • • • • • • • •
          DO I=1,2
              IRAWDT(I) = INOT(IRAWDT(I))
              I3RAWDT(I) = 0.0
              I2RAWDT(I) = 0
              I1RAWDT(I) = 0
              I1RAWDT(I) = IIBITS(IRAWDT(I),10,4)
              I2RAWDT(I) = IIBITS(IRAWDT(I),0,10)
              I3RAWDT(I) = (2.••I1RAWDT(I)) •( float(I2RAWDT(I)))
              TIME(I) = I3RAWDT(I)/32.
              FREQ(I) = 1000./TIME(I)
          END DO
          FREQUENCY(1,INDEX) = FREQ(1)
          FREQUENCY(2,INDEX) = FREQ(2)
C • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C      Convert frequency data to velocities
C • • • • • • • • • • • • • • • • • • • • • • • • • • • •
          CALL VELOCITY (FREQ,VELOC,TWO,MIX,ANSWER,GAM,DEL,KAP,D,l,Nfr,CAL_CONST)
          WRITE(5,30) VELOC(1),VELOC(2)
 30       FORMAT(1X,2F11.4)
       END DO
       CLOSE(5)
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C      Calculate average counter frequency and print to sceen for diagnostics
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
       FREQONE = 0.0
       FREQTWO = 0.0
       DO 34 I = 1,NSAMPLES
          FREQONE = FREQONE + FREQUENCY(1,I)
34        FREQTWO = FREQTWO + FREQUENCY(2,I)
       FREQ1_AVE = FREQONE/FLOAT(NSAMPLES)
       FREQ2_AVE = FREQTWO/FLOAT(NSAMPLES)
       WRITE(•,•)' '
       WRITE(•,•)'Average counter frequency # 1 =',FREQ1_AVE
       WRITE(•,•)'Average counter frequency # 2 =',FREQ2_AVE
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C      Write Probe Volume dimensions and parameters to a file
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
       IF(ANSWER.EQ.1)THEN
          OPEN(UNIT=15,FILE='Probe.dat',STATUS='NEW')
          WRITE(15,•)'Probe Volume Data'
          WRITE(15,•)'——————————————————————————————————————————————————————',
     &              '———————'
          DO 40 I = 1,2
             IF(BLUE_GREEN.EQ.1) DIRECTION = 'Horizontal Velocity'
             IF(BLUE_GREEN.EQ.0) DIRECTION = 'Vertical Velocity  '
             IF(I.EQ.2) THEN
                IF(BLUE_GREEN.EQ.1) DIRECTION = 'Vertical Velocity  '
                IF(BLUE_GREEN.EQ.0) DIRECTION = 'Horizontal Velocity'
             ENDIF
             WRITE(15,1)GAM(I),DIRECTION,DEL(I),KAP(I),D(I),l(I)•.001,Nfr
1            FORMAT(/1X,'Wavelength of light <microns> = ',f7.4,'   (Measuring',
     &              a19,')'//1x,'Fringe Spacing <microns> = ',f10.2//1x,'Lens',
     &              ' angle <deg> =',f7.2//1x,'Probe volume diameter <microns> = ',
     &              f10.5//1x,'Length of measuring volume <mm> = ',f10.4//1x,
     &               Number of Fringes =',i3/)
             WRITE(15,•)'——————————————————————————————————————————————————————',
     &                 '———————'
40        CONTINUE
          CLOSE(15)
       ENDIF
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C      Option to print a histogram of velocity data on screen for diagnostics
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
       WRITE(•,•)' '
       WRITE(•,•)'Do you want a Histogram? (1/0)'
```

```
      READ(*,*) HISTO
      END = NSAMPLES
      IF(HISTO.EQ.1)CALL HISTOGRAM(COUNT,END,BLUE_GREEN)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Again?
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(LUNTI,*)' '
      WRITE (LUNTI,*) 'Acquire LDV Data again?  (0=NO, 1=YES) '
      READ  (LUNTI,*) IYESNO
      WRITE(*,*)' '
      IF ( IYESNO.EQ.1 ) GO TO 100
C
      RETURN
      END
```

```
      SUBROUTINE VELOCITY(FREQ,VELOC,N,MIX,ANSWER,GAM,DEL,KAP,Dm,l,Nfr,
     &                       CAL_CONST)
C------------------------------------------------------------------------
C        Subroutine written by Maj Richard R Ryles, NPS Student, 9 Jun 87  |
C                                                 Last Update 29 Aug 87    |
C                                                                          |
C        This subroutine was written to convert the LDV raw data from frequency |
C        to velocities.                                                    |
C                                                                          |
C------------------------------------------------------------------------
      REAL    THETA,GAMMA,DELTA,d,f,E,KAPPA,Dsubm,lm,De,FREQ(N),VELOC(N),GAM(2)
      REAL    SHIFT,MIX(2),DEL(2),CAL_CONST(2),KAP(2),Dm(2),l(2)
      INTEGER IANS,ANSWER,Nfr
      IF(CAL_CONST(1).EQ.0.0) CAL_CONST(1) = 1.0
      IF(CAL_CONST(2).EQ.0.0) CAL_CONST(2) = 1.0
      PI = ASIN(1.0)*2.
      d = 13.                        ! Beam spacing (mm)
      f = 600.                       ! Focal length of transmitting lens (mm)
      E = 1.                         ! Beam expansion ratio
      KAPPA = ATAND(E*d/2./f)        ! Lens angle (deg)
      DO 5 I = 1,2
         GAMMA = GAM(I)              ! Wavelength of Laser (microns)
         DELTA = CAL_CONST(I) * GAMMA / (2.*SIND(KAPPA))  ! Fringe spacing
         De = 1.1                 ! Diameter of the incident laser beam  (mm)
         Dsubm = (4.*GAMMA*f/(PI*De*E)) ! Probe volume diameter (microns)
         lm = Dsubm / TAND(KAPPA)       ! Probe volume length (microns)
         Nfr = 1.27 * d / De            ! Number of fringes
         DEL(I) = DELTA                 ! Variables passed to output file
         KAP(I) = KAPPA                 !    "       "     "    "     "
         Dm(I) = Dsubm                  !    "       "     "    "     "
         l(I) = lm                      !    "       "     "    "     "
         SHIFT = 40.0                       ! = Bragg shift (MHz)
         FREQ(I) = FREQ(I) - SHIFT + MIX(I) ! MIX = Mixing Freq (MHz)
5        VELOC(i) = DELTA * FREQ(i)         ! FREQ = Adjusted Doppler Freq
      RETURN                                !           (MHz)
      END
```

```
      SUBROUTINE HISTOGRAM(COUNTER,END,BLUE_GREEN)
C-----------------------------------------------------------------------
C       Subroutine written by Maj Richard R Ryles, NPS Student, 10 Jun 87  |
C                                                       Update  29 Aug 87  |
C                                                                          |
C       This subroutine is intended to create two histograms of velocity   |
C       data from a two component LDV.                                     |
C                                                                          |
C-----------------------------------------------------------------------
      CHARACTER*4       STRING
      CHARACTER*22      LABELX                   ! variables for the axis
      CHARACTER*17      LABELY                   ! lables
      CHARACTER*29      LABELM
      INTEGER           STATUS,I,ISHADE,COUNTER,END
      INTEGER           N(3),BLUE_GREEN
      REAL              XCONTROL(4),XMIN,XMAX
      REAL              YCONTROL(4),COUNT(8)
      REAL              XLOW(8)
      REAL              XHIGH(8)
      REAL              Y(8),VELOCITY(8200,2),X(8200)
C
      N(1) = 1                                  ! number of data points
      N(2) = 1                                  ! number of columns of data
      N(3) = 1                                  ! number of rows of data
C * * * * * * * * * * * * * * * * * * * *
C     Read Velocity Data File
C * * * * * * * * * * * * * * * * * * * *
      OPEN(UNIT=5,FILE='VELOCITY.DAT',STATUS='OLD')
      IF(BLUE_GREEN.EQ.1)THEN
         DO 10 I = 1,END
10          READ(5,1) VELOCITY(I,1),VELOCITY(I,2)
      ELSE
         DO 20 I=1,END
20          READ(5,1) VELOCITY(I,2),VELOCITY(I,1)
      ENDIF
C * * * * * * * * * * * * * * * * * * *
C     Find the Min and Max values
C * * * * * * * * * * * * * * * * * * *
      CLOSE(5)
      DO 80 J = 1,2
         DO 30 K = 1,end
30          X(K) = VELOCITY (K,J)
      XMIN = 1000.0
      XMAX = -1000.0
      DO 40 I = 1,end
         IF(XMIN.GT.X(I)) XMIN = X(I)
40       IF(XMAX.LT.X(I)) XMAX = X(I)
C * * * * * * * * * * * * * * * * * * * *
C     Initialize X-Axis controls
C * * * * * * * * * * * * * * * * * * * *
      XCONTROL(1) = 5.5                         ! length of x-axis, in inches
      XCONTROL(2) = XMIN                        ! left most value on x-axis
      XCONTROL(3) = XMAX                        ! right most value on x-axis
      XCONTROL(4) = (XMAX-XMIN)/8.              ! increment of plotted points
C * * * * * * * * * * * * * * * * * * * *
C     Find y data from sample array
C * * * * * * * * * * * * * * * * * * *
      DO 50 I = 1,8
         Y(I) = 0.0
         XLOW(I) = XMIN + FLOAT(I-1)*XCONTROL(4)
         IF(I.EQ.8)THEN
         XHIGH(8) = XMAX
      ELSE
         XHIGH(I)= XMIN + FLOAT(I)*XCONTROL(4)
      ENDIF
50       COUNT(I) = XMIN + FLOAT(I)*XCONTROL(4)
C
      XTOTAL = 0.0
```

```
      DO 60 I = 1,end
           XTOTAL = X(I) + XTOTAL
           IF(X(I).GE.XMIN.AND.X(I).LT.COUNT(1)) Y(1) = Y(1) + 1.0
           IF(X(I).GE.COUNT(1).AND.X(I).LT.COUNT(2)) Y(2) = Y(2) + 1.0
           IF(X(I).GE.COUNT(2).AND.X(I).LT.COUNT(3)) Y(3) = Y(3) + 1.0
           IF(X(I).GE.COUNT(3).AND.X(I).LT.COUNT(4)) Y(4) = Y(4) + 1.0
           IF(X(I).GE.COUNT(4).AND.X(I).LT.COUNT(5)) Y(5) = Y(5) + 1.0
           IF(X(I).GE.COUNT(5).AND.X(I).LT.COUNT(6)) Y(6) = Y(6) + 1.0
           IF(X(I).GE.COUNT(6).AND.X(I).LT.COUNT(7)) Y(7) = Y(7) + 1.0
           IF(X(I).GE.COUNT(7).AND.X(I).LT.COUNT(8)) Y(8) = Y(8) + 1.0
60         IF(X(I).GE.COUNT(8).AND.X(I).LT.XMAX) Y(8) = Y(8) + 1.0
      XMEAN = XTOTAL/FLOAT(END)
C
      YMAX = 0.0
      DO 70 I = 1,8
           Y(I) = Y(I)/FLOAT(END)
           IF(YMAX.LE.Y(I))YMAX = Y(I)
70         Y(I) = 100. * Y(I)
      WRITE(*,2)YMAX*100.,XMEAN
C * * * * * * * * * * * * * * * * * *
C     Initialize Y axis controls
C * * * * * * * * * * * * * * * * * *
      YCONTROL(1) = 4.5                      ! height of graph in inches
      YCONTROL(2) = 0.0                      ! lower boundary value
      YCONTROL(3) = YMAX*120.                ! upper boundary value
      YCONTROL(4) = 5.0                      ! increment of plotted points
C * * * * * * * * * * * * * * * * * *
C     Histogram labels
C * * * * * * * * * * * * * * * * * *
      LABELX = '   VELOCITIES (mps)    '                ! x-axis label
           IF(COUNTER.GT.0)LABELX = 'FINAL VELOCITIES (mps)'        ! x-axis label
      LABELY = 'PERCENTAGE OF PTS'                      ! y-axis label
      LABELM = 'HORIZONTAL VELOCITY HISTOGRAM'          ! upper label
      IF(J.EQ.2) LABELM = ' VERTICAL VELOCITY HISTOGRAM'
      STRING = 'IXSY'                                   ! mode string
      ISHADE = 2                                        ! filling of graph
C * * * * * * * * * * * * * * * * * * * * * *
C     Set up coordinate system with values
C * * * * * * * * * * * * * * * * * * * * * *
      IF(XMIN.GE.XMAX)GOTO 90
      CALL LGP$PLOT(1,STRING,,0,N,LABELX,LABELY,STATUS,,,
     -               XCONTROL,YCONTROL,,LABELM)
      IF (STATUS .NE. LGP$_SUCCESS) THEN
           CALL LIB$SIGNAL(%VAL(STATUS))
      ENDIF
C * * * * * * * * * * * * * * * * * * *
C     Call routine and plot histogram
C * * * * * * * * * * * * * * * * * * *
      CALL LGP$HIST(1,XLOW,XHIGH,Y,8,STATUS,,ISHADE)
      IF (STATUS .NE. LGP$_SUCCESS) THEN
           CALL LIB$SIGNAL(%VAL(STATUS))
      ENDIF
      PAUSE 'Waiting for viewing/printing Histogram; Type CONT when finished'
      CALL LGP$TERMINATE_PLOT(1)             ! terminate plotting
80    CONTINUE
      RETURN
90    WRITE(*,*)' '
      WRITE(*,*)'*** No Histogram created: xmin>=xhigh ***'
      GOTO 80
C * * * * * * * * * * * * * * * * * *
C     Formats
C * * * * * * * * * * * * * * * * * *
1     FORMAT(1X,2F11.4)
2     FORMAT(/1X,'MAXIMUM PERCENTAGE OF SECTORS = ',F6.1//1X,'SAMPLE MEAN = ',
     &       F9.2)
      END
```

```
      SUBROUTINE REDUCE_DATA(NSAMP,COUNTER,END,MEAN,STD_DEV1,STD_DEV2,
     &                       BLUE_GREEN)
C-----------------------------------------------------------------------
C     Subroutine written by Maj Richard R Ryles, NPS Student, 12 Aug 87
C                                                     Update  29 Aug 87
C
C     This subroutine is excludes all velocity data from the LDV that fall
C     outside two standard deviations from the mean.
C-----------------------------------------------------------------------
      REAL•8 STD_DEV1,STD_DEV2,NEW_STD(2),TOTAL(2),SUM(2),X,Y
      REAL VELOC(2,10000),MEAN(2),UP,DOWN,TOLERANCE
      REAL NEW(2),VELO(2,20000)
      INTEGER NSAMP,COUNT(2),TOT,END,BLUE_GREEN
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Read velocity data from file
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
      OPEN(UNIT=3,FILE='VELOCITY.DAT',STATUS='OLD')
      DO 10 I = 1,NSAMP
10       READ(3,1)VELOC(1,I),VELOC(2,I)
      CLOSE(3)
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Calculate range of acceptable values i.e. all velocity
C     data that falls within two standard deviations of the mean
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
      DO 30 J = 1,2
         IF(J.EQ.1)THEN
            TOLERANCE = STD_DEV1•2.
            UP = MEAN(1) + TOLERANCE
            DOWN = MEAN(1) - TOLERANCE
         ENDIF
         IF(J.EQ.2)THEN
            TOLERANCE = STD_DEV2•2.
            UP = MEAN(2) + TOLERANCE
            DOWN = MEAN(2) - TOLERANCE
         ENDIF
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Flag each point outside of two standard deviations
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
         DO 20 I = 1,NSAMP
20          IF(VELOC(J,I).LT.DOWN.OR.VELOC(J,I).GT.UP)VELOC(J,I)=1000.0
30    CONTINUE
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Calculate new mean and standard deviations
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
      DO 60 J = 1,2
        TOTAL(J) = 0.D0
        COUNT(J) = 0
        SUM(J) = 0.D0
        DO 50 I = 1,NSAMP
          IF(VELOC(1,I).NE.1000.0.AND.VELOC(2,I).NE.1000.0) THEN
             TOTAL(J) = TOTAL(J) + VELOC(J,I)
             COUNT(J) = COUNT(J) + 1
          ENDIF
50      CONTINUE
        IF(COUNT(J).EQ.0) COUNT(J) = 1
        NEW(J) = TOTAL(J)/FLOAT(COUNT(J))
        DO 55 K = 1,NSAMP
          IF(VELOC(1,K).NE.1000.0.AND.VELOC(2,K).NE.1000.0) THEN
             X = VELOC(J,K) - NEW(J)
             Y = X•X•(1./FLOAT(NSAMP-1))
             SUM(J) = SUM(J) + Y
          ENDIF
55      CONTINUE
        NEW_STD(J) = SQRT(SUM(J))
60    CONTINUE
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C     Output new values to screen
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
```

```
      WRITE(*,*)' '
      WRITE(*,*)'New cleaned-up data'
      IF(BLUE_GREEN.EQ.1)THEN
           WRITE(*,*)'U dir',NEW(1),NEW_STD(1)
           WRITE(*,*)'V dir',NEW(2),NEW_STD(2)
      ELSE
           WRITE(*,*)'U dir',NEW(2),NEW_STD(2)
           WRITE(*,*)'V dir',NEW(1),NEW_STD(1)
      ENDIF
      MEAN(1) = NEW(1)
      MEAN(2) = NEW(2)
      STD_DEV1 = NEW_STD(1)
      STD_DEV2 = NEW_STD(2)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Write interim data file containing just good data
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      OPEN(UNIT=5,FILE='INTERIM.DAT',STATUS='NEW')
      DO 70 J = 1,2
         DO 70 I = 1, NSAMP
            IF(VELOC(1,I).NE.1000.0.AND.VELOC(2,I).NE.1000.0) THEN
               WRITE(5,2)VELOC(J,I)
            ENDIF
70    CONTINUE
      CLOSE(5)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Write final velocity data file for plotting
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      OPEN(UNIT=7,FILE='VELOCITY.DAT',STATUS='NEW')
      OPEN(UNIT=9,FILE='INTERIM.DAT',STATUS='OLD')
      TOT = COUNT(1) + COUNT(2)
      DO 80 I = 1, TOT
         IF(I.LE.COUNT(1))THEN
            READ(9,2) VELO(1,I)
         ELSE
            READ(9,2) VELO(2,I)
         ENDIF
80    CONTINUE
      CLOSE(9)
      END = COUNT(1)
      COUNTER = 1
      DO 100 I = 1, END
100      WRITE(7,1) VELO(1,I),VELO(2,COUNT(1)+I)
      CLOSE(7)
      RETURN
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     Format statements
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1     FORMAT(1X,2F11.4)
2     FORMAT(1X,F11.4)
      END
```

LIST OF REFERENCES:

1. American Institute of Aeronautics and Astronautics Paper 85-1769-CP, Dynamic Stall Progress in Analysis and Prediction, by L .W. Carr, 19-21 August 1985.

2. NASA Contractor Report 177386, Laser Doppler Velocimetry Primer by William D. Bachalo, December 1985.

3. Durst,F., Melling A., and Whitelaw,J. H., Principles and Practice of Laser-Doppler Anemometry, 2d ed., Academic Press, 1976.

4. Laser Doppler Velocimetry Instruction Manual, System 9100-7, TSI, Incorporated.

INITIAL DISTRIBUTION LIST

		No. copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor S. Bodapati, Code 67Bu Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	10
5.	Mr. Daniel Bencze Mail Stop 227-6 NASA Ames Research Center Moffett Field, California 94035	1
6.	Chief, Fluid Mechanics Laboratory Mail Stop 260-1 NASA Ames Research Center Moffett Field, California 94035	1
7.	Dr M.S. Chandrasekhara Mail Stop 260-1 NASA Ames Research Center Moffett Field, California 94035	1
8.	Captain H. Helin AFOSR/NA Bolling Air Force Base Washington, D. C. 20332-6448	1
9.	Dr. Robert E. Singleton U.S. Army Research Office P.O. Box 12211 Research Triangle Park, North Carolina 27709	1
10.	Major Richard R. Ryles, USA P.O. Box 42 Fort Rucker, Alabama 36362	3

END

DATE

FILMED

FEB.

1988