NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMPUTER-CONTROLLED DATA AQUISITION AND ANALYSIS

by

John Anthony DiMiceli

September 1986

Thesis Advisor: S. Bodapati

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for Public Release; Distribution is Unlimited

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, CA 93943-5000

7b. ADDRESS (City, State, and ZIP Code)
Monterey, CA 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)
COMPUTER-CONTROLLED DATA AQUISITION AND ANALYSIS

12 PERSONAL AUTHOR(S)
DIMICELI, JOHN A.

13a TYPE OF REPORT
MASTER'S THESIS

13b TIME COVERED
FROM ________ TO ________

14 DATE OF REPORT (Year, Month, Day)
1986 SEPTEMBER

15 PAGE COUNT
88

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
DATA AQUISITION, COMPUTER, AIRFOIL, SCANIVALVE

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

A computer-controlled data aquisition and analysis system was designed, built, and tested. All of the required elements of a complete data aquisition system were assembled, including transducers, amplifiers, signal conditioners, display devices, computer interface, computer, and related software. A digital multiplexer was constructed to multiplex up to 16, 16-bit channels for input into the computer. A computer program was written to bring the various parts of the system together. An experiment, Analysis of an Airfoil by Pressure Distribution, was used as a vehicle to test the system. The final output of the data aquisition system, including graphical information, compared favorably with previous results from an older data aquisition system in use.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
S. BODAPATI

22b TELEPHONE (Include Area Code)
408-646-2854

22c OFFICE SYMBOL
67Bu

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

Computer-Controlled Data Aquisition And Analysis

by

John Anthony DiMiceli
Lieutenant, United States Navy
B.S., Manhattan College, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
September 1986

ABSTRACT

A computer-controlled data aquisition and analysis system was designed, built, and tested. All of the required elements of a complete data aquisition system were assembled, including transducers, amplifiers, signal conditioners, display devices, computer interface, computer, and related software. A digital multiplexer was constructed to multiplex up to 16, 16-bit channels for input into the computer. A computer program was written to bring the various parts of the system together. An experiment, Analysis of an Airfoil by Pressure Distribution, was used as a vehicle to test the system. The final output of the data aquisition system, including graphical information, compared favorably with previous results from an older data aquisition system in use.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

A special thank you to Ted Dunton who provided invaluable insight and direction. His selflessness and patience were instrumental in helping me to realize the goals of this thesis project.

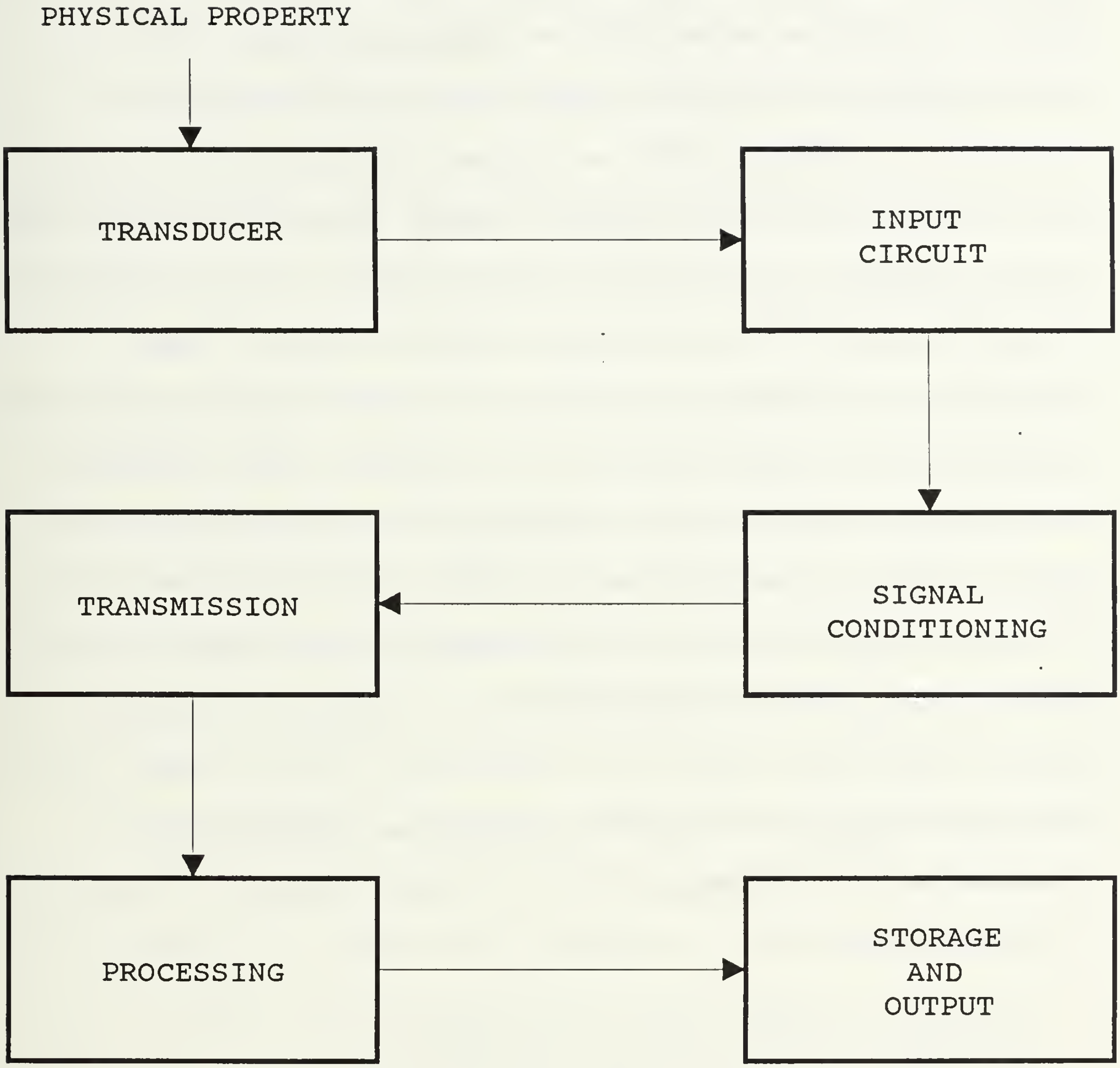
A special thank you to Dr. Bodapati who gave me encouragement and guidance when I needed it most.

I. INTRODUCTION

A. GENERAL DISCUSSION

The object of any data aquisition and analysis system is to collect the data, process them in a desired fashion, and record the results in a form suitable for storage, presentation, or additional subsequent processing. The simplest method of data aquisition might consist of a person, reading an instrument, and recording the observation on a data sheet. Complex data aquisition systems could incorporate the aquisition of thousands of data points, coupled with real-time analysis and final output on the device of your choice.

Figure 1 illustrates the basic building blocks to most data aquisition systems. Data aquisition begins with a physical property to be measured, such as mass, velocity, pressure, position, etc. A transducer is used to convert the measurable quantity into an electrical signal which is used as a direct representative of that physical property. The output of the transducer could be in the form of a varying voltage, current, or frequency. From this point onward until output is desired (since human beings cannot "read" electrical signals directly), the value of the original physical property remains in electrical form. An input circuit is used to ensure transducer compatibility to



Data Aquisition Basic Components

Figure 1

the rest of the electrical circuitry. Signal conditioning refers to various techniques used to reduce the effects of noise. Transmission involves transporting the representative electrical signal from one location to another. It really could be located at any point in the block diagram since all signals travel an arbitrary distance between processing functions. Processing involves molding the raw signal into a form that could be used to make comparisons with other signals of the same kind, or signals coming from different sources. Processing also includes data manipulation such as mathematical calculations, and type conversions such as analog-to-digital or current-to-voltage. Amplification could be used in the input circuit, signal conditioning, transmission, or processing steps. Lastly, the worked data (or if desired, the raw data) is stored or displayed so that it could be interpreted immediately through paper or crt display output, or used at a later time.

B. COMPUTER ASSISTED DATA AQUISITION

The digital computer presents some unique opportunities and techniques for data aquisition and analysis, and its use incurs many benefits. First, it has multipurpose and multitasking capabilities. It can record, manipulate or process, display and store data in a very efficient manner. It not only can receive information, but it can distribute information to control the various processes involved from simple on-off operations to complicated robotics. Second, it can make decisions, outlined in software, that allow it to be effective over a wide range of input parameters. Third, it is fast. Although the term "fast" is relative, and "fast" is getting faster every day, in comparison to the capabilities of a person to perform the same duties, it has great speed. Fourth, it excels at repetitive, boring tasks that often compromise the routine of an experiment, without forgetting steps. Fifth, it is relatively inexpensive in comparison with most high grade experimentation equipment, especially considering what it accomplishes. Lastly, it frees its human operator from data collecting to data interpretation, which is the task people (vs computers) excell at.

Using a computer, however, does not solve every problem. First, no computer ever designed an experiment. A human must perceive the need for information, determine what he needs to know, and how he is to obtain that information. A person must build the separate components to a data aquisition system and ensure their compatibility. A person is not relieved of the basic calibration and other data processing fine tuning that ensures correct results. Second, "Garbage in - Garbage out." is always true with computers. Errors can multiply quickly, or worse, can go unnoticed if the the results are within expectations.

Third, since transistor logic is standardized, the data must always be converted into a form that computer circuits can recognize. Fourth, the computer cannot interpret data beyond software algorithms. That is, it cannot decide if the results are meaningful. In the final analysis, no computer can guarantee meaningful results, it can only make it less tedious to obtain them.

C. THESIS GOALS

It was the goal of this thesis project to design, build, integrate the required system components, and operate a complete data aquisition system. The goal also included the writing of a computer program to aquire, analyze and display the data on-line, and store the required information for later processing. Additionally, the computer program must produce a hardcopy report to include tabular data and graphical output. A design requirement was that the system be as general as possible so that it could be used for a variety of tasks.

As a test case, a pre-designed experiment: Airfoil Performance By Pressure Distribution [Ref. 1], was chosen as a vehicle to test the system. Measurement requirements included pressure, temperature, angle of incidence. Data processing requirements included input signal conditioning, analog-to-digital conversion, and mathematical data reduction. Final output requirements included on-line displaying of raw data, reduced data and graphed data, and X-Y plotting of calculated results with paper hardcopy.

II. WIND TUNNEL AND INSTRUMENTATION

A. WIND TUNNEL GENERAL DESCRIPTION

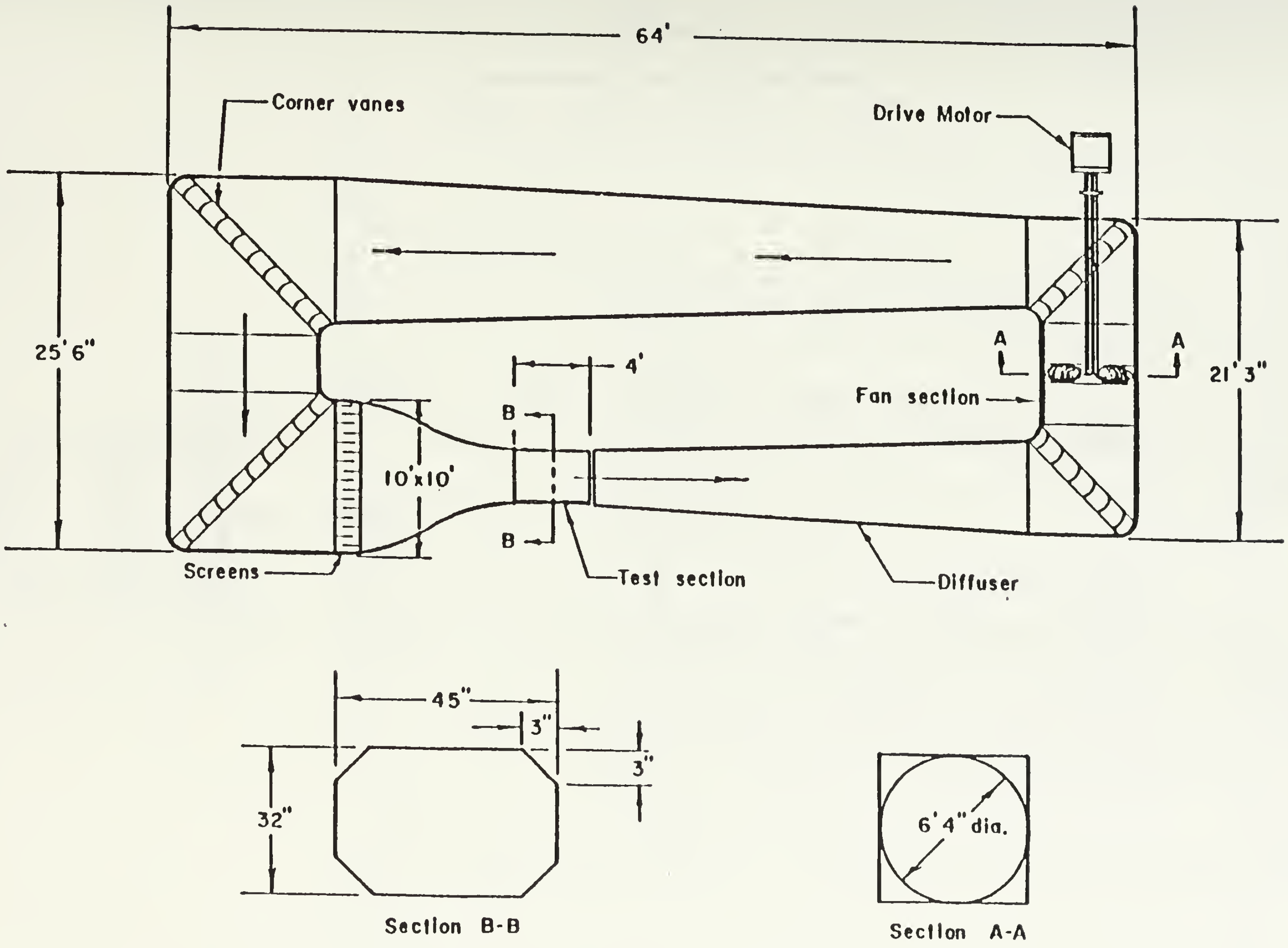
A complete description of the Department of Aeronautics Low-Speed Wind Tunnel Facility is found in Reference 1. Figure 2 shows the general layout of the tunnel. The wind tunnel has two static pressure manifolds that measure the static pressure in the settling chamber and the static pressure in the test section. The manifold in the settling chamber is located just aft of the screens [Fig. 2]. The other manifold is located at the entrance to the test section. Pressure from these manifolds are used to provide the total pressure (with appropriate corrections) and the free stream static pressure.

The airfoil tested was a NACA 66(215)-216 section. Figures 3 and 4 show the general shape of the airfoil as well as the locations of the static ports. The 36 static ports are tapped from leading edge to trailing edge, on both the upper and lower surfaces.

B. INSTRUMENTATION

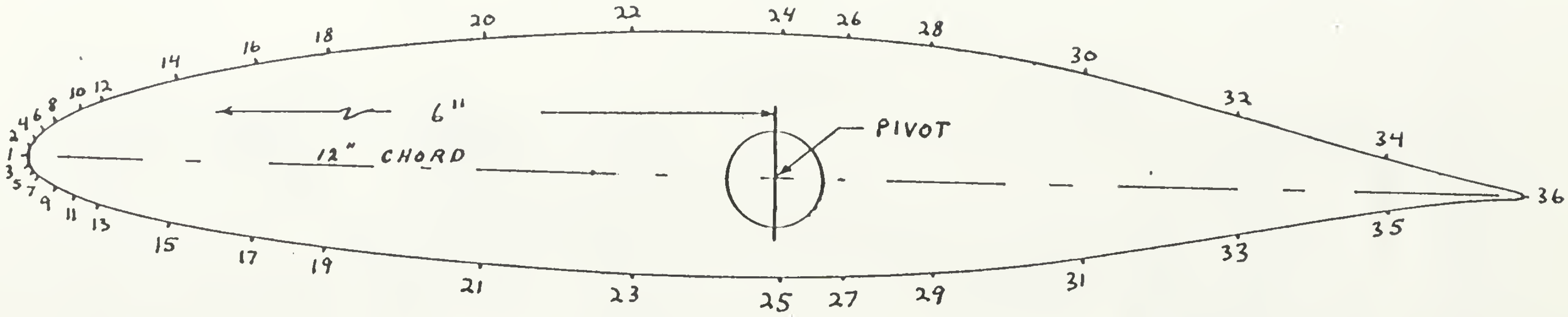
1. Scanivalve

One 48-port scanivalve was used to measure each of the 36 airfoil pressures, the settling chamber static pressure, and the free stream static pressure. The



Schematic of Horizontal Wind Tunnel
(No Scale)

Figure 2



Airfoil NACA 66(215)-216 with
Pressure Port Locations

Figure 3

Orifice	Station %c	Ordinate %c	Orifice	Station %c	Ordinate %c
1	0.00	0.00	3	0.33	0.90
2	0.33	1.20	5	0.83	1.28
4	0.83	1.68	7	1.00	1.43
6	1.25	1.97	9	2.18	1.93
8	2.25	2.62	11	3.17	2.32
10	3.63	3.27	13	4.83	2.77
12	5.35	3.92	15	9.80	3.86
14	10.0	5.38	17	15.30	4.68
16	16.0	6.78	19	20.0	5.27
18	20.3	7.55	21	30.4	6.10
20	30.7	8.80	23	40.2	6.48
22	40.4	9.38	25	50.0	6.50
24	50.1	9.44	27	55.0	6.32
26	55.0	9.24	29	60.0	6.04
28	60.0	8.82	31	70.0	4.84
30	70.0	7.17	33	80.2	3.04
32	80.0	4.64	35	90.2	1.20
34	90.0	1.96	--	95.0	0.43
--	95.0	0.76	36	100.0	0.00
L.E. radius 1.58%			Thickness at T.E. = 0.3%		

Airfoil Pressure Port
Coordinates

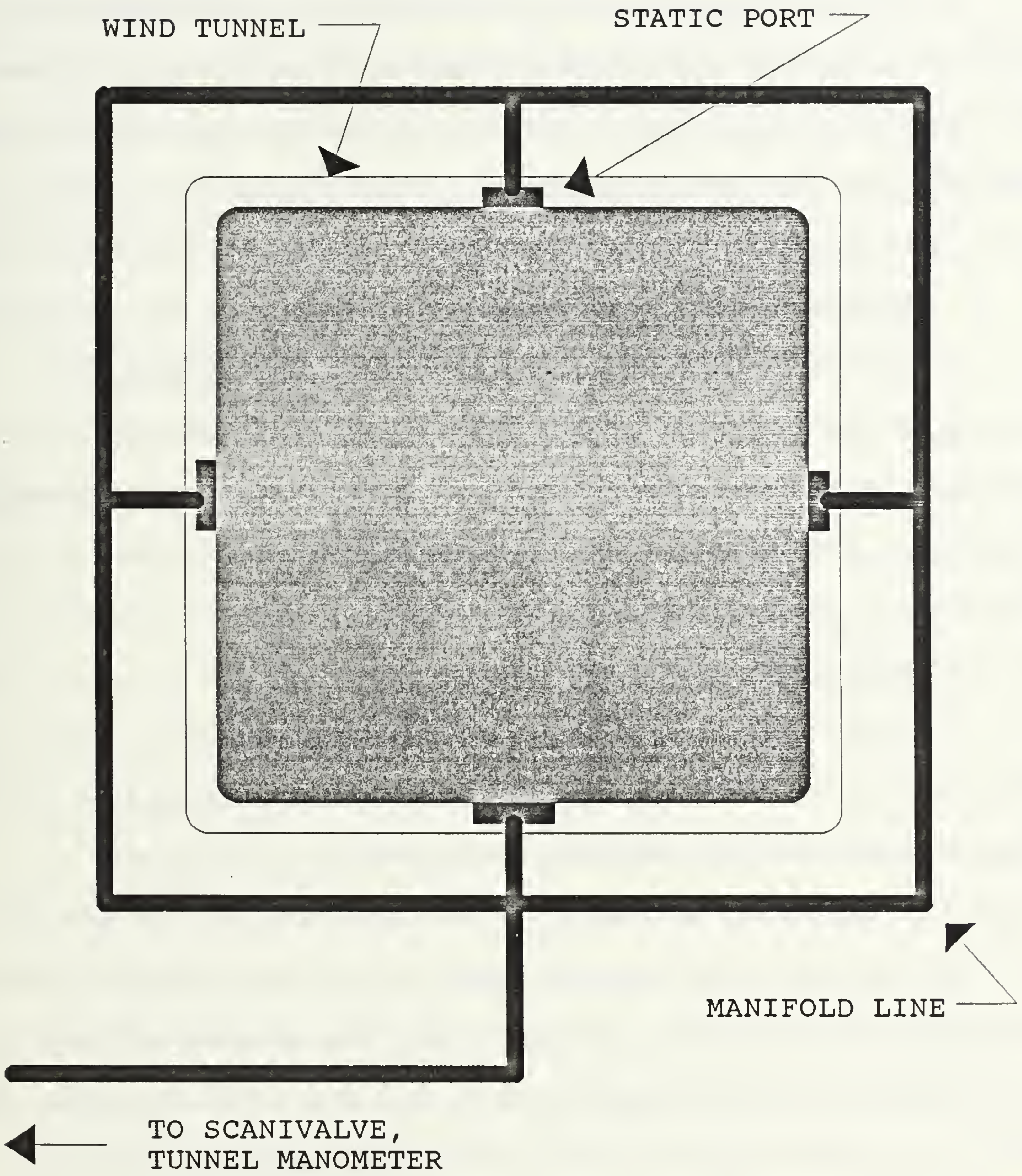
Figure 4

scanivalve was used because it provides an economical means of sensing the pressure of a multiple number of sources. It also simplifies calibration procedures in that only one transducer needs to be calibrated.

The scanivalve mechanism puts out a 7-bit binary coded decimal (BCD) signal that corresponds to the port (1 of 48) currently connected to the scanivalve transducer. This arrangement allows remote, electronic monitoring of the configuration of the port assembly of the scanivalve. This BCD signal is logic-reversed. That is, a "1" put out by the scanivalve would correspond to a "0" in TTL logic. For this reason, the scanivalve BCD output is inverted, bit by bit, before it is used for input into Hewlett-Packard digital indicators. The digital indicators provide a visual indication of the configuration of the scanivalve, apart from the automatic data aquisition process.

2. Settling Chamber Static Pressure

Settling chamber static pressure was taken from the static pressure manifold located in the settling chamber. The manifold consists of four static pressure ports located on the floor, ceiling, and side walls of the tunnel. Figure 5 shows the general design of the manifold. All four static pressure port are interconnected to reduce the effects of flow peculiarities.



Manifold Assembly

Figure 5

3. Free Stream Static Pressure

Tunnel free stream static pressure was taken from the other pressure manifold located at the entrance to the test section. It was constructed in the same manner as the manifold for the settling chamber static pressure, its location being the only difference.

4. Tunnel Dynamic Pressure

Tunnel Q indication was provided by a manometer, reading in centimeters of water. The manometer has, for its inputs, the pressures from the two static pressure manifolds described earlier. The manometer was a part of the established wind tunnel facility.

5. Temperature Sensing

Copper-constantan thermocouple probes were manufactured to provide inputs for the tunnel total temperature, and the ambient temperature. The probe measuring the total temperature was mounted in the tunnel wall of the settling chamber, just above the entrance door. The probe measuring the ambient temperature was mounted on a shelf built into the exterior of the tunnel wall between the settling chamber and the test section. Copper-constantan thermocouples were chosen for their sensitivity in the anticipated temperature ranges of tunnel operation.

6. Digital Thermometer

A digital thermometer was used to provide an output, in degrees fahrenheit, from the two copper-constantan thermocouples. This panel-type thermometer was mounted in the instrumentation rack and had facilities for manual operation as well as computer-controlled operation. It was desirable to have a visible means of temperature indication, independent from the automatic data aquisition process, so that an experimenter could check the system while the computer was involved in other tasks.

7. Calibration Pressure

Calibration pressure was provided by a manually operated u-tube manometer. The manometer had the capability of providing +- 60 cm of water of pressure for calibration. Cost considerations, lack of time, and the ready availability of the u-tube manometer prompted the choice to use it.

8. Calibration Pressure Measurement

Calibration pressure was measured by a differential pressure transducer. Its high input was connected to the calibration manometer, and its low input was vented to atmospheric pressure. This transducer was also a part of the wind tunnel facility and was available immediately.

9. Ambient Pressure

Ambient pressure was measured by an absolute pressure transducer that was simply vented to the ambient pressure of the room in which the wind tunnel was located. (Note: It was the intent of the design process for aquiring the various component transducers to use some very accurate

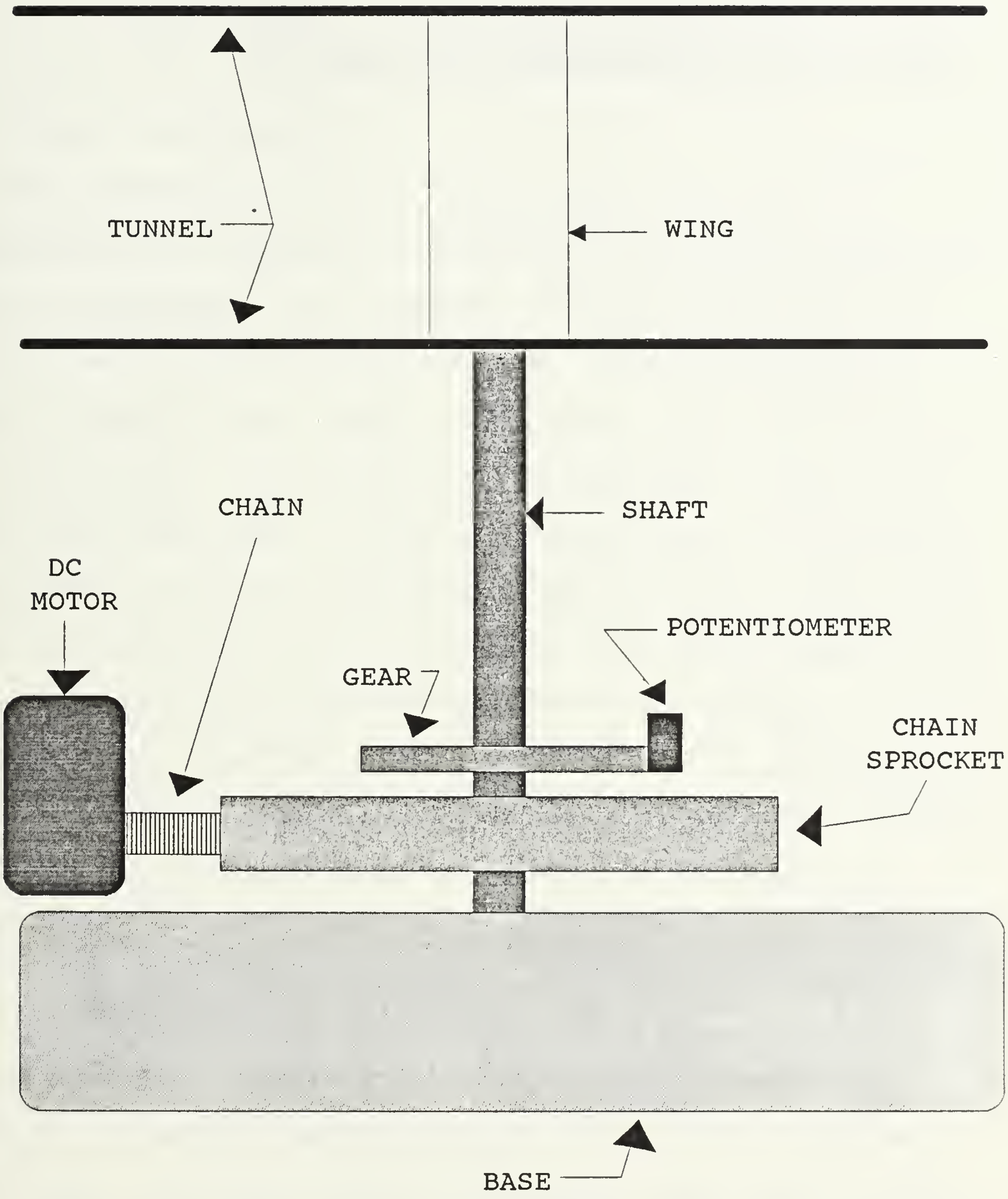
Digiquartz pressure transducers coupled to a pressure computer manufactured by Paroscientific. The delay of the delivery of these precise instruments prompted the use of the above mentioned transducers to measure calibration pressure and ambient pressure.)

10. Angle of Incidence

The angle of incidence was measured by a ten-turn potentiometer, coupled to the drive system of the wind tunnel turntable. Figure 6 shows the general layout of the turntable system. The wind tunnel turntable consisted of a DC-motor driving a large sprocket via chain drive. A shaft, concentrically attached to the large sprocket provided the means to turn the wing model. A simple relay system operated the DC-motor. The potentiometer was attached to a gear connected to the turntable. Its output was in millivolts.

11. Computer System

An IBM PC-AT was provided to accomplish the necessary data aquisition and control tasks. A device called the "Data Aquisition And Control Adapter" (henceforth called the "adapter") was provided to interface the computer to the analog/binary "outside world". The adapter consisted of four analog input channels, two analog output channels, one 16-bit input binary channel, one 16-bit output binary channel, a delay device, and a counter. Two adapters were installed inside the computer. They are labelled adapter



Turntable Assembly

Figure 6

#0, and adapter #1. Analog input requirements were + - 10 volts in the analog input channels. Binary input requirements were standard TTL logic.

III. COMPUTER - INSTRUMENT INTEGRATION

A. DIGITAL PANEL METERS

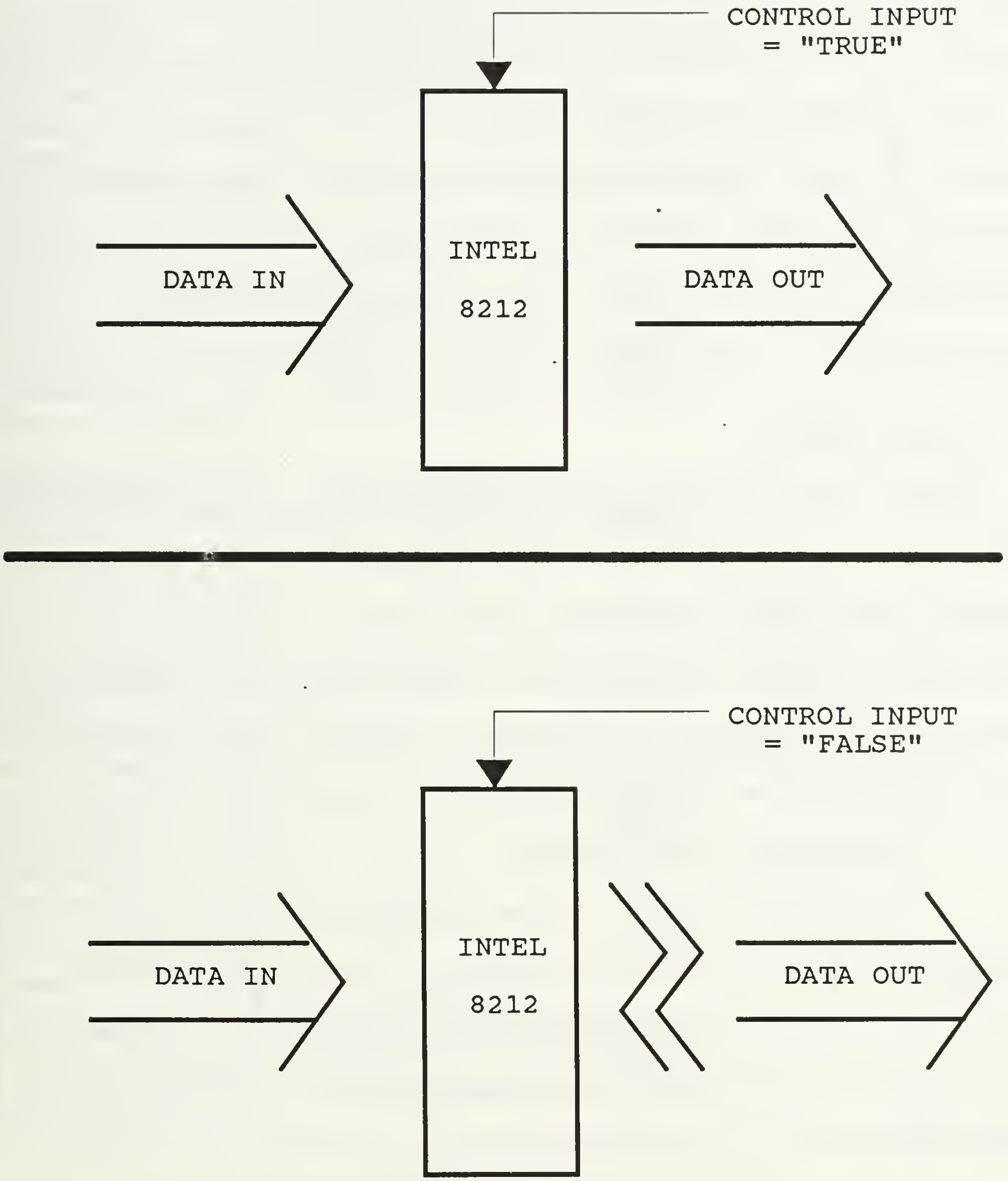
The design goal of this thesis was to use the Paroscientific digiquartz transducers and the Paroscientific pressure computer. The interface of this computer to the adapter would have been binary. For a temporary measure, generic digital panel meters (DPM) were utilized to function as substitutes. The Analog Devices DPM model #AD2024 was chosen for this task. It had a parallel BCD interface that was similar to that of the Paroscientific unit. These DPM's had slow conversion rates, but since they were installed to provide indications for the calibration and ambient pressure, and since this information was not time critical, the DPM's were considered appropriate for the task. A DPM was also used to provide an indication of the angle of incidence.

B. MULTIPLEXER DESIGN AND CONTROL

BCD encoded information is transported in parallel fashion (vs serial communications). Each digit requires four bits. Four full digits require 16 simultaneous binary signals to convey the information. Since the two data aquisition and control adapters contain two 16 bit channels between them, it quickly became apparent that 32 bits would not be sufficient to contain all the BCD channels required from the four DPM's, and the scanivalve port location indicators. A digital multiplexer was designed to multiplex up to 16, 16 bit channels into one 16 bit input into the computer.

The Intel 8212 8-bit input/output port was chosen as the prime component of the multiplexer. It allows a single digital input to control the throughput of data signals from its inputs to its outputs. For example, if a "1" is applied to its control input, data flows from its eight data inputs to the eight data outputs. Figure 7 illustrates this. If a "0" is applied to the control input of the 8212, then the data outputs "float" regardless of the signals at the data inputs.

Two 8212 chips complete a 16 bit input channel into the multiplexer. They are controlled as a pair. Overall control is maintained as follows: select one 16 bit channel by sending a "1" to the pair of 8212 chips controlling that channel, and by sending a "0" to all the other 8212's thereby preventing all other channels from interfering with the selected channel. Information can then pass unambiguously from any one 16 bit channel into the 16 bit digital input of one of the computer's data aquisition adapters (the other adapter's 16 bit digital input channel was not used).



Multiplexer Data Flow

Figure 7

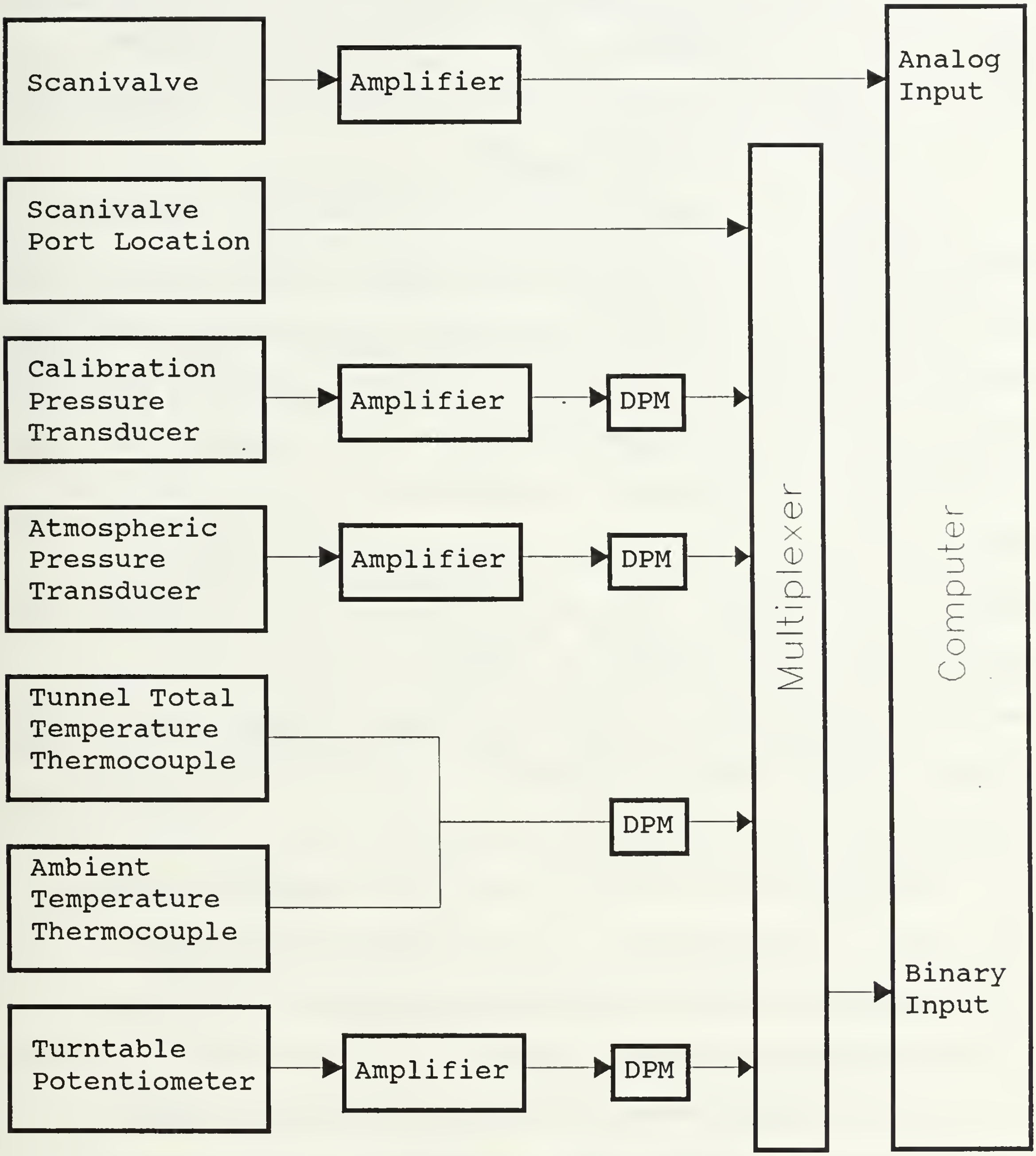
Overall control of the mulitplexer is performed through one of the computer data aquisition adapter 16 bit digital outputs. Thus sending a "2" will enable the second channel to send information; sending an "8" will enable the fourth channel to send information, and so forth. This digital multiplexer then is the key controlling device to obtain information from the DPM's, and the scanivalve port indicators.

C. SIGNAL PATHS

Figure 8 shows the general layout for signal paths from each transducer to the computer. There are four different signal paths that characterize the figure, the path for scanivalve output, the path for scanivalve port location, the path for calibration pressure, ambient pressure, and angle of incidence, and the path for temperature signals.

1. Scanivalve Signal Path

The signal path for the scanivalve, containing information about wing port pressure, free stream static pressure, and settling chamber static pressure, passes through the signal conditioner/amplifier. The signal is conditioned by a low pass filter with a cutoff of 10 hz. This removes most of the noise, including power line interference. The amplifier amplifies the signal from the



Signal Paths

Figure 8

millivolt range of the output of the transducer to the + 10 volt range of the adapter. The conditioned, amplified signal is input into analog input channel #0 of adapter #0.

2. Signal Path Through a DPM

Millivolt signals from the calibration transducer, ambient pressure transducer, and angle of incidence potentiometer are fed into the signal conditioner/amplifier. The gain of each is adjusted to calibrate each corresponding DPM in the desired engineering units. For example, the gain for the angle of incidence signal was adjusted at the amplifier so that the angle of incidence DPM reads out directly in degrees. The gain of the calibration signal was adjusted to allow its DPM to read in centimeters of water. Lastly, the gain of the ambient pressure signal was adjusted so that its DPM reads in lbs/in^2. All three signals are conditioned through the low pass filter with a cutoff of 10 hz. The DPM for each signal has a TTL compatible parallel BCD output. The computer receives these BCD signals via the multiplexer into the 16-bit binary input of adapter #0.

3. Temperature Signal Path

The outputs from the thermocouples were input directly into a special DPM, the Analog Devices AD2036 Digital Thermometer. This DPM contains the amplification, linearization, cold junction compensation necessary for accurate temperature measurements from the copper-constantan thermocouples. The DPM outputs directly in degrees fahrenheit and also has a TTL compatible parallel BCD output. The binary output from the AD2036 is input into the 16-bit binary input of adapter #0, via the multiplexer.

4. Port Location Signal Path

The computer receives scanivalve port location information from the same bit-inverters that generate the correct signal for the Hewlett-Packard port indicators. This BCD signal path also includes the multiplexer. The signal comes into the computer via the 16-bit binary input port of adapter #0.

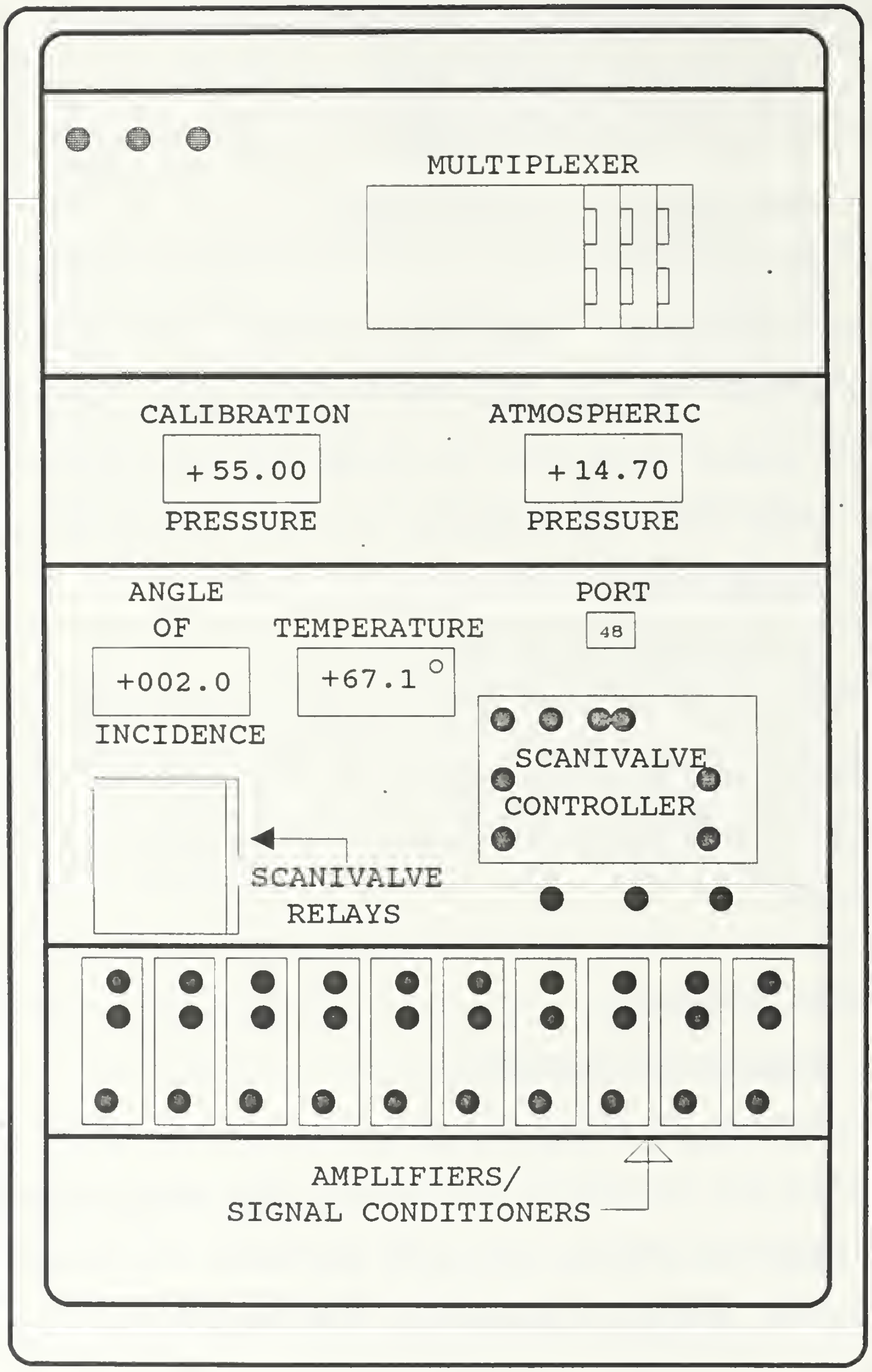
5. Instrumentation Rack

Figure 9 is a diagram of the instrumentation rack. All interconnections are made from the rear of the rack. Amplifier adjustments are made from the front of the amplifiers.

D. SYSTEM CONTROL

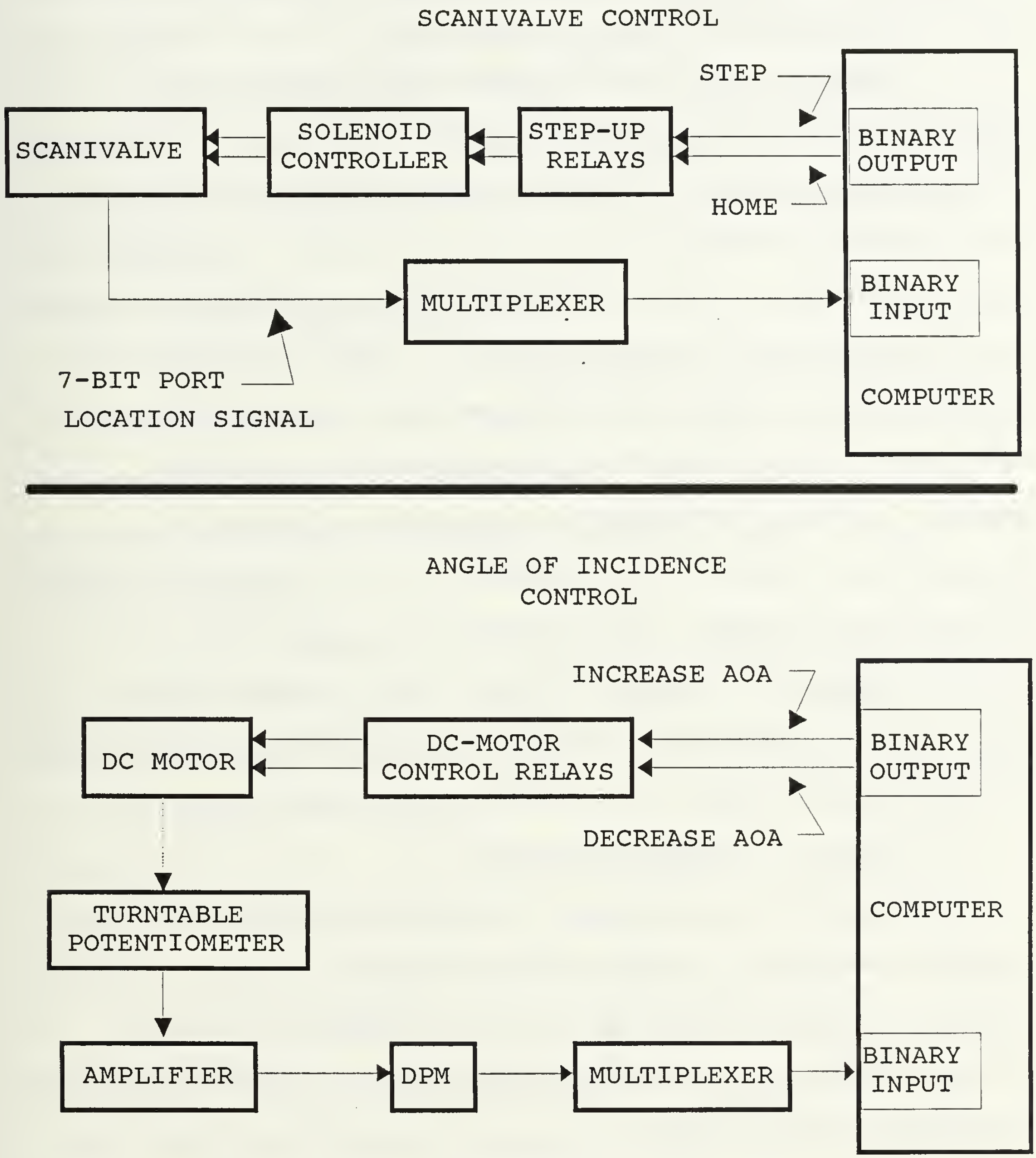
1. Scanivalve Control

Figure 10 shows the general control paths for the scanivalve and turntable position. The scanivalve consists of two separate units: one unit contains the transducer, motor drive, and port assembly, the second unit houses the solenoid controller. Control of the scanivalve is accomplished through the solenoid controller. The solenoid



Instrumentation Rack

Figure 9



System Control

Figure 10

controller allows two commands, STEP and HOME. STEP will "step" the scanivalve one port location. The HOME command sends the scanivalve to port number 48.

Computer control is accomplished through the solenoid controller. It will accept a 15 volt pulse to either STEP or HOME. A pulse is sent by using one of the 16-bit digital outputs. Bit position one (or the least significant bit) of the 16-bit binary output port of adapter #0 is reserved for the STEP command. STEP is accomplished by outputting a binary "1", followed by a binary "0". Similarly, bit position two is reserved for the HOME command. HOME is effected by outputting a binary "2", followed by a binary "0". A binary "2" will make the second bit position go high, while leaving all others low. Since, for the computer, a binary "1" or "high" is about five volts, an intermediate relay device was built to step up the voltage (to 15 volts) into the solenoid controller.

2. Angle of Incidence Control

The angle of incidence is controlled through the turntable assembly located beneath the test section. The DC motor is controlled through a relay system that allows manual control as well as computer control. Figure 10 shows the general arrangement.

Bit position three of the 16-bit binary output port of adapter #0 was reserved for angle of incidence - increase. By outputting a binary "4", the relay system will drive the DC motor in the positive direction. Similarly, bit position four is reserved for angle of incidence - decrease. By outputting a binary "8", the relay system will drive the DC motor in the negative direction. A binary "0" will cancel any previous commands and stop the turntable drive motor.

3. Temperature DPM Control

The AD2036 digital thermometer was different from the other DPM's in that it allowed six different inputs as opposed to one for the other DPM's. The temperature DPM also has provisions for manual control as well as for computer control.

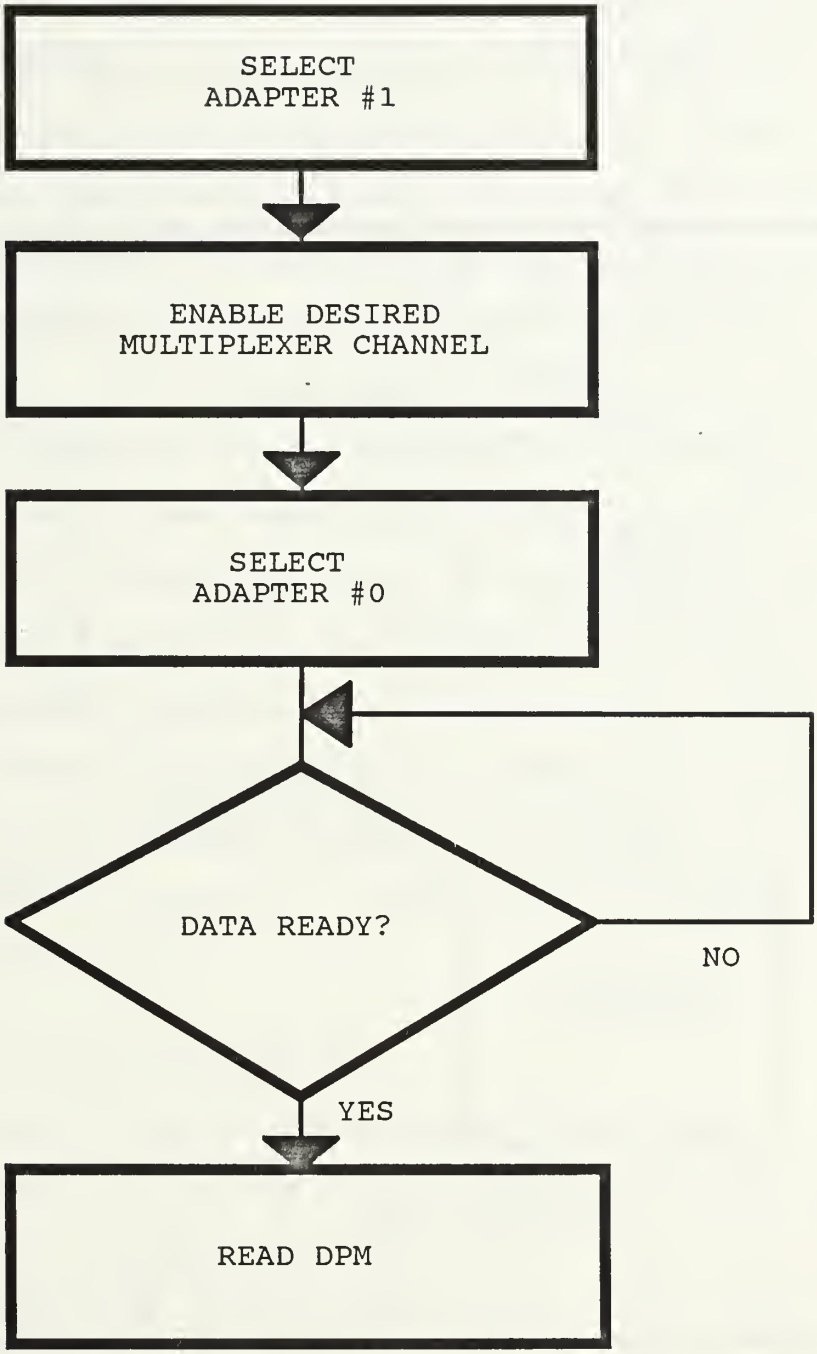
Each temperature channel was selectable through a BCD input. A binary "0" selected channel 0, a binary "5" selected channel 5, and so forth. On the binary output port of adapter #0, bit positions 5, 6, 7, were reserved for temperature channel selection. For example, outputting a binary "16" selected channel 1, outputting a binary "64" selected channel 4, and so on. Since, by design, a binary "0" was meant to reset any previous binary output commands in the data aquisition system, and since the AD2036 DPM calls its first channel: channel "0", any time a binary "0" is output, the temperature DPM will select channel "0" whether or not the computer is intending to read the AD2036 DPM.

IV. DATA AQUISITION

A. AQUIRING DIGITAL PANEL METER INFORMATION

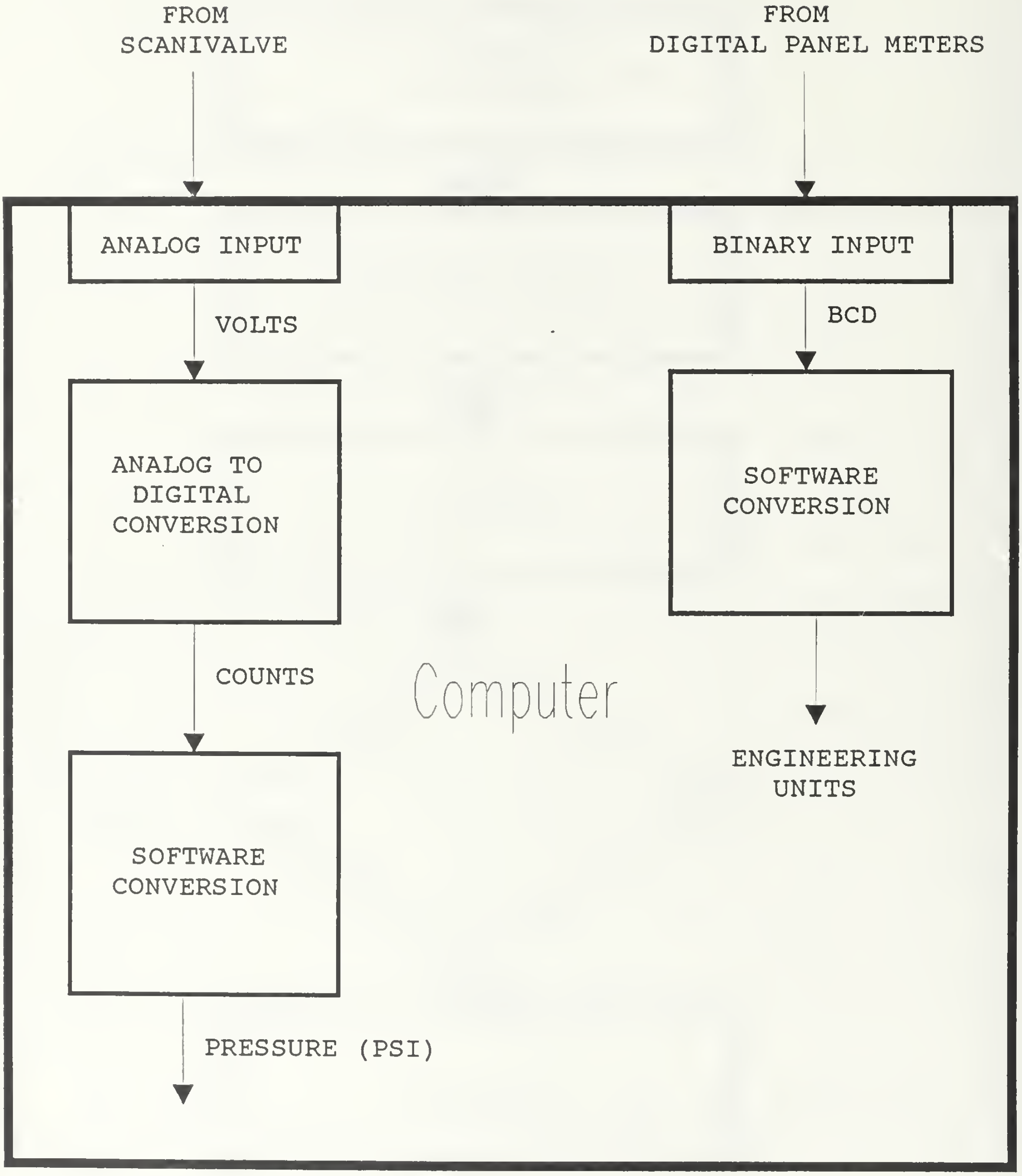
Figure 11 shows the general procedure for aquiring data from one of the DPM's. Figure 12 illustrates the data conversion procedure for obtaining engineering units. Adapter "0" was reserved to control the scanivalve, turntable position, and temperature channel. Adapter "1" was reserved to control multiplexer channel selection. Control sequence for reading a DPM goes as follows: (1) select adapter "1"; (2) through the digital output device, output a binary number corresponding to the multiplexer channel handling the DPM of choice; (3) select adapter "0"; (4) through the digital input device (information through the multiplexer), test for a "data ready" signal from the DPM; (5) when the "data ready" signal is true, read the DPM. The DPM will have ready at its parallel BCD outputs, the information that is currently displayed on the face of the meter.

Not all 16 bits of the digital input are assigned to numerical information. For example, the highest angle of incidence possible would be 360.0 degrees. In BCD code, that would require three full BCD digits and one "1/2" BCD digit. That is, the most significant BCD digit would only require two of its four binary bits since the highest number



DPM Data Aquisition

Figure 11



Computer Data Aquisition

Figure 12

required of that BCD digit would be a "3". This frees two bits of information that could be used for other purposes. Free bits are devoted to such uses as conveying the "data ready" signal, and the polarity signal. For each DPM, both the control information and the numerical data information inbound to the computer can be handled through one multiplexer channel. To read the temperature DPM, add step (3a): through the digital outputs, send the required binary number corresponding to the temperature channel desired.

All of the DPM information already exists in engineering units. When the data arrives at the computer through the binary input port, the computer does not "know" that it is in BCD. After the data is read from the binary input port into a variable in the computer program, it must be converted into decimal notation. A simple algorithm was written to accomplish this.

B. AQUIRING SCANIVALVE INFORMATION

Scanivalve pressure information passes through the signal conditioner/amplifier and then directly into one of the analog inputs of adapter "0". The onboard A/D converter produces an integer value in the range 0-4096. This number must be converted into engineering units in software. This is accomplished during the scanivalve calibration portion of the computer program.

1. Scanivalve Calibration

The scanivalve is calibrated by applying a known pressure source to the scanivalve transducer. The known pressure source was provided by the calibration manometer. The known value of the calibration pressure source is read from the calibration pressure DPM. Both transducers are referenced to ambient pressure. When the computer has obtained the above information, the scanivalve transducer can be calibrated and a calibration constant calculated.

The scanivalve is commanded to two ports, one for the calibration pressure, and the second for reference, or ambient pressure. The information returned is in counts (calcount, refcount, from the A/D converter). The following equation is used to calculate the calibration constant:

calconstant = (calcount - refcount) / cal pressure(PSI).

Calibration is complete when the calibration constant is calculated. The units of the calibration constant are counts per PSI.

2. Scanivalve Pressure Information

The pressure at any port may now be obtained by commanding the scanivalve to that port and obtaining the A/D count. The following equation is then used to convert pressure(count) into pressure(PSI):

pressure(PSI) = (pressure(count) - refcount) / calconst

3. Coefficient of Pressure Calculation

The coefficient of pressure is obtained by first calculating the tunnel dynamic pressure: Q. Q is obtained by commanding the scanivalve to the ports containing the tunnel settling chamber static pressure (P_2) and the free stream static pressure (P_1). Q is calculated by the following [Ref. 1:pp. A1-4]:

$$Q = ((P_2(PSI) - P_1(PSI)) / (1 - (A_2/A_1)^2)) * 1.075$$

where A_2 = area of the test section, A_1 = area of the settling chamber, and 1.075 = the tunnel calibration factor. The tunnel calibration factor was obtained by referencing Q_{ideal}, measured through the two static pressure manifolds, to Q_{true}, measured through a special calibrated pitot tube [Ref. 1:pp. A1-5]. The coefficient of pressure (C_p) follows:

$$C_p = (pressure(PSI) - pstat(PSI)) / Q.$$

V. DATA ANALYSIS

A. COMPUTER MATHEMATICAL INTEGRATION

The data reduction portion of the program requires various integrations to be performed. Since the pressure data can be varying widely, the common trapezoidal rule was avoided. The following method was developed to integrate the pressure coefficient: (1) A clamped cubic spline algorithm splines the data [Ref. 2:pp. 122-123]. To use the clamped cubic spline, derivatives at the end points are calculated first. The spline algorithm produces a third degree polynomial between each pair of data points; (2) Using Simpson's Rule [Ref. 2:pp. 188], each polynomial is integrated over the interval between each pair of data points; (3) The separate integral subtotals are summed to provide a final value.

B. DATA REDUCTION

1. Force Coefficients

C_P is calculated for each port on the wing model. By the integration procedure outlined in the previous section, the force coefficient in the Y-direction (C_Y), and the force coefficient in the X-direction (C_X) can be calculated [Ref. 1:pp. B2-5]. The following equations apply:

$$C_Y = \int_2^1 (C_{Pl} - C_{Pu})\, d(x/c)$$

$$C_X = \int_{-y_{1/c}}^{y_{2/c}} (C_{Pf} - C_{Pr})\, d(y/c)$$

where $C_{Pl} = C_P$ over the lower surface; $C_{Pu} = C_P$ over the upper surface; $C_{Pf} = C_P$ over the front surface; $C_{Pr} = C_P$ over the rear surface; (x/c) = x coordinate normalized to the chord length; (y/c) = y coordinate normalized to the wing thickness.

2. Moment Coefficient

In a similar manner, the moment coefficient about the leading edge (C_{Mle}) can be calculated using the following equation:

$$C_{Mle} = \int_{-y_{1/c}}^{y_{2/c}} (C_{Pf}-C_{Pr})*(y/c)\, d(y/c) - \int_2^1 (C_{Pl}-C_{Pu})(x/c)\, d(x/c)$$

3. Coefficient of Lift and Drag

The angle of incidence (alpha) is obtained from the angle of incidence DPM. The coefficient of lift (C_l) and coefficient of drag (C_d) can now be calculated using the following equations [Ref. 1:pp. B2-7]:

$$C_l = C_Y * \cos(alpha) - C_X * \sin(alpha)$$

$$C_d = C_X * \cos(alpha) - C_Y * \sin(alpha)$$

VI. RESULTS

A. COEFFICIENT OF PRESSURE

Figure 13 shows a sample output of a data run taken at 10 degrees angle of incidence and Reynolds number (Re) equal to 7.5 X 10^5. The data reveals a pressure leak in the model connections at about (x/c) = .72. Otherwise, the pressure distribution conforms to expectations.

B. LIFT COEFFICIENT

Figure 14 shows the results of a series of data runs performed at Re = 7.5 X 10^5, from -18° to +20° angle of incidence. The graph compares data from a previous lab setup, but under approximately similar conditions. Figure (14) also shows the more established NACA data. As shown in the graph, the experimentally obtained C_l is consistently below that of the NACA results. This may be accounted for by several reasons. First, the Reynolds number of the NACA data (at 2.9 X 10^6) is much higher than of the experimental data. Lower Reynolds numbers tend to lower the C_l. Second, there may be an error in the angle of incidence calibration which could shift the experimental results either left, or right. Third, there is a likely possibility that the wing models are not exactly similar. Fourth, the leak in one of the pressure lines has caused an erroneous data point that

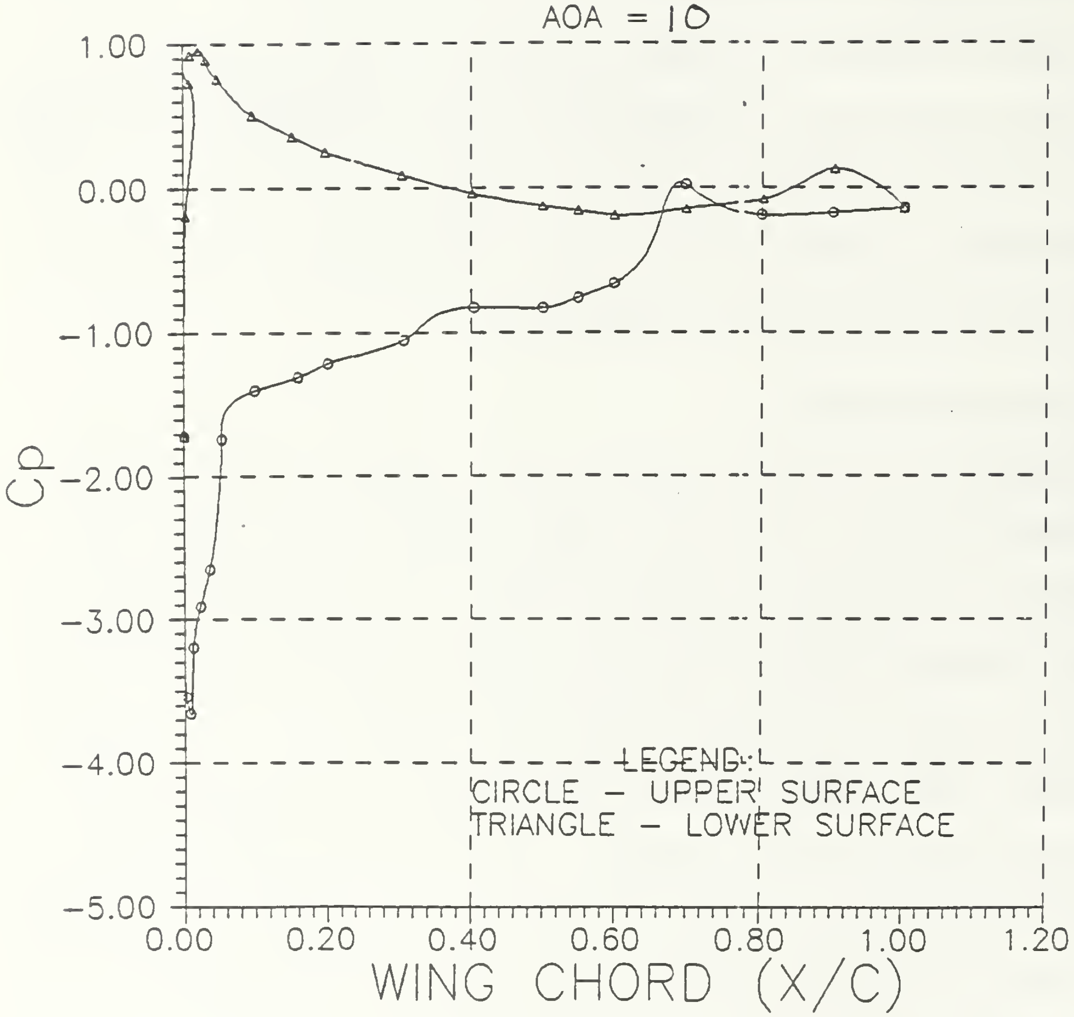
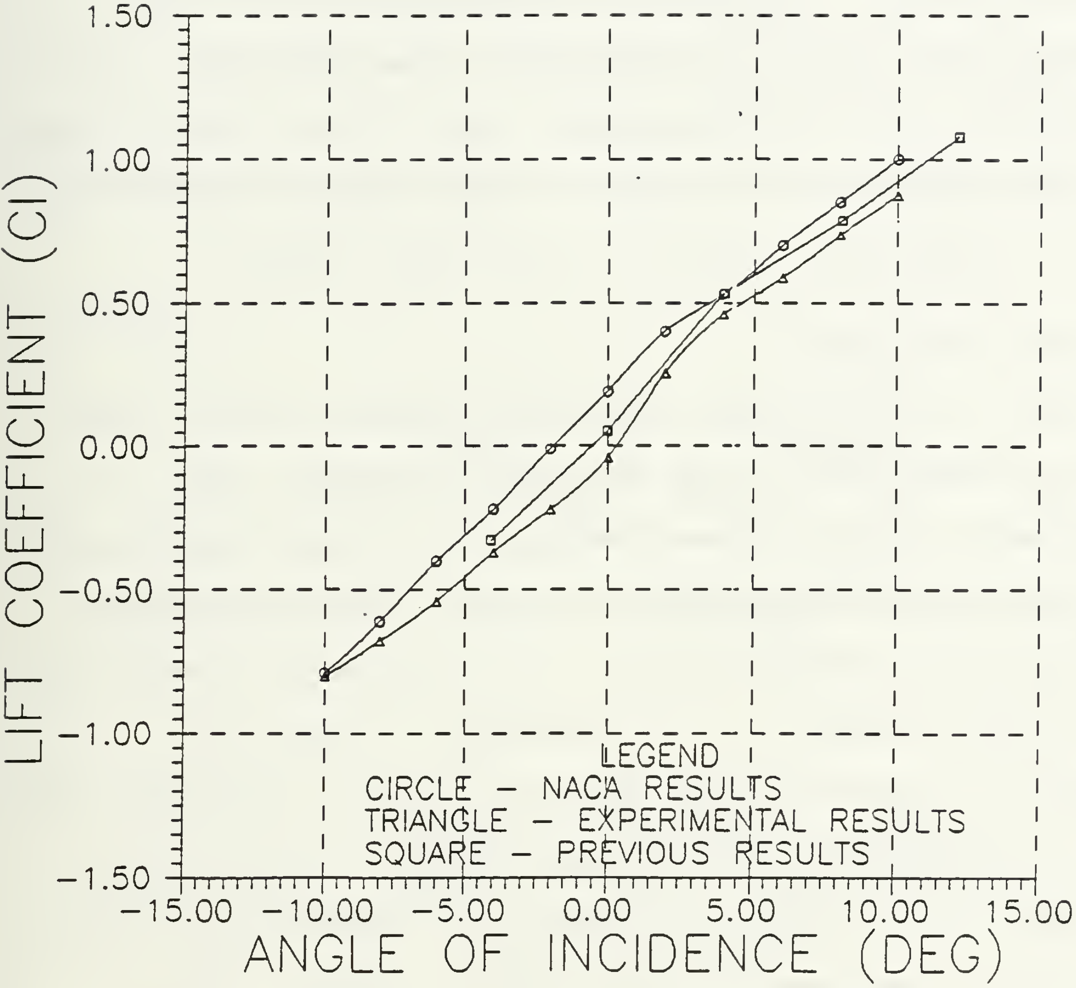

AOA = 10
1.00
0.00
−1.00
−2.00
−3.00
−4.00
−5.00
Cp
0.00
0.20
0.40
0.60
0.80
1.00
1.20
WING CHORD (X/C)
LEGEND:
CIRCLE – UPPER SURFACE
TRIANGLE – LOWER SURFACE

Figure 13



1.50
1.00
0.50
0.00
−0.50
−1.00
−1.50
LIFT COEFFICIENT (Cl)
−15.00
−10.00
−5.00
0.00
5.00
10.00
15.00
ANGLE OF INCIDENCE (DEG)
LEGEND
CIRCLE – NACA RESULTS
TRIANGLE – EXPERIMENTAL RESULTS
SQUARE – PREVIOUS RESULTS

Figure 14

is forcing the integral of the C_p over the upper surface to be higher than anticipated. The reason why the experimental data is further from the NACA data than the previous results is more puzzling. Perhaps the differences lie in the different techniques of integration. The previous calculations used the trapezoidal rule. Also, the previous data was measured at a slightly higher Reynolds number (8.9 X 10^5). This could account for the upward shift.

C. DRAG COEFFICIENT

Figure 15 shows more results calculated from the same data runs mentioned in the previous paragraph. The experimental results appear to generally agree with the NACA data but some discrepancies are noted. The following are postulated as possible reasons. The existence of negative drag, located about C_l = .48, suggests the overall difficulty in working with the small numbers associated with the C_d calculations. Noise present in the system destroys the resolution of the measurements and becomes magnified in the integrations used in calculating C_X and C_Y. The breakdown in the overall shape of the drag bucket about C_l = .48 indicates that the data is close to the experimental error. Lastly, the existence of the leak in one of the pressure lines may have influenced the drag toward negative numbers due to the erroneous higher pressure on the aft portion of the upper surface of the wing model. The

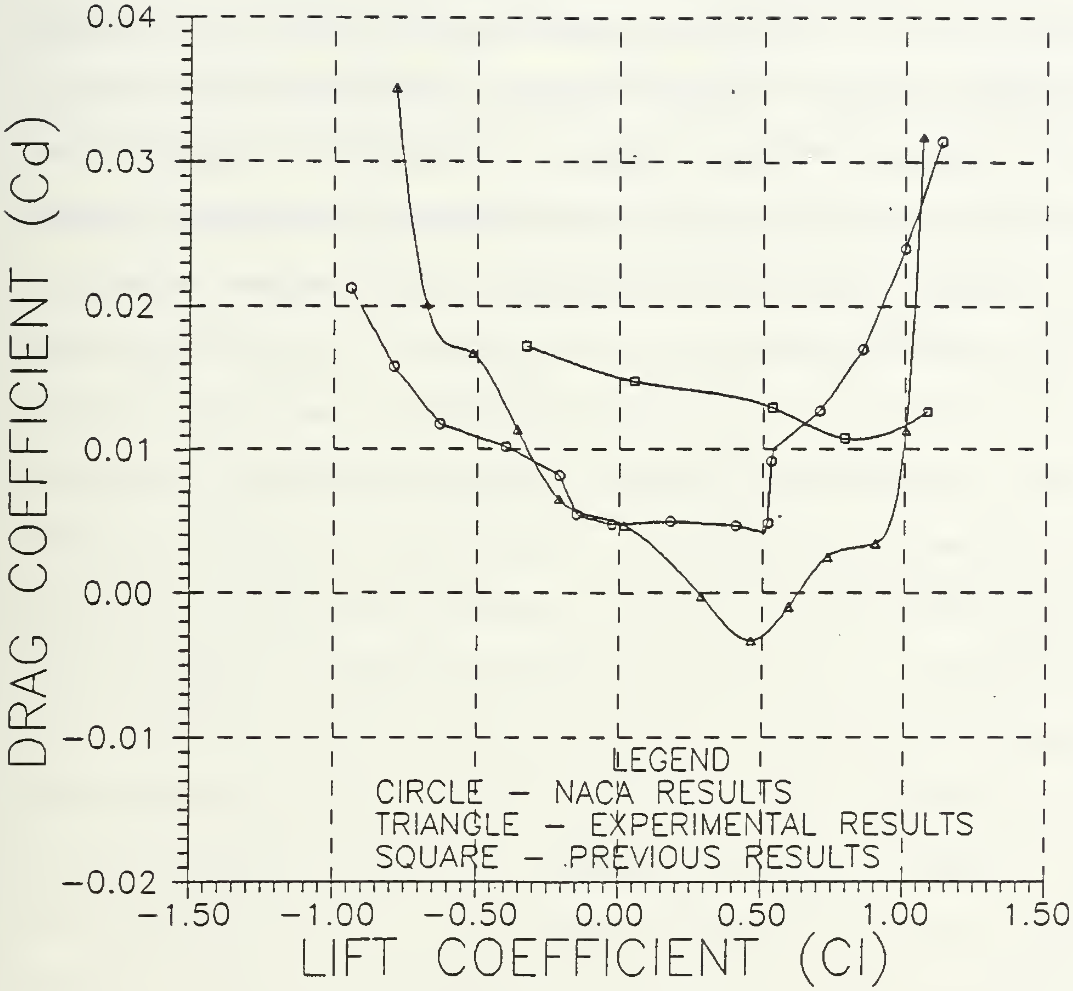

0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
DRAG COEFFICIENT (Cd)
−1.50
−1.00
−0.50
0.00
0.50
1.00
1.50
LIFT COEFFICIENT (Cl)
LEGEND
CIRCLE – NACA RESULTS
TRIANGLE – EXPERIMENTAL RESULTS
SQUARE – PREVIOUS RESULTS

Figure 15

experimental data appears closer to the NACA data than the previous results. The general shape has improved but data in the C_l= 0.0 - 1.0 range prohibits a more positive endorsement. One final comment: the NACA curves probably resulted from the smoothing over of many data runs. These experimental results resulted from a single run taken to test the data aquisition system.. A systematic approach to this experiment, coupled with experience in working with the various experimental variables should improve the overall quality of the results.

VII. CONCLUDING REMARKS

A data aquisition system has been designed that incorporates all of the necessary elements to complete the system from transducer to final output. All of the hardware requirements including transducers, input circuits, signal conditioning, transmission, processing, storage, and output have been assembled and integrated to ensure faithful representation of the original measurable quantity. A computer has been integrated into the design, including the computer interfaces such as a digital multiplexer, and analog to digital converters. A computer program has been written to bring the system to life, and to perform the later stages of signal processing necessary to complete the data analysis portion of the system. Hardware and software requirements of the final output have been assembled and include online display and hardcopy output of tabular data and graphical data. The lessons learned from this thesis project will be of value to forthcoming improvements to the Low Speed Wind Tunnel Facility at the Naval Postgraduate School.

VIII. RECOMMENDATIONS

In working with the tunnel apparatus, one deficiency became clear: the turntable positioning system is inadequate for precise measurements. First, there is too much play in its chain drive assembly. Angle of incidence adjustments that require direction reversals become blurred in the chain slack. The relay / DC motor system is imprecise and must rely on software timing to correctly position the turntable. Also, the relays were not designed for rapid on-off operation, allowing small increments of turntable adjustment. A precision stepper motor and new drive system are recommended changes.

Efficiency could be improved greatly with the addition of a computer controlled tunnel speed adjustment device. At present, the program user must manually adjust the tunnel Q. Also, the flexibility of the experiment would be improved by adding options to vary the tunnel Q as well as angle of incidence in the same data recording session.

The Scanivalve transducer used to obtain the experimental results has had known problems in past experiments. Another, more accurate transducer has been ordered that should improve the accuracy of the data. It is recommended that when the newer transducer arrives, the experiments be repeated.

The data aquisition and control adapter comes equipped with an internal 12-bit analog to digital converter. The manufacturer of the data aquisition and control adapter (Cyborg Corp.) has available higher resolution A/D converters that can easily be incorporated into the present system. Using higher resolution A/D converters with the new scanivalve transducer may improve the accuracy of the drag measurements.

The data aquisition system built for this thesis project began with a given assortment of equipment that reduced the overall flexibility and usefulness of the system from the start. The author has since learned that there are systems available from reputable manufacturers (such as Hewlett Packard, Omega Engineering, Fluke) that incorporate complete integrated designs from transducer to final output. These systems could provide the basis of an extremely flexible and capable data aquisition system with room to spare for growth and enhancement. They are recommended.

APPENDIX A

```
'MAIN PROGRAM

     'SCANPRO IS THE PROGRAM FOR THE SCANIVALVE EXPERIMENT.

     KEY OFF: CLS
     GOSUB INITIALIZE

     S$ = "1": SUBROUTINE = 1: ENDPRO$ = "1": ERRORR$ = "NO"
     WHILE ENDPRO$ <> "0"
          GOSUB GREETINGS
          IF ENDPRO$ = "0" THEN GOTO 1
          S$ = "1"
          WHILE S$ <> "0"
               ON SUBROUTINE GOSUB PROMPTAOA,_
                                                CALIBRATE,_
                                                SPEED,_
                                                DISPLAYVAR,_
                                                RECORD,_
                                                DISPLAYDATA,_
                                                FILEDATA,_
                                                REDUCEPLOT1,_
                                                REDUCEPLOT2,_
                                                PLOTCP
               SUBROUTINE = SUBROUTINE + 1
               IF SUBROUTINE = 11 THEN S$ = "0"
          WEND
          SUBROUTINE = 1
1  WEND
   COLOR 7,0: CLS: END
'***********************************************************
INITIALIZE:

     'ROUTINE INITIALIZE INITIALIZES CERTAIN DATA AQUISITION
          AND CONTROL
     'ADAPTER VARIABLES, AND OTHER DATA STORAGE VARIABLES
          NEEDED.

     'INPUTS: NONE
     'OUTPUTS: NONE

     CTRL% = 0: STOR% = 0: STAT% = 0: MODE% = 0: HNDSHK% = 0
     RATE = 500
     XORMSK% = &H0
     DIM DATAVAR%(50), COUNTS(50), PRESSURE(50),
         WANTAOA(80), WINDATA(3900)
     DIM X(20), Y(20), L(20), U(20), Z(20), B(20), C(20),
         D(20), G(3,4)
     DIM XWINGU(20), XWINGL(20), YWINGU(20), YWINGL(20),
         CPUPPER(20), CMLE(80)
     DIM CPLOWER(20), CX(80), CY(80), CL1(80), CD1(80),
         ALPHA1(80), CMC41(80)
     DIM CMAC(80), DATA3$(11), DATA2(10), DATA1(10), CP(38),
         XCOORD(38)
     DIM PLOTDU1(20), PLOTDL1(20), PLOTDU2(20), PLOTDL2(20)
     DIM PLOTDU3(20), PLOTDL3(20), PLOTAOA(3), PLOTCOUNT$(3)
     'NEXT LINE CONTAINS DATA FOR XWINGU()
2    DATA 0.0, .0033, .0083, .0125, .0225, .0363, .0535, .1,
         .16, .203, .307, .404, .501, .550, .6, .7, .8, .9,
          1.0
     'NEXT LINE CONTAINS DATA FOR XWINGL()
     DATA 0.0, .0033, .0083, .01, .0218, .0317, .0483, .098,
         .153, .2, .304, .402, .5, .55, .60, .7, .802, .902,
          1.0
     'NEXT LINE CONTAINS DATA FOR YWINGU()
     DATA 0.0, .012, .0168, .0197, .0262, .0327, .0392,
         .0538,
         .0678, .0755, .0880, .0938, .0944, .0924, .0882,
         .0717, .0464, .0196, 0.0
     'NEXT LINE CONTAINS DATA FOR YWINGL()
     DATA 0.0, .009, .0128, .0143, .0193, .0232, .0277,
         .0386,
         .0468, .0527, .0610, .0648, .065, .0632, .0604,
         .0484, .0304, .0120, 0.0
     FOR I = 1 TO 19: READ XWINGU(I): NEXT I
     FOR I = 1 TO 19: READ XWINGL(I): NEXT I
     FOR I = 1 TO 19: READ YWINGU(I): NEXT I
     FOR I = 1 TO 19: READ YWINGL(I): NEXT I
     RESTORE 2
     XCOORD(1) = 0: READ DUMMY
     FOR I = 2 TO 36 STEP 2: READ XCOORD(I): NEXT I
     READ DUMMY
     FOR I = 3 TO 35 STEP 2: READ XCOORD(I): NEXT I

     'DEFINE A FUNCTION THAT WILL BE USED FOR CUBIC SPLINE /
          INTEGRATE
     DEF FNSPLINE(I,JJ)=Y(I)+B(I)*(JJ-X(I))+C(I)*(JJ
     -X(I))^2+D(I)*(JJ-X(I))^3
RETURN
'***********************************************************
ERRORR:

     'THIS ERROR ROUTINE DETECTS WHEN THE DATA AQUISITION
     'AND CONTROL ADAPTER RETURNS AN ERROR CONDITION THROUGH
     'THE VARIABLE STAT%.   IF AN ERROR DOES OCCUR,
     'THE PROGRAM IS AUTOMATICALLY ABORTED. (NOT USER
     'FRIENDLY!)
     'INPUTS: STAT%
     'OUTPUTS: NONE

     IF STAT% <> 0 THEN
            SCREEN 0: LOCATE 12,10
            PRINT USING "EXECUTION ERROR ###"; STAT%
            LOCATE 20,10
            PRINT "CONSULT THE DATA AQUISITION AND CONTROL
            PROGRAMMING SUPPORT"
            LOCATE 21,10
            PRINT "MANUAL APPENDIX FOR INFORMATION ON THIS
            EXECUTION ERROR."
            LOCATE 23,27: PRINT "ENTER ANY KEY TO CONTINUE:"
            ERRORR$ = INPUT$(1): S$ = "0": GOTO 1
       END IF
RETURN
'************************************************************
SVPORT:

     'S.V. PORT LOCATION READS THE SCANIVALVE BCD PORT
          LOCATION

     'INPUTS: NONE
     'OUTPUTS: PORT

     ADAPT% = 1: RAWVALUE% = 1: DEVICE% = 8
     GOSUB BINOUT   'CALL BOUS, ENABLE MULTIPLEXER CHANNEL
     #1: S.V. PORT
     ADAPT% = 0:DEVICE% = 8: COUNT = 1
     ANDMSK% = &H7F 'MASK OUT UPPER 9 BITS, ALLOW BITS 0-6
     GOSUB BININ   'CALL BINM
     BCD = DATAVAR%(1): GOSUB BCD   'CALL BCD TO DECIMAL
     PORT = DECIMAL
RETURN
'************************************************************
BCD:

     'BCD TO DECIMAL CONVERTS THE BCD/BINARY NUMBER RECEIVED
          FROM BINM OR
     'THE S.V. PORT DISPLAY TO DECIMAL FOR NORMAL USE.

     'INPUTS: BCD
     'OUTPUTS: DECIMAL

     DECIMAL = 0
     FOR EXPONENT = 12 TO 0 STEP -4
          TRUNC = FIX(BCD / (2 ^ EXPONENT))
          DECIMAL = DECIMAL + (10 ^ (EXPONENT / 4) * TRUNC)
          BCD = BCD - (TRUNC * (2 ^ EXPONENT))
     NEXT EXPONENT
 RETURN
'***********************************************************
RESETT:

     'RESET SENDS A BINARY "0" OUT TO NULLIFY ANY "ONE'S"
           PREVIOUSLY SENT.
     'IT IS USED TO CLEAR RELAY COMMANDS SUCH AS STEP, HOME,
           AOA+, AOA-.

     'INPUTS: NONE
     'OUTPUTS: NONE

     ADAPT% = 0: RAWVALUE% = 0: DEVICE% = 8
     GOSUB BINOUT
RETURN
'***********************************************************
HOME:

     'HOME "HOMES" THE SCANIVALVE TO PORT NUMBER ONE.

     'INPUTS: NONE
     'OUTPUTS: NONE

     ADAPT% = 0: RAWVALUE% = 2: DEVICE% = 8: COUNT = 2000
     GOSUB BINOUT   'CALL BOUS, SEND THE SCANIVALVE TO PORT
           48
     GOSUB RESETT
     GOSUB DELAY   'WAIT TWO SECONDS, LET THE S.V. CATCH UP
     GOSUB STEPP   'STEP THE SCANIVALVE TO PORT 1
RETURN
'***********************************************************
STEPP:

     'STEPP "STEPS" THE SCANIVALVE TO THE NEXT PORT
           LOCATION AND UPDATES
     'THE VARIABLE "PORT".

     'INPUTS: NONE
     'OUTPUTS: PORT

     ADAPT% = O: RAWVALUE% = 1: DEVICE% = 8: COUNT = 100
     GOSUB BINOUT   'STEP SCANIVALVE TO NEXT PORT
     COUNT = 1: GOSUB DELAY: COUNT = 100
     GOSUB RESETT
     GOSUB DELAY   'WAIT FOR 1/40 SECOND (PREVENT S.V.
           BURNOUT)
     GOSUB SVPORT   'VERIFY LOCATION OF SCANIVALVE
RETURN
'***********************************************************
```

```
ANALOGIN:

     'READS, AVERAGES 50 SAMPLES FROM THE ANALOG CHANNEL
           SPECIFIED BY CHANLO%

     'INPUTS: CHANLO%
     'OUTPUTS: ANADATA

     ADAPT% = O: DEVICE% = 9: COUNT = 1000
     IF CHANLO% <> 0 THEN GOTO 3   'DELAY ONLY IF READING
           THE SCANIVALVE
     GOSUB DELAY   'DELAY FOR 1 SECOND TO LET PRESSURE
           STABILIZE
3  COUNT = 50
     GOSUB ANAIN   'READ 50 VALUES
     SUM = 0
     FOR I = 1 TO COUNT   'AVERAGE THE 50 VALUES
          SUM = SUM + DATAVAR%(I)
     NEXT I
     ANADATA = SUM / COUNT
RETURN
'************************************************************
READPORT:

     'READPORT POSITIONS THE SCANIVALVE TO A SINGLE PORT
           SPECIFIED BY
     '"WANTPORT", READS THE DATA, AND RETURNS THE DATA IN
           "ANADATA" AND COUNTS

     'INPUTS: WANTPORT
     'OUTPUTS: ANADATA, COUNTS()

     GOSUB SVPORT   'FIND S.V. CURRENT POSITION
     IF WANTPORT < PORT THEN GOSUB HOME
     WHILE WANTPORT > PORT
          GOSUB STEPP
     WEND
     CHANLO% = 0
     GOSUB ANALOGIN
     COUNTS(PORT) = ANADATA
RETURN
'************************************************************
BINOUT:

     'THIS ROUTINE OUTPUTS THE DECIMAL VARIABLE "RAWVALUE%"
           IN BINARY
     'VIA TERMINALS BOO - BO15

     'INPUTS: ADAPT%, DEVICE%, RAWVALUE% (HNDSHK% AND STAT%
           ARE INITIALIZED AT PROGRAM START)
     'OUTPUTS: NONE
     CALL BOUS (ADAPT%, DEVICE%, HNDSHK%, RAWVALUE%, STAT%)
     GOSUB ERRORR
RETURN
'***********************************************************
BININ:

     'THIS ROUTINE INPUTS BINARY SIGNALS FROM THE TERMINALS
           BI0 - BI15, AND
     'RETURNS THE VALUE IN THE ARRAY DATAVAR%.

     'INPUTS: ADAPT%, DEVICE%, ANDMSK%, COUNT.  (HNDSHK%,
           MODE%, STOR%,
     'OUTPUTS: DATAVAR%( ) (NOTE: OFTEN ONLY THE FIRST ITEM
           IN THE ARRAY DATAVAR%() IS REQUESTED)

     CALL BINM (ADAPT%, DEVICE%, HNDSHK%, MODE%, STOR%,
           ANDMSK%, XORMSK%, COUNT, RATE, DATAVAR%(1),
           STAT%)
     GOSUB ERRORR
RETURN
'***********************************************************
DELAY:

     'THIS ROUTINE USES THE DATA AQUISITION AND CONTROL
           ADAPTER TO DELAY "COUNT" MILLISECONDS.

     'INPUTS: ADAPT%, COUNT.  (STAT% IS INITIALIZED AT
           PROGRAM START)
     'OUTPUTS: NONE

     CALL DELAY (ADAPT%, COUNT, STAT%)
     GOSUB ERRORR
RETURN
'***********************************************************
ANAIN:

     'THIS ROUTINE INPUTS AN ANALOG VOLTAGE VIA TERMINALS
           A/D0 - A/D3, AND
     'RETURNS A COUNT IN THE RANGE (0 - 4096).  A READING IS
           TAKEN "COUNT"
     'NUMBER OF TIMES. EACH READING IS RETURNED IN THE
           ELEMENTS OF DATAVAR%.
     'CHANLO% SELECTS THE ANALOG CHANNEL TO BE READ (0 - 3).

     'INPUTS: ADAPT%, DEVICE%, CHANLO%, COUNT, RATE.
           (CTRL%, MODE%, STOR%, RATE - ARE INITIALIZED AT
           PROGRAM START)
     'OUTPUTS: DATAVAR%( )

     CALL AINM (ADAPT%, DEVICE%, CHANLO%, CTRL%, MODE%,
            STOR%, COUNT, RATE, DATAVAR%(1), STAT%)
      GOSUB ERRORR
RETURN
'*************************************************************
READAOA:

      'READ AOA READS THE CURRENT TURNTABLE POSITION.

      'INPUTS: NONE
      'OUTPUTS: AOA

      ADAPT% = 1: RAWVALUE% = 2: DEVICE% = 8
      GOSUB BINOUT   'ENABLE MULTIPLEXER CHANNEL #2: AOA
      ADAPT% = 0: COUNT = 1: ANDMSK% = &H8000
      GOSUB BININ   'DATA READY?
      'NOTE: IF "DATA READY?" IS TRUE, THE VALUE RETURNED
            WILL BE &H8000
      WHILE DATAVAR%(1) = 0
           GOSUB BININ
      WEND
      ANDMSK% = &H4000
      GOSUB BININ   'READ POLARITY
      'NOTE IF "POLARITY" IS TRUE, THE VALUE RETURNED WILL BE
            16384 (&H4000)
      POLARITY = DATAVAR%(1)
      ANDMSK% = &H3FFF
      GOSUB BININ   'READ AOA (FINALLY!)
      BCD = DATAVAR%(1)
      GOSUB BCD
      AOA = DECIMAL / 10
      IF POLARITY = 0 THEN AOA = AOA * -1
RETURN
'*************************************************************
AOACONTROL:

      'AOA CONTROL POSITIONS THE TURNTABLE TO THE LOCATION
            SPECIFIED BY
      '"WANTDEG"

      'INPUTS: WANTDEG
      'OUTPUTS: NONE

 4  GOSUB READAOA
      IF (AOA > 22) OR (AOA < -20) THEN GOTO 4
      WHILE (CINT(ABS(WANTDEG - AOA) * 10) / 10) >= .3
           ABSDIFF = ABS(WANTDEG - AOA): IF ABSDIFF > 38 THEN
                 ABSDIFF = 38
           ADAPT% = 0: DEVICE% = 8
           IF WANTDEG > AOA THEN RAWVALUE%=4: GOSUB BINOUT
           IF WANTDEG < AOA THEN RAWVALUE%=8: GOSUB BINOUT
           IF (CINT(ABSDIFF*10)/10) >=.4 THEN TIMING=100 ELSE
```

```
                TIMING=1
          COUNT=(ABSDIFF*TIMING)+190:IF COUNT>4000 THEN
                COUNT=4000
          GOSUB DELAY :GOSUB RESETT
          COUNT = 500
          GOSUB DELAY   'ALLOW TURNTABLE TO STABILIZE
5      GOSUB READAOA
          IF (AOA > 23) OR (AOA < -20) THEN GOTO 5
     WEND
RETURN
'***********************************************************
READCAL:

     'THIS ROUTINE READS THE CALIBRATION PRESSURE AND
           RETURNS THE VALUE IN
     '"CALPRESS", IN PSIG.

     'INPUTS: NONE
     'OUTPUTS: CALPRESS

     ADAPT% = 1: RAWVALUE% = 4: DEVICE% = 8
     GOSUB BINOUT   'ENABLE MULTIPLEXER CHANNEL #3, CAL.
           PRESSURE
     ADAPT% = 0 : COUNT = 1: ANDMSK% = &H8000
     GOSUB BININ   'DATA READY?
     'NOTE: IF "DATA READY?" IS TRUE, THE VALUE RETURNED
           WILL BE &H8000
     'IF FALSE, THE VALUE WILL BE 0.
     WHILE DATAVAR%(1) = 0
          GOSUB BININ
     WEND
     ANDMSK% = &H7FFF: GOSUB BININ   'READ CALIBRATION
           PRESSURE
     BCD = DATAVAR%(1): GOSUB BCD   'CALL BCD-TO-DECIMAL
     CALPRESS = (DECIMAL / 100) * .014223   'CONVERT CM
          -WATER TO PSIG
RETURN
'***********************************************************
READTEMP:

     'THIS ROUTINE READS ONE OF SIX CHANNELS OF THE
           TEMPERATURE METER.

     'INPUTS: WANTEMP (INTEGER 0 - 5)
     'OUTPUTS: TEMPER

     'NOTE: THE ANALOG DEVICES DIGITAL THERMOMETER DEFAULTS
           TO CHANNEL 0 UNLESS ANOTHER CHANNEL IS REQUESTED
           VIA THE COMPUTER, OR MANUALLY
     'SELECTED ON THE FRONT PANEL.
     IF WANTEMP >0 THEN TEMPER=0: GOTO ENDREADTEMP 'INVALID
```

```
          RESPONSE
     'IF NO THERMO-COUPLE IS CONNECTED TO THAT CHANNEL, SKIP
          ROUTINE.
     ADAPT% = 1: RAWVALUE% = 8: DEVICE% = 8
     GOSUB BINOUT   'ENABLE MULTIPLEXER CHANNEL #5,
          TEMPERATURES
     RAWVALUE% = WANTEMP * 16
     ADAPT% = 0
     GOSUB BINOUT   'SELECT A TEMPERATURE CHANNEL
     COUNT = 999: GOSUB DELAY   'WAIT 1/100 SEC, LET TEMP
          STABILIZE
     COUNT = 1: ANDMSK% = &H2000
     GOSUB BININ   'DATA READY?
     'NOTE: THE "DATA READY?" SIGNAL ON THIS METER GOES
          "HIGH" FOR 5 MSEC EVERY
     'CONVERSION CYCLE BUT REMAINS "LOW" OTHER TIMES.
     WHILE DATAVAR%(1) = &H2000
         GOSUB BININ
     WEND
     ANDMSK% = &H1FFF: GOSUB BININ   'READ TEMPERATURE
     BCD = DATAVAR%(1): GOSUB BCD   'CALL BCD-TO-DECIMAL
     TEMPER = DECIMAL / 10
     ENDREADTEMP:
     RETURN
'**********************************************************
READATMOS:

     'THIS ROUTINE READS ATMOSPHERIC PRESSURE AND RETURNS
          THE VALUE IN
     '"ATMOS", IN PSIA.

     'INPUTS: NONE
     'OUTPUTS: ATMOS

     ADAPT% = 1: RAWVALUE% = 16: DEVICE% = 8
     GOSUB BINOUT   'ENABLE MULTIPLEXER CHANNEL #5, ATMOS.
          PRESS.
     ADAPT% = 0: COUNT = 1: ANDMSK% = &H2000
     GOSUB BININ   'DATA READY?
     'NOTE: IF "DATA READY?" IS TRUE, THE VALUE RETURNED
          WILL BE &H2000
     'IF FALSE, THE VALUE WILL BE 0.
     WHILE DATAVAR%(1) = 0
         GOSUB BININ
     WEND
     ANDMSK% = &H1FFF: GOSUB BININ   'READ ATMOSPHERIC
          PRESSURE
     BCD = DATAVAR%(1): GOSUB BCD
     ATMOS = DECIMAL / 100
RETURN
'**********************************************************
```

```
CALIBRATE:

     'THIS ROUTINE DETERMINES THE CALIBRATION CONSTANT VIA A
           REFERENCE
     'PRESSURE FROM PORT 1 AND A CALIBRATION PRESSURE FROM
           PORT 48.

     'INPUTS: NONE
     'OUTPUTS: CALCONST

     COLOR 14,1: CLS
     LOCATE 5,29: PRINT "SCANIVALVE CALIBRATION"
     COLOR 10: LOCATE 10,13
     PRINT "USING THE HAND-CRANK ON THE CALIBRATION
           MANOMETER NEXT"
     LOCATE 12,13
     PRINT "TO THE COMPUTER, APPLY A PRESSURE BETWEEN 40 -
           50
     LOCATE 14,13
     PRINT "CENTIMETERS OF WATER.  THE COLUMN OF WATER ON
           THE  ";
     COLOR 12
     PRINT "LEFT "
     COLOR 10
     LOCATE 16,13
     PRINT "SHOULD RISE HIGHER THAN THE COLUMN ON THE
           RIGHT.";
     COLOR 7: LOCATE 23,28: PRINT "ENTER ANY KEY WHEN
           READY:"
     SS$ = INPUT$(1): LOCATE 23,28: PRINT "
           "
     COLOR 29: LOCATE 18,23
     PRINT "CALIBRATING SCANIVALVE TRANSDUCER . . ."
     GOSUB READCAL   'OBTAIN CALIBRATION PRESS IN PSIG
     'NOTE: HOME THE SCANIVALVE TO PORT 48 TO SAVE TIME:
     ADAPT% = 0: RAWVALUE% = 2: DEVICE% = 8: COUNT = 2000
     GOSUB BINOUT: GOSUB RESETT: GOSUB DELAY
     WANTPORT = 48: GOSUB READPORT   'OBTAIN CALIBRATION
           COUNTS
     CAL = ANADATA
     WANTPORT = 1: GOSUB READPORT   'OBTAIN REFERENCE COUNTS
     REF = ANADATA
     CALCONST = (CAL - REF) / CALPRESS
RETURN
'***********************************************************
DISPLAYVAR:

     'THIS ROUTINE DISPLAYS TUNNEL TEMP, AMBIENT TEMP,
           TUNNEL TOTAL PRESSURE,
     'TUNNEL FREESTREAM STATIC PRESSURE,  TUNNEL Q, TEST
           SECTION FREESTREAM
```

```
   'VELOCITY, BAROMETRIC PRESSURE.

   'INPUTS:TEMPER, CALCONST, ATMOS, PRESSURE, TUNTEMP,
         VELOCITY
   'OUTPUTS: AMBTEMP

   COLOR 1,1: CLS: LOCATE 12,38: COLOR 31: PRINT "WAIT. .
         ."
   GOSUB READATMOS   'OBTAIN BAROMETRIC PRESSURE
   WANTEMP = 0: GOSUB READTEMP   'OBTAIN AMBIENT TEMP
   AMBTEMP = TEMPER
   GOTO 7
   CALVEL:
   '********** MINI ROUTINE: CALCULATE VELOCITY **********
   WANTEMP = 1: GOSUB READTEMP   'OBTAIN TUNNEL TEMP
   TUNTEMP = TEMPER
   WANTPORT = 2: GOSUB READPORT   'OBTAIN TOTAL PRESSURE
   GOSUB PRESSURE   'CONVERT TO PSIA
   PTOTAL = PRESSURE
   WANTPORT = 3: GOSUB READPORT   'OBTAIN FREESTREAM
         STATIC PRESS
   GOSUB PRESSURE   'CONVERT TO PSIA
   PSTAT = PRESSURE
   Q = (PTOTAL - PSTAT) * 1.075      '1.075 = TUNNEL
         CALIBRATION
   IF Q < .001 THEN VELOCITY = 0: GOTO 6
   'NOTE: FACTOR .9902 DUE TO NONZERO VELOCITY IN THE
         SETTLING CHAMBER
   VELOCITY = SQR((2 * Q * 1715 * (TUNTEMP + 459.67)) /
         (PSTAT * .9902))
6  RETURN
7  COLOR 14,1: CLS
   LOCATE 5,25: PRINT "EXPERIMENTAL CONDITIONS:"
   COLOR 12
   LOCATE 8,15: PRINT "BAROMETRIC PRESSURE:  "; ATMOS;
         "PSIA"
   LOCATE 10,15: PRINT "TUNNEL TOTAL PRESSURE:  "; PTOTAL;
         "PSIA"
   LOCATE 12,15: PRINT "TUNNEL FREE STREAM PRESSURE:  ";
         PSTAT; "PSIA"
   LOCATE 14,15: PRINT "AMBIENT TEMPERATURE:  "; AMBTEMP;
         "F"
   LOCATE 16,15: PRINT "TUNNEL TEMPERATURE:  "; TUNTEMP;
         "F"
   LOCATE 18,15: PRINT "TUNNEL Q:  "; Q; "PSI"
   LOCATE 20,15: PRINT "FREE STREAM VELOCITY:  ";
         VELOCITY; "FT/SEC"
   LOCATE 22,15: PRINT "CALIBRATION CONSTANT:  "; CALCONST
   COLOR 10: LOCATE 24,20: PRINT "ENTER 1 TO CONTINUE, 0
         TO ABORT: "
   S$ = INPUT$(1)
```

```
RETURN
'*************************************************************
PRESSURE:

     'THIS ROUTINE CONVERTS THE RAW DATA IN COUNTS (ANADATA)
           TO PRESSURE IN
     ' PSIA.

     'INPUTS: ANADATA, REF, CALCONST, ATMOS
     'OUTPUTS: PRESSURE

     PRESSURE = ((ANADATA - REF) / CALCONST) + ATMOS

RETURN
'*************************************************************
SPEED:

     'THIS ROUTINE ALLOWS THE USER TO OBSERVE TUNNEL
           VELOCITY AS HE MANUALLY
     'ADJUSTS THE PITCH OF THE PROP SYSTEM.

     'INPUTS: TEMPER, CALCONST, ATMOS, PRESSURE
     'OUTPUTS: VELOCITY, KNOTS TO SCREEN

     COLOR 31,4:CLS:LOCATE 5,27: PRINT "ADJUST TUNNEL SPEED
           NOW."
     COLOR 14: LOCATE 8,30: PRINT "TUNNEL VELOCITY: "
     GOSUB READATMOS    'OBTAIN BAROMETRIC PRESSURE
     WANTPORT = 3: GOSUB READPORT:    'OBTAIN FREESTREAM
               STATIC PRESS

     GOSUB PRESSURE   'CONVERT TO PSIA
     PSTAT = PRESSURE
8    WANTPORT = 2: GOSUB READPORT    'READ TOTAL PRESSURE
     GOSUB PRESSURE   'CONVERT TO PSIA
     PTOTAL = PRESSURE
     WANTEMP = 1: GOSUB READTEMP   'OBTAIN TUNNEL
           TEMPERATURE
     TUNTEMP = TEMPER
          Q = (PTOTAL - PSTAT) * 1.075      '1.075 = TUNNEL
           CALIBRATION
     IF Q < .001 THEN VELOCITY = 0: GOTO 9
     'NOTE: .9902 DUE TO NONZERO VELOCITY IN THE SETTLING
           CHAMBER
     VELOCITY = SQR((2 * Q * 1715 * (TEMPER + 459.67)) /
           (PSTAT * .9902))
9    KNOTS = .5924 * VELOCITY
     COLOR 10: LOCATE 11,30:PRINT USING
           "####.##";VELOCITY;:PRINT " FT/SEC"
     COLOR 13: LOCATE 13,30: PRINT USING "####.##";
           KNOTS;:PRINT " KNOTS"
     COLOR 11: LOCATE 23,10
     PRINT "ENTER ANY SINGLE KEY WHEN FINISHED ADJUSTING
           TUNNEL SPEED:"
     S$ = INKEY$: IF S$ = "" THEN GOTO 8
     LOCATE 5,27: PRINT "                         "
     LOCATE 23,10
     PRINT "
           "
     LOCATE 23,25
     PRINT "ENTER 1 TO CONTINUE, 0 TO ABORT: "
     S$ = INPUT$(1)
RETURN
'*****************************************************************
GREETINGS:

     'THIS ROUTINE GREETS THE USER AND PROVIDES INITIAL
           ORIENTATION AND
     'INSTRUCTIONS.

     'INPUTS: NONE
     'OUTPUTS: NONE

     CLS: COLOR 31: LOCATE 12,38: PRINT "WAIT. . ."
     WANTDEG=0: GOSUB AOACONTROL   'RESET AOA TO 0.
     COLOR 0,7: CLS
     LOCATE 5,20: PRINT "A E R O L A B   W I N D   T U N N
           E L "
     LOCATE 7,25: PRINT "W I N G   E X P E R I M E N T"
     LOCATE 10,10: COLOR 4
     PRINT "WELCOME TO THE AEROLAB WIND TUNNEL FOR THE WING
           EXPERIMENT USING"
     PRINT "    AN IBM PC-AT TO CONTROL ALL OF THE DATA
           AQUISITION ACTIVITIES.  YOU"
     PRINT "    MUST MANUALLY START THE WIND TUNNEL AND
           MANUALLY ADJUST THE TUNNEL"
     PRINT "    SPEED TO OBTAIN THE DESIRED WIND VELOCITY.
           BUT FIRST, THE COMPUTER"
     PRINT "    WILL CALIBRATE THE SCANIVALVE TRANSDUCER."
     COLOR 1: LOCATE 23,23
     PRINT "ENTER 1 TO CONTINUE, 0 TO ABORT"
     ENDPRO$ = INPUT$(1)
RETURN
```

```
'***************************************************************
PROMPTAOA:

     'THIS ROUTINE PROMPTS THE USER TO INPUT THE DESIRED
           ANGLE OF INCIDENCE.

     'INPUTS: NONE
     'OUTPUTS: WANTAOA()

     CLOSE
     GOSUB 10: GOTO 12
10   COLOR 14,3: CLS
11   LOCATE 5,20: PRINT"A N G L E    O F    I N C I D E N C
           E"
     LOCATE 7,30: PRINT "S E L E C T I O N": RETURN
12   OPEN "I", 1, "AOADATA.DAT"
     I = 1
     WHILE NOT EOF(1)
          INPUT #1, WANTAOA1
          WANTAOA(I) = WANTAOA1
          I = I + 1
     WEND
     CLOSE 1
     COLOR 4: LOCATE 10,5
     PRINT "DO YOU WISH TO USE THE PREVIOUS SELECTIONS OF
           ANGLE OF INCIDENCE? "
     PRINT "                                        (ENTER 1
           FOR YES, 0 FOR NO)"
     LOCATE 5,75: S$ = INPUT$(1)
     IF S$ = "0" THEN COLOR 14: CLS: GOSUB 11: GOTO 13
     CLS:I=1:LOCATE 5,25:COLOR 5:PRINT"PREVIOUS AOA
           SELECTIONS:":LOCATE 7,1
     COLOR 4
     WHILE WANTAOA(I) <> -99
          PRINT WANTAOA(I); ",  ";
          I = I + 1
     WEND
     COLOR 0: LOCATE 23,10
     PRINT "ENTER 1 TO USE THESE AOA SELECTIONS, 0 TO ENTER
           NEW SELECTIONS: "
     S$ = INPUT$(1)
     IF S$ = "1" THEN GOTO 14
     GOSUB 10
13   COLOR 4: LOCATE 10,20: PRINT "ENTER DESIRED ANGLE OF
           INCIDENCE: "
     LOCATE 11,30: PRINT "(LIMITS ARE +20 TO -18)"
     COLOR 5: LOCATE 12,30: PRINT "(ENTER -99 TO STOP)"
     COLOR 4: LOCATE 15,20: PRINT "TOTAL NUMBER OF
           SELECTIONS ENTERED: "
     LOCATE 16,25: PRINT "(MAX NUMBER OF CHOICES = 77)"
     SELECTNUM = 0: SELECT = 0
     COLOR 1
     WHILE SELECT <> -99
          LOCATE 15,60: PRINT "    ": LOCATE 15,60: PRINT
           SELECTNUM
          LOCATE 10,58: PRINT "       ":LOCATE 10,58: INPUT
                " ", SELECT
          WANTAOA(SELECTNUM +1) = SELECT
          SELECTNUM = SELECTNUM + 1
          IF SELECTNUM > 76 THEN SELECT = -99:
                WANTAOA(SELECTNUM+1)=-99
     WEND
     LOCATE 20,6
     PRINT"ENTER 1 TO STORE THESE FOR LATER USE, 0 NOT TO
           STORE THEM:"
     LOCATE 21,6
     PRINT"                       (NOTE: OLD FILE WILL BE
           DESTROYED.)"
     LOCATE 20,70: S$ = INPUT$(1)
     IF S$ = "0" THEN GOTO 14
     OPEN "O", 1, "AOADATA.DAT"
     I = 1
     WHILE WANTAOA(I) <> -99
          PRINT #1, USING "### "; WANTAOA(I)
          I = I + 1
     WEND
     PRINT #1, USING "### "; WANTAOA(I)
     CLOSE 1
14   COLOR 0: LOCATE 23,10
     PRINT "
           "
     LOCATE 23,23: PRINT "ENTER 1 TO CONTINUE, 0 TO ABORT: "
     S$ = INPUT$(1)
     IF S$ = "0" THEN GOTO 15
     CLS: LOCATE 12,38: COLOR 31: PRINT "WAIT. . ."
     WANTDEG=WANTAOA(1): GOSUB AOACONTROL   'POSITION TO
           FIRST AOA
     COLOR 30,6:CLS:LOCATE 10,30:PRINT"START TUNNEL
           NOW":LOCATE 12,23:COLOR 12
     PRINT"SET APPROXIMATE Q IN CM OF WATER":COLOR 0:LOCATE
           23,23
     PRINT"ENTER 1 TO CONTINUE, 0 TO ABORT: ": S$=INPUT$(1)
15 RETURN
'***********************************************************
RECORD:

     'THIS ROUTINE CONTROLS THE ACTUAL RUNNING OF THE
           EXPERIMENT AND DATA RECORDING.

     'INPUTS: WANTAOA(), COUNTS(), PRESSURE, Q, VELOCITY,
           TUNTEMP, AMBTEMP PSTAT, PTOTAL, ATMOS
     'OUTPUTS: WINDATA(), ENDATA
     K = 0: M = 1 : RERUN$ = "FALSE"
     VELOCITY1 = VELOCITY
     WHILE WANTAOA(M) <> -99 AND S$ <> "0"
          CLS: LOCATE 12,35: PRINT "RESETTING AOA. . ."
          WANTDEG = WANTAOA(M): GOSUB AOACONTROL
          CLS: LOCATE 12,31: PRINT "TUNNEL STABILIZING. . ."
          COUNT=5000: GOSUB DELAY   'WAIT FOR 5 SECONDS,
               STABILIZE TUNNEL
          GOSUB CALVEL    'CALL CALCULATE VELOCITY
          IF ABS(VELOCITY1 - VELOCITY) > 2 THEN
                 SCREEN 0
                 GOSUB SPEED 'CALL SET TUNNEL VEL.
          END IF
          IF S$ = "0" THEN GOTO 17
          CALL SCRNPLT1
            LOCATE 22,5: PRINT "AOA =";WANTAOA(M)
          FOR J = 4 TO 39
                 WANTPORT = J: GOSUB READPORT
                 GOSUB PRESSURE
                 Q = (PTOTAL - PSTAT) * 1.075
                 IF Q < .001 THEN CP = 0: GOTO 16
                 CP = (PRESSURE - PSTAT) / Q
16               XPOS = INT(XCOORD(J-3) * 520) + 100
                 YPOS = INT(CP * 30 + 180) + 100
                 PSET (XPOS,YPOS): CIRCLE (XPOS,YPOS),2
                 L = (K * 50) + J: WINDATA(L) = CP
                 S1$ = INKEY$
                 IF S1$ <> "" THEN
                      LOCATE 23,31: PRINT "
                          "
                      COLOR 15: LOCATE 23,13
                      PRINT "ENTER 1 TO CONTINUE, 5 TO
                               REPEAT SAME RUN, 0 TO ABORT"
                      S$ = INPUT$(1)
                      IF S$ = "5" THEN
                           J=40
                           ELSEIF S$ = "0" THEN
                                SCREEN 0
                                J = 40
                      END IF
                      LOCATE 23,13
                      PRINT "
                               "
                      LOCATE 23,31: PRINT "ENTER ANY SINGLE
                               KEY TO PAUSE"
                      COLOR 14
                 END IF
                 IF ERRORR$ <> "NO" THEN ERRORR$="NO": J =
                          40: RERUN$ = "TRUE"
            NEXT J
          IF (S$ = "5") OR (S$ = "0") THEN GOTO 17
          DATA1(1)=Q: DATA1(2)=VELOCITY:
                   DATA1(3)=WANTAOA(M): DATA1(4)=TUNTEMP
          DATA1(5)=AMBTEMP: DATA1(6)=CALCONST:
                   DATA1(7)=CALPRESS: I = 1
          FOR J = 40 TO 46
               L = (K * 50) + J
               WINDATA(L) = DATA1(I)
               I = I + 1
          NEXT J
          DATA2(1)=ATMOS: DATA2(2)=PTOTAL: DATA2(3)=PSTAT:
                   I = 1
          FOR J = 1 TO 3
               L = (K * 50) + J
              WINDATA(L) = DATA2(I): I = I + 1
          NEXT J
          IF RERUN$ = "TRUE" THEN RERUN$ = "FALSE": GOTO
                   17
          M=M+1:  K=K+1
17   WEND
     ENDATA = K * 50
     SCREEN 0: COLOR 31,5: CLS: LOCATE 12,38: PRINT "WAIT. .
          ."
     WANTDEG = 0: GOSUB AOACONTROL   'RESET AOA TO 0.
     COLOR 1,3: CLS: LOCATE 12,15
     PRINT "FINISHED RECORDING DATA.  ";
     COLOR 20: PRINT "SHUT DOWN TUNNEL NOW."
     LOCATE 23,23: COLOR 0 : PRINT "ENTER 1 TO CONTINUE, 0
           TO ABORT:"
     S$ = INPUT$(1)
RETURN
'************************************************************
FILEDATA:

     'THIS ROUTINE FILES THE DATA LOCATED IN THE ARRAY
           WINDATA() AND STORES
     'IT ON A SEQUENTIAL FILE UNDER THE NAME  "WINDATA.TXT"

     'INPUTS: WINDATA(), ENDATA
     'OUTPUTS: NONE

     COLOR 14,6: CLS: LOCATE 5,30
     PRINT "F I L E    D A T A"
     COLOR 12: LOCATE 8,15
     PRINT "ENTER 1 TO SAVE THE RECORDED DATA, 0 NOT TO SAVE
           IT:"
     LOCATE 9,15
     PRINT "                    (NOTE: OLD DATA WILL BE
           DELETED.)"
     LOCATE 8,70: SAVEDATA$ = INPUT$(1)
     IF SAVEDATA$ = "0" THEN GOTO 18
     OPEN "O", 1, "WINDATA.TXT"
     FOR I = 1 TO ENDATA
          PRINT #1, USING "####.#### "; WINDATA(I)
     NEXT I
     CLOSE #1
18 RETURN
'************************************************************
DISPLAYDATA:

     'THIS ROUTINE DISPLAYS THE DATA AS RECORDED IN THE
           EXPERIMENT.

     'INPUTS: WINDATA(), ENDATA
     'OUTPUTS: INFORMATION TO THE SCREEN

     COLOR 14,2: CLS: LOCATE 1,25: PRINT "R E C O R D E D
           D A T A "
     COLOR 4: LOCATE 4,2
     LOCATE 4,10: PRINT "WING CP"
     FOR I = 1 TO 18
          IF I < 10 THEN J1$ = RIGHT$(STR$(I),1) ELSE J1$ =
                     RIGHT$(STR$(I),2)
          J2$ = RIGHT$(STR$(I + 18),2): ROW = I + 5
          LOCATE ROW,2:PRINT"P(";J1$;"):":LOCATE
                     ROW,20:PRINT"P(";J2$;"):"
          NEXT I
     COLOR 4:DATA3$(1)="BAROMETRIC PRESSURE(PSIA): ":
                DATA3$(2)="TOTAL PRESSURE(PSIA): "
     DATA3$(3)="FREESTREAM STATIC PRESS(PSIA): ":
                DATA3$(4)="TUNNEL Q(PSIA): "
     DATA3$(5)="TUNNEL VELOCITY(FT/SEC): ": DATA3$(6)="ANGLE
                OF INCIDENCE(DEG): "
     DATA3$(7)="TUNNEL TEMPERATURE(F): ": DATA3$(8)="AMBIENT
                TEMPERATURE(F): "
     DATA3$(9)="CALIBRATION CONSTANT: ":
                DATA3$(10)="CALIBRATION PRESSURE(PSIA): "
     FOR I = 1 TO 10
          ROW=I+3: LOCATE ROW,40: PRINT DATA3$(I)
     NEXT I
     COLOR 0: LOCATE 20,50: PRINT"ENTER 1 TO CONTINUE"
     LOCATE 21,50         : PRINT"      5 TO STOP"
     LOCATE 22,50         : PRINT"      0 TO ABORT"
     K = ENDATA / 50
     FOR I = 1 TO K
          IF I=K THEN COLOR 0:LOCATE 22,50:PRINT"
                     ":LOCATE 21,50: PRINT"      0 TO
                     ABORT":COLOR 1
          COLOR 11: LOCATE 3,34: PRINT"RUN:"; I: COLOR 1
          FOR J = 4 TO 21
               L = (((I-1) * 50) + J): ROW = J + 2
               LOCATE ROW,8:PRINT WINDATA(L): LOCATE ROW,26:
```

```
                          PRINT WINDATA(L+18)
          NEXT J
          FOR J = 1 TO 3
               L = ((I - 1) * 50) + J: ROW = J + 3
               LOCATE ROW,70: PRINT USING "###.###";
                          WINDATA(L)
          NEXT J
          FOR J = 40 TO 46
               L = ((I - 1) * 50) + J: ROW = J - 33
               LOCATE ROW,70: PRINT USING "####.###";
                          WINDATA(L)
          NEXT J
          LOCATE 22,70: S$ = INPUT$(1)
          IF S$ <> "1" THEN I = K
     NEXT I
RETURN
'***********************************************************
STARTSPLINE:

     'TO USE A CLAMPED CUBIC SPLINE, DERIVATIVES AT THE END
                POINTS MUST
     'BE EVALUATED:
     'ESTABLISH A MATRIX: G, THAT WILL BE USED TO SOLVE FOR
                THE
     'EQUATION OF A QUADRATIC AT EACH END POINT.
     'C1 + C2X + C3X^2 = Y(X), WHERE X, Y ARE KNOWN, C1,2,3
                ARE UNKNOWN.

     'INPUTS: X(), Y(), N
     'OUTPUTS: FP1, FP2

     START = 1: FINISH = 3: GOSUB 19         'FORM FIRST
                MATRIX
     GOSUB 20                                'SOLVE FOR
                CONSTANTS
     FP1 = 2 * C3 * X(1) + C2                 'FIRST
                DERIVATIVE

     START = N-2: FINISH = N: GOSUB 19       'FORM SECOND
                MATRIX
     GOSUB 20                                'SOLVE FOR
                CONSTANTS
     FP2 = 2 * C3 * X(N) + C2                 'SECOND
                DERIVATIVE
     RETURN   'NOW EXECUTE SPLINE!

19 'FORM MATRIX TO SOLVE FOR COEFFICIENTS OF A QUADRATIC:

     L = 1
     FOR I = START TO FINISH
          G(L,1) = 1: G(L,2) = X(I): G(L,3) = X(I)^2: G(L,4)
```

```
                     = Y(I)
     L = L + 1
     NEXT I
     RETURN

20 'CRAMERS RULE TO SOLVE FOR THE QUADRATIC CONSTANTS:

     D1 = G(1,1) * (G(2,2) * G(3,3) - G(2,3) * G(3,2))
     D2 = G(1,2) * (G(2,3) * G(3,1) - G(2,1) * G(3,3))
     D3 = G(1,3) * (G(2,1) * G(3,2) - G(2,2) * G(3,1))
     DET = D1 + D2 + D3
     D1 = G(1,1) * (G(2,4) * G(3,3) - G(2,3) * G(3,4))
     D2 = G(1,4) * (G(2,3) * G(3,1) - G(2,1) * G(3,3))
     D3 = G(1,3) * (G(2,1) * G(3,4) - G(2,4) * G(3,1))
     C2 = (D1 + D2 + D3) / DET
     D1 = G(1,1) * (G(2,2) * G(3,4) - G(2,4) * G(3,2))
     D2 = G(1,2) * (G(2,4) * G(3,1) - G(2,1) * G(3,4))
     D3 = G(1,4) * (G(2,1) * G(3,2) - G(2,2) * G(3,1))
     C3 = (D1 + D2 + D3) / DET
RETURN
'************************************************************
EXECUTESPLINE:

     'THIS ROUTINE TAKES THE DERIVATIVES FROM "START CUBIC
                SPLINE", X(), AND
     'Y(), AND COMPUTES A SPLINE FOR THAT INTERVAL

     'INPUTS: FP1, FP2, X(), Y()
     'OUTPUTS: B(), C(), D() (CONSTANTS FOR FNSPLINE() )

     ALPHA1 = 3 * (Y(2) - Y(1)) / (X(2) - X(1)) - 3 * FP1
     ALPHAN = 3 * FP2 - 3 * (Y(N) - Y(N-1)) / (X(N) - X(N
               -1))
     DEF FNALPHA(I)=3*((Y(I+1)*(X(I)-X(I-1)))-(Y(I)*(X(I+1)
               -X(I-1)))+(Y(I-1)*(X(I+1)-X(I))))/((X(I+1)
               -X(I))*(X(I)-X(I-1)))
     L(1) = 2 * (X(2) - X(1))
     U(1) = .5
     Z(1) = ALPHA1 / (2 * (X(2) - X(1)))
     B(1) = FP1
     FOR I=2 TO N-1
          L(I) = 2 * (X(I+1) - X(I-1)) - (X(I) - X(I-1)) *
                     U(I-1)
          U(I) = (X(I+1) - X(I)) / L(I)
          Z(I) = (FNALPHA(I) - (X(I) - X(I-1)) * Z(I-1)) /
                     L(I)
     NEXT I
     L(N) = (X(N) - X(N-1)) * (2 - U(N-1))
     Z(N) = (ALPHAN - (X(N) - X(N-1)) * Z(N-1)) / L(N)
     C(N) = Z(N)
     FOR L=N-1 TO 1 STEP -1
```

```
          C(L) = Z(L) - U(L) * C(L+1)
          B(L)=((Y(L+1)-Y(L))/(X(L+1)-X(L)))-((X(L+1)
                    -X(L))*(C(L+1)+2*C(L))/3)
          D(L) = (C(L+1) - C(L)) / (3 * (X(L+1) - X(L)))
     NEXT L
RETURN
'************************************************************
INTEGRATE:

     'THIS ROUTINE WILL INTEGRATE THE DATA GIVEN TO IT IN
               THE FORM OF X,Y
     'COORDINATES BY CREATING A SERIES OF THIRD DEGREE
               POLYNOMIALS (VIA THE
     'SPLINE ROUTINE) AND THEN INTEGRATING EACH POLYNOMIAL
               OVER  EACH INTERVAL
     'AND LASTLY SUMMING EACH SUBTOTAL FOR A FINAL ANSWER.

     'INPUTS: X(), Y(), N; USES THE FUNCTION: FNSPLINE()
     'OUTPUTS: INTEGRAL

     GOSUB STARTSPLINE
     GOSUB EXECUTESPLINE

     'USE SIMPSON'S RULE (EXACT FOR THIRD DEGREE
               POLYNOMIALS)

     INTEGRAL = 0: SUM = 0
     FOR I = 1 TO N-1
          JJ = X(I): HH = (X(I + 1) - X(I)) / 2
          SUM = FNSPLINE(I,JJ)
          JJ = X(I) + HH: SUM = SUM + FNSPLINE(I,JJ) * 4
          JJ = X(I + 1): SUM = SUM + FNSPLINE(I,JJ)
          INTEGRAL = INTEGRAL + ((HH / 3) * SUM)
     NEXT I
RETURN
'************************************************************
CALCY:

     'THIS ROUTINE CALCULATES THE FORCE COEFFICIENT  IN THE
               "Y" DIRECTION

     'INPUTS: CPUPPER(), CPLOWER(), XWINGU(), XWINGL()
     'OUTPUTS: CY, [ X(), Y(), N, FOR "INTEGRATE" ]

     FOR I = 1 TO 19
          X(I) = XWINGU(I)
          Y(I) = CPUPPER(I)
     NEXT I
     N = 19
     GOSUB INTEGRATE
     CY = INTEGRAL
     FOR I = 1 TO 19
          X(I) = XWINGL(I)
          Y(I) = CPLOWER(I)
     NEXT I
     N = 19
     GOSUB INTEGRATE
     CY = INTEGRAL - CY    'CY = INTEGRAL(CPLOWER - CPUPPER)
                dX/C
     CY(J+1) = CY
RETURN
'***********************************************************
REDUCEPLOT1:

     'THIS ROUTINE CONTROLS THE DATA REDUCTION AND PLOTTING
                DECISIONS.

     'INPUTS: WINDATA(), XWINGU(), XWINGL(), YWINGU(),
                YWINGL(), ENDATA
     'OUTPUTS: CPUPPER(), CPLOWER()

     'REDUCE DATA AND PLOT FOR EACH AOA SEPARATELY:

     K = ENDATA / 50
       COLOR 4,7: CLS: LOCATE 8,20
       PRINT "ENTER 1 TO PLOT CP, 0 NOT TO PLOT CP: "
       PLOTFLAG$ = INPUT$(1): IF PLOTFLAG$="0" THEN GOTO 21
       COLOR 1: LOCATE 10,20: LIMIT = K: IF LIMIT>3 THEN
                LIMIT=3
       PRINT "YOU MAY PRINT UP TO"; LIMIT; "PLOT(S) OF CP.":
                LOCATE 12,20
       COLOR 2: PRINT "ENTER THE NUMBER OF PLOTS REQUESTED:"
       S$=INPUT$(1): PLOTNUM= VAL(S$):
       FOR I = 1 TO PLOTNUM
            LOCATE 14,13: COLOR 5
            PRINT "ENTER THE AOA CORRESPONDING TO THE CP
                     PLOT REQUESTED:"
            LOCATE 14,70: INPUT " ", PLOTAOA(I)
            LOCATE 14,68: PRINT "        "
       NEXT I
       PLOTCOUNT$(1) = "NO": PLOTCOUNT$(2) = "NO":
                PLOTCOUNT$(3) = "NO"
       PLOTCOUNT = 1
       COLOR 0: LOCATE 23,23: PRINT "ENTER 1 TO CONTINUE, 0
                TO ABORT:"
       S$ = INPUT$(1): IF S$ = "0" THEN RETURN

21     COLOR 14,1: CLS:LOCATE 10,33: PRINT "REDUCING DATA. .
                ."
     FOR J = 0 TO (K - 1)
          'SEPARATE OUT UPPER SURFACE DATA FROM LOWER
                     SURFACE DATA:
```

```
          CPUPPER(1) = WINDATA((J * 50) + 4)
          CPLOWER(1) = CPUPPER(1)
          CPUPPER(19) = WINDATA((J * 50) + 39)
          CPLOWER(19) = CPUPPER(19)
          FOR I = 2 TO 18
               CPUPPER(I) = WINDATA((J * 50) + (2 * I) + 1)
               CPLOWER(I) = WINDATA((J * 50) + (2 * I) + 2)
          NEXT I

          'IF PLOTTING IS DESIRED, SET UP DATA FILES

          IF PLOTFLAG$ = "0" THEN GOTO 24
          FOR I = 1 TO PLOTNUM
               IF WANTAOA(J+1) = PLOTAOA(I) THEN
                     PLOTCOUNT$(PLOTCOUNT) = "YES"
                     PLOTCOUNT = PLOTCOUNT + 1
               END IF
          NEXT I
          IF PLOTCOUNT$(1) <> "YES" THEN GOTO 22
          FOR I = 1 TO 19
               PLOTDU1(I) = CPUPPER(I)
               PLOTDL1(I) = CPLOWER(I)
          NEXT I
22    IF PLOTCOUNT$(2) <> "YES" THEN GOTO 23
            FOR I = 1 TO 19
                      PLOTDU2(I) = CPUPPER(I)
                      PLOTDL2(I) = CPLOWER(I)
            NEXT I
23    IF PLOTCOUNT$(3) <> "YES" THEN GOTO 24
            FOR I = 1 TO 19
                      PLOTDU3(I) = CPUPPER(I)
                      PLOTDL3(I) = CPLOWER(I)
            NEXT I

24    COLOR 12,1: LOCATE 12,38:PRINT "RUN: "; J+1
          GOSUB CALCY
          GOSUB CALCX
          ALPHA = (WINDATA((J*50) +42) * 3.1415926#) / 180
          CL = (CY * COS(ALPHA)) - (CX * SIN(ALPHA))
          CD = (CX * COS(ALPHA)) + (CY * SIN(ALPHA))
          GOSUB CALCMLE
          GOSUB CALBLOCK
     NEXT J
RETURN
'************************************************************
CALCX:

     'THIS ROUTINE CALCULATES THE FORCE COEFFICIENT IN THE
                "X" DIRECTION.

     'INPUTS: CPUPPER(), CPLOWER(), YWINGU(), YWINGL(),
                INTEGRAL
     'OUTPUTS: CX, [ X(), Y(), N, FOR "INTEGRATE"]
     '
     FOR I = 1 TO 13                                  'UPPER FRONT
          X(I) = YWINGU(I): Y(I) = CPUPPER(I)
     NEXT I
     N = 13: GOSUB INTEGRATE
     CX = INTEGRAL
     FOR I = 1 TO 13                                  'LOWER FRONT
          X(I) = YWINGL(I):  Y(I) = CPLOWER(I)
     NEXT I
     N = 13:  GOSUB INTEGRATE
     CX = CX + INTEGRAL
     FOR I = 1 TO 7                                    'UPPER REAR
          X(I) = YWINGU(I + 12):  Y(I) = CPUPPER(I + 12)
     NEXT I
     N = 7:  GOSUB INTEGRATE
     CX = CX + INTEGRAL
     FOR I = 1 TO 7                                    'LOWER REAR
          X(I) = YWINGL(I + 12):  Y(I) = CPLOWER(I + 12)
     NEXT I
     N = 7: GOSUB INTEGRATE
     CX = CX + INTEGRAL
     CX(J+1) = CX
RETURN
'************************************************************
CALCMLE:

     'THIS ROUTINE CALCULATES THE COEFFICIENT OF MOMENT
                ABOUT THE LEADING EDGE.

     'INPUTS: CPUPPER(), CPLOWER(), XWINGU(), XWINGL(),
                YWINGU(), YWINGL()
     'OUTPUTS: CMLE, [ X(), Y(), N, FOR "INTEGRATE" ]

     FOR I = 1 TO 19
          X(I) = XWINGU(I):  Y(I) = CPUPPER(I) * XWINGU(I)
     NEXT I
     N = 19:  GOSUB INTEGRATE
     CMLE = INTEGRAL
     FOR I = 1 TO 19
          X(I) = XWINGL(I):  Y(I) = CPLOWER(I) * XWINGL(I)
     NEXT I
     N = 19:  GOSUB INTEGRATE
     CMLE = CMLE - INTEGRAL
     FOR I = 1 TO 13
          X(I) = YWINGU(I):  Y(I) = CPUPPER(I) * YWINGU(I)
     NEXT I
     N = 13:  GOSUB INTEGRATE
     CMLE = CMLE + INTEGRAL
     FOR I = 1 TO 13
```

```
          X(I) = YWINGL(I):  Y(I) = CPLOWER(I) * YWINGL(I)
     NEXT I
     N = 13:  GOSUB INTEGRATE
     CMLE = CMLE - INTEGRAL
     FOR I = 1 TO 7
          X(I) = YWINGU(I+12):  Y(I) = CPUPPER(I+12) *
                     YWINGU(I+12)
     NEXT I
     N = 7:  GOSUB INTEGRATE
     CMLE = CMLE + INTEGRAL
     FOR I = 1 TO 7
       X(I) = YWINGL(I+12):  Y(I) = CPLOWER(I+12) *
                     YWINGL(I+12)
     NEXT I
     N = 7:  GOSUB INTEGRATE
     CMLE = CMLE - INTEGRAL
     CMLE(J+1) = CMLE
RETURN
'************************************************************
CALBLOCK:

     'THIS ROUTINE CALCULATES SOLID BLOCKING, WAKE BLOCKING,
                FOR 2-D MODELS
     'AND CORRECTS THE AERODYNAMIC PARAMETERS: CL, CD, Q,
                CMC/4.

     'INPUTS: CL, CD, WINDATA()
     'OUTPUTS: CL1, CD1, CL1(), CD1(), ALPHA1, ALPHA1(),
                CMC41, CMC41()

     EPSILON = .004981 + (CD / (2 * 3.75))
     'NOTE: .004981 = (K1 * MODEL VOLUME) / (C^1.5)  [LAB
                MANUAL PG 8-2]
     SIGMA = ((3.14159)^2 / 48) * (1 / 3.75)^2      '[LAB
                MANUAL PG 8-4]
     CL1 = CL * (1 - SIGMA - (2 * EPSILON))
                'CL1=CORRECTED VALUE OF CL
     CL1(J + 1) = CL1
     CD1 = CD * ((1 - (3 * .004981) - (2 * (CD / (2 *
                3.75)))))
     CD1(J + 1) = CD1
     'CD1=CORRECTED VALUE OF CD
     Q1 = WINDATA((J+50) + 40) * (1 + (2 * EPSILON))'
                Q1=CORRECTED VALUE OF Q
          CMC4 = CMLE + (CL / 4)     'CMC4=MOMENT
                COEFFICIENT ABOUT QUARTER CHORD
     ALPHA = WINDATA((J*50) + 42)
     ALPHA1 = ALPHA +
                (((57.3*SIGMA)/(2*3.14159))*(CL+(4*CMC4)))
     ALPHA1(J+1) = ALPHA1
                'ALPHA1=CORRECTED VALUE OF AOA
     CMC41 = (CMC4 * (1 - (2 * EPSILON))) + ((SIGMA * CL1) /
                4)
     CMC41(J+1) = CMC41                          'CMC41=CORRECTED
                VALUE OF CMC4
RETURN
'************************************************************
REDUCEPLOT2:

     'THIS ROUTINE REDUCES THE THE DATA THAT REQUIRES INPUT
                FROM EACH SEPARATE
     'AOA RUN; HANDLES PLOTTING AND HARDCOPY DECISIONS FOR
                THIS DATA ALSO.

     'INPUTS: CMLE(), CY(), K, ALPHA1(), CD1(), CL1(),
                CMC41()
     'OUTPUTS: XAC, CMAC();  DATA TO THE PRINTER

     IF K = 1 THEN GOTO 25   'NO CALCULATIONS IF ONLY ONE
                RUN
     'COMPUTE THE AERODYNAMIC CENTER (XAC) BY FINDING THE
                SLOPE OF
     'CMLE VS CY, BY USING LEAST SQUARES FIT [LAB MANUAL PG
                B2-10]:
     '
     SUM1 = 0: SUM2 = 0: SUM3 = 0: SUM4 = 0
     FOR I = 1 TO K
          SUM1 = SUM1 + (CY(I) * CMLE(I))
          SUM2 = SUM2 + CY(I)
          SUM3 = SUM3 + CMLE(I)
      SUM4 = SUM4 + (CY(I))^2
     NEXT I
     DENOMINATOR = ((K * SUM4) - (SUM2^2))
     IF DENOMINATOR = 0 THEN
          XAC = 0
          ELSE XAC = -(((K * SUM1) - (SUM2 * SUM3)) /
                     DENOMINATOR)
     END IF
     FOR I = 1 TO K
          CMAC(I) = CMLE(I) + (CL1(I) * XAC)
     NEXT I
25   COLOR 14,4: CLS: LOCATE 12,2
     PRINT "ENTER 1 TO OUTPUT A HARDCOPY OF CP, CY, CX, XAC,
                CMLE, CL, CD, CMC/4, CMAC,"
     LOCATE 14,2
     PRINT "
                ENTER 0 NOT TO OUTPUT:"
     HARDCOPY$ = INPUT$(1)
     IF HARDCOPY$ = "0" THEN GOTO 26   'SKIP HARDCOPY OUTPUT
     FOR I = 1 TO K
          LPRINT "AOA: ";ALPHA1(I)
          LPRINT "PORT:", "   CP:"
          FOR II = 1 TO 36
               LPRINT II, WINDATA((I-1) * 50 +II + 3)
          NEXT II
          LPRINT: LPRINT
          LPRINT "CY=", CY(I): LPRINT "CX=", CX(I): LPRINT
                    "XAC=", XAC
          LPRINT "CMLE=", CMLE(I): LPRINT "CL=", CL1(I):
                     LPRINT "CD=", CD1(I)
          LPRINT "CMC/4=", CMC41(I): LPRINT "CMAC=", CMAC(I)
          LPRINT CHR$(12)
     NEXT I
26   COLOR 7,0: CLS
RETURN
'***********************************************************
PLOTCP:

     'THIS ROUTINE TAKES THE ARRAY VARIABLES SET UP WITH THE
                X-Y COORDINATES
     'AND CALLS THE GRAPH PROGRAM VIA THE SHELL STATEMENT

     'INPUTS:
     'OUTPUTS:

     CLOSE
     IF PLOTFLAG$ = "0" THEN GOTO 27
     OPEN "O", #1, "\GRAPH\PLOTDATU.DAT"
            FOR I = 1 TO 19
                 PRINT #1, USING "#.####^^^^   "; XWINGU(I);
                          PLOTDU1(I)
            NEXT I
            CLOSE #1
            OPEN "O", #1, "\GRAPH\PLOTDATL.DAT"
            FOR I = 1 TO 19
                 PRINT #1, USING "#.####^^^^   "; XWINGL(I);
                          PLOTDL1(I)
            NEXT I
            CLOSE #1
            SHELL "CPPLOT.BAT"

          IF PLOTNUM < 2 THEN GOTO 27
          OPEN "O", #1, "\GRAPH\PLOTDATU.DAT"
          FOR I = 1 TO 19
               PRINT #1, USING "#.####^^^^   "; XWINGU(I);
                          PLOTDU2(I)
          NEXT I
          CLOSE #1
          OPEN "O", #1, "\GRAPH\PLOTDATL.DAT"
          FOR I = 1 TO 19
               PRINT #1, USING "#.####^^^^   "; XWINGL(I);
                          PLOTDL2(I)
          NEXT I
          CLOSE #1
          SHELL "CPPLOT.BAT"

          IF PLOTNUM < 3 THEN GOTO 27
          OPEN "O", #1, "\GRAPH\PLOTDATU.DAT"
          FOR I = 1 TO 19
               PRINT #1, USING "#.####^^^^   "; XWINGU(I);
                     PLOTDU3(I)
          NEXT I
          CLOSE #1
          OPEN "O", #1, "\GRAPH\PLOTDATL.DAT"
          FOR I = 1 TO 19
               PRINT #1, USING "#.####^^^^   "; XWINGL(I);
                     PLOTDL3(I)
          NEXT I
          CLOSE #1
          SHELL "CPPLOT.BAT"
27   RETURN

SUB SCRNPLT1  STATIC

     GOSUB DIMARR
     GOSUB HIRES
     GOSUB VERTICALGRID
     GOSUB HORIZONTALGRID
     GOSUB NUMBERY
     GOTO ENDSCRNPLT1
'************************************************************
DIMARR:

     DIM GRID(10)
     DIM XGRID(20), YGRID(10), LCHAR$(65), E(3)
RETURN
'************************************************************
HIRES:

     SCREEN 9 : COLOR 14,5: CLS
     WINDOW (0,0) - (639,349)
     LINE (100, 100) - (620, 340),, B
RETURN
'************************************************************
VERTICALGRID:

          COLOR 10
     FOR I = 1 TO 7
       XGRID(I) = I * 65 + 100
       LINE (XGRID(I), 340) - (XGRID(I), 98),,,&HEEEE
     NEXT I
          LOCATE 19,13: PRINT "0.0": LOCATE 19,28: PRINT
                "0.25"
          LOCATE 19,45: PRINT "0.5": LOCATE 19,61: PRINT
```

```
                "0.75"
          LOCATE 19,77: PRINT "1.0"
          COLOR 12: LOCATE 21,38: PRINT "WING CHORD (X/C)"
          COLOR 15: LOCATE 23,31: PRINT "ENTER ANY SINGLE
                KEY TO PAUSE"
RETURN
'************************************************************
HORIZONTALGRID:

     GRID(1)=-6:GRID(2)=-4:GRID(3)=
               -2:GRID(4)=0:GRID(5)=2:GRIDNUM=5
     SCALEY = (GRID(GRIDNUM) - GRID(1)) / 240
     YCONSTANT = GRID(1) / SCALEY
     COLOR 10
     FOR I = 1 TO GRIDNUM
          YGRID(I) = INT(GRID(I) / SCALEY - YCONSTANT) + 100
          IF GRID(I) = 0 THEN LINE (98,YGRID(I))
                   -(620,YGRID(I)): GOTO 28
          IF (I=1) OR (I=GRIDNUM) THEN GOTO 28
          LINE (98, YGRID(I)) - (620, YGRID(I)),,,&HF0F0
28   NEXT I
          COLOR 12: LOCATE 10,5: PRINT "CP"
RETURN
'************************************************************
SIDESTRING:
     'ROUTINE TO PRINT A STRING AT A SPECIFIC SCREEN
                POSITION, SIDEWAYS

     'VARIABLES: WORD$  -       STRING TO BE PRINTED
     '           LONG   -       LENGTH OF WORD$
     '           C$     -       INDIVIDUAL CHARACTERS OF
                WORD$
     '           ASCII  -       ASCII DECIMAL VALUE OF C$
     '           X      -       X POSITION OF FIRST
                CHARACTER
     '           Y      -       Y POSITION OF FIRST
                CHARACTER
     'INPUTS: WORD$, X, Y
     'OUTPUTS: NONE

     'DEFINE THE SIDEWAYS CHARACTER SET:
     COLOR 14
     LCHAR$(1) = "BU7"
                'BLANK
     LCHAR$(9) = "BL2 L5 H2 BR9 G1 BU3 BR1 BU2"
                '(
     LCHAR$(10)= "BL9 BU2 E2 R5 F2 BU4"
                ')
     LCHAR$(14)= "BU1 BL5 BU1 U3 BR5 BU2"
                '-
     LCHAR$(15)= "U0 BU2"
```

```
          '.
LCHAR$(17)= "BL1 L7 H1 U2 E1 R7 F1 ND2 BU3"
          '0
LCHAR$(18)= "BU2 NL9 BU2"
          '1
LCHAR$(19)= "BL8 L1 U4 R2 F4 R3 U4 BU2"
          '2
LCHAR$(20)= "BL8 H1 U2 E1 R2 F1 ND2 E1 R3 F1 D2 NG1
          BU5"   '3
LCHAR$(21)= "BL4 L5 BU4 R5 ND3 R4 BU2"
          '4
LCHAR$(22)= "BL5 NU3 L4 U4 BR4 R5 F1 D2 G1 BU4 BR1 BU2"
          '5
LCHAR$(23)= "BL1 L7 H1 U2 BR4 ND3 E1 R3 F1 ND2 BU3"
          '6
LCHAR$(24)= "BL9 U4 R5 F4 BU6"
          '7
LCHAR$(25)= "BL1 L3 BL1 L3 H1 U2 E1 R2 F1 ND2 E1 R3 F1
          ND2 BU3" '8
LCHAR$(26)= "BL6 L2 H1 U2 E1 R3 ND3 R5 BU2"
          '9
LCHAR$(34)= "L7 H2 E2 R2 ND3 R5 BU2"
          'A
LCHAR$(35)= "L9 U3 E1 R3 ND3 R4 F1 ND2 BU3"
          'B
LCHAR$(36)= "BL1 L7 H1 U2 E1 BR6 R1 F1 ND2 BU3"
          'C
LCHAR$(37)= "L9 U3 E1 R7 F1 ND2 BU3"
          'D
LCHAR$(38)= "L9 NU4 BR4 NU3 BR5 U4 BU2"
          'E
LCHAR$(39)= "L9 NU4 BR4 NU3 BR5 BU6"
          'F
LCHAR$(40)= "BL1 L7 H1 U2 E1 BR3 ND4 R4 F1 ND2 BU3"
          'G
LCHAR$(41)= "L9 BU4 R4 ND3 R5 BU2"
          'H
LCHAR$(42)= "BL9 U2 NU2 R9 ND2 U2 BU2"
          'I
LCHAR$(43)= "BL1 L2 BU4 NL6 R2 F1 ND2 BU3"
          'J
LCHAR$(44)= "L9 BU4 F4 E4 R1 BU2"
          'K
LCHAR$(45)= "NL9 U4 BU2"
          'L
LCHAR$(46)= "L9 E2 NR1 H2 R9 BU2"
          'M
LCHAR$(47)= "L9 U1 R3 U1 R3 U1 R3 U1 NL9 BU2"
          'N
LCHAR$(48)= "BL1 L7 H1 U2 E1 R7 F1 ND2 BU3"
          'O
```

```
     LCHAR$(49)= "L9 U3 E1 R2 F1 ND2 BU1 BR5 BU2"
               'P
     LCHAR$(50)= "BL1 L7 H1 U2 E1 R8 NG2 ND3 BU2"
               'Q
     LCHAR$(51)= "L9 U3 E1 R2 F1 D2 E3 R2 BU2"
               'R
     LCHAR$(52)= "BL8 H1 U2 E1 BD4 R2 E1 U2 E1 R3 F1 D2 G1
               BU4 BR1 BU2" 'S
     LCHAR$(53)= "BL9 U2 NR9 U2 BR9 BU2"
               'T
     LCHAR$(54)= "BL1 NL8 BU4 NL8 BR1 BD1 ND2 BU3"
               'U
     LCHAR$(55)= "BU3 G1 L1 G1 L6 BU4 R6 F1 R1 F1 BU4"
               'V
     LCHAR$(56)= "NL9 H2 NL1 E2 NL9 BU2"
               'W
     LCHAR$(57)= "L3 H2 G2 L2 BU4 R2 F1 BR2 E1 R3 BU2"
               'X
     LCHAR$(58)= "BU2 L5 G2 L2 BU4 R2 F1 BR6 BU3"
               'Y
     LCHAR$(59)= "BL9 L1 U4 R3 F4 R3 U4 NL1 BU2"
               'Z

     Y = -Y + 349
     MOVE$ = "BM" + STR$(X) + "," + STR$(Y)
     DRAW "X" + VARPTR$(MOVE$)
     LONG = LEN (WORD$)
     IF NEGFLAG$ = "Y" THEN START = 1 ELSE START = 2
     FOR J = START TO LONG
          C$ = MID$(WORD$, J, 1)
          ASCII = ASC(C$) - 31
          DRAW "X" + VARPTR$(LCHAR$(ASCII))
     NEXT J
RETURN
'************************************************************
NUMBERY:

     'THIS ROUTINE WILL TAKE THE VALUES OF THE Y-AXIS
               GRIDLINES AND
     'NUMBER THE Y-AXIS WITH SIDEWAYS CHARACTERS.

     'INPUTS: GRIDNUM, YGRID()
     'OUTPUTS: WORD$

      = 96
     FOR I = 1 TO GRIDNUM
          X = 96
          Y = YGRID(I) - 5: IF GRID(I)>=10 THEN Y = Y - 3
          WORD$ = STR$(GRID(I))
          IF GRID(I)<0 THEN NEGFLAG$ = "Y" ELSE NEGFLAG$ =
                    "N"
          IF LEN(WORD$) = 2 THEN WORD$ = STR$(GRID(I)) + "."
                     + "0"
          GOSUB SIDESTRING
     NEXT I
RETURN
ENDSCRNPLT1:
END SUB
```

LIST OF REFERENCES

1. Laboratory Manual for Low Speed Wind Tunnel Testing, Department of Aeronautics, Naval Postgraduate School, October 1983.

2. Burden, R. L., Faires, J. D., and Reynolds, A. C., Numerical Analysis, Prindle, Weber & Schmidt, 1978.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Satyanarayana Bodapati 10
   M. S. 260-1
   NASA Ames Research Center
   Moffet Field, CA 94035

4. M. F. Platzer, Chairman 1
   Department of Aeronautics
   Code 67
   Naval Postgraduate School
   Monterey, Ca 93943-5100

5. John A. DiMiceli 5
   1265 Leahy Road
   Monterey, CA 93940

6. Ted Dunton 1
   Department of Aeronautics
   Code 67
   Naval Postgraduate School
   Monterey, Ca 93943-5100

118

220165

Thesis
D57646 **Dimiceli**
c.1 Computer-controlled data acquisition and analysis.