NAME OF AUTHOR......Roger W. Toogood..............

TITLE OF THESIS......Data Acquisition System
......for Aerodynamic..............
......Experiments..............

UNIVERSITY......Alberta (Edmonton)..........

DEGREE FOR WHICH THESIS WAS PRESENTED......M.Sc..........

YEAR THIS DEGREE GRANTED......1974..................

Permission is hereby granted to THE NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed)......Roger Toogood......

PERMANENT ADDRESS:

......8722-119 ST......
......Edmonton......
....

DATED......June 14......1974

NL-91 (10-68)

THE UNIVERSITY OF ALBERTA

DATA ACQUISITION SYSTEM FOR AERODYNAMIC EXPERIMENTS

by

© ROGER W. TOOGOOD

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL ENGINEERING

EDMONTON, ALBERTA

Fall, 1974

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled DATA ACQUISITION SYSTEM FOR AERODYNAMIC EXPERIMENTS submitted by ROGER W. TOOGOOD in partial fulfilment of the requirements for the degree of Master of Science.

David Marsden
Supervisor

. . . .

. . . .

. . . .

Date June 12/74

ABSTRACT

A portable data acquisition system was developed for experimental investigations of aerodynamic models. The system records the test data in digital form on a standard cassette tape recorder, which is later played back for analysis by computer. The early conversion of data from analogue to digital form maintained overall accuracies to ± 1 percent. The system is notable in that it eliminates the need for any manual data handling, and is not restricted to the present application.

The data acquisition system was used in wind tunnel and free flight testing of low Reynolds number airfoils. Some suggestions for improvements to the system arise from a discussion of these tests, and comparison to theoretical results.

ACKNOWLEDGEMENTS

The author wishes to express his thanks and appreciation first to Dr. D. J. Marsden for his patience and guidance throughout the course of this work. He also gratefully acknowledges the financial support of this research by the National Research Council.

Thanks are also due to the staff of the Mechanical Engineering Shop and Technical Services, in particular Bill Diachuk, Gary Knull, and Wayne Sherrard, for their work in the construction and maintenance of the electronics, and to Larry Lewis for his assistance in programming the PDP 11/20.

The author would also like to thank Martha Laverty for her careful typing of this thesis.

Lastly, the author would like to thank all persons who have offered encouragement and understanding. Except for my parents, these are, unfortunately, too numerous to name here.

TABLE OF CONTENTS

			Page
CHAPTER I	INTRODUCTION		1
CHAPTER II	DATA RETRIEVAL		4
	2.1	Transducers	4
	2.2	Electronic Data Handling and Control	7
	2.3	Data Recording	11
	2.4	Power Supply	12
	2.5	Data Transfer to Coded Tape	12
	2.6	Calibration and Accuracy	16
CHAPTER III	DATA ANALYSIS		25
CHAPTER IV	APPLICATIONS OF THE DATA ACQUISITION SYSTEM		36
	4.1	Test Configurations	36
	4.2	Results	39
	4.3	Discussion	45
	4.4	Suggested Improvements and Further Work	47
CHAPTER V	CONCLUSION		49
BIBLIOGRAPHY			51
APPENDIX A	Electronic Circuit Diagrams		52
APPENDIX B	Computer Programs		56
APPENDIX C	Sample Computer Outputs		84

LIST OF FIGURES

Figure		Page
1	Data Acquisition System (Schematic)	5
2	Central Electronics Schematic	8
3a	Signal Conditioning and Counter (Schematic)	14
3b	PDP 11/20 Count Initiation and Interrupt Servicing	15
4	Calibration Curve for Temperature	18
5	Calibration Curve for Airspeed (Dynamic Pressure)	19
6	Calibration Curve for Static Pressure	20
7	Calibration Curve for Scanivalve Pressure Transducer	21
8	Calibration Curve for Position	22
9	Calibration Curve for Wake Pitot Pressure	23
10	SCANREAD and TRAVREAD Flowchart	26
11a	SCANPLOT (MAIN) Flowchart	29
11b	SCANPLOT (FLOUT) Flowchart	30
11c	SCANPLOT (FLIN) Flowchart	31
12a	SCANLIFT (MAIN) Flowchart	32
12b	SCANLIFT (FLOUT) Flowchart	33
12c	SCANLIFT (FLIN) Flowchart	34
13	TRAVDRAG Flowchart	35
14	Results from Series III	40
15	Results from Series IV	41
16	Results from Series V	42
17	Series III Pressure Distributions	43
18	Series IV Pressure Distributions	44

LIST OF FIGURES (CONTINUED)

Figure		Page
19	Crystal Oscillator and Shaper	53
20	Scanning and Solenoid Stepping Circuit	54
21	Multiplexer and A/D Conversion Circuit	55

CHAPTER I

INTRODUCTION

A large quantity of data which is mainly repetitive in nature must be collected and analysed when testing aerodynamic models. Performing this function manually is extremely time consuming and may easily result in errors. In the special case of free flight testing, it is extremely difficult to collect this data manually. A system which automatically collects and records series of data at a high speed becomes necessary. If, in addition, the system can be interfaced with a computer for data analysis it becomes even more valuable. These were the incentives for the construction of the system described in this paper.

Free flight testing has been carried out with models mounted on trucks and airplanes (1). The advantages of this method of testing are that large scale models can be used, that wind tunnel constraints are avoided, and that free air turbulence is simulated exactly. The most serious disadvantages are the problems of operating equipment in the field and the great dependence on weather. The use of a data acquisition system such as is described in this paper best alleviates the first difficulty; the second one cannot be helped.

There are many data acquisition systems available commercially. However, several restrictions and limitations placed on the system due to its application in free flight testing made it desirable to construct a system suited specifically for these tests.

Apart from physical limitations on the size and weight of the system, and the necessity that it be operated under battery power, it was of prime importance that the system be easily operated with a minimum of supervision by the pilot of the carrier. It was necessary, therefore, to provide some kind of automatic control on the system. This would be necessary even if a commercial system were used. However, the nature of the data was such that discrete monitoring of several data channels (rather than continuous monitoring) was necessary. This sequencing function could also be built directly into the system controls, rather than modifying a commercial system. The construction of a system also ensured that the system could be easily interfaced with the data analysis computers. The system could also be constructed so as to provide digital signals recorded on tape which could be easily read directly into the computer.

A commercial data system, outside of limitations on space and weight, would require extensive hardware modifications and additions for its use in the present application. Therefore, for reasons of uniqueness of application, assured compatability with other hardware, and the availability of extensive computer resources for analysis, the total design and construction of a data acquisition system was undertaken.

The two chief goals of the designed system were that the process of data recording and analysis be automated as much as possible, hence no manual processing of data be required, and that the system be portable, with the application to free flight testing in mind.

The designed system is an extension of an idea presented by Whitfield (2), in which a much simpler data recording system used in the performance testing of gliders was described. A fundamental difference of the present system compared to Whitfield's is the use of standard transducers which produce a simple analog DC voltage output signal. Since the central electronics accepts a standard voltage input (usually 0 to 5 volts) any transducer having similar output characteristics can be used. This greatly increases the flexibility of the system making it useful for a wide range of applications. For example, studies in vibration, fluid mechanics and heat transfer, to name a few, could all be carried out using this data acquisition system.

The system itself operates in two parts: data recording and data analysis. The former is dependent largely on the hardware system and is described in Chapter II. The data analysis phase is most dependent on the software package developed for the large ground based computer. It is discussed in Chapter III. Following the system description, two applications are presented in Chapter IV, along with discussion of results and some suggested improvements to the system.

CHAPTER II

DATA RETRIEVAL

For aerodynamic testing, the minimum set of experimental parameters which must be measured are airspeed (or dynamic pressure), static pressure, angle of attack, local static pressure at points on the model surface, temperature, and velocity at points in the wake of the model. Many measurements of local static pressure and wake velocity must be made at each test configuration to give the pressure distribution and wake velocity profile. From this data, values of lift, moment and drag can be found, and a complete performance analysis can be made.

The system developed to perform these measurements is shown schematically in Figure 1. The system operates under the control of the central electronics module. Due to the complexity of the system, the major components will be discussed separately.

2.1 TRANSDUCERS

The transducers used for measurement of the parameters mentioned above were standard models. Their general features will be included here, as they affect other components in the system.

The dynamic and static pressures were measured with Validyne capacitance-type pressure transducers. The dynamic pressure was measured directly, while the static pressure, in free flight tests, was measured relative to a sample of air contained in an insulated and sealed vacuum bottle.

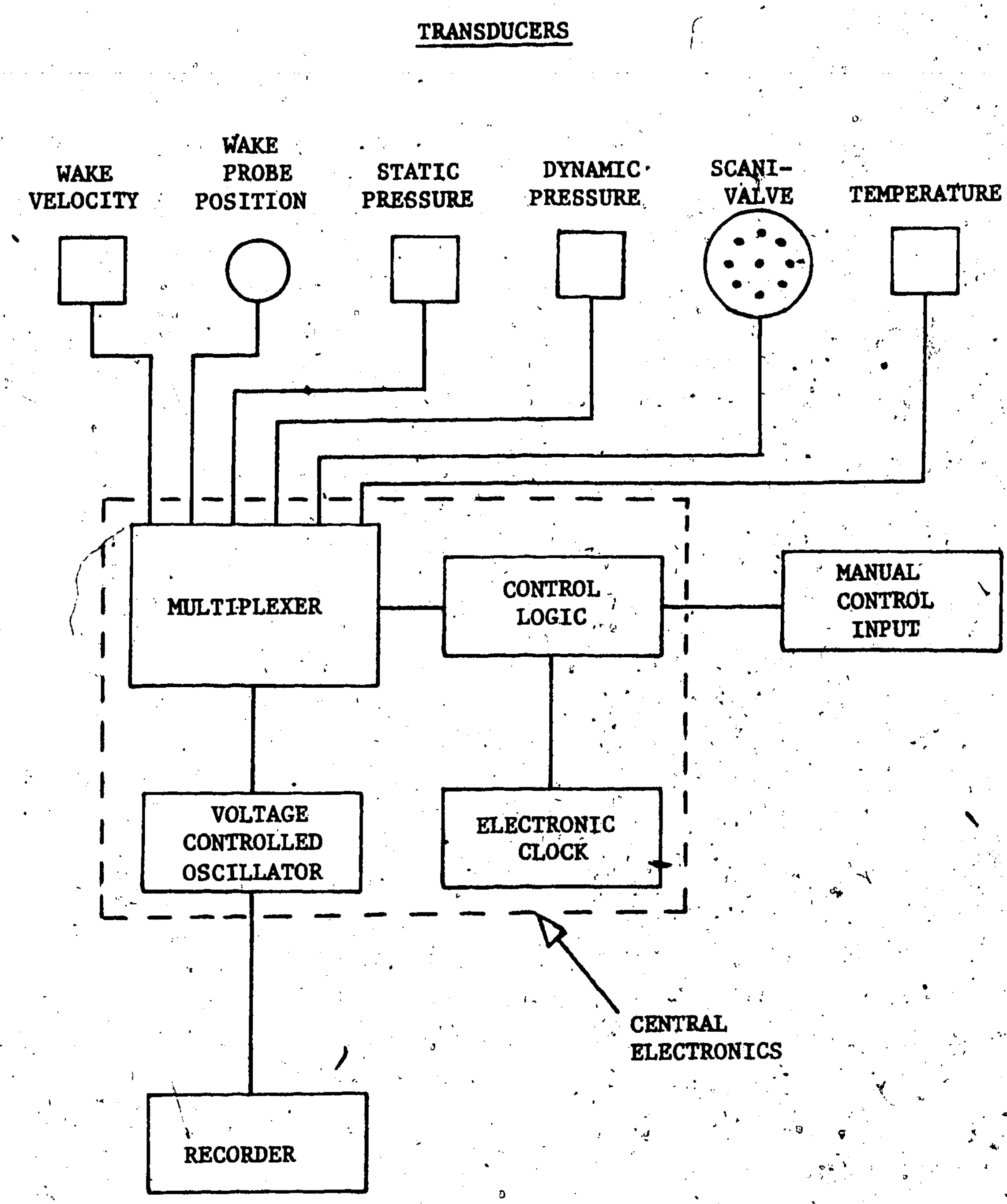


Figure 1. Data Acquisition System Schematic

Temperature was measured using a TSI Telethermometer transducer.

A scanivalve was used for measurement of local static pressure at points on the model surface. This instrument is a combination transducer and solenoid drive mechanism. Up to 48 pressure tubes, coming from taps on the model, can be connected to the ports on the head of the scanivalve. Each port is connected in turn, by means of a channel in the scanivalve rotor, to the transducer cavity. The rotor is moved from port to port by the solenoid drive mechanism. After a complete revolution of the rotor, a limit switch indicates that all the ports have been monitored. This instrument was ideal for this application because of its compactness and simplicity of operation.

The angle of attack was measured, in the free flight tests, by a yaw probe mounted on the model support structure. The scanivalve was used to monitor the pressures coming from the two sides of the yaw probe.

Two types of transducers were used for the measurement of velocity at points in the wake. The first was a constant temperature hot film anemometer. This instrument was used in hopes of getting information on the free stream turbulence level, which is of some importance to low Reynolds number airfoils. However, extreme difficulty was experienced in using this instrument, both in operation and analysis of data. Quantitative information on turbulence levels could not be obtained directly with the type of recorder used. An FM recorder is necessary for this. Also, since the anemometer had to be calibrated for both temperature and density variations, the analysis was quite difficult. Following these problems, a standard pressure transducer and Pitot probe were substituted to measure the dynamic head in the wake directly. This greatly increased the reliability of the data and simplified the analysis. The probes for these transducers were mounted on a stem behind the model. The stem could be traversed through the wake by a geared variable speed motor drive. The position of the probe in the wake was monitored by a 10 turn potentiometer coupled directly to the motor shaft. The potentiometer was supplied with a constant voltage from the central electronics, and the output voltage was a linear function of the probe position.

2.2 ELECTRONIC DATA HANDLING AND CONTROL

The heart of the data acquisition system is the data handling and control system. The function of this component is to sequentially monitor the transducer signals and produce the digital data signals sent to the recorder. The control circuitry also operated the scanivalve drive mechanism and responded to the start and stop signals from the operator control and the cycle limit switches. A block diagram of the central electronics is shown in Figure 2 and circuit diagrams are in the appendix.

The operation of the electronics in the treatment of data was as follows: A multiplexer circuit sequentially gated each of the transducer signals to a voltage controlled oscillator. The transducer signals were amplified and bias voltage applied where needed to operate the oscillator in a higher frequency range. Strict timing for the multiplexing process was achieved by using an electronic clock based on a very high frequency (262.144 KHz.) crystal oscillator.

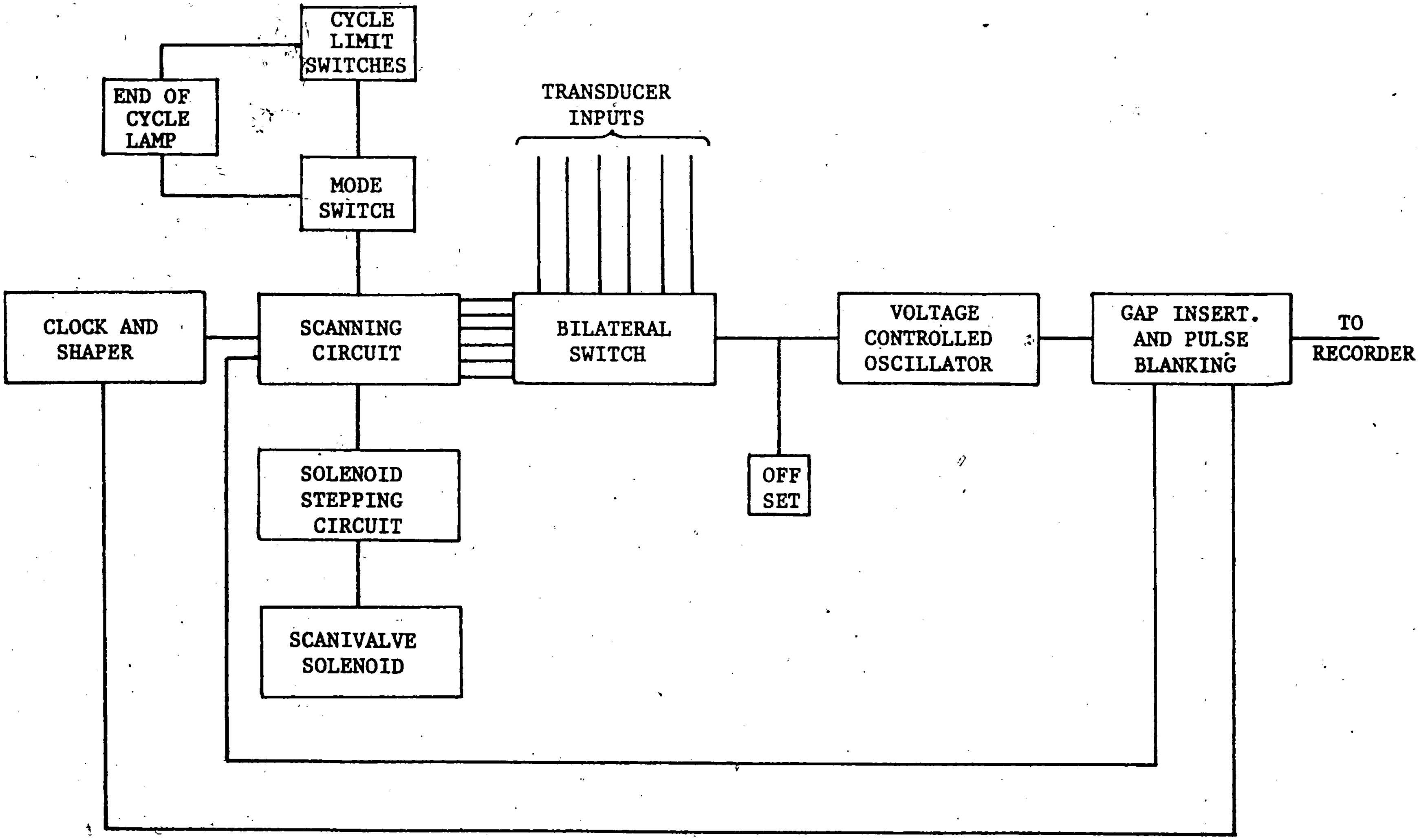


Figure 2. Central Electronics Schematic

Each transducer signal was gated through to the voltage controlled oscillator for a period of 289.05 milliseconds. The multiplexer then closed off the transducer signal and, after a short delay of 39.06 milliseconds, connected the next transducer in sequence through to the oscillator. The signal gap was used to separate data points as will be explained in section 2.5. Each transducer signal was monitored in turn and the multiplexing process repeated until all the data was recorded.

The voltage controlled oscillator produces an output frequency proportional to the input voltage. Input voltages from 0 to 10 volts produced output frequencies from 0 to 10 KHz. The oscillator was operated in the range of 300 to 3,000 Hz. During the gap between transducer signals, output from the oscillator was blanked to leave a clean break between data points on tape. The output of the oscillator was then amplified and sent directly to the recording head of the data recorder.

The resulting record of data on tape consisted of a series of data bursts of varying frequencies separated by short gaps, in repeated groups of four or five. The number of bursts in each group was determined by the mode of operation of the system. In the wake traverse mode, five transducers were active. In mode two, the model pressure distribution was scanned and four transducers were used.

The number of cycles in each burst was therefore a digital representation of the voltage output of the transducers and hence of the physical quantity being measured. Tape speed on recording or playback was immaterial, since each data burst had been written on tape for a precisely controlled time period. That is, neither amplitude nor frequency of the signal as it appeared on tape mattered, as it was the number of cycles in the burst which carried information. The means by which these data bursts were read back and reduced to useful data is discussed in later sections.

When operating in the second mode, the scanivalve drive mechanism was controlled by the electronics package. Immediately after the scanivalve transducer signal was monitored and recorded, the scanivalve was stepped to the next port. The transducer was not monitored again for slightly more than a second, giving ample time for equilibrium to be reached at the new pressure.

The completion of data monitoring for a test condition was indicated to the central electronics by limit switches installed in the scanivalve drive and the traverse motor drive mechanisms. When these limits were met, the multiplexing and recording functions were terminated and an indicator lamp on the operator's control box signalled the end of the cycle.

The overall control of the system was through a hand-held box connected by multiconductor cable to the central electronics. This box held all the necessary controls for power, mode selection, traverse direction and speed, cycle start switches, and the end of cycle indicator lamp. The operator could therefore move around while data recording was in progress and remain in full control of the system.

The electronics remained idle while the next test condition was set up and until the signal to start recording was received from the control box.

2.3 DATA RECORDING

The output signal from the central electronics could be sent directly to a real-time computer to give an immediate read-out of data. However, to give portability to the system, a medium for temporary data storage was required. Ideally, this recorder would have written data directly in a suitable code to computer-compatible magnetic tape. This was unacceptable, however, due to the size of such a recorder and the required power loads. A small self-powered cassette tape recorder was selected as the temporary storage device. This necessitated a further step in data processing — the transfer of data from cassette to coded magnetic tape on a small ground based computer. This transfer phase will be discussed in section 2.5.

It was mentioned previously that the data output signal was sent directly to the recording heads of the cassette recorder. The recorder input signal did not require amplification and the internal circuitry of the recorder would have added noise and distortion to the signal. The recorder's playback circuits were used, however, with no difficulty.

The recorder used was a two channel model. The first channel was used exclusively for the test data. The second channel was used for a voice commentary on current test conditions, recording start and other information to identify the data files. Originally, it was planned that the second channel would also record the information on turbulence from the anemometer. This was impractical due to an automatic level control in the internal circuitry of the recorder.

The recorder was mounted so as to be easily removable from the instrument box. The use of cassettes for data storage was advantageous in the ease with which they could be manipulated and stored.

2.4 POWER SUPPLY

Power was supplied to all components in the data system from a set of 12 volt batteries. Two motorcycle batteries and one automobile battery were used. The former were connected in series to provide 12 and 24 volts to components requiring DC power. The larger automobile battery was used to drive an inverter which produced 110 volts AC for the other components. When operating in the lab, all components could be operated under external power to conserve the charge in the batteries. The traverse motor was powered separately by a 9 volt dry cell battery.

Under internal (battery) power, the data acquisition system could be run continuously for two to three hours, depending on initial charge in the batteries. After this time, or if the batteries were low, performance of the system became erratic. This was usually indicated first by the scanivalve solenoid drive failing to step to the next position, causing the read cycle to take inordinately long to complete. However, power failure proved to be of little problem as all necessary data could be recorded in much less than two hours.

2.5 DATA TRANSFER TO CODED TAPE

With the raw test data recorded on the cassette, the last step in the retrieval process was to transfer this data to computer-compatible tape in a usable form. This was done on a small real-time computer, a PDP 11/20. A functional schematic of the transfer process is shown in Figure 3.

The cassette recorder was interfaced with the computer through a simple signal conditioner. Each pulse from the cassette activated a trigger in the conditioner, sending a more uniform and sharply defined pulse to the computer. The threshold for trigger activation could be adjusted so as to eliminate noise induced triggering. This was monitored by visual inspection of the output signal displayed on an oscilloscope. As each pulse reached the computer, a count register was incremented, so that the number of cycles in the burst would be obtained.

The signal conditioner module also sensed the gap between data bursts. When no signal arrived within 30 milliseconds (set by operator), an interrupt was sent to the computer. This caused the current total in the count register to be stored in core memory, and the register to be reset to zero for the next burst.

When a certain number of count totals, specified by the operator as the block length, usually 40, were stored in core memory, the entire block was written on nine track magnetic tape in a code and format compatible with the large data processing computer (IBM 360/67).

When the data set for a test condition had been completely read into the computer, the counting function was terminated and all remaining totals in memory were written to tape. This termination occurred on the generation of an interrupt from the teletype console by the operator. The resulting file on magnetic tape consisted of several blocks of numbers representing the number of cycles in each

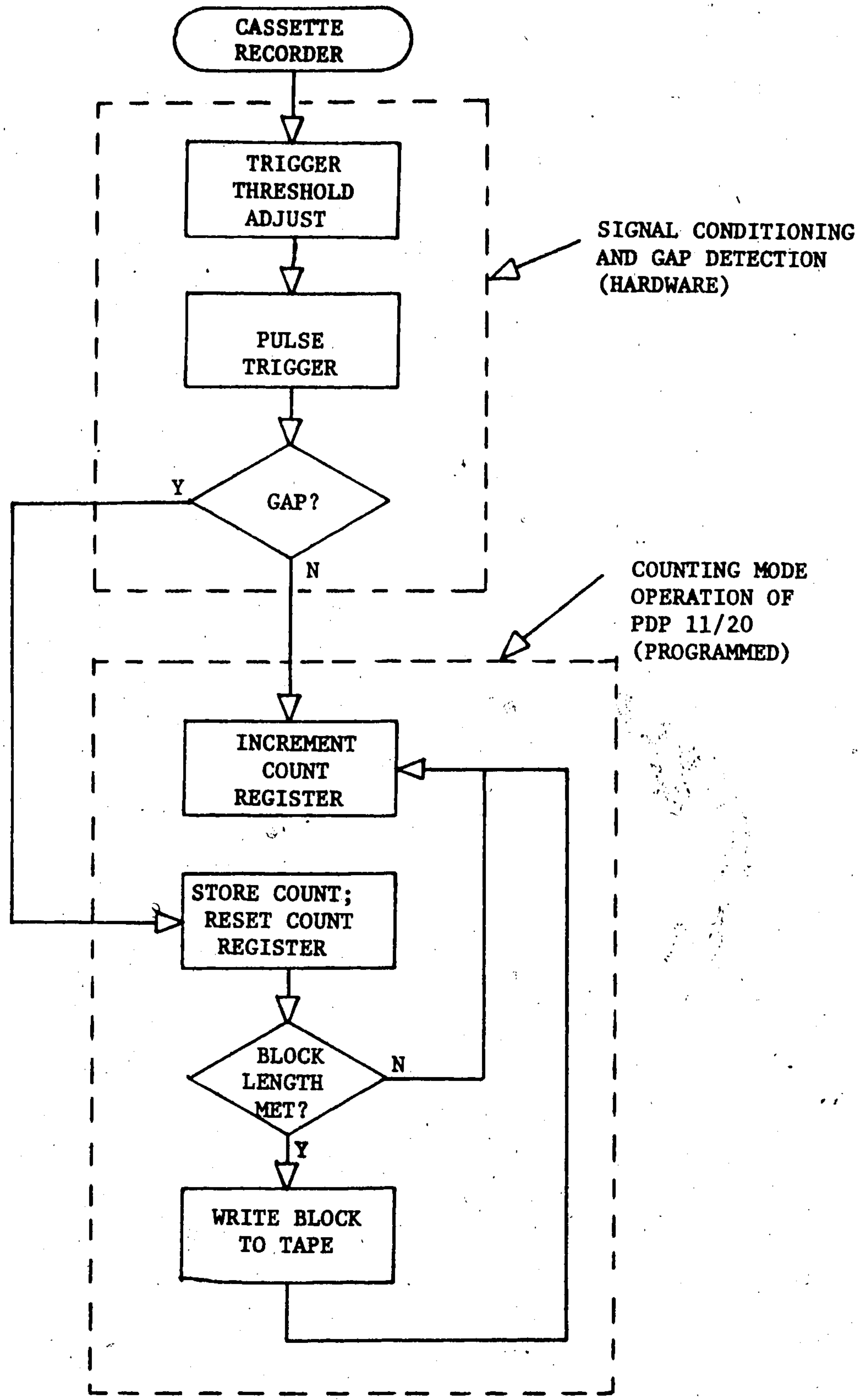


Figure 3a. Signal Conditioning and PDP 11/20 Counter (Schematic)

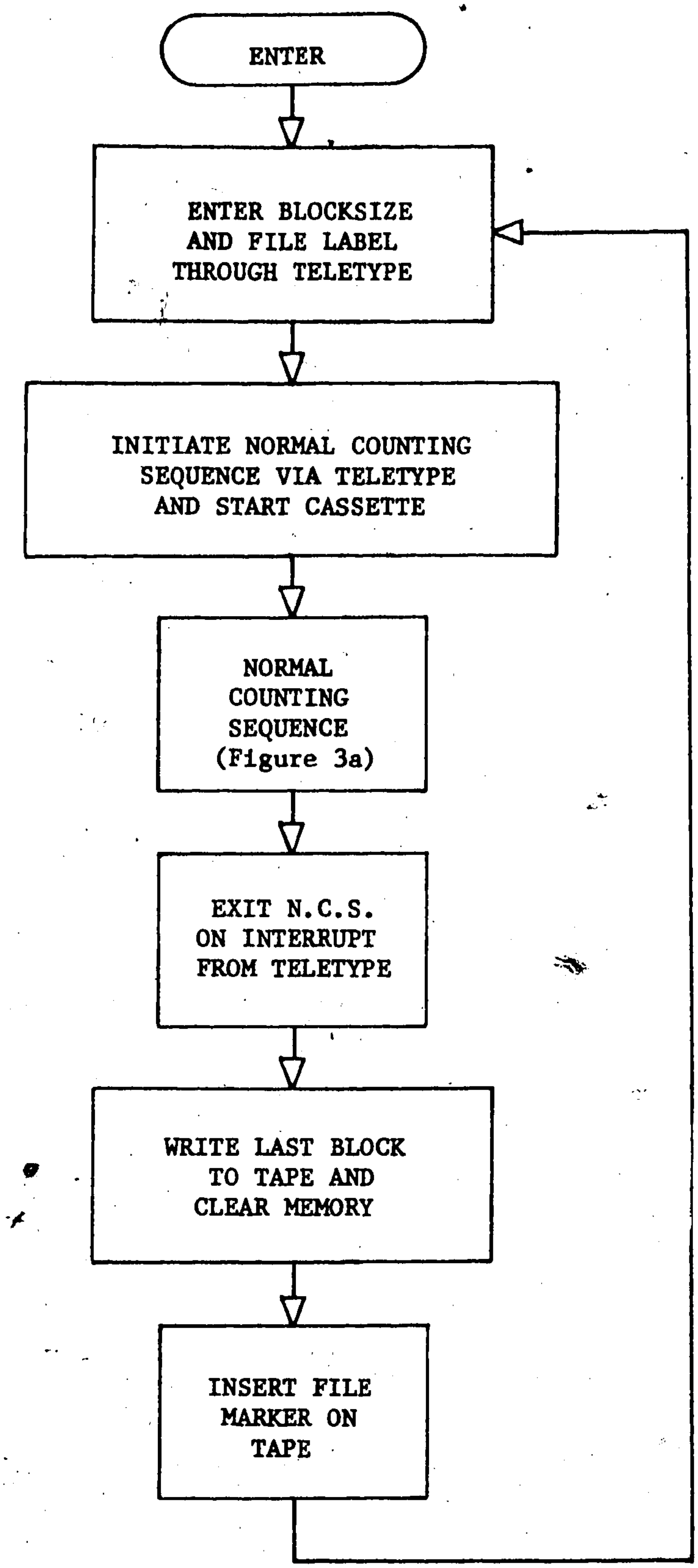


Figure 3b. PDP 11/20 Count Initiation and Interrupt Servicing

burst of the cassette record read. These numbers formed a data vector, with no distinction between each number to indicate its transducer origin. The job of sorting these numbers into separate vectors was done in the programs described in Chapter III.

After the terminate interrupt had been processed, the computer prompted on the teletype for the desired blocksize and file label for the next data set. When the counting function for the next data set was initiated through the teletype, this file label was written on the magnetic tape as the first block in the new data file. The label could be recovered on the large computer to give details of the test conditions.

The program for the PDP 11/20 is written in Assembler and is included in the appendix. This program resided on a magnetic tape in core image form and was loaded into the computer before each data transfer session.

With the digital representation of data on tape, the next phase in the data analysis could be carried out on the larger data processing computer. This was largely a software dependent function and is described in the next chapter.

2.6 CALIBRATION AND ACCURACY

Before any testing was carried out, the system had to be calibrated. This was done by setting up known physical inputs to the transducers and operating the system as if test data were being recorded. Using a small auxiliary program which simply read the coded magnetic tape, a list of counts for each transducer for each input condition could be obtained. These count totals were used to determine the calibration relations, samples of which are shown in Figures 4 through 9. The use of the derived calibration relations is discussed in more detail in the next chapter, but it should be mentioned here that the system was entirely recalibrated before every test series. This procedure required approximately half an hour.

The accuracy of the system can be described in three ways: precision, sensitivity, and repeatability. The precision of the data system refers to how accurately the computed value of the physical quantity compares with the actual value. Sensitivity relates to the smallest perceptable change in the computed value as a percentage of the full range of values. Repeatability refers to the degree with which the same measured value results from repetitive measurements of the same physical quantity.

The precision of the data system depends almost entirely on the calibration procedure and the resulting calibration relation. However, two factors other than the exactness of the calibration affect the precision. The first is the fact that the analog signals from the transducers were, effectively, integrated over the gate-open time. Thus, small variations in the physical quantity were averaged over that time period. Secondly, the count total arrived at by the small computer was accurate to only plus or minus one count. In a total range of five hundred counts (an average value) this represents a system induced error of ± 0.2 percent, which is the lower bound on the precision. The upper bound on the precision depended on the care and accuracy with which the calibration was accomplished. This is estimated to be on the order of 1 percent of full scale.

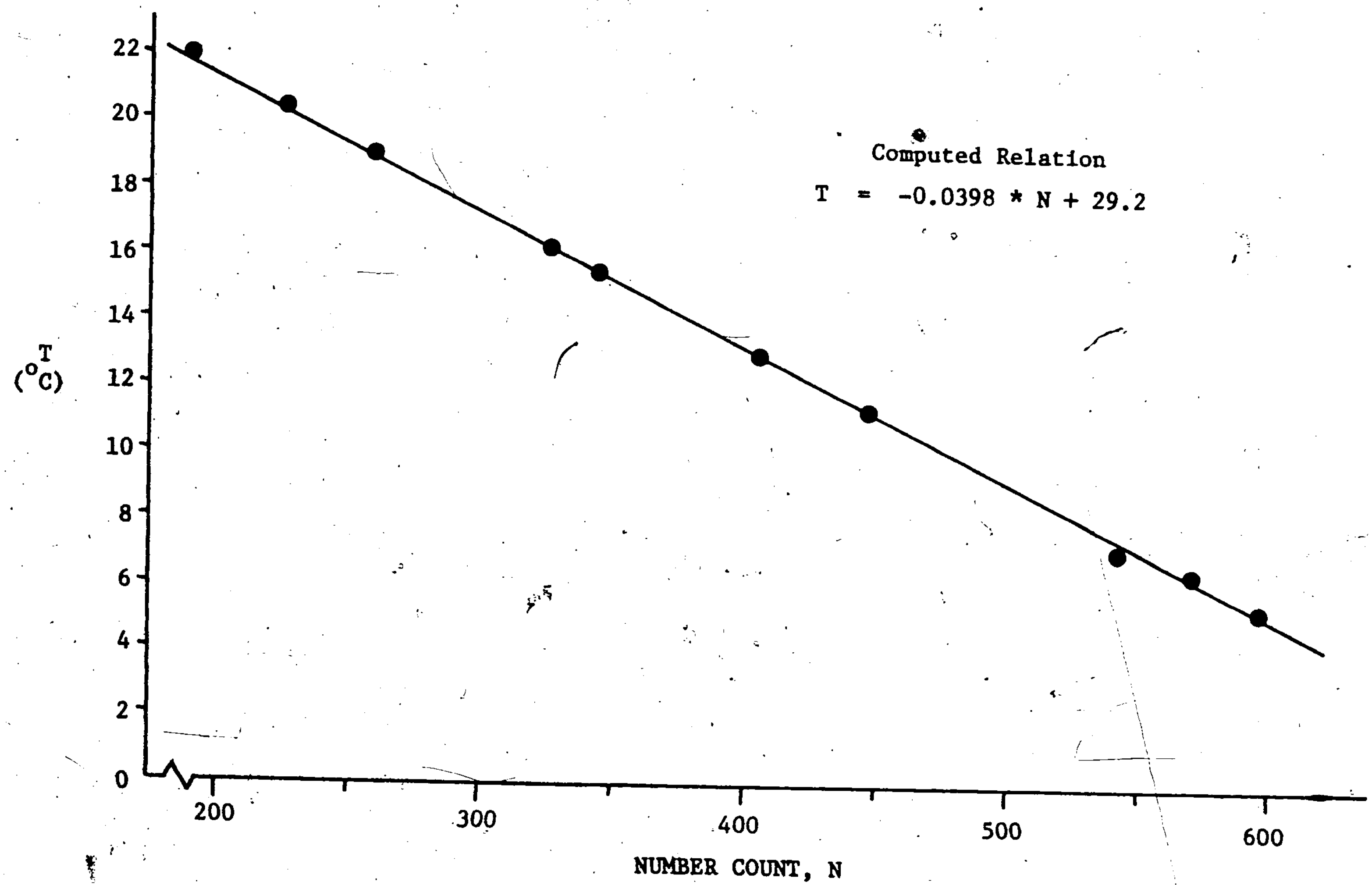


Figure 4. Calibration Curve for Temperature

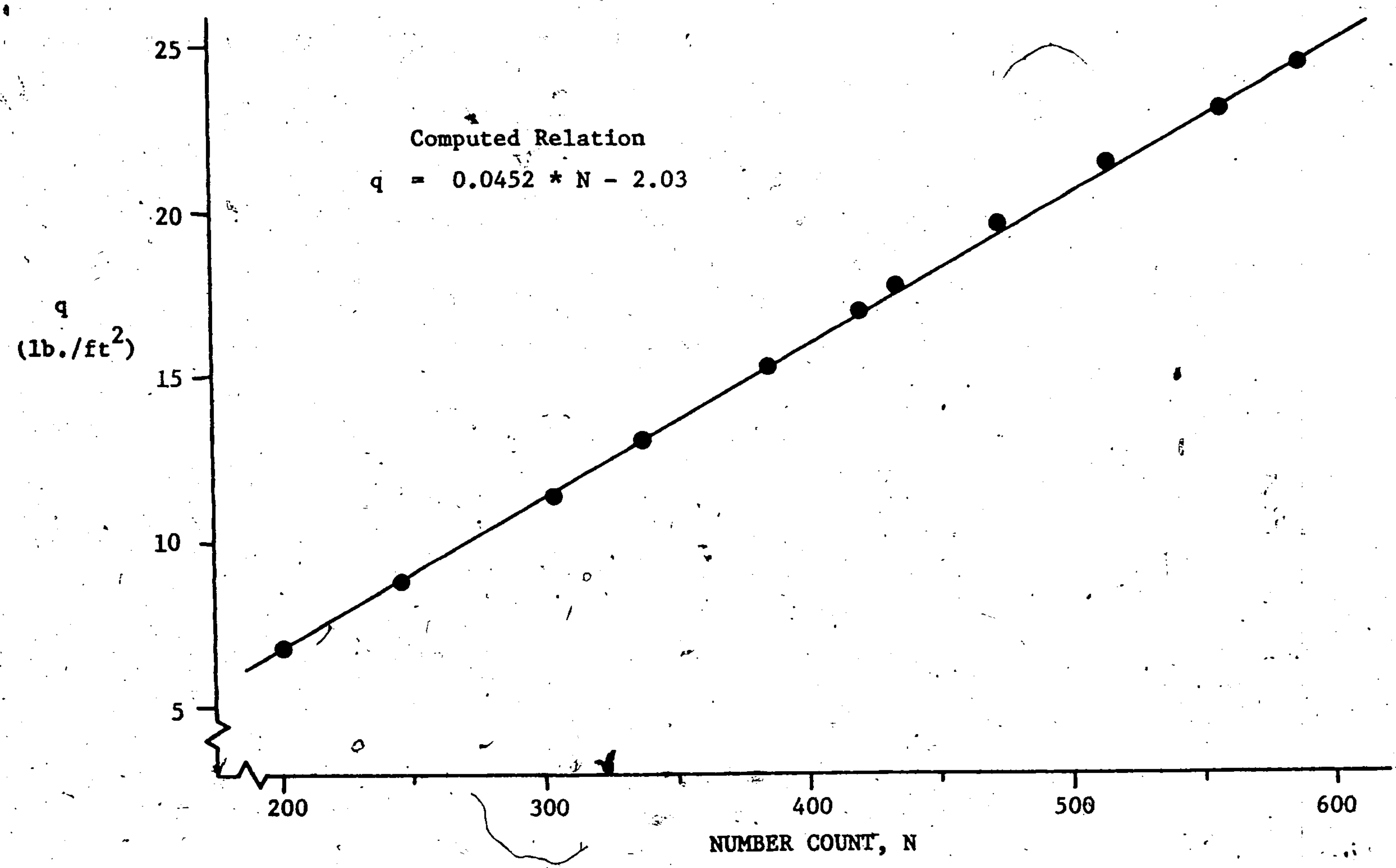


Figure 5. Calibration Curve for Airspeed (Dynamic Pressure)

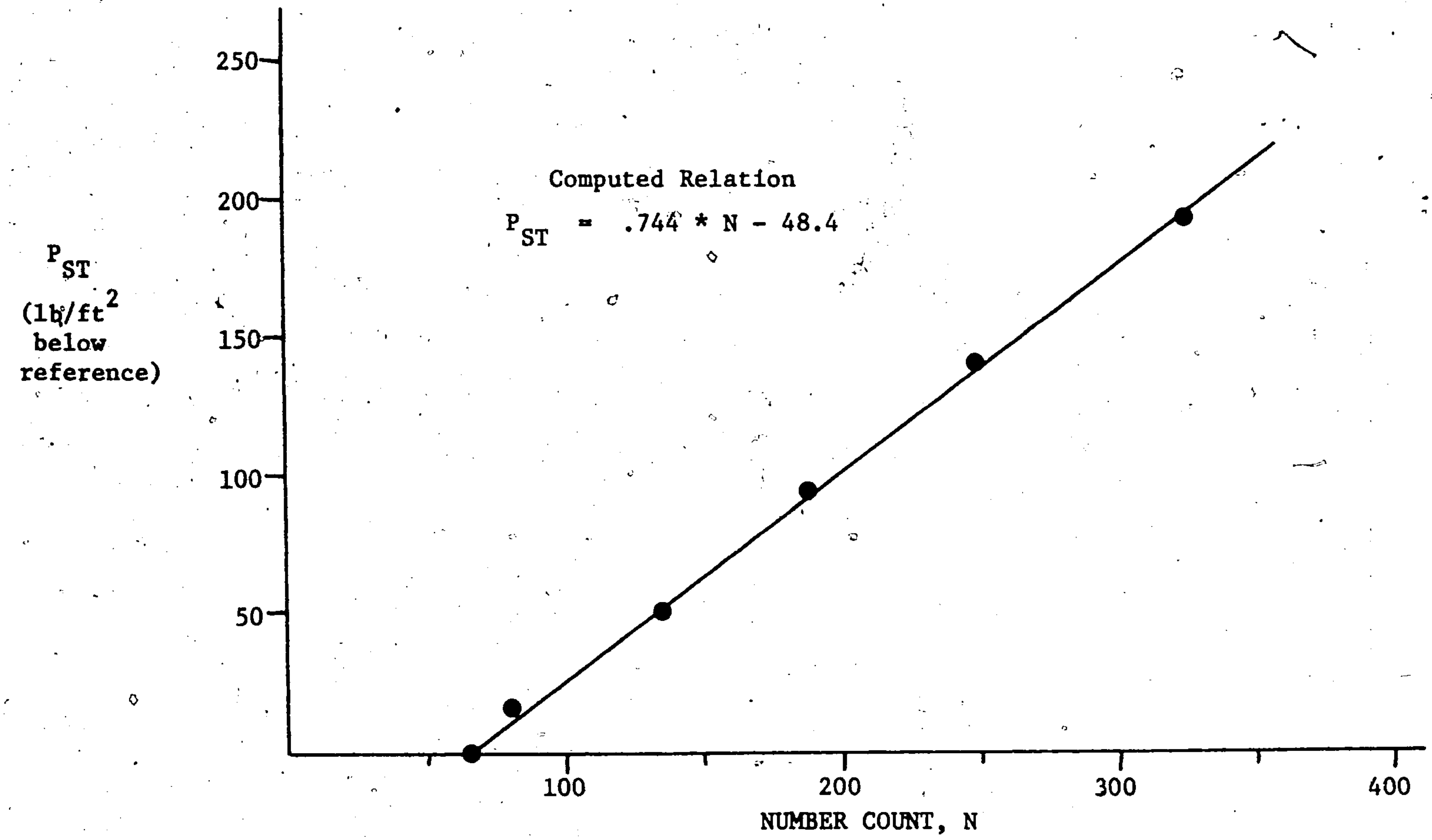


Figure 6. Calibration Curve for Static Pressure

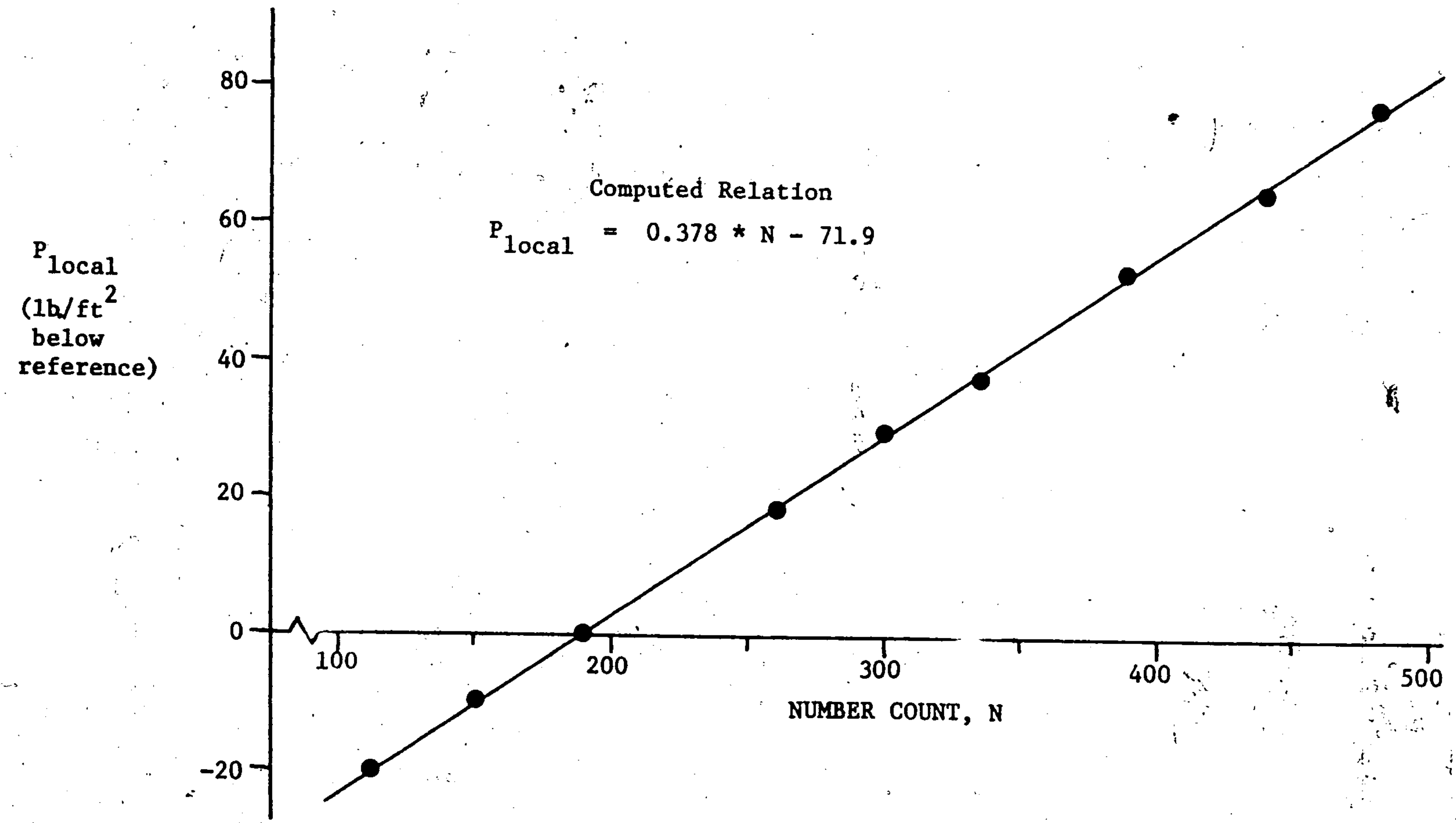


Figure 7. Calibration Curve for Scanivalve Pressure Transducer

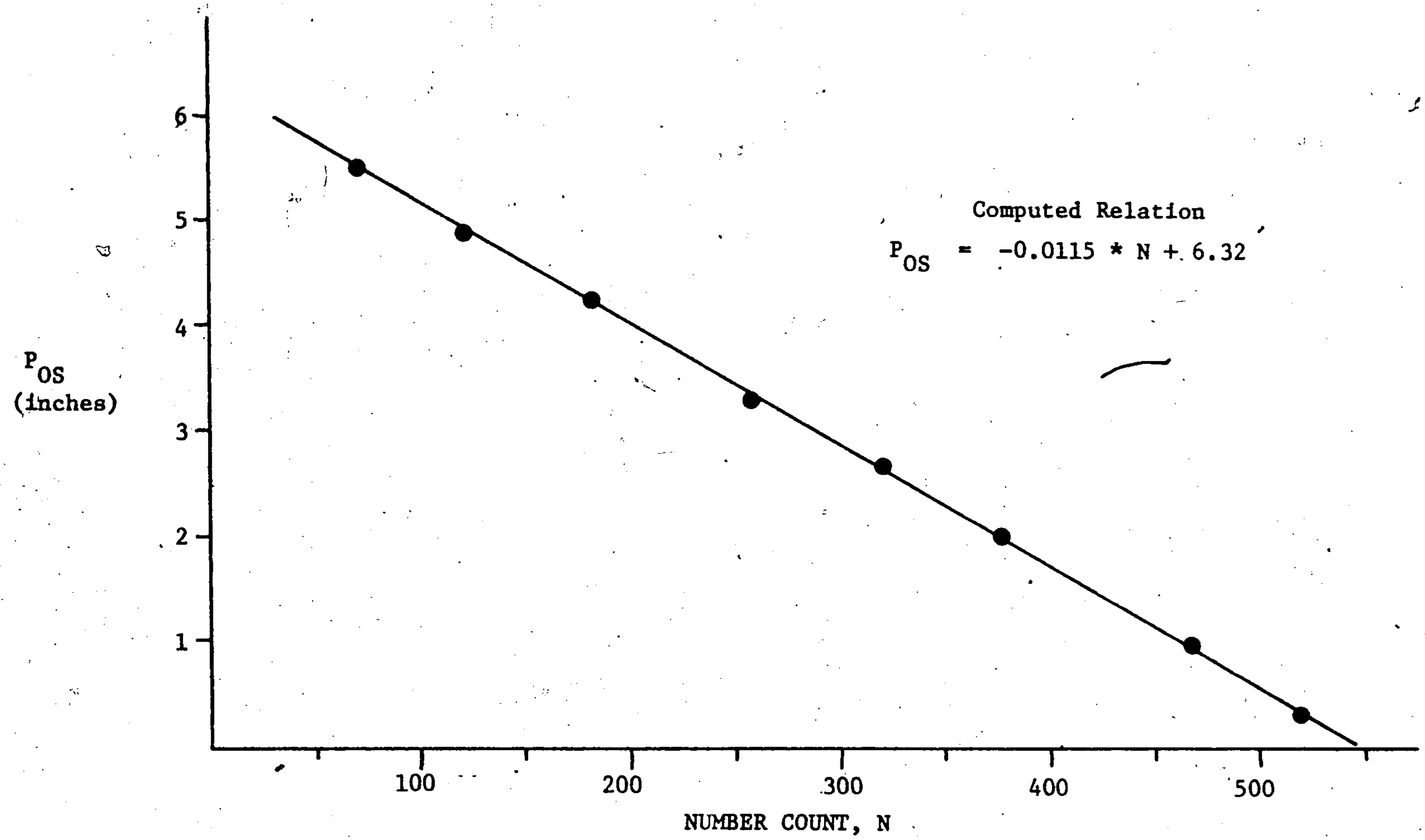


Figure 8. Calibration Curve for Position

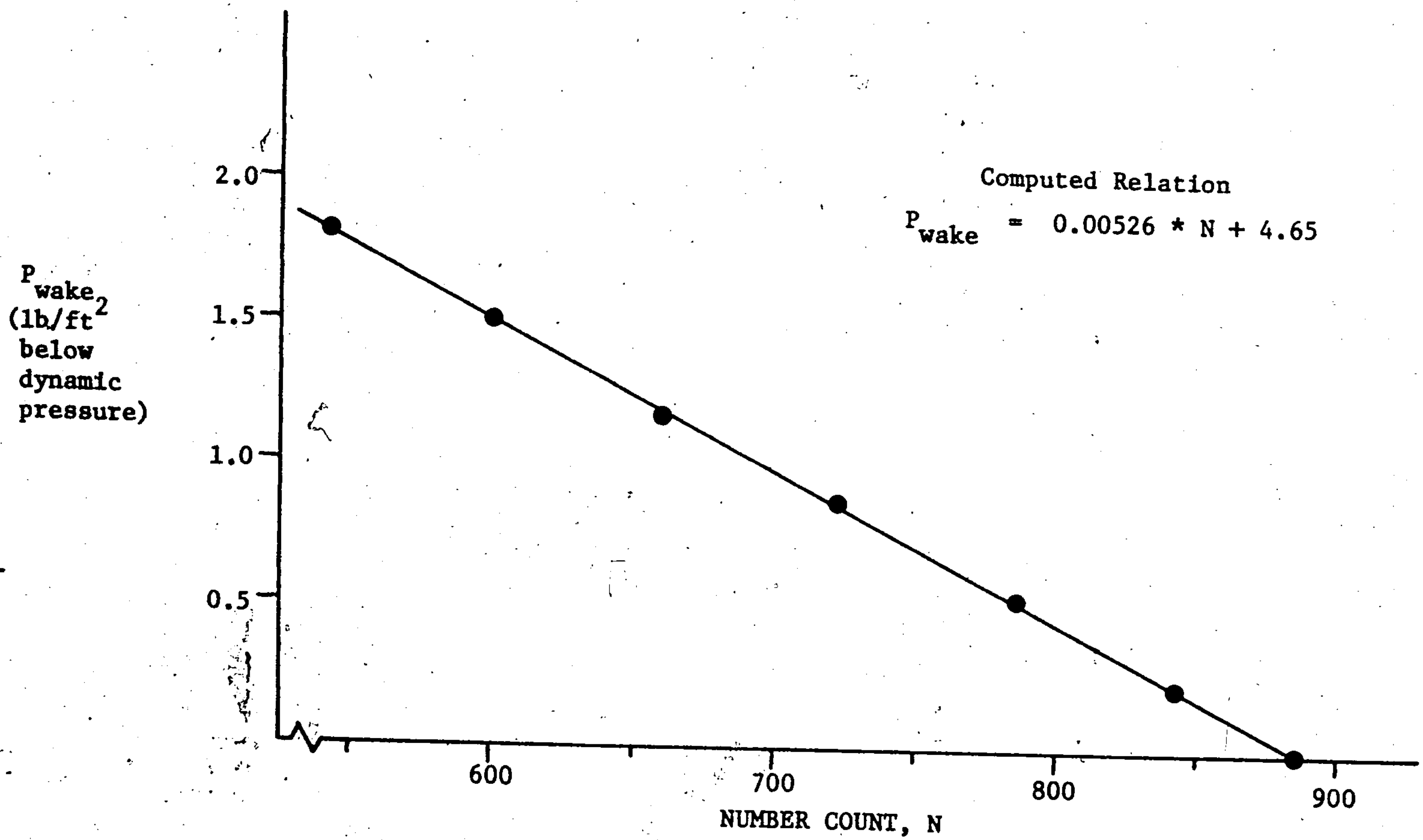


Figure 9. Calibration Curve for Wake Pitot Pressure

In this situation, sensitivity and repeatability are related. It was found that repeated measurements of the same physical quantity resulted in count totals that differed by, at most, ± 2 from the average. Thus, the smallest significant perceptable change is produced by a change of three in the count total. This represents a sensitivity of ± 0.6 percent of the full scale of five hundred. Since the sensitivities of each transducer were much lower than this, the ± 0.6 percent of full scale repeatability was the limitation, also, to the system sensitivity.

The comparison of the sensitivity and precision shows that even if great care was used in obtaining the calibration relations, the overall sensitivity of the system limits accuracies to ± 0.6 percent of full scale values. However, in view of the fact that final results depended on integrations of these values (for example, the lift coefficient derived from the pressure distribution), the accuracy of the final results were estimated to be on the order of ± 1 percent. This is quite good accuracy for experimental testing.

CHAPTER III

DATA ANALYSIS

Starting with the magnetic tape records of count totals, the task of data analysis breaks down into five main parts. The first is to sort the string of numbers on tape into vectors, one for each physical quantity being measured. Next, using suitable calibration relations, the actual values of the physical quantities can be calculated from the count totals. With these basic values, other physical quantities can be derived and secondary parameters calculated. Fourth, the results are made available for any further analysis desired by storing them on disk. Lastly, final analyses such as lift and drag calculations and plotting can be carried out. This last step was carried out by separate programs described below.

The first four steps are included in both the programs SCANREAD and TRAVREAD which are listed in the appendix. A flowchart for these programs is shown in Figure 10. The complete specifications for input format, peripheral device assignment, and execution commands under MTS (Michigan Terminal System) control are included in the appendix listings. It was necessary to use two programs for this first phase of data reduction because of the differing formats of data for the two operating modes of the recording system. That is, mode one (traverse) records five physical quantities, while mode two (scan) records only four.

Provision was made in these two programs for calibration

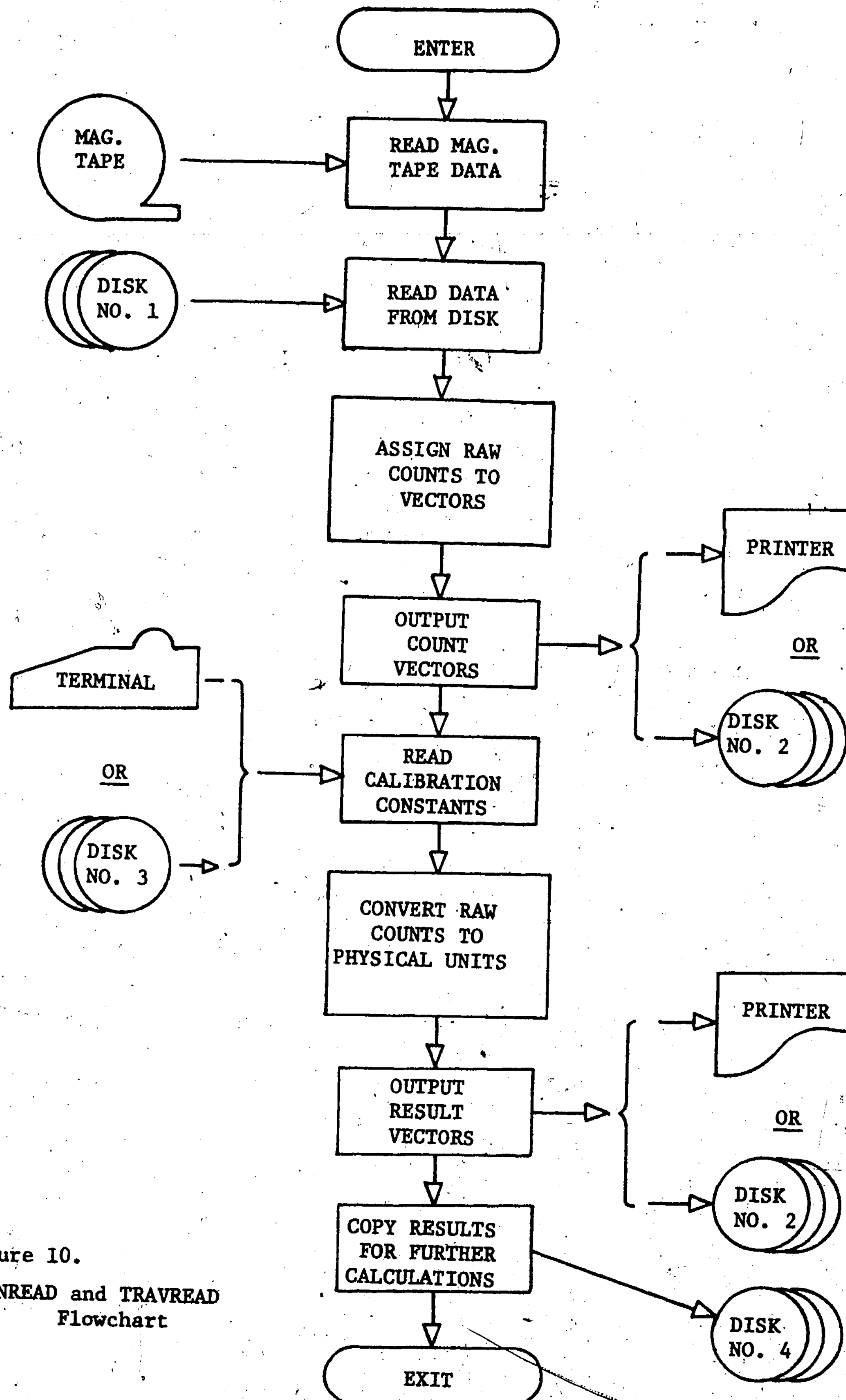


Figure 10.

SCANREAD and TRAVREAD Flowchart

constants to be included with each set of data rather than being built into the program. This was done by placing the appropriate factors in a disk file and reading them into the program during execution. Alternatively, the values could be entered through the terminal, if that was the *SOURCE* default. The derivation of the calibration relations was described in section 2.6. All the relations (with the exception of the hot film anemometer calibration) were linear. The anemometer required a two stage reduction, both of which were non-linear. It was found that the linear calibrations were most affected by a zero shift, while the proportionality constant stayed the same. Thus, by obtaining the count values at zero conditions only, a rough but reasonably accurate calibration could be obtained.

Once the primary physical values had been determined using the calibration relations, they could be used in combinations to derive the secondary values such as airspeed, density, altitude and pressure coefficients.

The fourth step mentioned above, and the last operation in SCANREAD and TRAVREAD, was to output these results to the printer and to transfer them to disk for further analysis. Accompanying this data to disk was a list of parameters describing the test condition, for example, airfoil name, angle of attack, location of static pressure taps on the model, and flap configuration.

Several programs were used for final analysis of the data written to disk by SCANREAD and TRAVREAD. The program SCANPLOT uses output data from SCANREAD to plot the pressure distribution on the model. The program SCANLIFT, using the same data as SCANPLOT, calculates the lift and moment coefficients. Each of these were set up to process several complete sets of data in a single run. The data set was a compounded list obtained by the execution of SCANREAD on several raw data files. The program TRAVDRAG reads a compounded data list, produced by several runs of TRAVREAD, to calculate the drag coefficient. Flowcharts for these analysis programs are shown in Figures 11 through 13.

All the above-mentioned programs were stored in object form on disk. Once a data tape had been produced, the task of analysis reduced to issuing a set of instructions for the computer to run a certain program with a certain data file. The programs are entirely compatable with each other regarding internal data exchange formats, and instruction queuing. In only a few minutes, a set of instructions could be assembled which would produce all the desired final results, including complete listing of raw and reduced data, pressure distribution plots, and lift, drag and moment coefficients.

The programs as written rely heavily on the very efficient and extensive peripheral device management programs resident in MTS. (In running SCANREAD, seven devices were used for input/output operations.) Also, extensive use was made of system and mathematical library subroutines, some of which may not be available on all installations. The resident subroutines used are listed in the appendix.

This completes the description of the data acquisition system. The next chapter describes two applications of the system, followed by a discussion of results and suggested improvements.

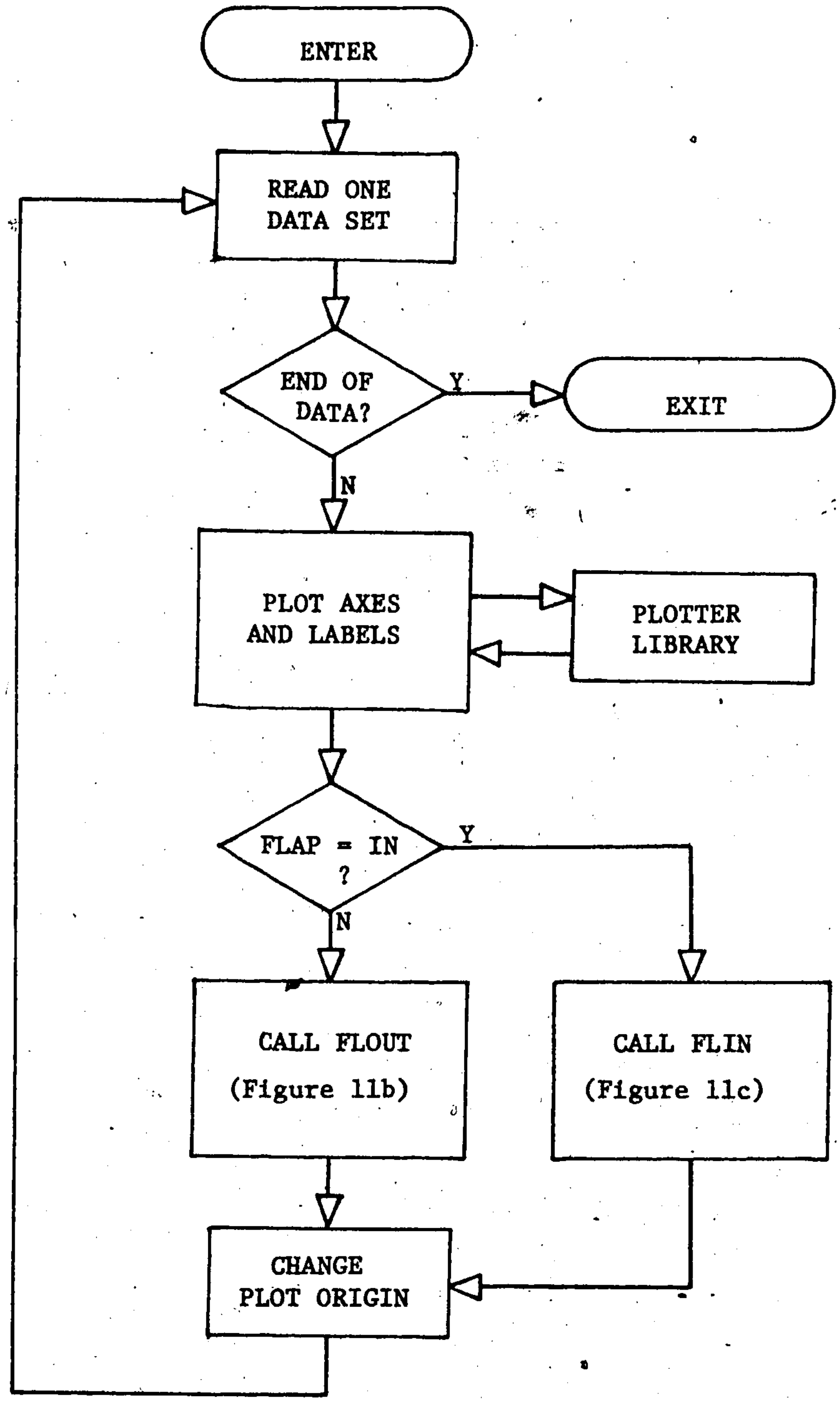


Figure 11a. SCANPLOT (MAIN) Flowchart

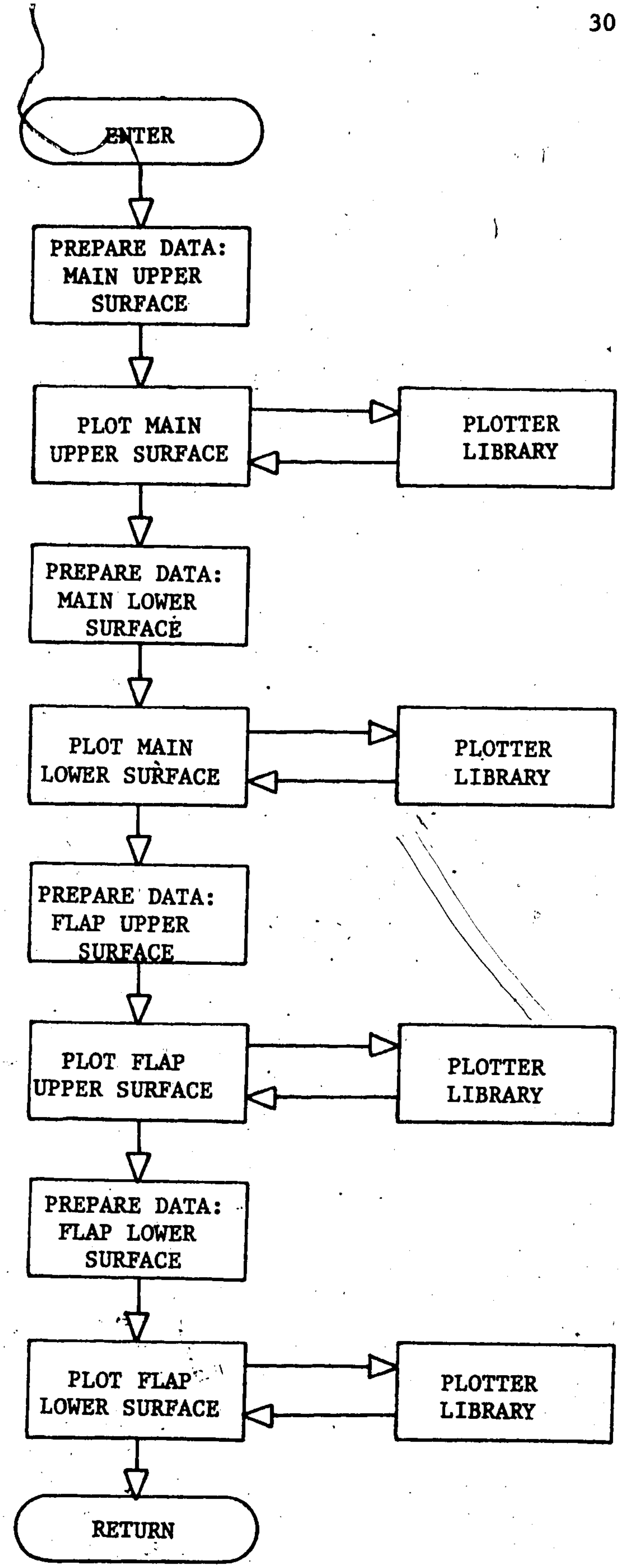


Figure 11b.
SCANPLOT (FLOUT) Flowchart

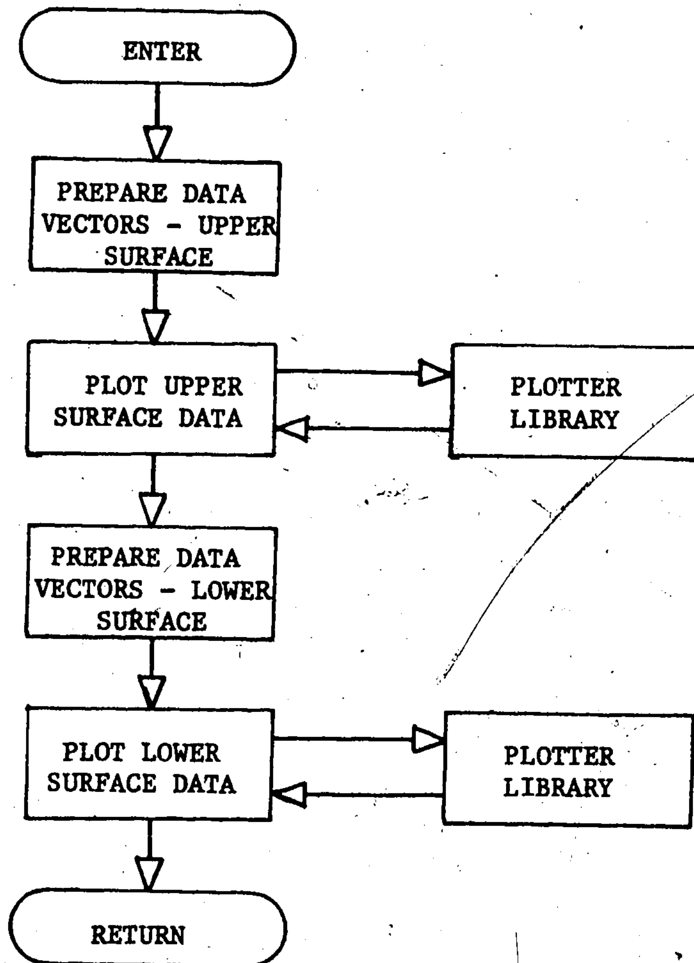


Figure 11c. SCANPLOT (FLIN) Flowchart

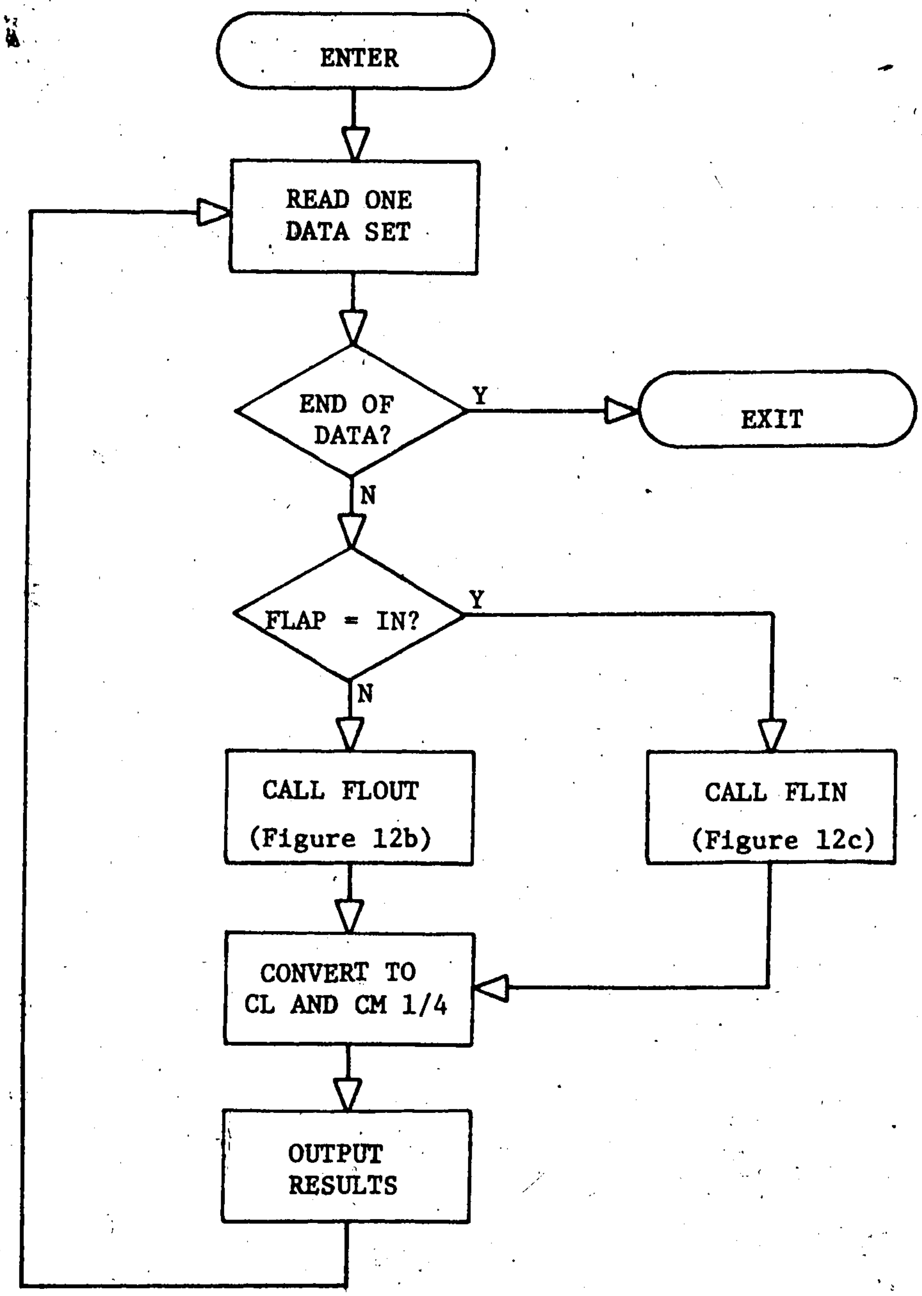


Figure 12a. SCANLIFT (MAIN) Flowchart

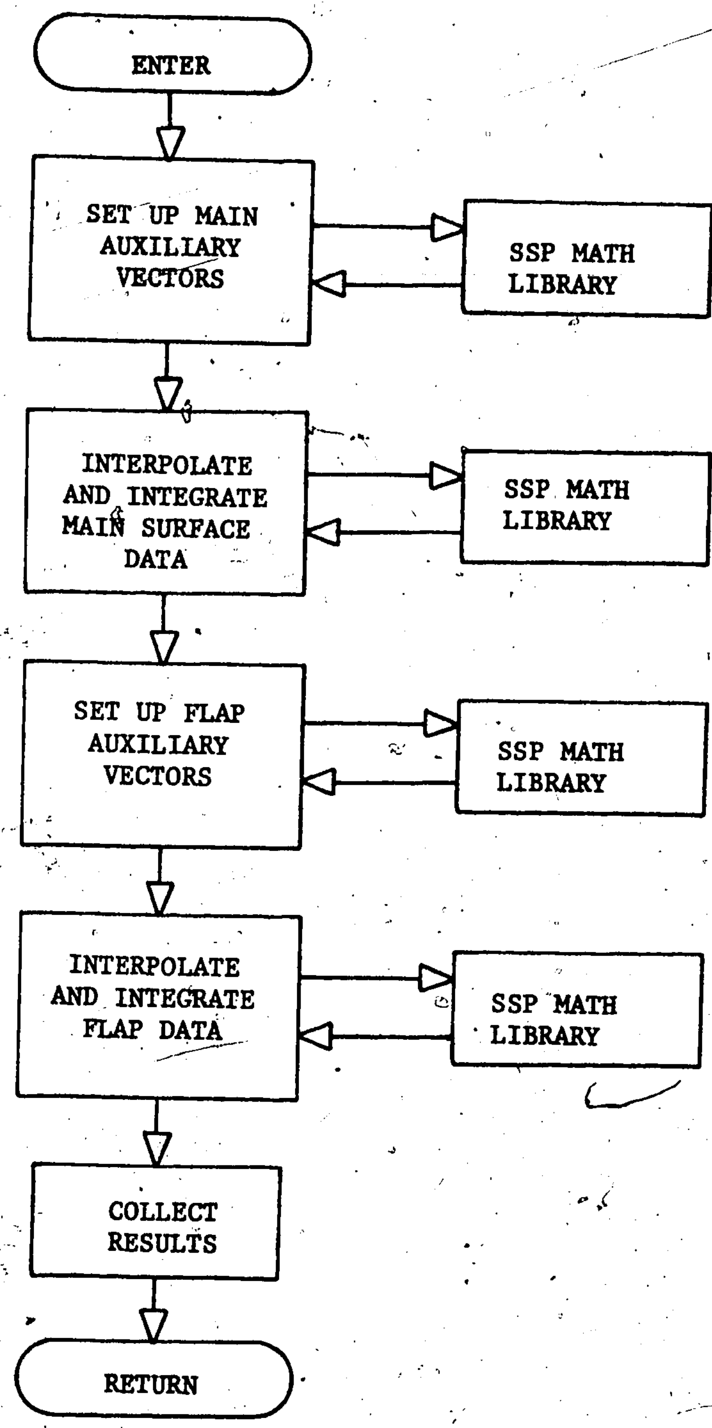


Figure 12b. SCANLIFT (FLOUT) Flowchart

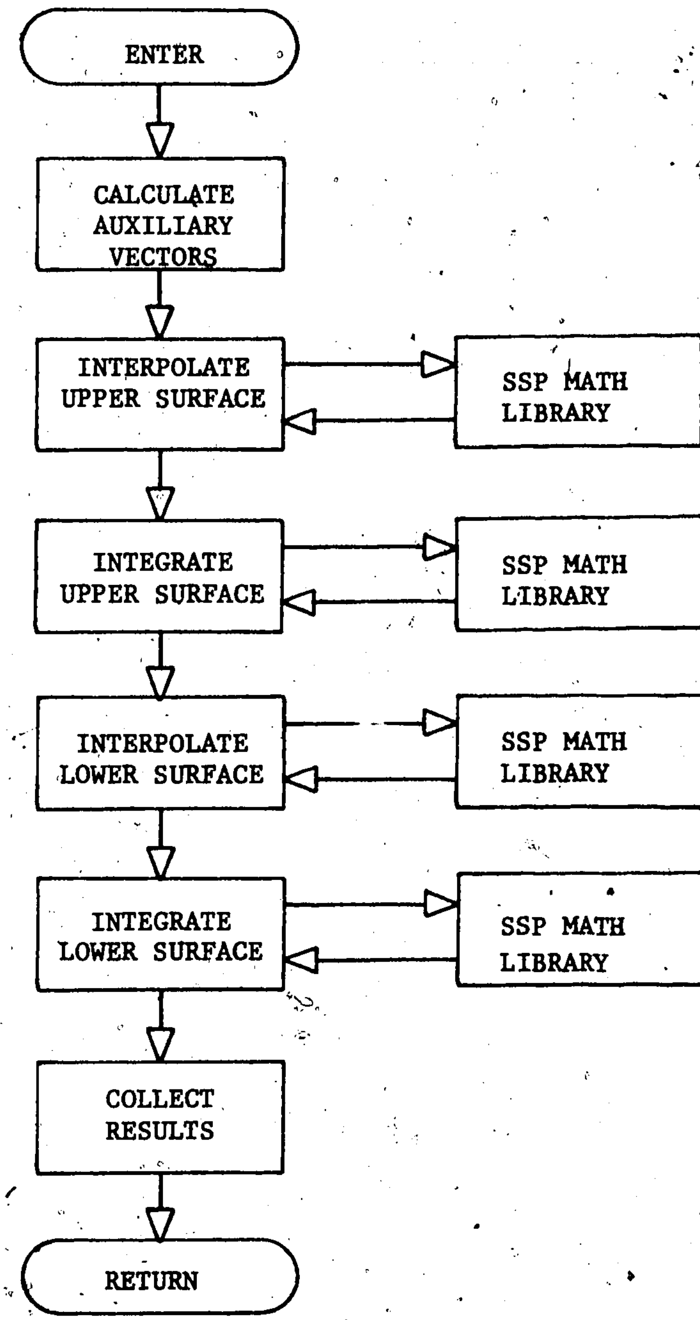


Figure 12c. SCANLIFT (FLIN) Flowchart

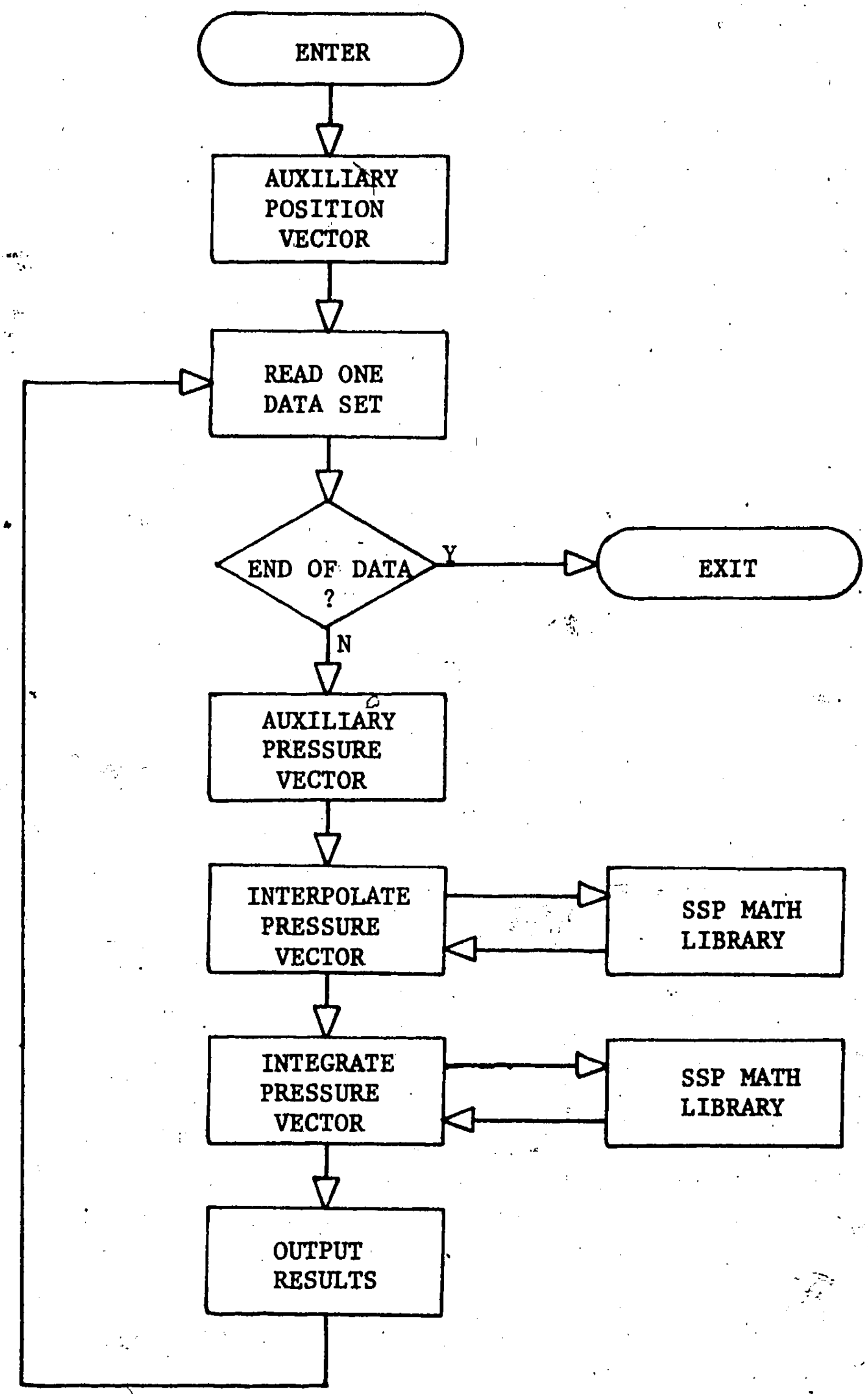


Figure 13. TRAVDRAG Flowchart

CHAPTER IV

APPLICATIONS OF THE DATA ACQUISITION SYSTEM

4.1 TEST CONFIGURATIONS

The data acquisition system was used in two applications with some minor variations in power supply and transducer arrangements. The first was free flight testing of an FX-61-163 profile at low Reynolds number. The second was wind tunnel testing of the same profile with a 35 percent chord slotted flap adjustable to any position.

The free flight tests were performed on a two dimensional model having a span of 48 inches and a chord of 18 inches. The model was mounted between rectangular endplates 54 inches above the wing of the carrier sailplane. The angle of attack was adjustable from the cockpit by means of a pushrod and lever mechanism. With this model configuration, Reynolds numbers based on model chord of from 0.6×10^6 to 1.1×10^6 could be achieved. The data acquisition system was mounted in the rear seat of the sailplane, with the control box in the cockpit accessible to the pilot.

The normal test procedure was to tow the carrier sailplane to 3,000 feet above ground, during which time measurements could be taken. Following release, more measurements were taken as the sailplane descended in free flight. An entire flight lasted from 25 to 35 minutes allowing a complete range of angles of attack to be covered.

A very large number of problems were faced in performing the flight tests. Generally, these arose from the operation of a fairly sophisticated electronic system far from the usual support equipment available in the laboratory. Numerous problems arose in the development of the system itself, many of which did not appear until after the system was mounted in the sailplane or had been aloft for the test. Since the airfield was some distance from support equipment, a great deal of time was lost in dismantling and transporting the system between the airfield and the laboratory. The dependence on weather proved to be another great problem.

Specifically, the greatest difficulty with the free flight tests was the operation of the hot film anemometer. Originally planned to be used to get an indication of the free stream turbulence as well as measuring the wake velocity, the anemometer required much more attention than was possible in a free flight test. Also, the output signal during the tests was found to be well outside the range of values obtained in the laboratory, upsetting the electronics considerably and destroying many sets of data. This was probably caused by the extreme deviations of air temperature and density in the test conditions from those of the calibration conditions. This posed no great problem, however, since the wake traverse data could be obtained using a Pitot probe and a standard pressure transducer. This substitution was then made on later tests. Since it was only hoped to get qualitative information on free stream turbulence with the original system (quantitative data would have required much more equipment, and be even harder to operate), the loss of this information was not considered serious.

Preliminary results obtained in the free flight tests pinpointed some further problems. Foremost among these was the determination of the angle of attack of the model. As mentioned previously, the angle of attack was measured with a yaw probe mounted on the model endplates, with pressure lines leading to the scanivalve. Thus, the angle of attack was monitored once in scan mode and not at all in traverse mode. Although the pressure distributions measured during these tests were fairly smooth, there was a very large scatter in plots of lift coefficient against angle of attack. This could have been caused by changes in the attitude of the sailplane. Other than the location of the pushrod, there was no way of determining exactly what the angle of attack was at each point in a test.

Lacking the time to make the necessary changes to the data acquisition system to correct these problems, and also because of poor weather, it was decided to terminate the free flight tests.

For the second application of the data system, a series of wind tunnel tests were set up. The model tested was again the FX-61-163 with 35 percent slotted flap, but with only a 12 inch chord. The flap was adjustable to any setting. There were nearly twice as many pressure taps on this model compared to the flight test model to obtain more definition in the pressure distribution. The data system was run under external power to eliminate the periodic need for battery removal and recharging.

It was possible to complete a series of tests in the wind tunnel covering a full range of angle of attack at a given speed and flap setting and record all the data in less than half an hour. The transfer of data from cassette to magnetic tape took equally as long.

With all the data on tape, only several minutes were required to assemble a set of instructions for analysis on the large computer. A discussion of these results follows in the next section.

4.2 RESULTS

The wind tunnel tests were performed at flap settings of 0, 10, and 20 degrees, labelled Series III, IV, and V respectively. (Series I and II were preliminary tests used in setting up the test equipment.) The Reynolds number for all tests was 0.8×10^6 based on chord. The final results for these tests are shown in Figures 14 to 16. A few sample pressure distributions as they are plotted by the computer are shown in the appendix. Several of these plots have been overlaid and are shown in Figures 17 and 18.

It should be noted that all data shown is uncorrected for wall constraints. The use of a semi-open top, as suggested in (4), makes the theoretical wall corrections negligible. More experimental tests must be carried out to see if that is, in fact, the case.

The present series of data show very little scatter, both in the pressure distributions and in the integrated lift and drag results, and are reasonably close to the values reported for this airfoil (3). Differences are on the order of 10 percent. It was felt that the chief sources of error in the final results were with the tunnel itself, since the model was quite large. This is a serious problem when testing high lift airfoils, and further efforts are being made to develop the test facilities. These efforts are directed at boundary layer control on the tunnel walls, and further investigation of the semi-open top as mentioned above. The development of this

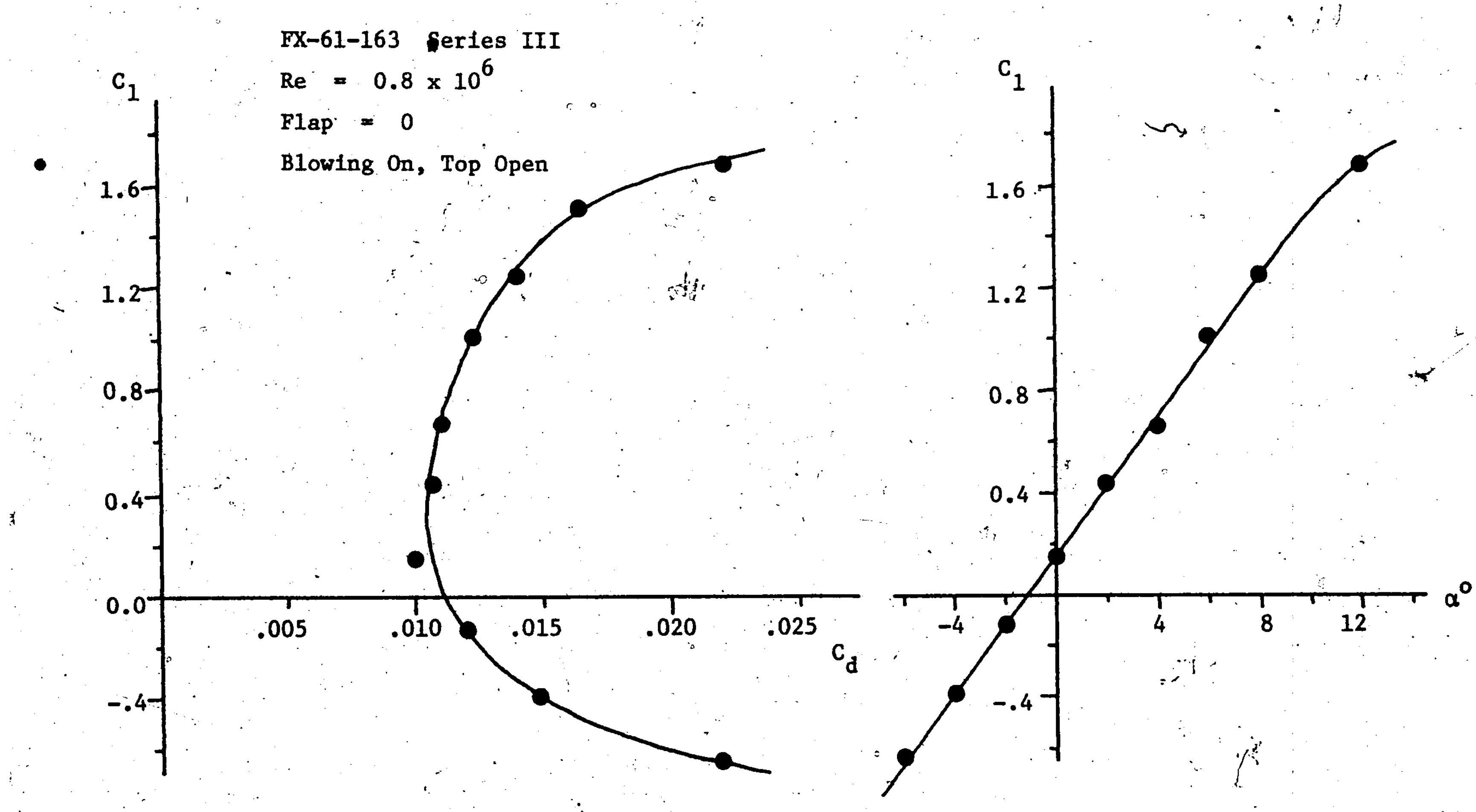


Figure 14. Results from Series III

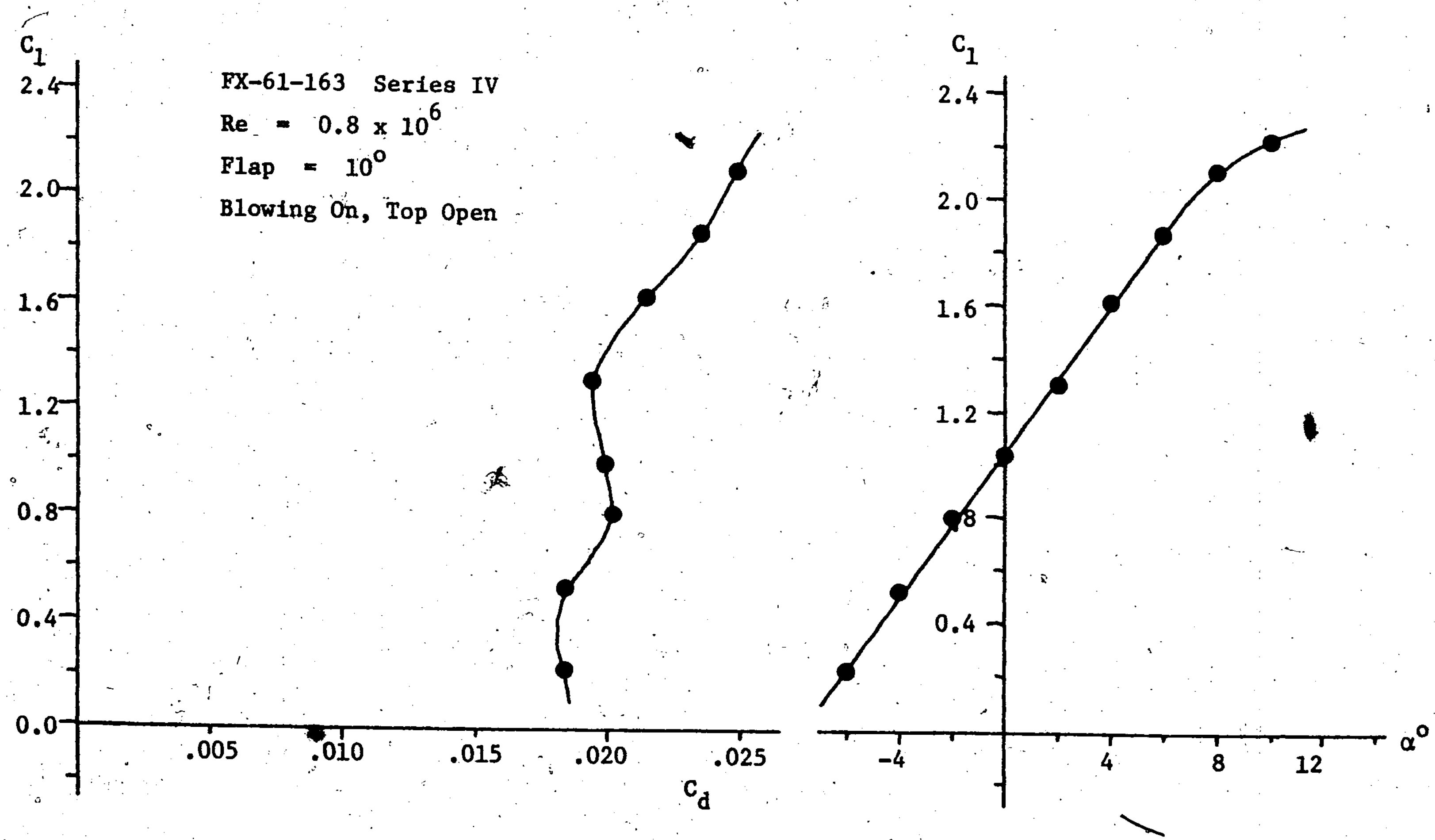


Figure 15. Results from Series IV

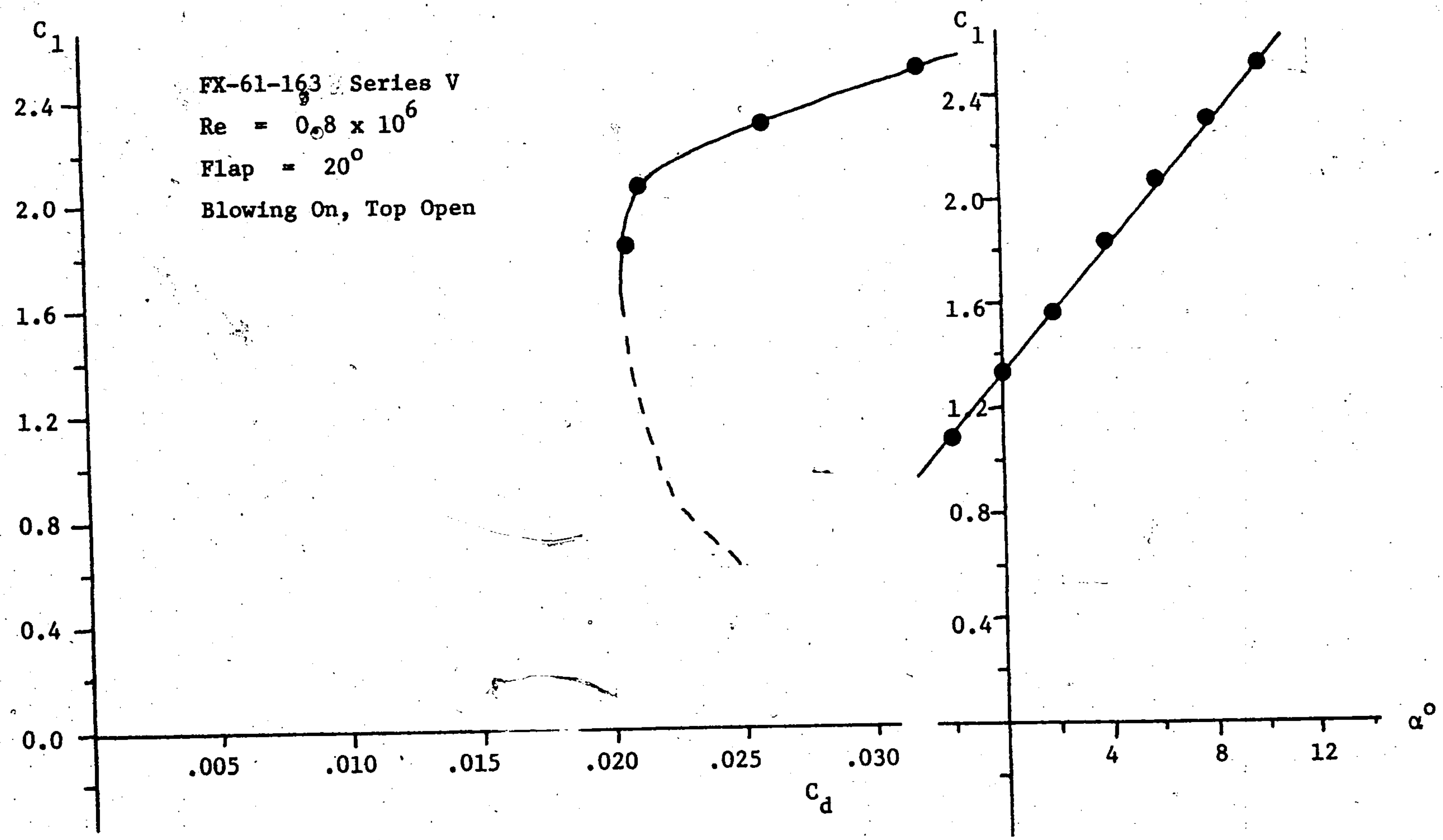


Figure 16. Results from Series V

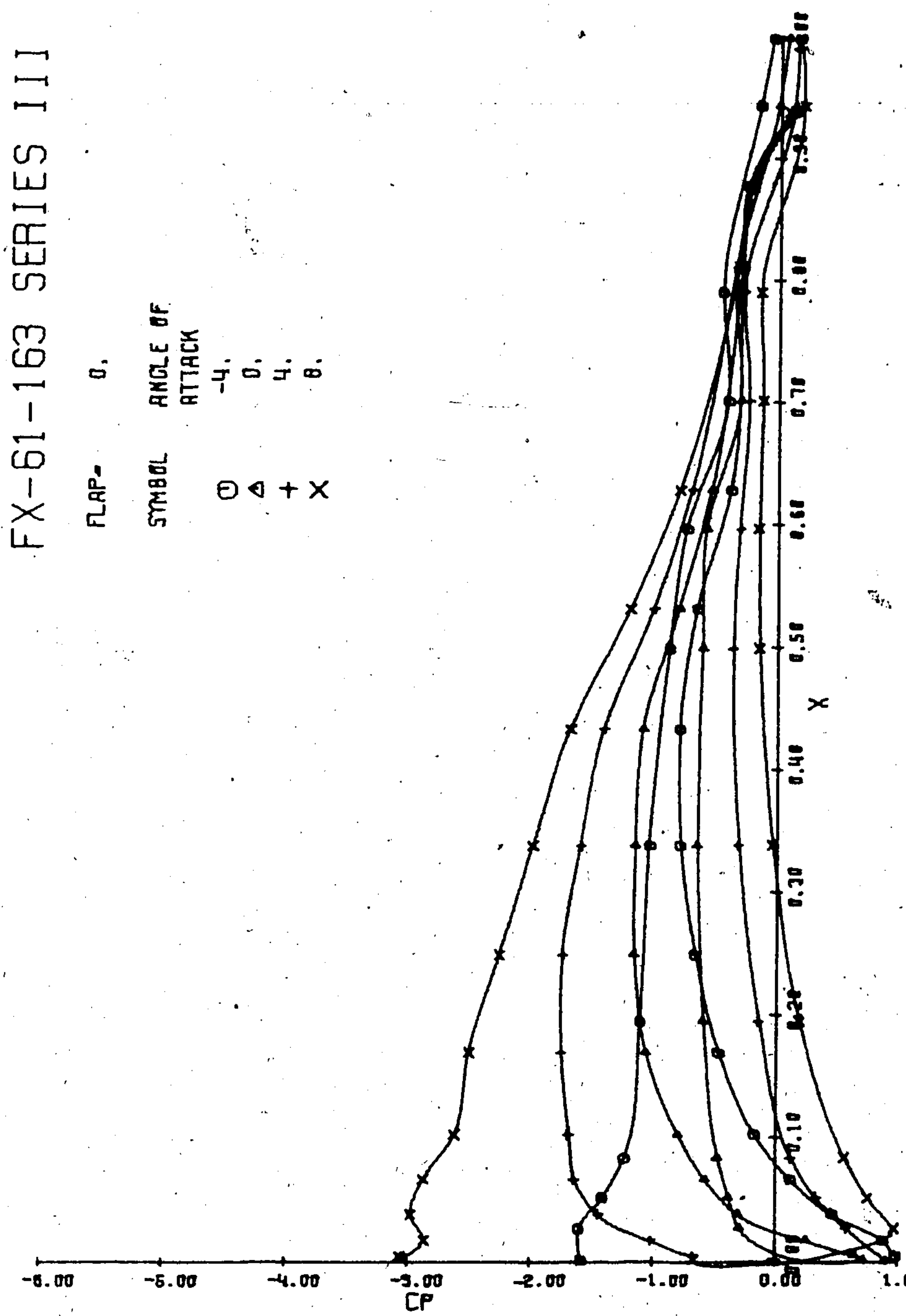


Figure 17. Series III Pressure Distributions

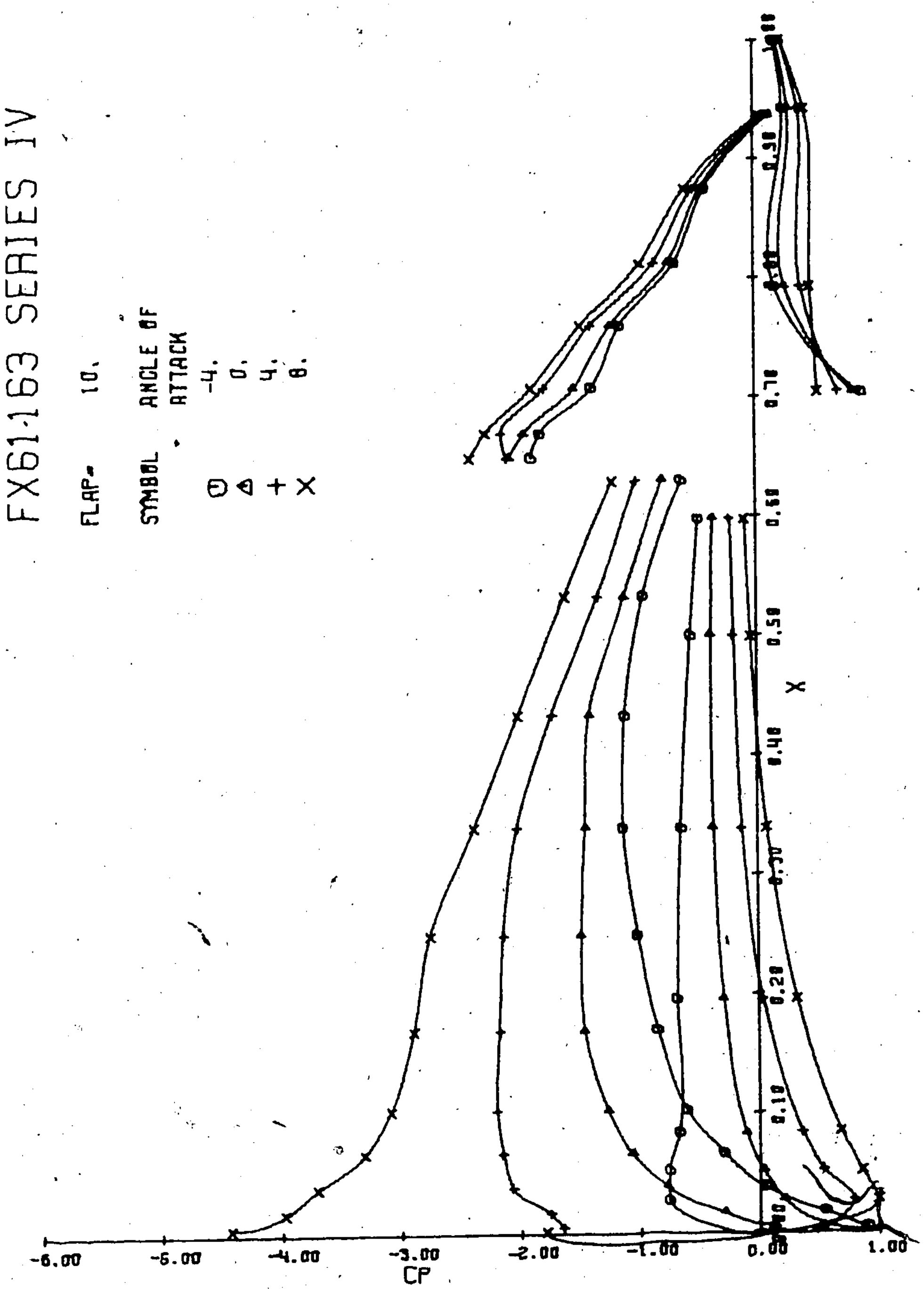


Figure 18. Series IV Pressure Distributions

facility should make it possible to investigate the airfoil performance at high angles of attack, going beyond the stalling point.

4.3 DISCUSSION OF RESULTS

It should be noted, referring to Figure 19, that the pressure distributions were measured and plotted separately for the upper and lower surfaces and do not join. Since the pressure taps did not extend to the trailing edge of the main airfoil, values of pressure coefficient had to be extrapolated in this region for the calculation of lift. This was done in the program SCANLIFT, to approximately 80 percent chord. From the appearance of the plots, this should cause no great problem as the pressure recovery in this region is quite smooth. A further point of interest is that, on the flap, the exact location of the stagnation point can only be inferred from the data given. In fact, the computer-drawn plots may be misleading in some cases, as the plotter only connects the input data points, and cannot infer the location of such strategic points as the stagnation point. This is a minor disadvantage of a completely automated system, that is, the system can handle the general case flawlessly, while errors may arise in treatment of some specific cases. This disadvantage can only be overcome by providing more data points to complete the curves (more taps on the model), by very sophisticated programming, or by manual editing of results.

Despite the fact that the results obtained from these two test configurations were somewhat disappointing, the data acquisition system was able to show its value in this type of experimental work. With minor modifications to the test facilities, it should be possible to accurately measure the airfoil performance characteristics.

Through the use of the data acquisition system as originally conceived, and an appreciation of the difficulties involved in the test procedures, several possible improvements to the system arise. None of these are major, and do not change the central idea of the system.

Firstly, it would be possible to decrease the gate-open time of the multiplexer and at the same time increase the voltage offset of the transducer signals. The former would speed up the recording process while the latter would increase the frequency output of the oscillator, thereby maintaining the accuracy. The real time computer used to count the cycles can operate much faster than the range where it was used. However, for flight testing, speeding up the recording of data would increase the effects of instabilities of the carrier, as the analog signal integration period would be much shorter.

Secondly, the method of measuring the angle of attack in free flight tests must be changed. It should be monitored continually through the test. This involves the addition of another transducer and the rearrangement of sequencing in the multiplexer. Also, since temperature is the most stable parameter, it need be monitored only once for each test condition.

Since one of the ideas of free flight testing is to simulate atmospheric turbulence exactly, it should be measured by placing an RMS meter on the anemometer output, to give a DC signal for recording. The anemometer must also be temperature compensated. However, in view of the difficulties experienced with the anemometer, this would require very special care.

The process of data transfer to magnetic tape is slow due to the necessary delays in setting up the cassette at the desired position, entering file label through the teletype and starting the program. This could possibly be improved to be a continuous read-out. However, the problems of file identification and data sorting would be immense. With practice, it was possible to set up the computer in the gaps between data sets and the cassette could be run continuously.

Lastly, the analysis programs could be improved so as to decrease the amount of supervision needed. It would be possible to queue the computer by key words in the file label. In effect, the programs as written would become subroutines under a master operating program. It would also be possible, since the small computer was not operating near capacity, to do the first four stages in the data reduction at the same time as the raw data was being read in and counted. However, the PDP 11/20 does not have the sophisticated subroutine library and input/output management program that is available on the larger IBM 360/67. The program for the PDP 11/20 must also be written in Assembler rather than FORTRAN.

4.4 SUGGESTED IMPROVEMENTS AND FURTHER WORK

From the preceding discussion, the following recommendations can be made to improve the system performance:

1. The transducer input sequence should include an in-sequence measurement of angle of attack for free flight testing.
2. If information on turbulence is desired, an RMS voltmeter should be used on the anemometer output to give a DC voltage signal which would be monitored like any other transducer.

3. The possibility of continuous playback of the cassette should be investigated more fully.
4. The data reduction process on the large computer could be streamlined to simplify the operating instructions.

These changes, which do not drastically alter the makeup of the data acquisition system, would be quite easily implemented and would improve the performance and usefulness of the system.

The data acquisition system has shown its usefulness for this kind of repeated testing. It is therefore suggested that it be used to make a complete study of airfoil performance in the wind tunnel under many different wall and flap configurations, and boundary layer control. It should be possible, then, by comparing experimental results with theory to develop the airfoil testing facility so as to provide completely dependable and accurate results.

CHAPTER V

CONCLUSIONS

The data acquisition system described in this report has shown itself to be a useful and efficient research tool, thereby achieving its objective. Its operation is extremely flexible, allowing fixed or mobile operation, and it is not limited to aerodynamic testing or to the specific transducers used here. Since no manual processing of large masses of data is required, either in recording, transfer to computer punch cards or files, or analysis, a great deal of time is saved and processing errors are eliminated. The system therefore allows much more efficient use of test facilities. The system also combines the use of inexpensive and low powered measurements with the use of the large data processing computer.

The present system was estimated to have an accuracy of ± 1 percent. The conversion from analogue transducer signals to digital representations at an early stage eliminates many sources of error and assures the best possible accuracy consistent with the transducer calibration and stability.

It was shown that, even with automatic data recovery, the difficulties of performing free flight tests are quite large and numerous, though not insurmountable. The wind tunnel tests indicate that more work must be done to accurately assess the effects of the tunnel walls on the results.

Data acquisition systems such as described here will become more prominent in future research. With advances in technology and the growing abundance of minicomputers it will become increasingly possible to perform experimental research quickly and accurately without great expense. Especially, any investigation requiring the repeated monitoring of several data channels should be considered as requiring a data acquisition system such as described in this report.

BIBLIOGRAPHY

1. Nettleton, T. R. "Handling Qualities Research in the Development of STOL Utility Transport Aircraft", Canadian Aeronautics and Space Journal, Volume 12, No. 3 (1966).

2. Whitfield, G. R. "A Simple Data Recording System with Computer Analysis", Radio Electronics Engineer, Volume 40, No. 5 (1970).

3. Althaus, D. "Stuttgarter Profilkatalog I", Institut fur Aerodynamik und Gasdynamik der Universitat Stuttgart (1972).

4. Williams, C. D. "A Theory for Wind Tunnel Wall Corrections", M.Sc. Thesis, University of British Columbia (1973).

APPENDIX A

ELECTRONIC CIRCUIT DIAGRAMS

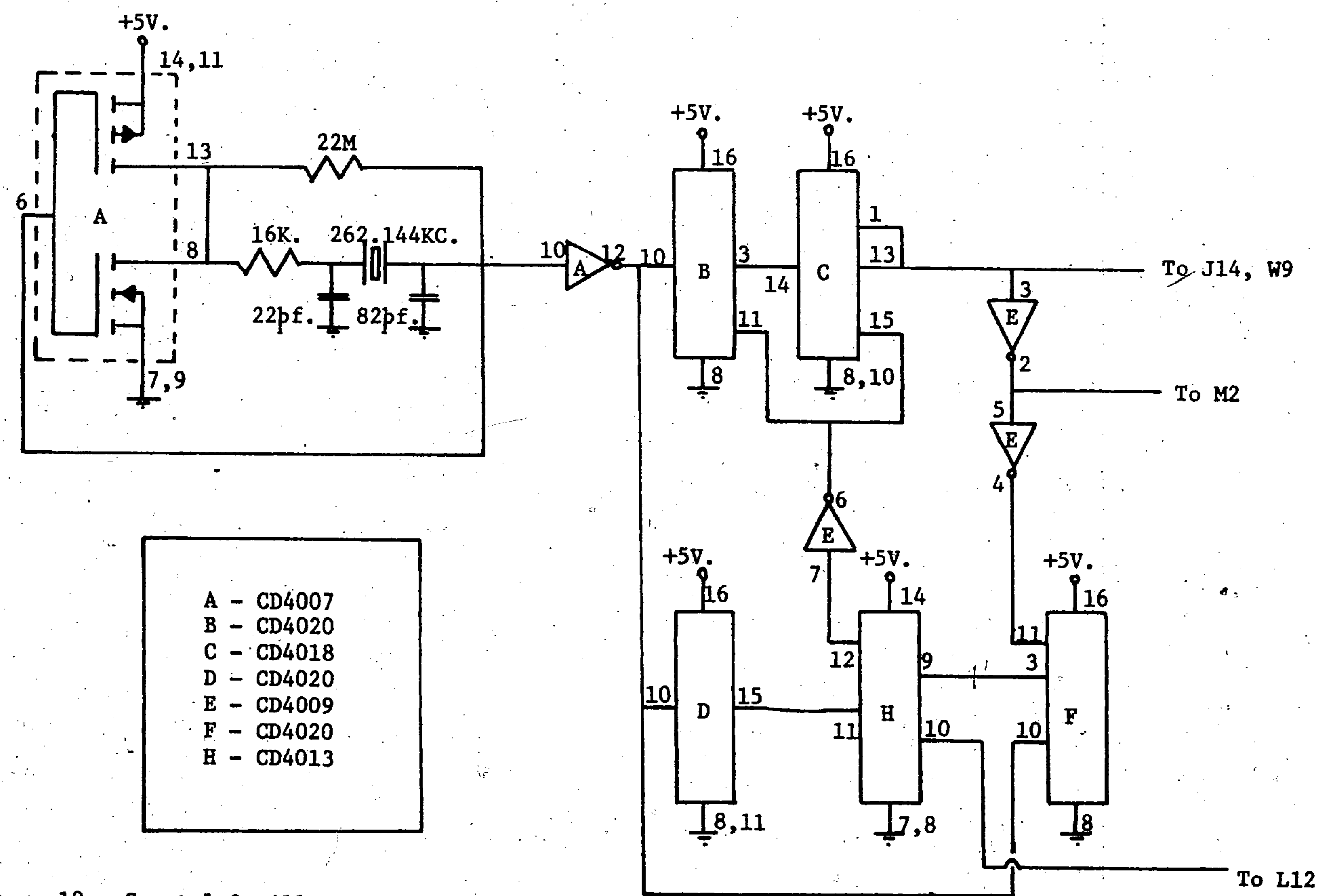


Figure 19. Crystal Oscillator and Shaper

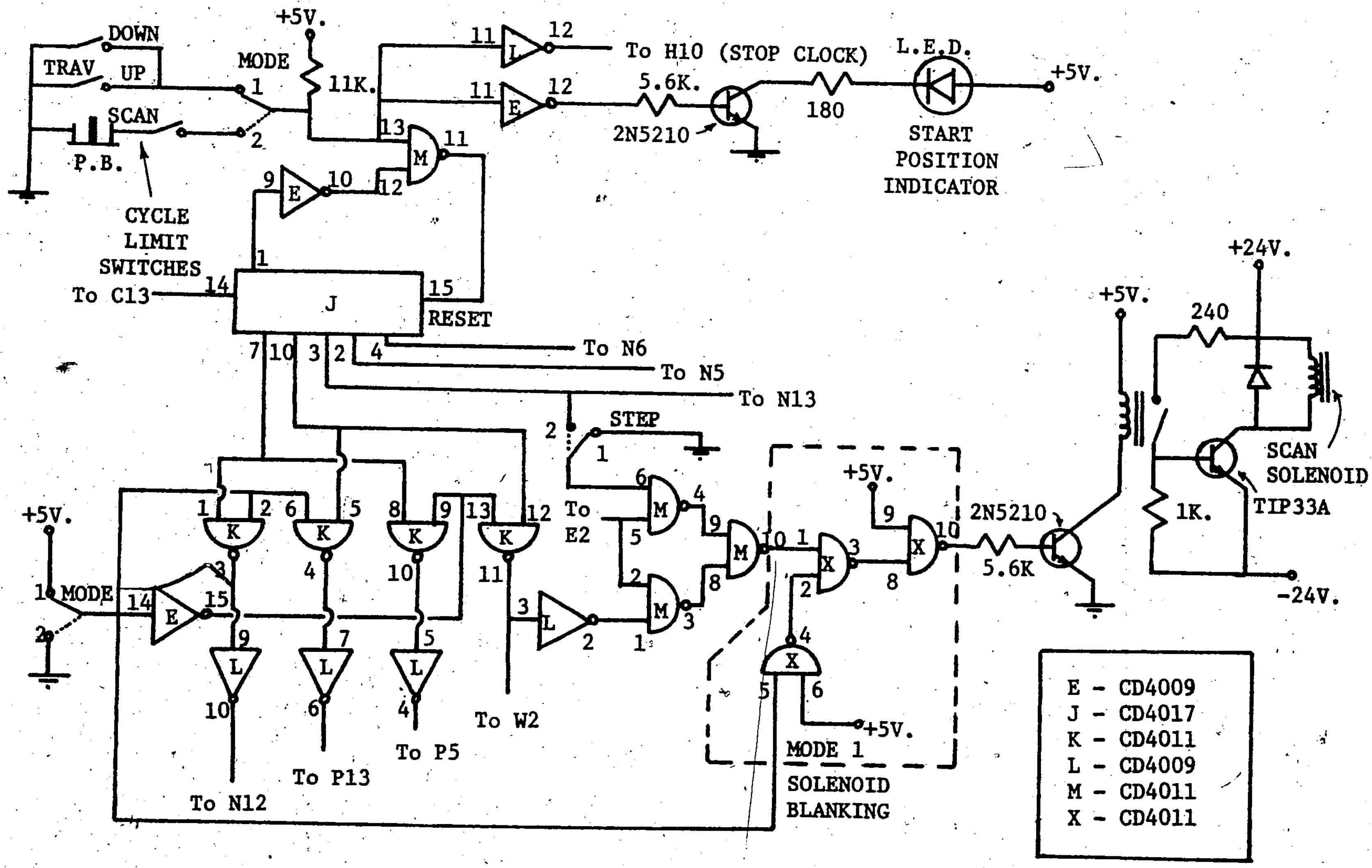


Figure 20. Scanning and Solenoid Stepping Circuit.

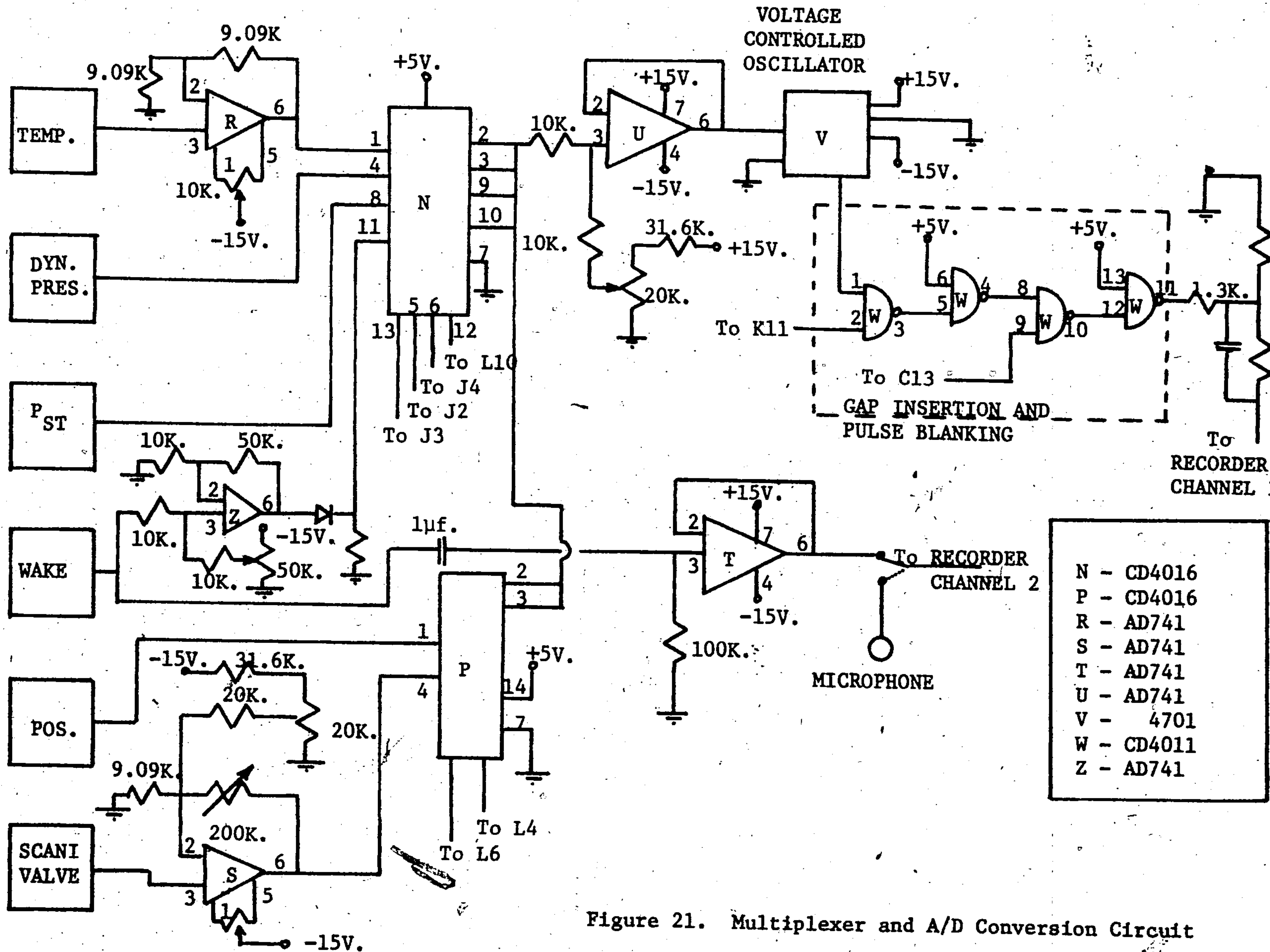


Figure 21. Multiplexer and A/D Conversion Circuit

APPENDIX B

COMPUTER PROGRAMS

COMPUTER PROGRAMS

Following is a list of all system subroutines used in the program package listed in this appendix. These subroutines were available in the Scientific Subroutine Package library (*SSP MATHLIB) or in the Calcomp plotter library (*PLOTLIB) available through MTS. Complete documentation for these subroutines is available in "System/360 Scientific Subroutine Package" published by IBM, in "The Michigan Terminal System Volume 3: System Subroutines" published by the University of Alberta Computing Services, and in "Calcomp Users Manual" also published by the University of Alberta Computing Services.

Name	Function
ALI	- Aitken-Lagrange interpolation at point specified, using ordered vectors produced by ATSG.
ATSG	- sets up ordered vectors of data points for ALI.
AXIS	- plots, labels and dimensions a linear axis.
CONVAE*	- converts characters coded on magnetic tape in ASCII to EBCDIC which is IBM 360/67 internal code.
FLINE	- plots smooth curve through data points.
PLOT	- changes plot origin.
PLOTS	- opens a plot file.
QSF	- integration of equally spaced function values using Simpson's rule.
READ	- reads data blocks from magnetic tape.
SYMBOL	- plots specified character string.

* This subroutine was not available through MTS and was written by L. Lewis, Tech. Services, U. of A.

```
C               PROGRAM NAME: SCANREAD
C
C               OBJECT NAME: SCANREADOB
C
C        THIS PROGRAM OBTAINS DATA FROM MAGNETIC TAPE RECORDS
C        FOR SCAN MODE OPERATION AND REDUCES THE RAW COUNTS TO
C        PHYSICAL QUANTITIES.  THIS IS FOLLOWED BY OUTPUT TO
C        TEMPORARY DISK SPACE FOR USE IN SUBSEQUENT CALCULATIONS
C        AND/OR LISTING.  UNDER 'MTS' TEMPORARY FILES NEED NOT
C        BE 'CREATED' BUT THEIR CONTENTS ARE DESTROYED AFTER
C        'SIGNOFF'.
C
C        TO RUN THIS PROGRAM FROM THE TERMINAL, THE FOLLOWING
C        DEVICES MUST BE SPECIFIED IN THE RUN COMMAND:
C
C           UNIT 0 MAGNETIC TAPE DEVICE (EG. *T*)
C                4 OUTPUT TEMPORARY FILE FOR FURTHER CALCULATIONS
C                5 (DEFAULTS TO *SOURCE*)
C                6 (DEFAULTS TO *SINK*)
C                8 OUTPUT TEMPORARY FILE FOR LISTING OF RESULTS
C                9 INPUT FILE OF X-COORDINATES, FLAP, ETC.
C
C
C        [illegible]NREADOB 0=*T* 4=-S1 8=-L1 9=FX61163
C
C        [illegible]IS PROGRAM *BATCH*, THESE ASSIGNMENTS MUST BE
C        [illegible]TO:
C
C           UNIT 5 DISK FILE OF CALIBRATION PARAMETERS
C                8 *SINK*
C
C        EXAMPLE:
C        $RUN SCANREADOB 0=*T* 4=-S1 5=CALFILE 8=*SINK* 9=FX51163
C
C        AFTER SEVERAL RUNS OF SCANREADOB, OUTPUT DATA ON UNIT 4
C        CAN BE COLLECTED INTO ONE FILE BY USING THE INSTRUCTION:
C
C        $COPY -S1+-S2+...+-S9 TO SCANDATA
C
C        THEN FILE SCANDATA BECOMES INPUT TO SCANPLOT AND
C        SCANLIFT.  HOWEVER, THE LAST LINE IN SCANDATA MUST BE:
C
C         END
C
C        TO OBTAIN A COMPLETE LISTING OF RESULTS:
C
C        $COPY -L1+-L2+...+-L9 TO SCANLIST
C        $LIST SCANLIST
C
C
      INTEGER*2 IN(40),LEN,LAB(80)
      INTEGER NAS(50),NSCAN(50),WORK(200),NAME(5),NPST(50),NTP(50)
      DIMENSION Q(50),SCAN(50),CP(50),X(50),PST(50),TP(50)
C
C      READ IN THE LABEL BLOCK FROM TAPE
```

```
C
      CALL READ(LAB,LEN,0,LNR,0,&100)
C      CALL CONVAE(LAB,LEN)
      WRITE(6,6000)
C      WRITE(6,6001) ,(LAB(I),I=1,LEN)
      WRITE(6,6002)
      READ(5,5000) ALPHA
      WRITE(6,6003) ALPHA
C
C      READ IN THE FIRST DATA BLOCK
C
      CALL READ(IN,LEN,0,LNR,0,&101)
      LEN2=LEN/2
C
C      INITIALIZE SUBSCRIPTS FOR DATA SEARCH
C
      K=1
      DO 3 L=2,LEN2
      WORK(K)=IN(L)
      K=K+1
3     CONTINUE
C
C      READ IN THE REMAINING DATA BLOCKS
C
4     CALL READ(IN,LEN,0,LNR,0,&102)
      LEN2=LEN/2
      DO 5 L=1,LEN2
      WORK(K)=IN(L)
      K=K+1
5     CONTINUE
      GO TO 4
C
102   LEN2=LEN/2
      DO 6 L=1,LEN2
      WORK(K)=IN(L)
      K=K+1
      IF(K.GT.200) GO TO 7
6     CONTINUE
C
C      READ INPUT DATA FROM DISK FILE
C
7     READ(9,9000) NPTS,FLAP,(NAME(I),I=1,5)
      READ(9,9001) NMU,NML,NFU,NFL
      READ(9,9002) (X(I),I=1,NPTS)
C
C      ASSIGN COUNT VECTORS
C
      J=1
      DO 8 I=1,NPTS
      NAS(I)=WORK(J)
      NPST(I)=WORK(J+1)
      NSCAN(I)=WORK(J+2)
      NTP(I)=WORK(J+3)
      J=J+4
8     CONTINUE
```

```
C
C      OUTPUT THE COUNT VECTORS TO DISK FOR LISTING
C
      WRITE(8,8000) (NAME(I),I=1,5)
      WRITE(8,8001) ALPHA,FLAP
      WRITE(8,8002)
      WRITE(8,8003) ((I,NAS(I),NSCAN(I),NPST(I),NTP(I)),I=1,NPTS)
C
C      READ IN CALIBRATION CONSTANTS
C
      WRITE(6,6007)
      READ(5,5001) QSL,QINT
      WRITE(6,6008) QSL,QINT
      WRITE(6,6009)
      READ(5,5001) SSL,SINT
      WRITE(6,6010) SSL,SINT
      WRITE(6,6011)
      READ(5,5001) PSL,PINT
      WRITE(6,6012) PSL,PINT
      WRITE(6,6013)
      READ(5,5001) TSL,TINT
      WRITE(6,6014) TSL,TINT
C
C      CONVERT RAW COUNTS TO PHYSICAL UNITS
C
      DO 9 I=1,NPTS
      Q(I)=QSL*NAS(I)+QINT
      PST(I)=PSL*NPST(I)+PINT
      TP(I)=TSL*NTP(I)+TINT
      SCAN(I)=SSL*NSCAN(I)+SINT
      CP(I)=1.0+SCAN(I)/Q(I)
9     CONTINUE
C
C      COPY VECTORS TO DISK FOR LATER LISTING
C
      WRITE(8,8004)
      WRITE(8,8005) ((I,Q(I),SCAN(I),CP(I),PST(I),TP(I)),I=1,NPTS)
      WRITE(8,8006)
C
C      OUTPUT RESULTS TO DISK FOR FURTHER CALCULATIONS
C
      WRITE(4,4000) (NAME(I),I=1,5)
      WRITE(4,4001) ALPHA,FLAP,NPTS,NMU,NML,NFU,NFL
      WRITE(4,4002) (X(I),I=1,NPTS)
      WRITE(4,4003) (CP(I),I=1,NPTS)
      WRITE(6,6004)
      GO TO 999
C
C
4000 FORMAT(5A4)
4001 FORMAT(2F4.0,5I3)
4002 FORMAT(16F5.4)
4003 FORMAT(10F8.3)
5000 FORMAT(F4.0)
5001 FORMAT(2F8.3)
```

```
6000 FORMAT(/'FILE LABEL:')
6001 FORMAT(/40A2)
6002 FORMAT(/'ENTER ALPHA:')
6003 FORMAT(/'CONTINUING WITH ALPHA=',F6.0)
6004 FORMAT(/'DATA COPIED TO DISK FOR FURTHER CALCULATIONS')
6005 FORMAT(/'END-OF-FILE ENCOUNTERED IN LABEL BLOCK')
6006 FORMAT(/'END-OF-FILE ENCOUNTERED.IN FIRST DATA BLOCK')
6007 FORMAT(/'ENTER Q CALIBRATION SLOPE AND INTERCEPT:')
6008 FORMAT('CALIBRATION IS:  Q=',F8.5,'*NAS+',F5.2)
6009 FORMAT(/'ENTER SCAN CALIBRATION SLOPE AND INTERCEPT:')
6010 FORMAT('CALIBRATION IS:  SCAN=',F8.5,'*NSCAN+',F5.2)
6011 FORMAT(/'ENTER PST CALIBRATION SLOPE AND INTERCEPT:')
6012 FORMAT('CALIBRATION IS:  PST=',F8.4,'*NPST+',F6.1)
6013 FORMAT(/'ENTER TP CALIBRATION SLOPE AND INTERCEPT:')
6014 FORMAT('CALIBRATION IS:  TP=',F8.4,'*NTP+',F5.1)
8000 FORMAT(//'DATA FOR TEST OF',2X,5A4)
8001 FORMAT(/'TEST CONDITIONS: ALPHA=',F6.0,'  FLAP=',F6.0)
8002 FORMAT(/'RAW COUNTS'/T3,1HI,T8,3HNAS,T18,5HNSCAN,T29,
    14HNPST,T40,3HNTP)
8003 FORMAT(I3,4I10)
8004 FORMAT(/'PHYSICAL QUANTITIES*'/4HHOLE,T10,1HQ,T20,4HSCAN,
    1T30,2HCP,T38,3HPST,T48,1HT)
8005 FORMAT(I3,3X,F7.3,3X,F7.3,3X,F7.2,3X,F7.2,4X,F4.1)
8006 FORMAT(/'* QUANTITIES NOT REQUIRED ARE MASKED ******')
9000 FORMAT(I3,F4.0,5A4)
9001 FORMAT(4I4)
9002 FORMAT(16F5.4)
C
C     SPECIAL EXITS
C
100  WRITE(6,6005)
     GO TO 999
101  WRITE(6,6006)
C
C
999  STOP
     END
C
```

```
C
C            PROGRAM NAME: SCANPLOT
C
C            OBJECT NAME: SCANPLOTOB
C
C     THIS PROGRAM READS DATA WRITTEN TO DISK BY SCANREAD AND
C     PLOTS THE PRESSURE DISTRIBUTION FOR EACH SET OF DATA.
C     NOTE: TERMINATION OCCURS WHEN ' END' IS ENCOUNTERED IN
C     THE DATA FILE ON UNIT 5 (THE INPUT DATA) IN DATA LINE
C     ONE.  I/O UNITS SPECIFIED IN THE RUN COMMAND ARE:
C
C        UNIT 5 FILE OF INPUT DATA
C             6 (DEFAULTS TO *SINK*) OUTPUT LIST
C             9 TEMPORARY DISK FILE TO BE USED WITH *CALCOMPQ
C
C     THE PLOTTER SUBROUTINE LIBRARY MUST BE USED.
C
C     A TYPICAL RUN COMMAND WOULD BE:
C
C     $RUN SCANPLOTOB+*PLOTLIB 5=SCANDATA 9=-TEMP
C
C     IF RC=0 ON RETURN FROM SCANPLOTOB, THE PLOTTING IS
C     COMPLETED BY:
C
C     $RUN *CALCOMPQ SCARDS=-TEMP
C
      DIMENSION WORK(2048),X(50),CP(50)
      INTEGER END/' END'/,IN/'   0.'/,ALPHA,FLAP,NAME(5)
      CALL PLOTS(WORK,8192)
      CALL PLOT(2.0,2.0,-3)
      COUNT=1.0
      FACT=0.667
C
C     'FACTOR' SUBROUTINE SCALES THE PHYSICAL SIZE OF THE
C     RESULTING PRESSURE DISTRIBUTION PLOTS.  FACT=1 WILL
C     PRODUCE A PLOT 10 INCHES BY 6 INCHES. THE MAXIMUM
C     ALLOWABLE SIZE FOR THIS PROGRAM.
C
      CALL FACTOR(FACT)
C
C     READ INPUT DATA
C
1     READ(5,5000) (NAME(I),I=1,5)
      IF(NAME(1).EQ.END) GO TO 999
      READ(5,5001) ALPHA,FLAP,NPTS,NMU,NML,NFU,NFL
      WRITE(6,6000) ALPHA
      READ(5,5002) (X(I),I=1,NPTS)
      READ(5,5003) (CP(I),I=1,NPTS)
C
C     PLOTTING OF AXES AND LABELS
C
      CALL AXIS(0.0,0.0,2HCP,-2,6.0,0.0,-5.0,1.0,10.0)
      CALL AXIS(5.0,0.0,1HX,-1,10.0,90.0,0.0,0.1,10.0)
      CALL PLOT(5.0,0.0,-3)
      DO 2 I=1,5
      Y=FLOAT(I)/1.2
```

```
      CALL SYMBOL(-5.C,Y,C.250,NAME(I),90.0,4)
2     CONTINUE
      CALL SYMBOL(-4.5,1.0,0.125,6HALPHA=,90.0,6)
      CALL SYMBOL(-4.5,2.C,C.125,ALPHA,90.0,4)
      CALL SYMBOL(-4.0,1.0,C.125,5HFLAP=,90.0,5)
      CALL SYMBOL(-4.0,2.0,0.125,FLAP,90.C,4)
C
      IF(FLAP.EQ.IN) GO TO 20
C
C      PLOT SUBROUTINE FOR FLAP=OUT
C
      CALL FLOUT(X,CP,NMU,NML,NFU,NFL)
      GO TO 40
C
C      PLOT SUBROUTINE FOR FLAP=IN
C
20    CALL FLIN(X,CP,NMU,NML)
C
C      SET UP PLOT ORIGIN FOR NEXT PLOT
C
40    IF(COUNT.LT.0.0) GO TO 60
      CALL PLOT(-5.0,12.0,-3)
      GO TO 70
60    CALL PLOT(7.0,-12.0,-3)
70    COUNT=-COUNT
      GO TO 1
C
C
5000  FORMAT(5A4)
5001  FORMAT(2A4,5I3)
5002  FORMAT(16F5.4)
5003  FORMAT(10F8.3)
6000  FORMAT(/'PLOTTING FOR ALPHA=',2X,A4)
C
C      PROGRAM EXIT
C
999   IF(COUNT.GT.0.0) GO TO 80
      CALL PLOT(7.0,-12.C,-3)
80    CALL PLOT(0.C,0.0,999)
      STOP
      END
C
C
C
C
      SUBROUTINE FLIN(X,CP,NMU,NML)
C      THIS SUBROUTINE PLOTS FOR FLAP=IN CASE
      DIMENSION X(50),CP(50),XGR(30),CPGR(30)
C
C      UPPER SURFACE
C
      DO 1 I=1,NMU
      XGR(I)=X(I)
      CPGR(I)=CP(I)
1     CONTINUE
```

```
      CALL DIST(XGR,CPGR,NMU,1)
C
C      LOWER SURFACE
C
      DO 2 I=1,NML
      J=NMU+I
      XGR(I)=X(J)
      CPGR(I)=CP(J)
2     CONTINUE
      CALL DIST(XGR,CPGR,NML,2)
C
      RETURN
      END
C
C
C
C
      SUBROUTINE FLOUT(X,CP,NMU,NML,NFU,NFL)
C      THIS SUBROUTINE IS FOR FLAP=OUT CASES
      DIMENSION X(50),CP(50),XGR(20),CPGR(20)
C
C      MAIN UPPER SURFACE
C
      DO 1 I=1,NMU
      XGR(I)=X(I)
      CPGR(I)=CP(I)
1     CONTINUE
      CALL DIST(XGR,CPGR,NMU,1)
C
C      MAIN LOWER SURFACE
C
      DO 2 I=1,NML
      J=NMU+I
      XGR(I)=X(J)
      CPGR(I)=CP(J)
2     CONTINUE
      CALL DIST(XGR,CPGR,NML,2)
C
C      FLAP UPPER SURFACE
      DO 3 I=1,NFU
      J=NMU+NML+I
      XGR(I)=X(J)
      CPGR(I)=CP(J)
3     CONTINUE
      CALL DIST(XGR,CPGR,NFU,1)
C
C      FLAP LOWER SURFACE
      DO 4 I=1,NFL
      J=NMU+NML+NFU+I
      XGR(I)=X(J)
      CPGR(I)=CP(J)
4     CONTINUE
      CALL DIST(XGR,CPGR,NFL,2)
C
      RETURN
```

```
      END
C
C
C
C
      SUBROUTINE DIST(X,CP,N,K)
      DIMENSION X(30),CP(30)
      X(N+1)=0.0
      X(N+2)=0.1
      CP(N+1)=0.0
      CP(N+2)=1.0
      CALL FLINE(CP,X,-N,1,1,K)
      RETURN
      END
C
```

```
C                 PROGRAM NAME: SCANLIFT
C
C                 OBJECT NAME: SCANLIFTOB
C
C      THIS PROGRAM USES THE DATA PRODUCED BY SCANREAD TO
C      CALCULATE THE LIFT AND MOMENT COEFFICIENTS.  THE
C      INPUT DATA IS THE SAME AS THAT FOR SCANPLOT.  THE
C      PROGRAM MAY BE RUN THROUGH *BATCH* OR THROUGH THE
C      TERMINAL.
C
C      THE PROGRAM REQUIRES SUBROUTINES FROM *SSPMATHLIB.
C
C      A TYPICAL RUN COMMAND WOULD BE:
C
C      $RUN SCANLIFTOB+*SSPMATHLIB 5=SCANDATA
C
C
      INTEGER END/' END'/,IN/'  0.'/,NAME(5)
      DIMENSION X(50),CP(50)
C
C      READ DATA DEPOSITED BY SCANREAD
C
1     READ(5,5000) (NAME(I),I=1,5)
      IF(NAME(1).EQ.END) GO TO 999
      READ(5,5001) ALPHA,FLAP,NPTS,NMU,NML,NFU,NFL
      READ(5,5002) (X(I),I=1,NPTS)
      READ(5,5003) (CP(I),I=1,NPTS)
      WRITE(6,6000) (NAME(I),I=1,5)
      WRITE(6,6001) ALPHA,FLAP
      IF(FLAP.NE.0.0) GO TO 2
C
C      INTEGRATION SUBROUTINE FOR FLAP=IN CASE
C
      CALL FLIN(X,CP,NPTS,NMU,NML,CN,CMLE)
      GO TO 3
C
C      INTEGRATION SUBROUTINE FOR FLAP=OUT CASE
C
2     CALL FLOUT(X,CP,NPTS,NMU,NML,NFU,NFL,CN,CMLE)
C
C      CALCULATE CL AND CM1/4
C
3     CL=CN*COS(ALPHA*3.1416/180.)
      CMQ=CMLE+CL/4.
C
C      OUTPUT THE RESULTS
C
      WRITE(6,6002) CN,CMLE
      WRITE(6,6003) CL,CMQ
      GO TO 1
C
C
5000 FORMAT(5A4)
5001 FORMAT(2F4.0,5I3)
5002 FORMAT(16F5.4)
```

```
5003 FORMAT(10F8.3)
6000 FORMAT(//'TEST OF',2X,5A4)
6001 FORMAT('ALPHA=',F6.0,2X,'FLAP=',F6.0)
6002 FORMAT('CN=',F7.2/'CMLE=',F7.3)
6003 FORMAT('CL=',F7.2/'CM1/4=',F7.3)
C
C
999   STOP
      END
C
C
C
      SUBROUTINE FLIN(X,CP,NPTS,NMU,NML,CN,CMLE)
      DIMENSION X(50),CP(50),XI(51),CPI(51),WORK(100),ARG(8),
     1VAL(8),XU(25),CPU(25),XL(25),CPL(25),Z(51)
      EPS=1.0E-2
      H=0.02
C
C      SET UP AUXILIARY VECTORS
C
      DO 1 I=1,51
1     XI(I)=(I-1)*H
      DO 2 I=1,NMU
      XU(I)=X(I)
2     CPU(I)=CP(I)
      DO 3 I=1,NML
      J=NMU+I
      XL(I)=X(J)
3     CPL(I)=CP(J)
C
C      INTERPOLATE CP VECTORS AND INTEGRATE FOR UPPER SURFACE
C
      LOOP=4
      DO 4 I=1,51
      CALL ATSG(XI(I),XU,CPU,WORK,NMU,1,ARG,VAL,8)
      CALL ALI(XI(I),ARG,VAL,CPI(I),8,EPS,IER)
      IF(IER.EQ.0) GO TO 4
      WRITE(6,6000) LOOP,I,IER
4     CONTINUE
      CALL QSF(H,CPI,Z,51)
      SUMLUP=Z(51)
      DO 5 I=1,51
5     CPI(I)=CPI(I)*XI(I)
      CALL QSF(H,CPI,Z,51)
      SUMMUP=Z(51)
C
C      DO LIKEWISE FOR LOWER SURFACE
C
      LOOP=6
      DO 6 I=1,51
      CALL ATSG(XI(I),XL,CPL,WORK,NML,1,ARG,VAL,8)
      CALL ALI(XI(I),ARG,VAL,CPI(I),8,EPS,IER)
      IF(IER.EQ.0) GO TO 6
      WRITE(6,6000) LOOP,I,IER
6     CONTINUE
```

```
      CALL QSF(H,CPI,Z,51)
      SUMLLO=Z(51)
      DO 7 I=1,51
7     CPI(I)=CPI(I)*XX(I)
      CALL QSF(H,CPI,Z,51)
      SUMMLO=Z(51)
C
C      COLLECT RESULTS
C
      CN=SUMLLO-SUMLUP
      CMLE=SUMMLO-SUMMUP
C
C
6000 FORMAT(' IN FLIN LOOP',I5,' AT POINT',I4,' IER=',I4)
C
C
      RETURN
      END
C
C
C
      SUBROUTINE FLOUT(X,CP,NPTS,NMU,NML,NFU,NFL,CN,CMLE)
      DIMENSION X(50),CP(50),XI(40),CPI(40),WORK(100),ARG(8),
     1VAL(8),XMU(30),XML(30),XFU(20),XFL(20),CPMU(30),CPML(30),
     2CPFU(20),CPFL(20),Z(50)
      EPS=1.0E-2
      H=0.02
C
C      MAIN SURFACE
C
      DO 1 I=1,32
1     XI(I)=(I-1)*H
      DO 2 I=1,NMU
      XMU(I)=X(I)
2     CPMU(I)=CP(I)
      DO 3 I=1,NML
      J=NMU+I
      XML(I)=X(J)
3     CPML(I)=CP(J)
C
C      INTERPOLATE CP VECTORS AND INTEGRATE
C
      LOOP=4
      DO 4 I=1,32
      CALL ATSG(XI(I),XMU,CPMU,WORK,NMU,1,ARG,VAL,8)
      CALL ALI(XI(I),ARG,VAL,CPI(I),8,EPS,IER)
      IF(IER.EQ.0) GO TO 4
      WRITE(6,6000) LOOP,I,IER
4     CONTINUE
      CALL QSF(H,CPI,Z,32)
      SUMMUP=Z(32)
      DO 5 I=1,32
5     CPI(I)=CPI(I)*XI(I)
      CALL QSF(H,CPI,Z,32)
      ADDMUP=Z(32)
```

```
      LCOP=6
      DO 6 I=1,32
      CALL ATSG(XI(I),XML,CPML,WORK,NML,1,ARG,VAL,8)
      CALL ALI(XI(I),ARG,VAL,CPI(I),8,EPS,IER)
      IF(IER.EQ.0) GO TO 6
      WRITE(6,6000) LCOP,I,IER
 6    CONTINUE
      CALL QSF(H,CPI,Z,32)
      SUMMLO=Z(32)
      DO 7 I=1,32
 7    CPI(I)=CPI(I)*XI(I)
      CALL QSF(H,CPI,Z,32)
      ADDMLO=Z(32)
C
C      FLAP
C
      EPS=0.10
      DO 8 I=1,NFU
      J=NMU+NML+I
      XFU(I)=X(J)
 8    CPFU(I)=CP(J)
      DO 9 I=1,NFL
      J=NMU+NML+NFU+I
      XFL(I)=X(J)
 9    CPFL(I)=CP(J)
      DO 10 I=1,17
 10   XI(I)=(I-1)*H+0.65
C
C      INTERPOLATE CP VECTORS AND INTEGRATE
C
      LOOP=11
      DO 11 I=1,17
      CALL ATSG(XI(I),XFU,CPFU,WORK,NFU,1,ARG,VAL,NFU)
      CALL ALI(XI(I),ARG,VAL,CPI(I),NFU,EPS,IER)
      IF(IER.EQ.0) GO TO 11
      WRITE(6,6000) LOOP,I,IER
 11   CONTINUE
      CALL QSF(H,CPI,Z,17)
      SUMFUP=Z(17)
      DO 12 I=1,17
 12   CPI(I)=CPI(I)*XI(I)
      CALL QSF(H,CPI,Z,17)
      ADDFUP=Z(17)
      LOOP=13
      DO 13 I=1,17
      CALL ATSG(XI(I),XFL,CPFL,WORK,NFL,1,ARG,VAL,NFL)
      CALL ALI(XI(I),ARG,VAL,CPI(I),NFL,EPS,IER)
      IF(IER.EQ.0) GO TO 13
      WRITE(6,6000) LOOP,I,IER
 13   CONTINUE
      CALL QSF(H,CPI,Z,17)
      SUMFLO=Z(17)
      DO 14 I=1,17
 14   CPI(I)=CPI(I)*XI(I)
      CALL QSF(H,CPI,Z,17)
```

```
      ADDFLO=Z(17)
C
C      COLLECT RESULTS
C
      CN=SUMMLO-SUMMUP+SUMFLO-SUMFUP
      CMLE=ADDMLO-ADDMUP+ADDFLO-ADDFUP
C
6000  FORMAT('IN FLOUT LOOP',I4,2X,'AT POINT',I5,' IER=',I5)
C
      RETURN
      END
C
C
```

```
C              PROGRAM NAME: TRAVREAD
C
C              OBJECT NAME: TRAVREADOB
C
C     THIS PROGRAM READS MAGNETIC TAPE RECORDS FOR TRAV MODE
C     OPERATION AND REDUCES THE RAW COUNTS TO PHYSICAL UNITS.
C     THIS IS FOLLOWED BY OUTPUT TO TEMPORARY DISK SPACE FOR
C     USE IN SUBSEQUENT CALCULATIONS.  THE PROGRAM MAY BE RUN
C     FROM THE TERMINAL.  HOWEVER, TO MINIMIZE I/O THROUGH THE
C     TERMINAL, RESULTS ARE COPIED TO TEMPORARY DISK SPACE FOR
C     LATER LISTING ON THE LINE PRINTER.  THE 'RUN' COMMAND
C     MUST CONTAIN REFERENCE TO THE FOLLOWING UNITS:
C        UNIT 0 MAGNETIC TAPE (EG *T*)
C             4 OUTPUT DISK FILE FOR SUBSEQUENT CALCULATIONS
C             5 (DEFAULTS TO *SOURCE*)
C             6 (DEFAULTS TO *SINK*)
C             8 OUTPUT DISK FILE FOR LISTING OF RESULTS
C             9 INPUT DATA FILE (SAME AS FOR SCANREAD)
C
C     A TYPICAL RUN COMMAND FROM THE TERMINAL WOULD BE:
C
C     $RUN TRAVREADOB 0=*T* 4=-T1 8=-L1 9=INDATA
C
C     A TYPICAL RUN COMMAND THROUGH *BATCH* WOULD BE:
C
C     $RUN TRAVREADOB 0=*T* 4=-T1 5=CALFILE 8=*SINK* 9=FX61163
C
C     AFTER SEVERAL RUNS OF TRAVREADOB, DATA OUTPUT ON UNIT 4
C     MAY BE COLLECTED INTO ONE DISK FILE FOR TRAVDRAG USING:
C
C     $COPY -T1+-T2+...+-T9 TO TRAVDATA
C
C     THEN, TRAVDATA BECOMES INTPUT DATA FOR TRAVDRAG.  THE
C     LAST LINE IN DRAGDATA MUST BE ' END' TO TERMINATE THE
C     EXECUTION OF TRAVDRAG PROPERLY.
C
C     A COMPLETE LISTING OF RESULTS MAY BE OBTAINED BY
C
C     $COPY -L1+-L2+...+-L9 TO TRAVLIST
C     $LIST TRAVLIST
C
C
C
      INTEGER*2 IN(40),LEN,LAB(80)
      INTEGER NAS(50),NWAKE(50),NPOS(50),WORK(200),NAME(5),
     1NPST(50),NTP(50)
      DIMENSION Q(50),WAKE(50),POS(50),PST(50),TP(50)
C
C     READ THE LABEL BLOCK
C
      CALL READ(LAB,LEN,0,LNR,0,&100)
C     CALL CONVAE(LAB,LEN)
      WRITE(6,6000)
C     WRITE(6,6001) (LAB(I),I=1,LEN)
      WRITE(6,6002)
```

```
      READ(5,5000) ALPHA
      WRITE(6,6003) ALPHA
C
C      READ IN THE FIRST DATA BLOCK AND PERFORM DATA SEARCH
C
      CALL READ(IN,LEN,0,LNR,C,&101)
      LEN2=LEN/2
      K=1
      DO 3 L=2,LEN2
      WORK(K)=IN(L)
      K=K+1
3     CONTINUE
C
C      READ IN THE REMAINING DATA BLOCKS
C
4     CALL READ(IN,LEN,0,LNR,0,&6)
      LEN2=LEN/2
      DO 5 L=1,LEN2
      WORK(K)=IN(L)
      K=K+1
      IF(K.GT.200) GO TO 8
5     CONTINUE
      GO TO 4
C
C
6     LEN2=LEN/2
      DO 7 L=1,LEN2
      WORK(K)=IN(L)
      K=K+1
      IF(K.GT.200) GO TO 8
7     CONTINUE
C
C      READ INPUT DATA FROM DISK
C
8     READ(9,9000) NPTS,FLAP,(NAME(I),I=1,5)
C
C      ASSIGN COUNTS TO WORKING VECTORS
C
      K=K/5
      J=1
      DO 9 I=1,K
      NAS(I)=WORK(J)
      NPST(I)=WORK(J+1)
      NWAKE(I)=WORK(J+2)
      NPOS(I)=WORK(J+3)
      NTP(I)=WORK(J+4)
      J=J+5
9     CONTINUE
C
C      OUTPUT COUNT VECTORS TO DISK
C
      WRITE(8,8000) (NAME(I),I=1,5)
      WRITE(8,8001) ALPHA,FLAP
      WRITE(8,8002)
      WRITE(8,8003) ((I,NAS(I),NWAKE(I),NPOS(I),NPST(I),NTP(I)),I=1,
```

```
C
C      READ IN CALIBRATION CONSTANTS
C
      WRITE(6,6007)
      READ(5,5001) QSL,QINT
      WRITE(6,6008) QSL,QINT
      WRITE(6,6009)
      READ(5,5001) WSL,WINT
      WRITE(6,6010) WSL,WINT
      WRITE(6,6011)
      READ(5,5001) PSL,PINT
      WRITE(6,6012) PSL,PINT
      WRITE(6,6013)
      READ(5,5001) PSTSL,PSTINT
      WRITE(6,6014) PSTSL,PSTINT
      WRITE(6,6015)
      READ(5,5001) TSL,TINT
      WRITE(6,6016) TSL,TINT
C
C -----CONVERT COUNTS TO PHYSICAL UNITS
C
      DO 10 I=1,K
      Q(I)=QSL*NAS(I)+QINT
      WAKE(I)=WSL*NWAKE(I)+WINT
      POS(I)=PSL*NPOS(I)+PINT
      PST(I)=PSTSL*NPST(I)+PSTINT
      TP(I)=TSL*NTP(I)+TINT
10    CONTINUE
C
C      COPY RESULTS TO DISK FOR LATER LISTING
C
      WRITE(8,8004)
      WRITE(8,8005) ((I,Q(I),WAKE(I),POS(I),PST(I),TP(I)),I=1,K)
      WRITE(8,8006)
C
C      OUTPUT RESULTS FOR FURTHER CALCULATIONS
C
      WRITE(4,4000) (NAME(I),I=1,5)
      WRITE(4,4001) ALPHA,FLAP,K
      WRITE(4,4002) (WAKE(I),I=1,K)
      WRITE(4,4003) (POS(I),I=1,K)
      WRITE(4,4002) (Q(I),I=1,K)
      WRITE(6,6004)
      GO TO 999
C
C
4000  FORMAT(5A4)
4001  FORMAT(2F4.0,I3)
4002  FORMAT(10F8.4)
4003  FORMAT(16F5.3)
5000  FORMAT(F4.0)
5001  FORMAT(2F8.3)
6000  FORMAT(/'FILE LABEL:')
6001  FORMAT(/40A2)
6002  FORMAT(/'ENTER ALPHA')
```

```
6003 FORMAT(/'CONTINUING WITH ALPHA=',F6.0)
6004 FORMAT(/'DATA COPIED TO DISK FOR FURTHER CALCULATIONS.')
6005 FORMAT(/'END-OF-FILE ENCOUNTERED IN LABEL BLOCK.')
6006 FORMAT(/'END-OF-FILE ENCOUNTERED IN FIRST DATA BLOCK.')
6007 FORMAT(/'ENTER Q CALIBRATION SLOPE AND INTERCEPT:')
6008 FORMAT('CALIBRATION IS:  Q=',F8.5,'*NAS+',F5.2)
6009 FORMAT(/'ENTER WAKE CALIBRATION SLOPE AND INTERCEPT:')
6010 FORMAT('CALIBRATION IS:  W=',F8.5,'*NWAKE+',F5.2)
6011 FORMAT(/'ENTER POS CALIBRATION SLOPE AND INTERCEPT:')
6012 FORMAT('CALIBRATION IS:  POS=',F8.5,'*NPOS+',F5.2)
6013 FORMAT(/'ENTER PST CALIBRATION SLOPE AND INTERCEPT:')
6014 FORMAT('CALIBRATION IS:  PST=',F8.4,'*NPST+',F6.1)
6015 FORMAT(/'ENTER TP CALIBRATION SLOPE AND INTERCEPT:')
6016 FORMAT('CALIBRATION IS:  TP=',F7.2,'*NTP+',F6.2)
8000 FORMAT(//'DATA FOR TEST OF',2X,5A4)
8001 FORMAT(/'TEST CONDITIONS:  ALPHA=',F6.0,'  FLAP=',F6.0)
8002 FORMAT(/'RAW COUNTS'/T3,1HI,T10,3HNAS,T19,5HNWAKE,T30,
    14HNPOS,T40,4HNPST,T50,3HNTP)
8003 FORMAT(I3,5I10)
8004 FORMAT(/'PHYSICAL QUANTITIES*'//T3,1HI,T10,1HQ,T20,4HWAKE,
    1T30,3HPOS,T40,3HPST,T50,2HTP)
8005 FORMAT(I4,F10.2,2F10.3,4X,F6.1,4X,F6.1)
8006 FORMAT(/'* QUANTITIES NOT REQUIRED ARE MASKED *****')
9000 FORMAT(I3,F4.0,5A4)
C
C
C      SPECIAL EXITS
C
100   WRITE(6,6005)
      GO TO 999
101   WRITE(6,6006)
C
C      EXIT
C
999   STOP
      END
C
C
```

```
C           PROGRAM NAME: TRAVDRAG
C
C           OBJECT NAME: TRAVDRAGOB
C
C     THIS PROGRAM CALCULATES THE DRAG COEFFICIENT FROM THE
C     WAKE DATA PRODUCED BY TRAVREAD.  IT MAY BE RUN THROUGH
C     THE TERMINAL OR THROUGH *BATCH*.
C
C     IT REQUIRES SEVERAL SUBROUTINES FROM *SSPMATHLIB.  A
C     TYPICAL RUN COMMAND WOULD BE:
C
C     $RUN TRAVDRAGOB+*SSPMATHLIB 5=TRAVDATA
C
C     THE PROGRAM TERMINATES ON READING ' END' IN THE FIRST
C     LINE OF A DATA SET.
C
C
      INTEGER END/' END'/,NAME(5)
      DIMENSION WAKE(50),Q(50),POS(50),D(50),Y(100),P(100),
     1ARG(8),VAL(8),Z(100),POSF(50)
      EPS=1.0E-2
      H=0.005
C
C      SET UP AUXILIARY POSITION VECTOR
C
      DO 1 I=1,100
1     Y(I)=(I-1)*H
C
C      READ DATA DEPOSITED BY TRAVREAD
C
2     READ(5,5000) (NAME(I),I=1,5)
      IF(NAME(1).EQ.END) GO TO 999
      READ(5,5001) ALPHA,FLAP,K
      READ(5,5002) (WAKE(I),I=1,K)
      READ(5,5003) (POS(I),I=1,K)
      READ(5,5004) (Q(I),I=1,K)
C
C      CALCULATE AUXILIARY VECTORS
C
      DO 3 I=1,K
      D(I)=WAKE(I)/Q(I)
3     POSF(I)=POS(I)/12.
C
C      INTERPOLATE VALUES AT 0.005 FOOT INTERVALS AND INTEGRATE
C
      DO 4 I=1,100
      CALL ATSG(Y(I),POSF,D,WORK,K,1,ARG,VAL,8)
      CALL ALI(Y(I),ARG,VAL,P(I),8,EPS,IER)
      IF(IER.EQ.0) GO TO 4
      WRITE(6,6000) I,IER
4     CONTINUE
      CALL QSF(H,P,Z,100)
      CD=Z(100)
C
C      OUTPUT THE RESULTS
```

```
C
      WRITE(6,6001) (NAME(I),I=1,5)
      WRITE(6,6002) ALPHA,FLAP
      WRITE(6,6003) CD
      GO TO 2
C
C
5000  FORMAT(5A4)
5001  FORMAT(2F4.0,I3)
5002  FORMAT(10F8.4)
5003  FORMAT(16F5.3)
5004  FORMAT(10F8.4)
6000  FORMAT('AT POINT',I5,' IER=',I5)
6001  FORMAT(//'TEST OF',2X,5A4)
6002  FORMAT(' ALPHA=',F6.0,' FLAP=',F6.0)
6003  FORMAT(' DRAG COEFFICIENT=',F9.5//)
C
C
999   STOP
      END
```

```
FO       EQU       0
R1       EQU       1
R2       EQU       2
R3       EQU       3
R4       EQU       4
R5       EQU       5
SP       EQU       6
PC       EQU       7
BUFSIZE  EQU       1024+32
MINSIZE  EQU       40
MAXSIZE  EQU       1024
MECH     CSECT
*                  SET INTERRUPT ADDRESS
         MOV       @#W"414",R0
         MOV       #MECHTCP,@R0
*                  REQUEST BLOCK SIZE TO START
MECH1    MOV       #W"140115",R0
         MOV       #BLKREQEL,R1
         MOV       #80,CURSIZE
         MOV       @#W"402",R2
         EMT       0
*                  WAIT FOR OPERATOR INPUT
WAIT     MOV       #W"042000",R0
         EMT       1
*                  CLOCK INTERRUPT?
         BIT       #W"002000",R0
         BNE       WAIT                 BRANCH IF IT IS
*                  SHOULD WE STOP?
         MOV       #DATA,R0
         JSR       PC,TOKEN
```

```
         DC       A"ERROR"
         CMPB     @R0,#C' '
         BNE      TESTDEL           BRANCH IF NOT
*        DELETE CORE
         MOV      #W"10000C",RC
         MOV      @#W"410",R2
         EMT      0
*        STOP
         EMT      4
*        DELETE PREVIOUS FILE OR START NEW ONE?
TESTDEL  CMPB     @R0,#C'D'         DELETE PREVIOUS FILE?
         BNE      GETSIZE           BRANCH IF NOT
*
*        DELETE PREVIOUS FILE
*
DELFILE  MOV      #2,R3             # OF FILE MARKS TO BACKSPACE OVER
DELFILE1 MOV      #X'8008',R0       BACKSPACE COMMAND
         MOV      @#W"406",R2
         EMT      0
         MOV      #W"020000",RG     WAIT FOR COMPLETEION
         EMT      1
         BIT      #X'0020',R1       AT BEGINNING OF TAPE?
         BNE      PROBLMSG          BRANCH IF AT BOT
         BIT      R1,#X'4000'       AT END OF FILE?
         BEQ      DELFILE1          BRANCH IF NOT AT EOF
         DEC      R3                BACKSPACED OVER REQUIRED # OF EOF'S?
         BNE      DELFILE1          BRANCH IF NOT
         MOV      #X'8008',R0       FORWARD SPACE OVER LAST EOF
         EMT      0
*        PRINT FILE DELETED MESSAGE
```

```
PRDELMSG  MOV     #DFLMSG,R1
          BR      PRMSG
*
*         START NEW FILE - GET BLOCKSIZE
*
GETSIZE   JSR     PC,NUMBER
          DC      A"ERROR"
          MOV     R1,BLKSIZE
*         IS BLOCKSIZE IN CORRECT RANGE
          CMP     R1,#MINSIZE
          BLT     ERROR                BRANCH IF NOT IN RANGE
          CMP     R1,#MAXSIZE
          BLE     ISSTERM              BRANCH IF IN RANGE
*
*         ERROR IN INPUT - PRINT ERROR MESSAGE
*
ERROR     MOV     #ERRMSG,R1
          BR      PRMSG
*
*         ISSUE REQUEST TO TERMINATE MESSAGE
*
ISSTERM   MOV     #W"140115",R0
          MOV     #TERMSGEL,R1
          MOV     @#W"402",R2
          EMT     0
*         INITIALIZE BUFFER FOR WRITE
          MOV     #40,R3
          MOV     #DATA,CURPOS
          MOV     #DATA+80,R4
          CLR     R5
```

```
*                 FILL OUT LABEL WITH BLANKS
                  MOV       #81,R0
                  SUB       CURSIZE,R0
                  MOV       #DATA+81,R1
         FILL     MOVB      #' ',-(R1)
                  DEC       R0
                  BNE       FILL
*                 WRITE LABEL
                  JSR       PC,WRITE
*                 INITIALIZE CLOCK
         LOADCNT  CLR       @#W"172540"                 RESET CLOCK
                  CLR       @#W"172542"                 CLEAR COUNTER SET BUFFER
                  MOV       #W"37",@#W"172540"          SET COUNTER RUNNING WITH NO INTERRUPT
*                 WAIT FOR DATA
                  MOV       #W"042000",R0
                  EMT       1
*                 WAS THERE DATA?
                  BIT       #W"002000",R0
                  BEQ       TERMFILE                    BRANCH IF NO DATA
*                 STORE DATA
                  MOV       R1,(R4)+                    STORE DATA
                  CMP       R4,#DATAEND                 UPDATE BUFFER POINTER
                  BNE       *+6
                  MOV       #DATA,R4                    POINT BACK TO BUF BEGINNING
                  INC       R3                          INCREMENT CURRENT SIZE
*                 WRITE RECORD?
                  CMP       R3,BLKSIZE                  IS BLOCK FULL?
                  BNE       LOADCNT                     BRANCH IF IT ISN'T
*                 WRITE RECORD
                  JSR       PC,WRITE
```

CENTRAL STORES HF-585

```
         BR      LOADCNT
*        TERMINATE FILE - WRITE LAST RECORD
TERMFILE JSR     PC,WRITE              WRITE LAST RECORD
         JSR     PC,WRITE              WAIT FOR COMPLETION OF LAST WRITE
*        WRITE FILE MARK
         MOV     #X"8006",R0
         MOV     @#W"406",R2
         EMT     0
         MOV     #W"020000",R0         WAIT FOR COMPLETEION
         EMT     1
*        ISSUE FILE TERMINATED MESSAGE
         MOV     #FILETERM,R1
PRMSG    MOV     #W"100115",R0
         MOV     @#W"402",R2
         EMT     0
         JMP     MECH1
*
*        TAPE WRITE SUBROUTINE
*
WRITE    TST     R5                    WAITING FOR PREVIOUS WRITE?
         BEQ     NOWAIT                BRANCH IF NOT
*        WAIT FOR PREVIOUS WRITE
         MOV     #W"020000",R0
         EMT     1
*        WRITE CURRENT RECORD
         MOV     R5,CURPOS             STORE BUF BEG IN PARM LIST
         CLR     R5
NOWAIT   MOV     R3,CURSIZE            STORE RECORD SIZE
         BEQ     WRITERET              BRANCH IF NO RECORD TO WRITE
         CLR     R3
```

```
                MOV     R4,R5
                MOV     #X'8004',R0          WRITE CURRENT RECORD
                MOV     #TAPELIST,R1
                MOV     @#W"406",R2
                EMT     0
WRITERET        RTS     PC
*
*               NUMBER SUBROUTINE - CONVERT CHARACTER STRING TO NUMBER
*               R0 POINTS TO 1ST DIGIT
*               RETURNS TO ADDR FOLLOWING CALL IF 1ST DIGIT NOT NUMERIC
*
NUMBER          CLR     R1                   WILL HOLD NUMBER
CKNUMB1         CMPB    @R0,#W"060"          IS 1ST DIGIT NUMERIC?
                BLT     RETURN1              BRANCH IF ERROR
                CMPB    @R0,#W"071"
                BGT     RETURN1              BRANCH IF ERROR
CKNUMB2         CMPB    @R0,#W"060"          AT END OF NUMBER?
                BLT     RETURN0              BRANCH IF AT END
                CMPB    @R0,#W"071"
                BGT     RETURN0              BRANCH IF AT END
                MOV     R1,R2                MULTIPLY NUMBER BY 10
                ASL     R1
                ASL     R1
                ADD     R2,R1
                ASL     R1
                MOVB    (R0)+,R2             ADD NEW DIGIT
                BIC     #W"177060",R2        CLEAR ZONE BITS
ADDNUM          ADD     R2,R1
                BR      CKNUMB2              GET NEXT DIGIT
*
```

CENTRAL STORES HF-588

```
*            TOKEN SUBROUTINE - FIND 1ST NON - BLANK IN STRING
*            PC POINTS TO STRING
*            RETURNS TO ADDR FOLLOWING CALL IF CARRIAGE RETURN IS 1ST NON - BLANK
*
TOKEN        CMPB      (P[illegible]"040"
             BEQ       TOKEN
             CMPB      -(PC),#W"015"
             BEQ       RETURN1
RETURNO      ADD       #2,@SP              NORMAL RETURN
             RTS       PC
RETURN1      MOV       @(SP)+,PC
*
*            DATA FOR MECH PROGRAM
*
BLKREQEL     DC        A"0",A"MECHMSG",A"CURSIZE"
TERMSGEL     DC        A"0",A"TERMSG",A"0"
MECHMSG      DC        F"13",E'ENTER BLKSIZE'
TERMSG       DC        F"27",E'ENTER ''M'' TO TERMINATE FILE'
DELMSG       DC        F"21",E'PREVIOUS FILE DELETED'
FILETERM     DC        F"23",E'CURRENT FILE TERMINATED'
ERRMSG       DC        F"14",E'ERROR IN INPUT'
TAPELIST     DC        A"BUFFER"
BLKSIZE      CC        2F"0"
CURPOS       DC        A"DATA"
BUFFER       DC        A"BUFSIZE"
CURSIZE      DC        F"0"
DATA         EQU       *
             ORG       *+BUFSIZE*2
DATAEND      EQU       *
MECHTCB      DC        A"0",A"MECHSTK-16",W"370",F"1",F"0"
MECHSTK      EQU       *+26
             DC        11F"0",A"MECH",W"0"
             END       MECHTCB
END OF FILE
```

APPENDIX C

COMPUTER OUTPUTS

$CON *T* POSN *14*
$R SCANREADOB 0=*T* 4=-A 5=ZEROS 8=*SINK* 9=FXFZERO
17:27.13

FILE LABEL:

ENTER ALPHA:

CONTINUING WITH ALPHA= 0.

DATA FOR TEST OF FX-61-163 SERIES II

TEST CONDITIONS: ALPHA= 0. FLAP= 0.

RAW COUNTS

I	NAS	NSCAN	NPST	NTP
1	467	244	29	27
2	467	285	29	27
3	469	341	29	27
4	468	372	29	27
5	467	396	29	28
6	468	423	29	27
7	466	430	29	28
8	466	437	29	27
9	470	426	29	27
10	468	400	29	27
11	471	367	29	28
12	468	349	29	27
13	467	340	29	27
14	467	300	29	27
15	469	246	29	28
16	467	350	29	27
17	470	362	30	27
18	470	366	29	27
19	469	378	29	28
20	468	387	29	28
21	472	381	30	27
22	469	375	29	27
23	466	346	29	27
24	469	351	29	28
25	472	315	29	28
26	470	305	29	28

ENTER Q CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: Q=-0.08240*NAS+59.88

ENTER SCAN CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: SCAN=-0.20200*NSCAN+38.70

ENTER PST CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: PST=*********NPST+******

ENTER TP CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: TP=*********NTP+******

PHYSICAL QUANTITIES*

HOLE	Q	SCAN	CP	PST	T
1	21.399	-10.588	0.51	*******	****

C3

86

2	21.399	-18.870	0.12	*******	****
3	21.234	-30.182	-0.42	*******	****
4	21.317	-36.444	-0.71	*******	****
5	21.399	-41.292	-0.93	*******	****
6	21.317	-46.746	-1.19	*******	****
7	21.482	-48.160	-1.24	*******	****
8	21.482	-49.574	-1.31	*******	****
9	21.152	-4[illegible].352	-1.24	*******	****
10	21.317	-42.100	-0.97	*******	****
11	21.070	-35.434	-0.68	*******	****
12	21.317	-31.798	-0.49	*******	****
13	21.399	-29.980	-0.40	*******	****
14	21.399	-21.900	-0.02	*******	****
15	21.234	-10.992	0.48	*******	****
16	21.399	-32.000	-0.50	*******	****
17	21.152	-34.424	-0.63	*******	****
18	21.152	-35.232	-0.67	*******	****
19	21.234	-37.656	-0.77	*******	****
20	21.317	-39.474	-0.85	*******	****
21	20.987	-38.262	-0.82	*******	****
22	21.234	-37.050	-0.74	*******	****
23	21.482	-31.192	-0.45	*******	****
24	21.234	-32.202	-0.52	*******	****
25	20.987	-24.930	-0.19	*******	****
26	21.152	-22.910	-0.08	*******	****

* QUANTITIES NOT REQUIRED ARE MASKED ******

DATA COPIED TO DISK FOR FURTHER CALCULATIONS

STOP 0

17:27.15 .713 RC=0

$R TRAVREADOB 0=*T* 4=-B 5=ZEROT 8=*SINK* 9=FXFZERO
17:27.17

FILE LABEL:

ENTER ALPHA

CONTINUING WITH ALPHA= 0.

DATA FOR TEST OF FX-61-163 SERIES II

TEST CONDITIONS: ALPHA= 0. FLAP= 0.

RAW COUNTS

I	NAS	NWAKE	NPOS	NPST	NTP
1	496	830	570	29	28
2	497	829	540	29	28
3	497	829	506	29	27
4	496	829	473	29	27
5	498	829	440	29	27
6	500	827	411	29	27
7	498	830	386	29	27
8	499	827	359	29	27
9	497	827	334	29	28
10	497	824	308	29	27
11	499	800	284	29	27
12	496	640	273	29	27
13	494	452	267	29	27
14	499	311	261	30	27
15	496	118	254	29	27
16	498	32	247	29	27
17	497	31	239	29	28
18	498	30	231	29	27
19	498	100	224	29	27
20	498	246	218	29	27
21	496	372	212	29	27
22	498	519	206	29	27
23	498	646	201	29	27
24	494	729	193	29	28
25	500	789	186	29	27
26	498	813	179	29	27
27	496	822	158	29	27
28	495	824	132	29	27
29	500	823	105	29	27
30	499	825	77	30	28

ENTER Q CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: Q=-0.08240*NAS+59.88

ENTER WAKE CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: W=-0.00500*NWAKE+ 4.25

ENTER POS CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: POS=-0.01100*NPOS+ 6.16

ENTER PST CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: PST=*********NPST+******

FNTER TP CALIBRATION SLOPE AND INTERCEPT:
CALIBRATION IS: TP=********NTP+*******

C5
88

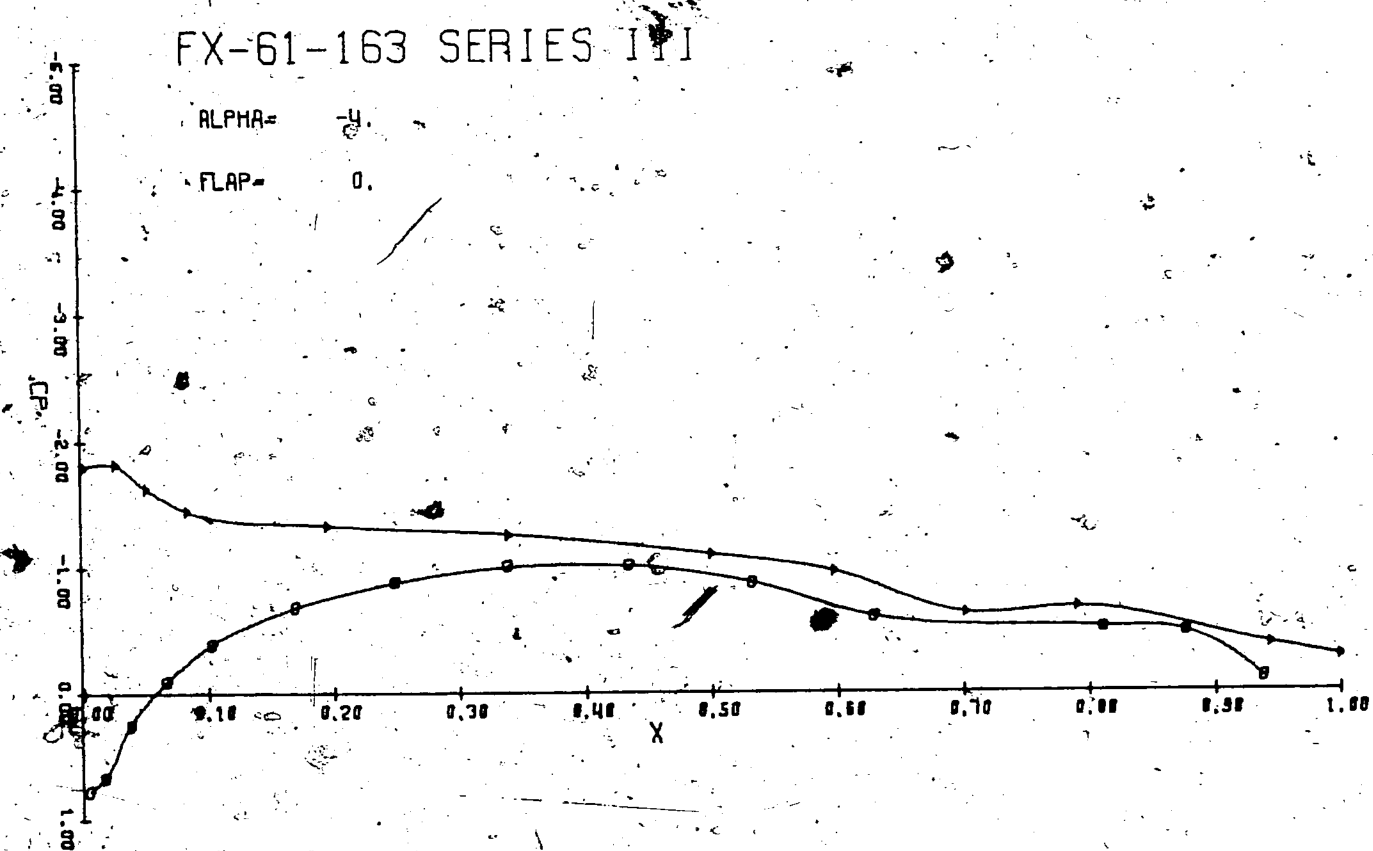
PHYSICAL QUANTITIES*

I	Q	WAKE	POS	PST	TP
1	19.01	0.100	-0.110	******	******
2	18.93	0.105	0.220	******	******
3	18.93	0.105	0.594	******	******
4	19.01	0.105	0.957	******	******
5	18.84	0.105	1.320	******	******
6	18.68	0.115	1.639	******	******
7	18.84	0.100	1.914	******	******
8	18.76	0.115	2.211	******	******
9	18.93	0.115	2.486	******	******
10	18.93	0.130	2.772	******	******
11	18.76	0.250	3.036	******	******
12	19.01	1.050	3.157	******	******
13	19.17	1.990	3.223	******	******
14	18.76	2.695	3.289	******	******
15	19.01	3.660	3.366	******	******
16	18.84	4.090	3.443	******	******
17	18.93	4.095	3.531	******	******
18	18.84	4.100	3.619	******	******
19	18.84	3.750	3.696	******	******
20	18.84	3.020	3.762	******	******
21	19.01	2.390	3.828	******	******
22	18.84	1.655	3.894	******	******
23	18.84	1.020	3.949	******	******
24	19.17	0.605	4.037	******	******
25	18.68	0.305	4.114	******	******
26	18.84	0.185	4.191	******	******
27	19.01	0.140	4.422	******	******
28	19.09	0.130	4.708	******	******
29	18.68	0.135	5.005	******	******
30	18.76	0.125	5.313	******	******

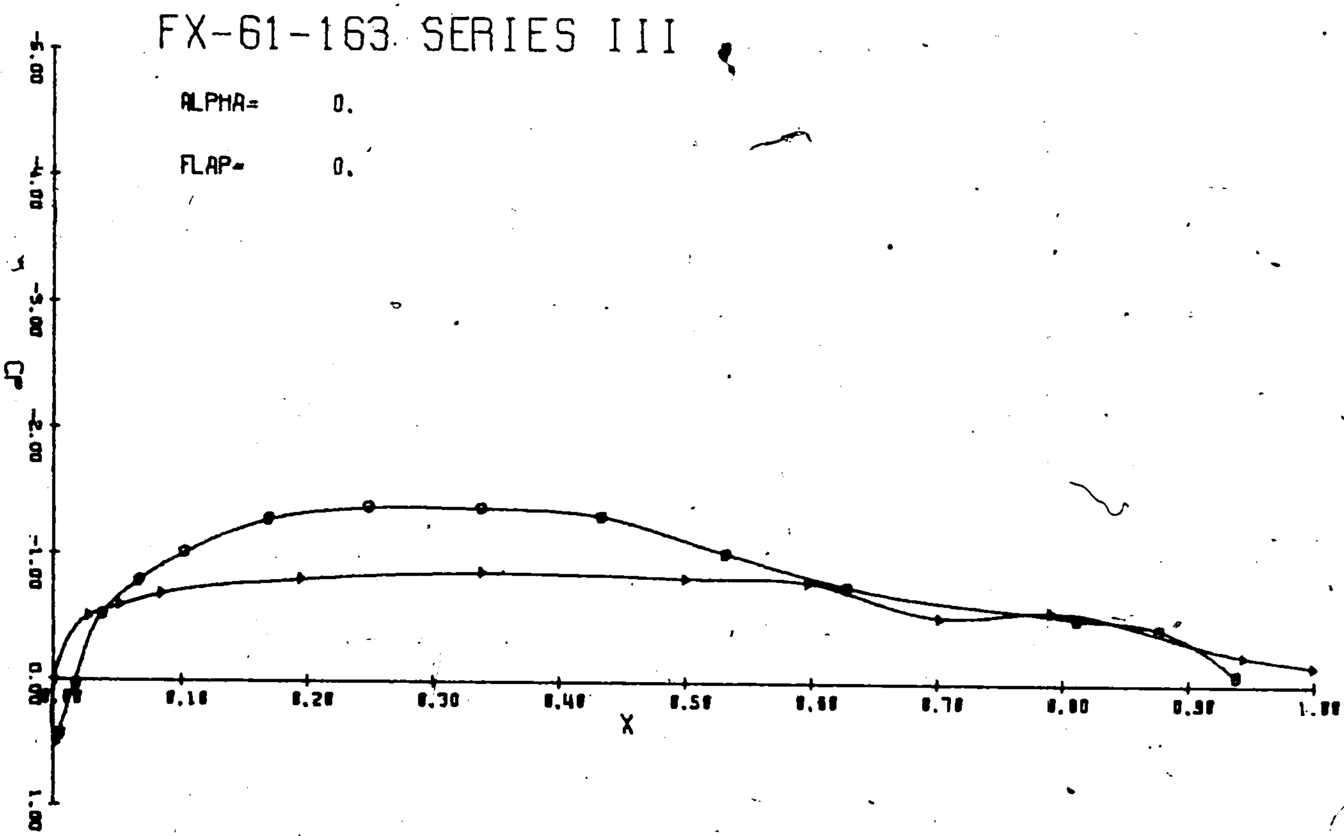
* QUANTITIES NOT REQUIRED ARE MASKED *****

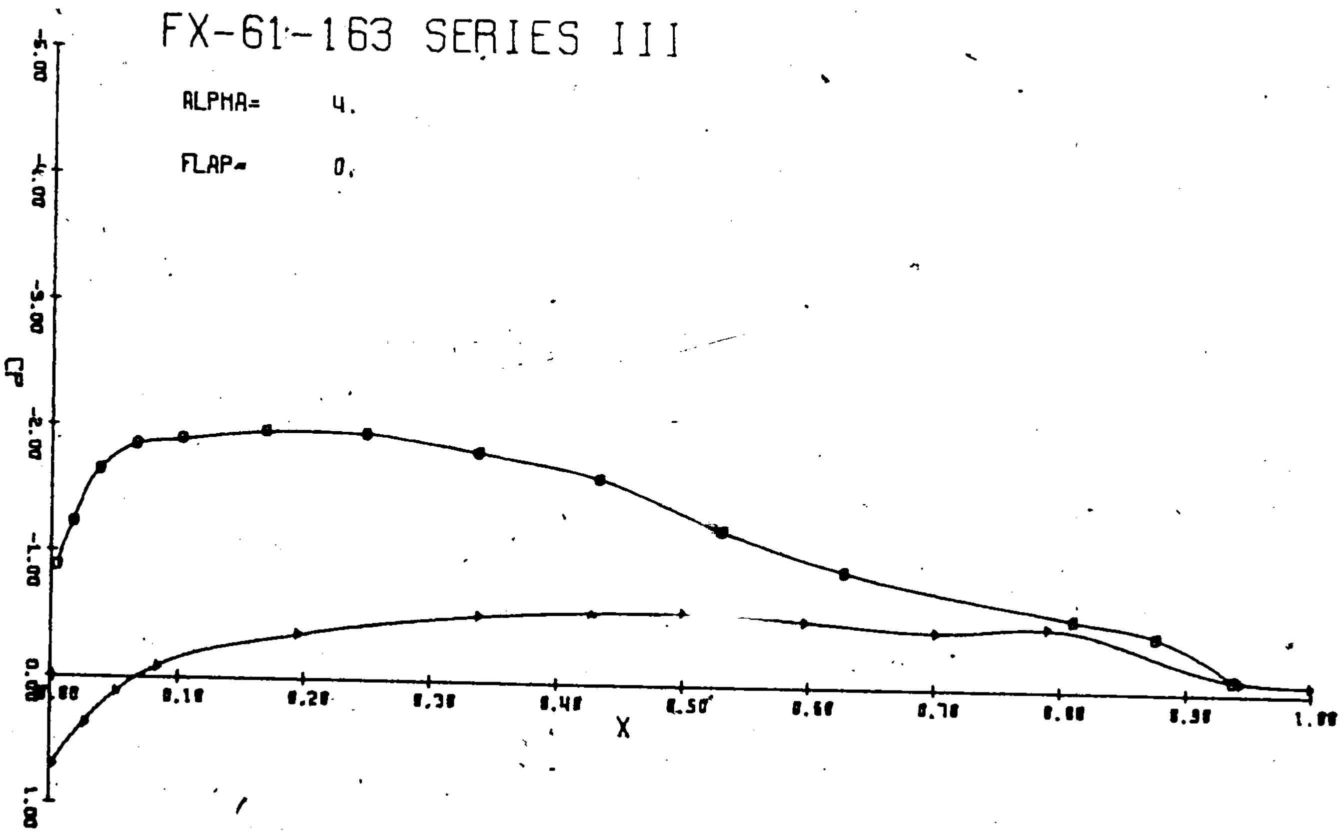
DATA COPIED TO DISK FOR FURTHER CALCULATIONS.

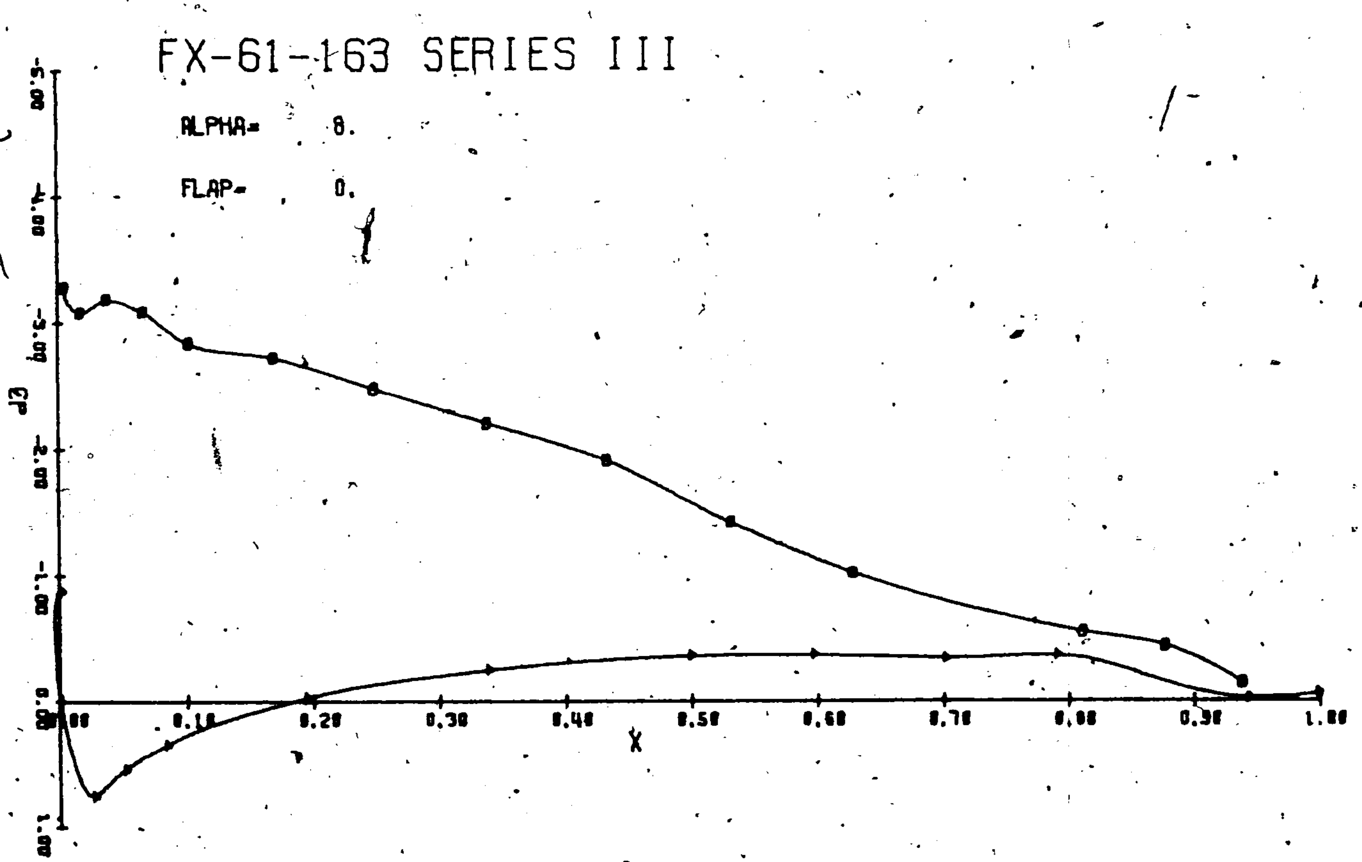
STOP 0
17.27.21 .766 RC=0

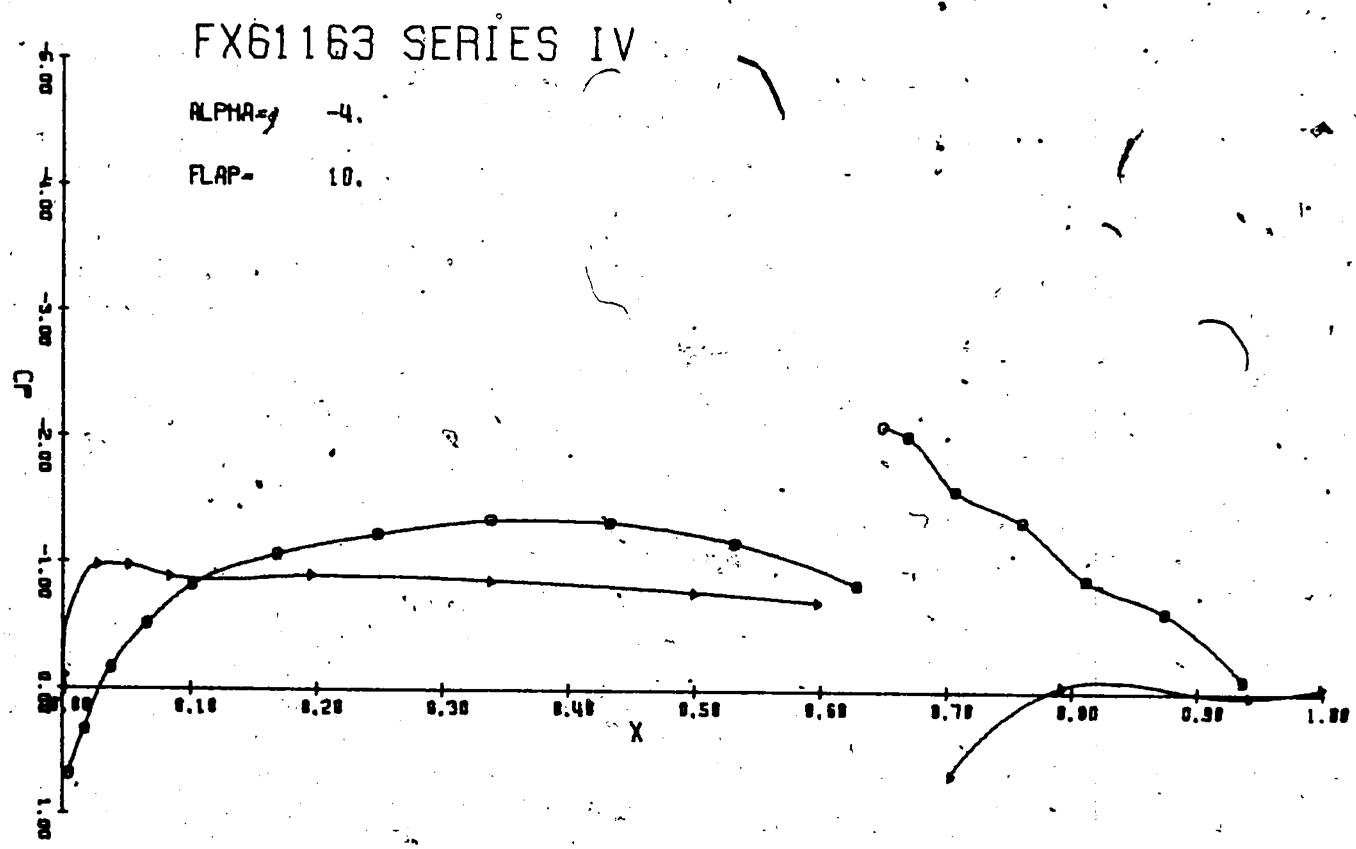


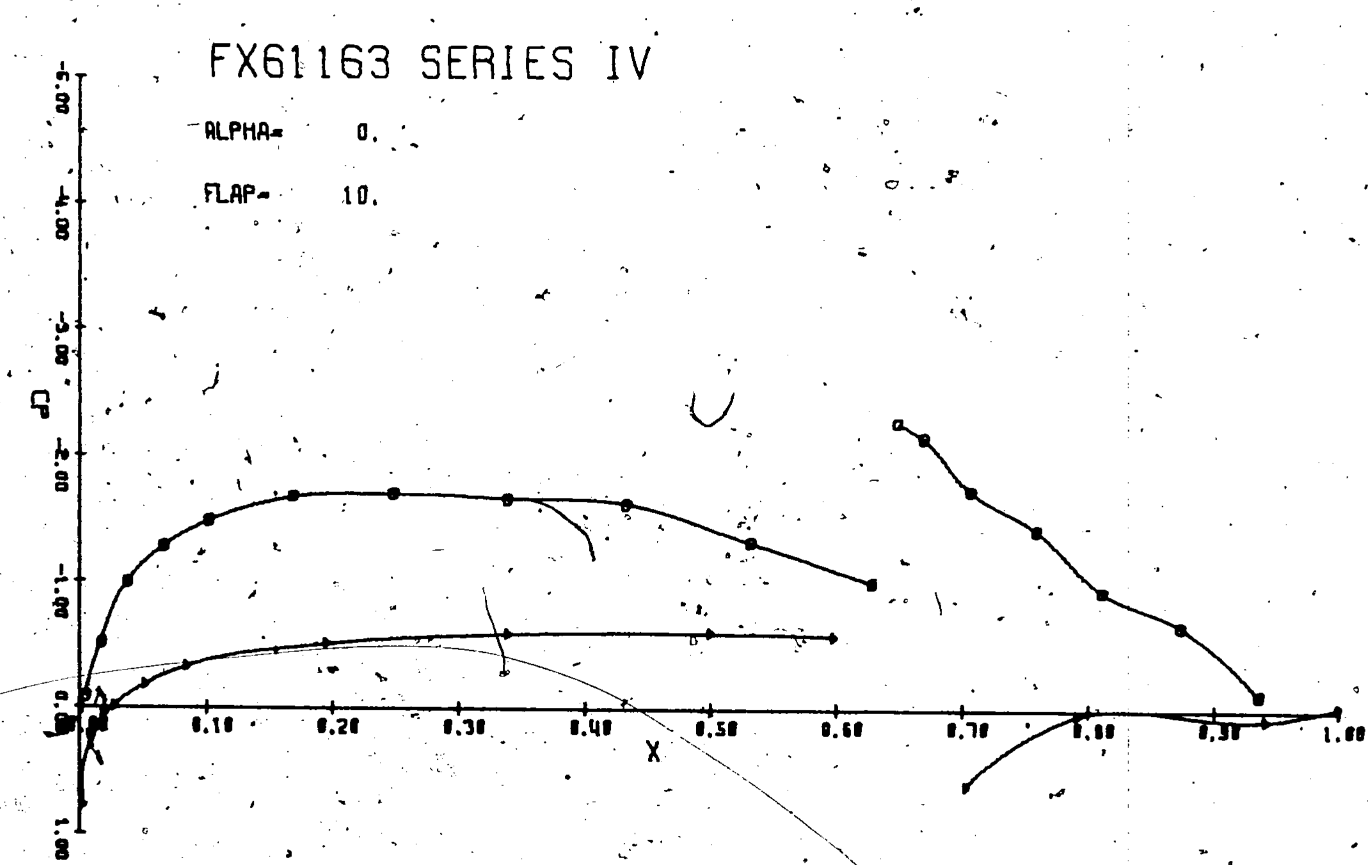
FX-61-163 SERIES III
ALPHA= -4.
FLAP= 0.
CP
X
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00

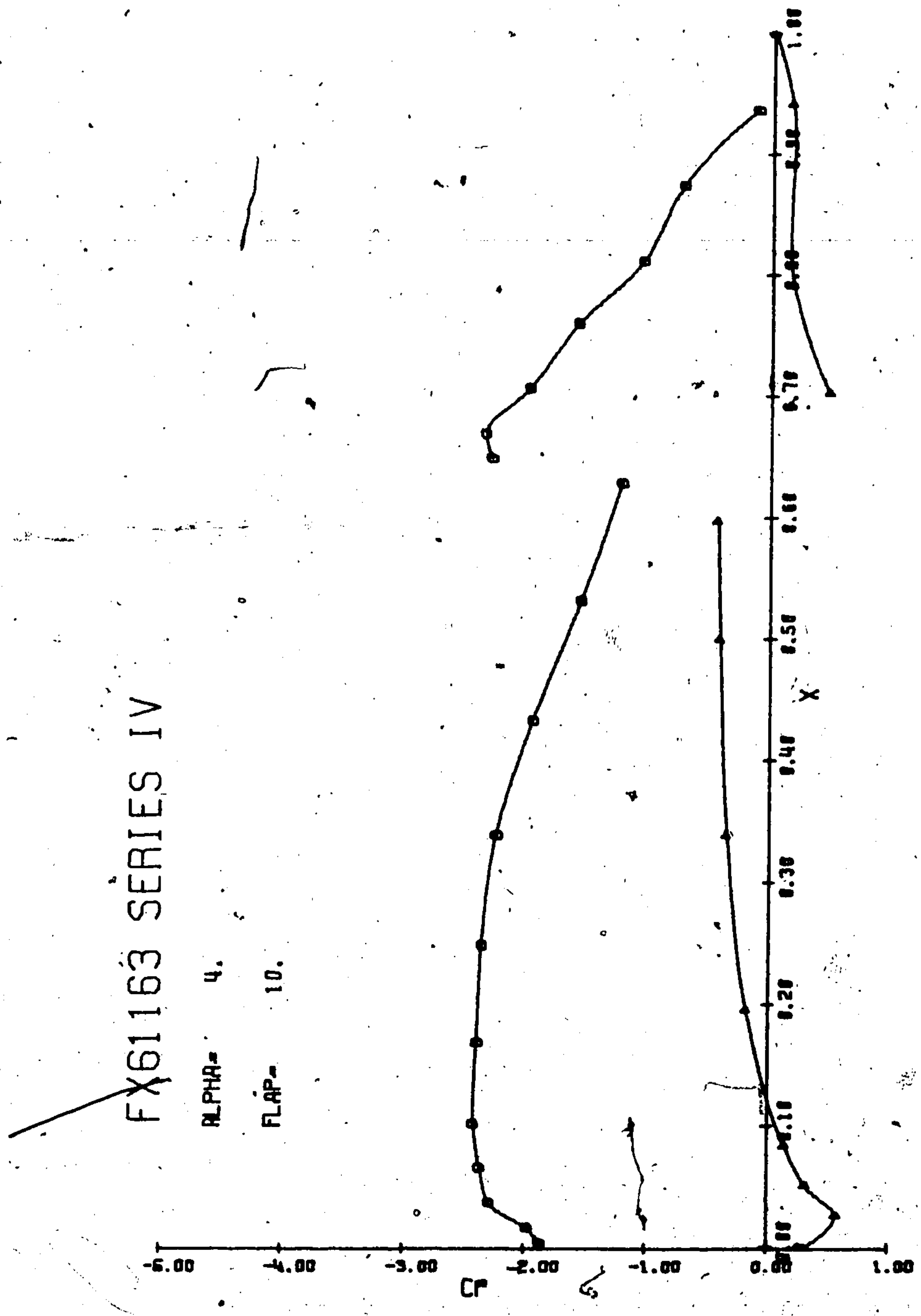

FX-61-163 SERIES III
ALPHA= 0.
FLAP= 0.
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
CP
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
X

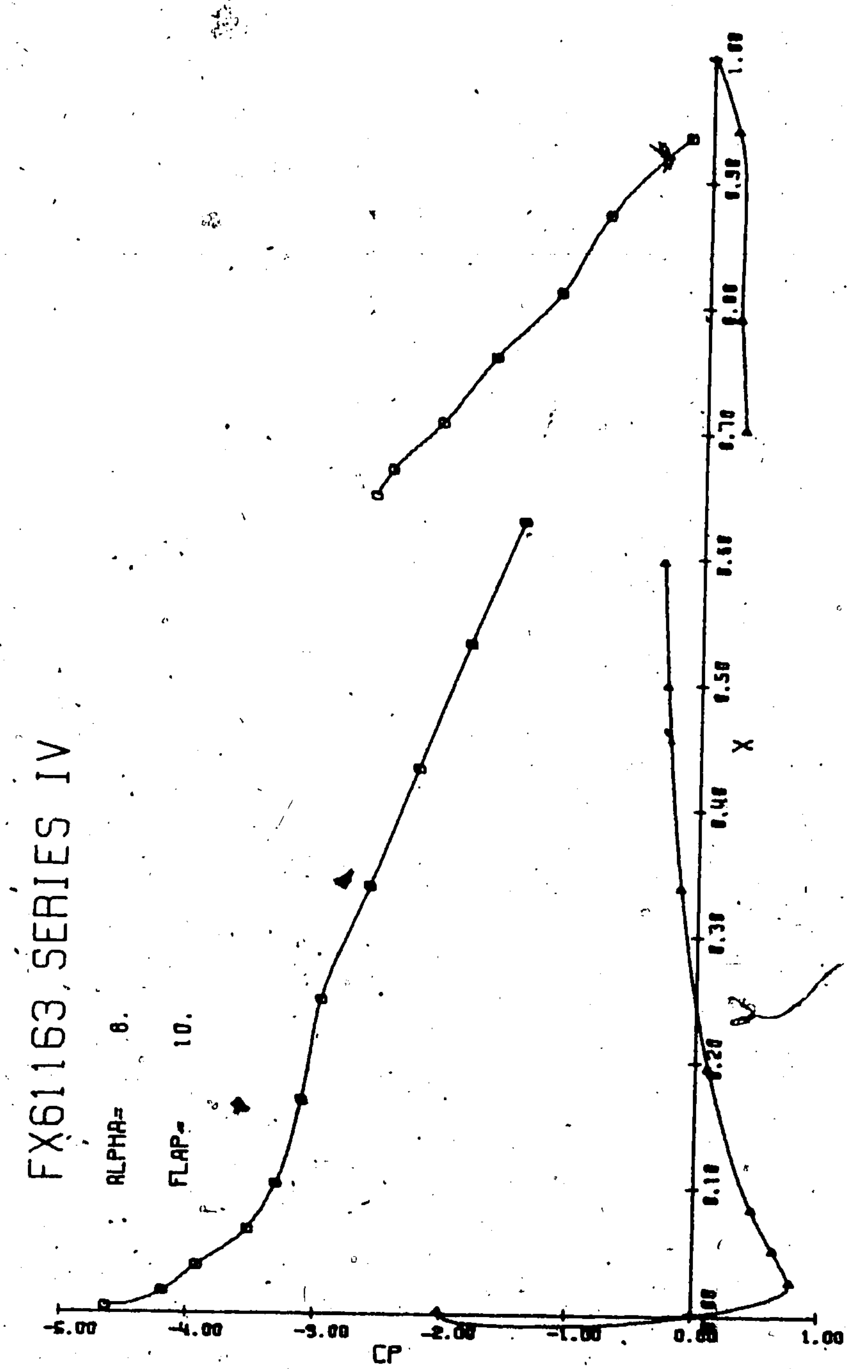

FX-61-163 SERIES III
ALPHA= 4.
FLAP= 0.
CP
X
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00


FX-61-163 SERIES III
ALPHA= 8.
FLAP= 0.
CP
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
X
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00


FX61163 SERIES IV
ALPHA= -4.
FLAP= 10.
CP
-6.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
X
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00


FX61163 SERIES IV
ALPHA= 0.
FLAP= 10.
CP
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
X
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00


FX61163 SERIES IV
ALPHA= 4.
FLAP= 10.
-6.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
CP
X
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00


FX61163 SERIES IV
ALPHA= 8.
FLAP= 10.
CP
X
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00