4-3-79

REPORT NO. NADC-79052-60

STRUCTURING OF DATA SYSTEMS

Prepared by
Bruce D. Fritchman
Lehigh University
Bethlehem, Pennsylvania

9 MARCH 1979

Contract No.
N62269-77-C-0347

Contract Monitor: John G. Nelson

FINAL REPORT
AIRTASK NO. A310310C/001A/7R04101001

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared for
NAVAL AIR SYSTEMS COMMAND
Department of the Navy
Washington, DC 20361

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteeth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: Edward J. Sturm DATE: 9 March 1979

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NADC-79052-60	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: Structuring of Data Systems: Psychophysiological Data from the Dynamic Flight Simulator	5. TYPE OF REPORT & PERIOD COVERED: Final Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Bruce D. Fritchman, Lehigh University John G. Nelson, Contract Monitor (Code 60031)	8. CONTRACT OR GRANT NUMBER(s): N62269-77-C-0347
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Lehigh University Bethlehem, PA 18015	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: AIRTASK A310310C/001A/ 7R04101001
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Air Systems Command Department of the Navy Washington, DC 20361	12. REPORT DATE: 9 MARCH 1979 13. NUMBER OF PAGES: 34
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Office of Naval Research Resident Representative University of Pennsylvania, David Rittenhouse Laboratory, 209 S. 33rd St., Philadelphia, PA 19104	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computer Aided Experimentation
Digital Data Collection
Data Analysis
Data Compression
Psychophysiological Data Processing
Data Systems
Pattern Recognition
Feature Extraction
Software Development
Dynamic Flight Simulation

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This investigation considered the methods and problems of collection, storage, processing and display of psychophysiological data from dynamic flight simulation in the Naval Air Development Center human centrifuge, using analog, mini, and remote main-frame computers. An integrated data processing system is described. Problems associated with data collection and storage are evaluated, and alternate solutions discussed. Anticipated problems in the development of data processing software are examined, and

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

the applicable basic technology identified. Approaches to the critical problem of orderly and systematic development and maintenance of software for semi-open-shop operations is examined.

TABLE OF CONTENTS

LIST OF FIGURES

STRUCTURING OF DATA SYSTEMS

1.0 Introduction

This investigation considers the problems which are likely to arise during the collection, storage, processing, and display of data generated in the performance of dynamic flight simulator (DFS) tests using the NADC centrifuge. These tests involve the use of 3 different computers: (1) an analog computer which controls the centrifuge and gondola, (2) a large mainframe computer (CDC 6600) used for real-time data processing, and (3) a new V77-600 mini-computer used to collect data, refresh graphic displays in real-time, and to process data off-line.

Section 2 of this report describes an integrated data processing system which should meet the goals of dynamic flight simulation for a long time to come. It is basically the system currently being implemented. Section 3 evaluates the problems associated with data collection and storage, and discusses alternate solutions to these problems. Section 4 examines the problems which lie ahead in the development of data processing software and identifies the basic technology which is applicable to this development. Finally Section 5 examines the problem of developing software in such a way that unsophisticated users may have access to it and that the signal processing library will expand in an orderly and useable fashion.

2.0 System Configuration

A complete system showing the analog control and display functions as well as the data system is illustrated in Figure 1. This is basically the system presently being developed at NADC. The portion of the system enclosed by dotted lines is the data system. It is this system which is of primary interest in this report.

The data system involves the collection and processing of data obtained from the dynamic flight simulator (DFS). The data is collected by sampling a number of analog signals (analog-to-digital conversion). These samples are compressed and stored on disc or tape or they are immediately processed using one or both of the two digital computers shown.

The CDC 6600 is a large mainframe computer capable of very fast operation using long computer words and having 128,000 words of core memory. The V77-600 is a 16 bit mini-computer, slow in comparison to the 6600, having 64,000 words of semiconductor memory. The two computers are physically separated from one another by a distance of about 1/4 mile. They will be linked together by a number of different communication channels. The primary data channels will be 5 million bit per second optical links in each direction.* There will also be coaxial cables and shielded cables containing multiple twisted pairs. The coaxial cables are primarily for the transmission of analog, video signals, though any excess capacity could be used for data transmission. Similarly, the shielded twisted pairs are to be used for the transmission of relatively narrow bandwidth analog signals, but they can also be used for low-speed data lines, perhaps at rates up to 9600 bits per second.

* The 5Mbit rate is an upper limit on the data transfer rate. End constraints determined by the communication/computer interfaces may result in a lower data transfer rate.

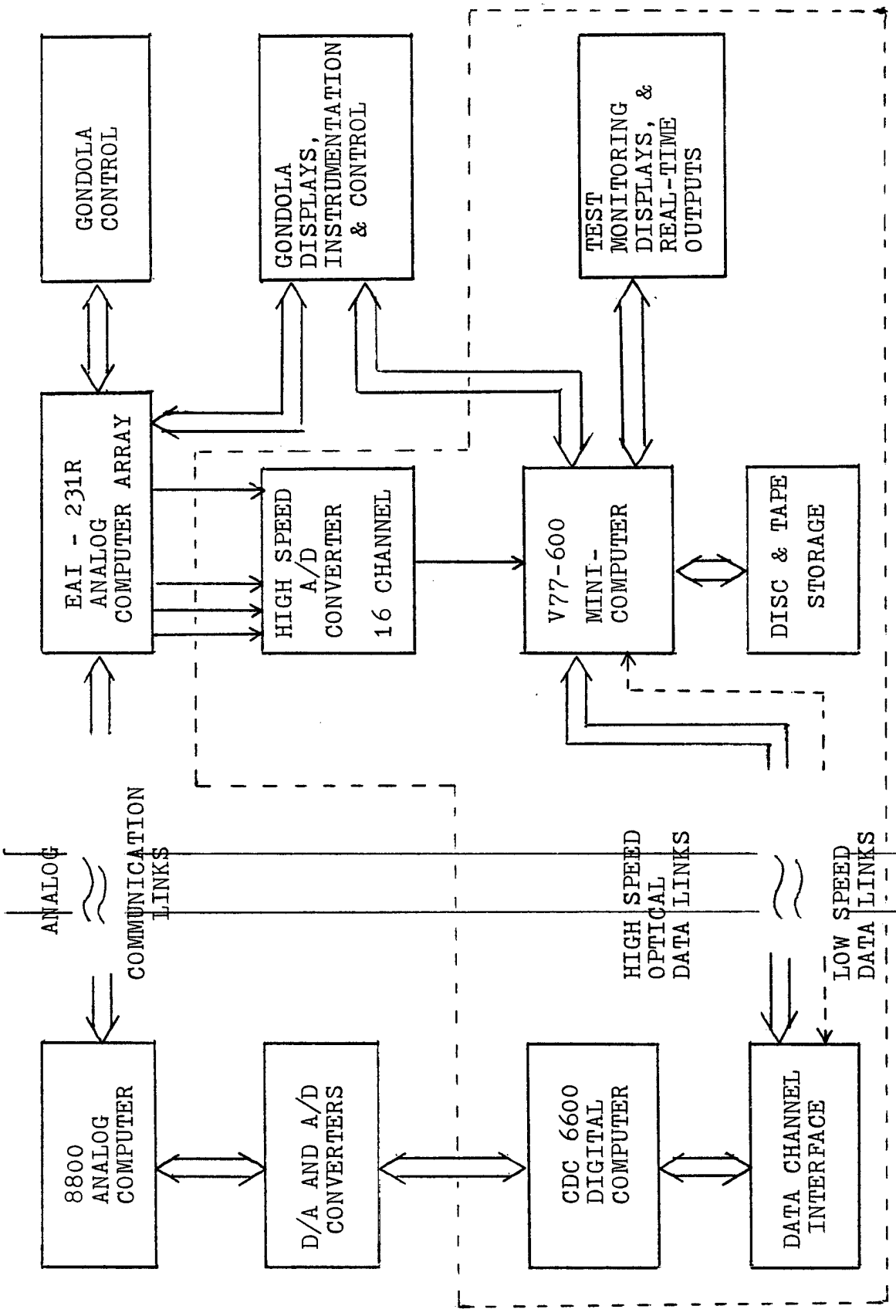

FIGURE 1 DATA PROCESSING SYSTEM

The system shown is already in the implementation phase. Due to the two very different computers included in the system, the distribution of necessary computation will be primarily determined by the type of computation required. Though the most economical use of the system is to perform as much computing and program development using the mini-computer as possible, there is no alternate to using the CDC 6600 for complicated real-time data processing. Moreover, due to its location, its direct interface with the analog-to-digital (A/D) converter, and its large disc storage capacity, the mini-computer will have to perform the data collection and data storage functions as well as the graphic display and refresh functions. When operating in real-time, these operations will utilize nearly all of the mini-computer capacity. Hence sophisticated, real-time signal analysis will have to be performed on the CDC 6600. However, when off-line processing is acceptable, the mini-computer should be used to the maximum extent possible. When not busy with data collection and graphic display functions, the mini-computer has substantial capacity to perform most, if not all, off-line data processing functions.

Since scheduling and operation of the CDC 6600 computer is costly and wasteful when tests must be aborted due to improper design or equipment failure, the mini-computer should be used to perform system and test design checks before the costly 6600 computer is brought on line.

3.0 Data Collection System

There are many factors to be considered in the design of the data collection subsystem. These include a number of potential problems related to analog-to-digital conversion of the time-continuous signals. Without care it is quite possible to overwhelm the rest of the system with data, thereby rendering it ineffective. There must be sufficient flexibility to simultaneously handle digital and analog signals having different bandwidths. It must also be possible to efficiently transfer the collected data to an appropriate Central Processing Unit (CPU) for further processing or to a nonvolatile storage medium for future analysis. Moreover, data collection, processing and storage must be accomplished without losing data during real-time operation.

The way a signal is handled by the data collection and processing system depends on the type of signal being processed. There are a number of ways of classifying the signals measured during tests. From the user's point of view, signals might be classified according to their source of origin. Thus they may be grouped as physiological, environmental, and performance signals. Typical signals which are of potential interest are given in Figure 2 according to this classification. From the point of view of data collection, there is another more important classification. Signals may be grouped as analog or digital. Furthermore, their bandwidths are important, especially the bandwidths of the analog signals.

The signals measured during tests of systems and human subjects are of varying bandwidths ranging from 50 Hertz to 2000 Hertz. Most of the signals are analog, but some, especially performance related signals, are digital. Thus a variety of signals must be dealt with in such a way that little or no information is lost in the data collection process.

BIOMEDICAL DATA

Crew Systems Department - NAVAIRDEVCEN

Environmental

Ambient Temperature

Wall Temperature

Ambient Pressure

PH_2O - R.H.

Air Velocity

Gaseous Comp
PO_2, PCO_2, PN_2 ...

G-Mag. and Dir.

Noise

Vibration (Buffet)

Light Level

Physiological (non-invasive)

ECG - Heart Rate
Systol. Time Interval

EEG

EMG

Blood Pressure

Blood Flow

Arterial Sat.

Resp. - Rate, Expir. Gas Comp.

Imped. Plethys.

Carotid Pulse

Esoph. Press.

Biochemistry

Blood - RBC, CBC, Hb ...

Urine - Vol, pH

Performance

Psycho-motor
Tracking
Sim. Combat

Visual
PLL (Peripheral Light Loss)

Mechanical

Mental
Arith.

Duration (Extremes)

DFS (Cent.)

1 ss - 5 min. to 60 min.

Eject. Tower & Horiz. Accel.

1 ss - Pre-run - 10 min.
Post-run - 10 min.

Alt. Chamber

6 ss - 2 weeks

Thermal Chamber

1 - 3 ss - 8 hr.

Underwater Tank

1 ss - 1 hr.

FIGURE 2 BIOMEDICAL DATA

Basically there is little problem with the digital signals. By using the proper computer interface circuits designed to operate at transfer rates compatible with the data rates generated by the digital sources, no information should be lost under normal operating conditions. Through the digital communication interface, the digital test data can be entered directly into the computer memory.

Unfortunately, the analog signals can not be dealt with so easily. First, the analog signals must be converted to digital signals. It is this analog-to-digital (A/D) conversion process which may cause problems. Any information lost in the A/D conversion can not be recovered by signal processing at other points in the system. Consequently the A/D conversion process establishes the upper limits on the overall system performance. Due to the critical nature of this problem, it is evaluated more carefully in the next section.

3.1 Analog-To-Digital (A/D) Conversion

Most of the physiological and environmental signals occur naturally as analog signals. Since data processing is to be entirely digital, these signals must be reduced to digital form. To appreciate the limitations of the A/D conversion process it is useful to view it as a two step process.* The first step consists of determining the amplitude of the measured signals at specific instants uniformly separated in time. The second step consists of converting the amplitude samples into numerical values which can be processed by both computer systems.

* Reference (1) is an excellent tutorial paper on sampling theory and contains 1005 references to all aspects of sampling.

There are two important considerations which must be dealt with when implementing these two steps. The first is the rate at which the samples are taken. If no information is to be lost, then samples must be taken at a rate equal to or faster than twice the frequency of the maximum frequency component of the signal being sampled, i.e., $f_0 \geq 2B$, where f_0 is the sampling rate and B is the bandwidth of the signal to be sampled when measured in Hertz. Thus for example, a signal having a bandwidth of 100Hz must be sampled at a rate of at least 200 samples per second. The second step in the A/D conversion process consists of converting sample amplitudes into numerical values. Since, however, digital computers can not represent numbers with infinite precision, the sample amplitudes are quantized and coded into finite binary sequences which represent the numerical values which can then be handled by the computers. Theoretically and practically any procedure which converts from a continuum of values to a finite set of values will introduce error, usually referred to as quantization error. Of course the more binary digits used to approximate the true amplitudes the smaller is the quantization error.

One convenient way to evaluate the quantization error is to consider the resulting quantized sequence as having been produced by adding an error sequence to the sequence of true values. From this point of view the effect of the quantization process is similar to the effect of adding a noisy signal to the analog signal before sampling. Thus quantization noise can be evaluated from the same point of view as other noise processes* introduced in the testing system.

The A/D converter which will be available with the V77-600 minicomputer has a maximum sampling rate of 50 thousand samples per second, and each sample is quantized to 13 binary digits. When considering

---

* Strictly speaking it is necessary to consider the statistical distributions when dealing with noise signals in the evaluation of their effects on the signal. In many situations, however, neglecting the distributions does not lead to erroneous conclusions.

the quantization process as the addition of a noisy signal to true signal, the equivalent signal-to-noise* (SNR) power ratio due only to quantization error is 88db. Of course other noise sources will be present in the system. These will be generated within the sensors, the communication links, and the transducers. When these noise signals are combined with the quantization noise, the resulting SNR can be expected to be less than 88db. Actually 88db is a very good SNR. It is reasonable to expect that the quantization noise will be less than the noise generated elsewhere in the system, and consequently these external sources will limit system performance and not the quantization noise when 13 bits are used. It may be concluded that the A/D system which will be available with the V77-600 mini-computer will be quite good from the point of view of quantization noise and sampling rate. There are, however, two potential problems which need further consideration.

One problem is related to the bandwidth of the signals being measured. As already pointed out these signals have bandwidths ranging between 50 Hertz and 2000 Hertz. In some cases the bandwidth of a sensor may be significantly greater than the bandwidth of the signal it is intended to measure. This results in the measurement of the signal of interest as well as the inclusion of interfering and noisy signals in the sensor output. If the sensor output is sampled at a rate consistent with its bandwidth, a higher than necessary sampling rate is required. Moreover, since the samples represent the signal to be measured as well as the extraneous signals, their use in the data processing system may lead to inaccurrate results. Alternatively, if the samples are taken at a rate consistent with the bandwidth of the signal of interest, distortion of the samples will result due to aliasing. The solution to this problem is to first filter the sensor out-

---

* SNR is a frequently used and usually very effective measure of performance.

put to eliminate out-of-band extraneous signals. Then the narrower bandwidth signal at the filter output can be sampled at a rate consistent with the bandwidth of the signal of interest. At the present time no provision has been made to provide appropriate filtering. Just how serious a problem this might be will depend on the characteristics of the sensors being used. It is recommended, however, that this problem be more fully investigated in order to determine its impact on the accurracy of the digital processing system.

A second problem is related to the A/D converter sampling rate. Its maximum sampling rate of 50 thousand samples per second can be divided among 16 different channels, thereby permitting simultaneous sampling of 16 different signals. What is not clear, however, is whether the total sampling rate must be equally divided among the 16 channels or whether different channels may be sampled at different rates in such a way that the total does not exceed 50 thousand samples per second. The latter feature is highly desirable and in fact is necessary if considerable unwanted and unnecessary data is not to be generated. Since the test signals will have varying bandwidths, ranging from 50Hz to 2000Hz, they require varying sampling rates. Of course any signal can always be sampled at a higher rate than the required minimum. However, increasing the rate above the minimum does not increase the accurracy of the representation, but it does generate unnecessary data which must be processed and stored. Most computer controlled A/D converters do permit variable sampling rates on different channels. The rates are generally selectable through program control. Unfortunately, the information available about the V77-600 A/D converter does not clearly indicate whether or not variable sampling rates will be permitted on parallel channels. It is a potential problem which will have to be examined when the system is installed. If the variable rate is not available, then software will have to be developed to purge the unwanted samples as part of the data compression software.

The final consideration of the A/D conversion process is the limitation to 16 parallel channels. As shown in Fig. 2, there are potentially more than 16 signals which are of interest. Actually the system will permit measurement of more than 16 signals, since some of them are digital. The digital signals need not be entered through the A/D converter (though they may and in some cases this might be desirable), but can be entered directly through a digital communication interface. For the foreseeable future it would appear the data collection system will be more than adequate. If more than 16 signals need to be sampled in the future, the problem can be overcome by first processing some of the narrowband signals using a separate low-speed A/D converter coupled to a micro-computer. These samples might then be entered through the digital communication interface port. Alternatively, it appears to be possible to expand the A/D conversion capability to as many as 64 channels, though there is no information available at this time on the maximum sampling rate for more than 16 channels.

3.2 Data Compression

The two step sampling and coding process described in the preceding section is generally referred to as Pulse Code Modulation (PCM). What is important to note is that when the A/D converter is operating at its maximum capacity it produces 50 thousand, 13 bit samples each second. This is a raw bit rate of 650 thousand bits per second. The 13 bit samples are converted into 50 thousand 16 or 32 bit computer words per second, depending on whether or not they are represented as integer or floating point numbers. Thus as many as 100 thousand bytes* per second can be generated by the A/D converter. If it is necessary to store all of this data on the disc, then the 80 Megabyte disc would be full in 800 seconds, assuming the entire disc were available for storage. Of course it is not, since it must store the operating system, required data processing software, and graphic display software. Thus, when operating at maximum capacity, the A/D conversion process has the

*A byte is generally considered to be 8 binary digits.

capability of overwhelming the data storage facilities. Moreover, indications are that very limited data storage facilities are available with the CDC 6600 computer when operating in real time. This was determined from discussions with personnel from the computer center at NADC. Consequently, the 6600 cannot be used for storage back-up.

Using the disc storage available to the V77-600 mini-computer, it is clear the amount of raw data must somehow be reduced when operating the data collection system at or near capacity. In most situations this problem can be partially solved or avoided altogether by being careful to sample each signal at a rate close to its allowable minimum as determined by its bandwidth, i.e., at close to the Nyquist rate. This will not cause any loss of information, while minimizing the amount of data generated. The philosophy, the faster the better, is not suitable in the data processing system being considered.

In most situations the raw data can be considerably reduced by using one of the many effective data compression techniques. Depending upon the wave shape of the signal being sampled, it is possible to reduce the number of binary digits needed to represent a sample, over that required by PCM, without increasing the quantization error. Such techniques are referred to as data compression. There are many such techniques. Which one is best depends on the characteristics of the signal to which it is applied. For example, a signal such as an ECG, which has a regular structure and a relatively large dynamic range, might benefit from differential PCM (DPCM). Using this procedure the difference in the sample amplitudes from one sample to another is coded instead of the absolute sample amplitude as in PCM. If the differences do not vary as greatly as the absolute sample amplitudes, then fewer binary digits can be used to represent the differences with the same level of quantization noise. This procedure may be further improved by adaptively adjusting the quantization range from one part of the signal to another. Thus if significant changes in a signal are

localized, a further reduction in the number of binary digits needed to represent it is possible. This technique is referred to as adaptive DPCM (ADPCM). Delta Modulation (DM) and adaptive DM (ADM) are two additional techniques which are applicable to signals having properties favorable to the application of DPCM and ADPCM. Still another approach which has been very effective for signals, such as speech, which have a wide dynamic range is linear predictive coding (LPC). In this approach past samples of the signal are used to predict the value of the next sample. Then the difference between the actual and predicted value is coded. This approach allows the quantizer to take advantage of trends in the signal to reduce the number of binary digits needed to represent a sample. This idea can be carried still further to include past and future trends over many points. Procedures of this type are referred to as tree coding. A theoretical basis for the determination of the minimum representation for a specified level of distortion has been developed. It is referred to as Rate Distortion Theory, and it can be a very useful tool in assessing the effectiveness of a specific algorithm applied to a signal having specific characteristics.

A number of excellent papers describing these techniques are available. There are two excellent survey articles which have appeared in recent years. In 1973, Bayless, Campanella, and Goldbert (2) reviewed a number of different types of digitization and compression techniques, as did Jayant (3) in 1974. Jayant's paper includes 75 references. In a volume edited by Davisson and Gray in 1976 (4) there appears a collection of some of the most significant data compression papers. Of particular interest, Atal and Hanauer (5) set the ground work for LPC; Bellow, Lincoln and Gish (6) describe a method of using statistical analysis to adapt DM; Jelinek and Anderson (8) introduce tree encoding which is extended by Berger and Jelinek (9), Viterbi and Omura (10), and Anderson and Bodie (11). An important tree encoding algorithm is also described by Gallager in (12).

Data compression techniques can be implemented in hardware or software. Thus it might be possible to process the incoming data using the V77-600 mini-computer through the application of relatively simple algorithms. This is possible providing sufficient processor time exists to collect the raw data, compress it, transfer the compressed data to the mini-computer disc or to another computer, and generate and refresh the graphic displays. Quite clearly the V77-600 mini-computer will be very busy during real time tests. Since the Vortex operating system permits more than one precessor to operate on memory, a second CPU could be used to handle the data compression processing while the first attends to the other processing. Another approach might be to use dedicated micro-processors to perform the task on specific signals. This could be an inexpensive and highly effective solution to the problem. At this time, however, there is relatively little understanding of the actual data requirements of the data collection system. Moreover, it will probably be sometime after the system is operational before the requirements can be fully assessed. At this point it is only necessary to recognize that such techniques are available and that they can be implemented in hardware or software. It should also be recognized that to implement them in software will require a substantial amount of software development by individuals knowledgeable about data compression techniques. It would be a backward step to try to reinvent techniques which are already highly developed.

3.3 Data Transfer

There is still another important consideration in the implementation of the data collection system. The mini-computer is configured so that the A/D is a peripheral device. In general, peripheral devices are linked to the computer by buses which are connected to the CPU and memory through interface devices. The interface devices are a combination of special hardware and software. Data can be communicated from the A/D converter to the computer in two different ways. One method is for the A/D to set a flag indicating it has data to enter.

The computer then interrupts what it is doing and accepts the data from the periferal device (in this case the A/D converter). Each periferal device generates interrupts. Moreover, interrupts may be generated from within the computer. Generally, prioities may be set on the interrupts so that more critical devices can be accessed more quickly. For example, the Power Fail interrupt will probably have the highest priority so that data and programs can be transferred to nonvolatile memory before the power goes down.

When the A/D converter is operating at its maximum rate of 50 thousand samples per second, it produces a new sample every 20 microseconds. Thus if this data is to be entered through interrupts, the computer must be capable of processing interrupts this quickly. In fact, it must be capable of a somewhat faster rate if interrupts generated by other devices are to be processed. If the computer can not process the A/D interrupts fast enough, data will be lost. At this writing information about the maximum rate of interrupt processing using external clocking is not available.

A second method of transferring the A/D data to the computer is to enter it directly into a specified area of the computer memory, i.e., into a memory buffer specifically set aside to accept the data. This is referred to as direct memory access (DMA), and it avoids interrupting the central processor unit. DMA is faster than using interrupts and has the added advantage of allowing the CPU to continue other processing while A/D data is entering memory. While this method is likely to be the best one for entering high speed A/D data, there are still some potential problems which might be encountered.

When entering data through the DMA channel, it is entered directly into the memory buffer. Since data will be produced continuously from the beginning to the end of a test, the memory buffer will quickly fill up. When this occurs the contents of the buffer must be transferred

elsewhere, i.e., to another computer, the disc, or tape unit. To accomplish the transfer the CPU must intervene. That is, a program which carries out the desired transfer must be run. The data can then be transferred at the maximum transfer rate of the device which is receiving the data. In the case of the disc, which is the most likely destination, the transfer rate is 604.8 thousand words per second. This is a rate more than 12 times faster than the maximum rate at which data enters the buffer. Of course the transfer can begin only after the disc heads are properly positioned. While in principle there appears to be no problem in reading the data into and out of the memory buffer, some mini-computer operating systems artificially create a problem. Some systems block the DMA channel when the CPU program, which empties the buffer, is swapped into the memory. This causes data to be lost, because the buffer cannot empty instantaneously. At this writing there is insufficient data to determine whether such a problem will occur with the V77-600 Vortex operating system. All indications are that the problem will not occur, however, it is prudent to be aware of the potential difficulty. If the problem does occur, the 64K memory should be sufficient to permit a modification or reconfiguration of the operating system to prevent its occurrence. The data collection system using DMA entry is illustrated in Figure 3.

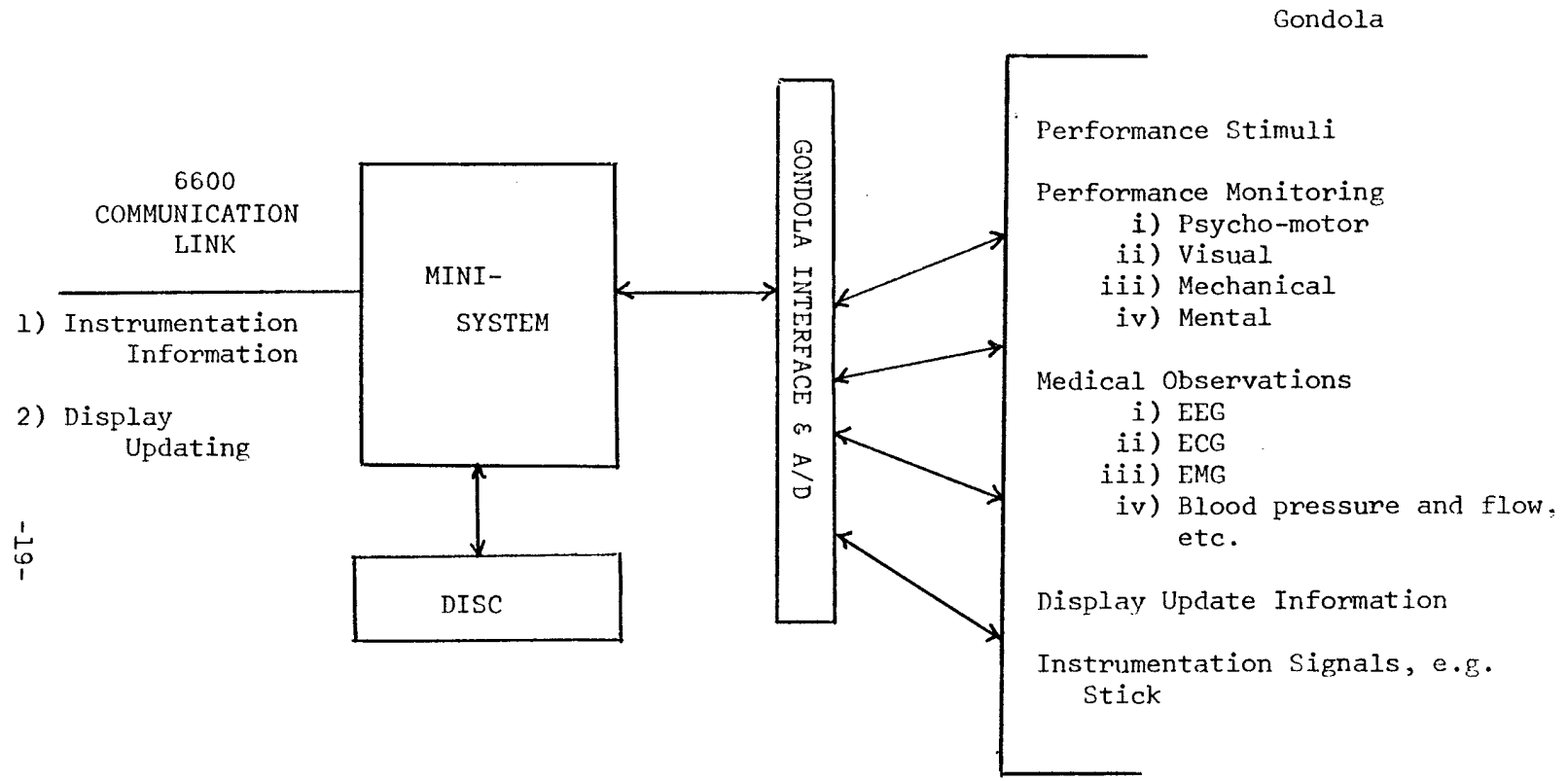

FIGURE 3 DATA COLLECTION SYSTEM

4.0 Data Processing

After the data has been collected and organized in files of appropriate structure, the data may be processed to extract the information deemed useful by the experimenters. Data processing is the heart of the system.

Before data processing can be accomplished with any degree of efficiency and effectiveness, it is necessary to quantify the procedures by which desired information is to be extracted. Ultimately, algorithms must be constructed and programs written to efficiently process the data.

In general, the problem of acquiring the desired information from the raw data is commonly referred to as feature extraction. In one sense feature extraction is similar to data compression. In the process of extracting features all data which is not relevent to a given feature can be discarded. If the data is only to be used to extract a feature or given set of features, then the data can usually be greatly compressed, thereby reducing transfer and storage problems. There is, however, an important distinction which can be made between data compression and feature extraction. If only the data relevent to a set of specific features is retained, then the original raw data can not usually be reconstructed from the extracted features. This is in contrast to the data compression problem previously described where data is reduced with the objective of restoring the raw data or the original signal with a specified and hopefully tolerable level of distortion. As a simple example, if the only features of a set of data are the mean value and standard deviation, saving only these quantities will result in an enormous reduction of data which must be handled. If, however, it is later desired to extract other features, it is generally not possible to do so, since the raw data can not usually be reconstructed from these two statistics. Another example may be der-

ived from speech processing. If only the intelligibility of the speech is of interest, it has been shown that data can be reduced by a factor of 200. With this level of data reduction however, other features are lost such as speaker identification and speech quality. Moreover, these features can not be reconstructed from the reduced data. While feature extraction is a necessary step in the processing of raw data, care must be exercised so as not to use it as a data compression technique if the data is likely to be needed to extract different features at a future time.

In many experimental situations a number of identifiable outcomes are possible. Each of these outcomes may be considered to be a specific feature. The problem is them to perform the experiment to determine which outcome has occurred. For example, a subjects ECG may react within normal bounds or it may not during a stressful situation. In this kind of problem there are a number of well defined features. The data processing problem is then to process the data so the outcome can be properly classified. This type of processing is referred to as pattern recognition.* There is a well developed theoretical framework which has been developed to solve pattern recognition problems. The techniques may be deterministic or probabilitistic depending upon the nature of the problem.

In the simplest pattern recognition systems, the features are described deterministically, and the data are classified by matching to the known features. Template matching used in the recognition of alpha-numeric characters is an example of this approach, where each letter is described by a specific template. The idea of a template can be extended to any type of pattern, including patterns found directly in the measured sample values or patterns of parameters derived from the measured values. Application of this approach presupposes that there is a nominal deterministic description which can be obtained

---

* An excellent summary of pattern recognition techniques - both statistical and syntactic - appears in Reference (13) along with a bibliography containing 304 references.

for each pattern. For many real problems, experience shows that well defined deterministic patterns do not exist, or if they do exist their description depends on so many parameters as to make them impractical to describe. Moreover, many patterns can only be described probabilistically, in which case template matching is inappropriate.

Alternate approaches to the recognition problem are required when any or all of the following conditions exist:

1. The relationship between measured data and physical variables is very complex,
2. The number of parameters required to describe a feature is very large,
3. Features can only be described probabilistically.

One approach which is very effective when one or more of these conditions are present is to view the measured sample values, or parameters derived from them, as components of a vector in a multi-dimensional space. Features are then recognized as clusters of points in the space. In such a system, features can be learned by determining the boundaries between clusters and by identifying bounded regions with specific patterns. What is particularly interesting about this approach is the idea of learning features which might be too difficult or impossible to describe analytically or which can only be described probabilistically. The learning process might be based on measured data which is known to characterize a specific pattern. This is described as learning with a teacher. Learning might also be based directly on the measured pattern clusters. In this case, as more patterns are processed, the fault boundaries are up-dated to best fit the cluster distributions. This is described as learning without a teacher. In many systems both types of learning are incorporated, i.e., learning with a teacher initially to roughly establish cluster boundaries using a small number of samples, and then learning without a teacher to fine-tune the location of the boundaries. In both cases the boundaries are usually obtained using an algorithm which attempts to mini-

mize the number of classification errors or some cost function related to the cost of misclassification.

Another approach which is applicable to many difficult pattern recognition problems is the linguistic or syntatic method. Typically a pattern is represented by a coded structure or structures called strings. A grammar, of the type used to describe formal languages, is devised that generates the string-encoded images. Since the grammar can be used in the inverse mode as an acceptor, the patterns can be classified by parsing their strings with the generation rules of the grammar. Parsing is the process of accepting the string as generated by a particular grammar and giving its structural derivation. An added dimension has evolved with the development of stochastic languages. This added dimension is obtained through production rules. A measure in the form of a string probability can then be designated for each string accepted by the grammars.

In general several grammars are designed with different probability assignments and with overlapping languages. The classification procedure then involves finding the structural derivation of the unknown string and hence the probability it was generated by each grammar. The grammar yielding the largest probability of generation is used to classify the unknown pattern. Using this method of classification, each grammar represents a pattern. Each grammar may permit many different strings and so a pattern is not represented by any one string. As in the case of the preceding approach to pattern recognition, the grammars may be changed adaptively as more data are acquired.

Some of the signal processing problems which will be of interest in tests involving the DFS have already been investigated from this point of view with some success. Thus it is likely that many of the algorithms which will eventually be used will fall in this category of signal processing technology. What is already known is that subtle features and sophisticated classification requires substantial computation. This knowledge has significant impact on the structuring of

the processing.

During the performance of a test, the mini-computer has two basic functions which it must perform in real-time. It must collect and transfer data and it must generate and refresh graphic displays. Furthermore, if at all possible, it should provide some processing of the data to compress it. In such an environment, there will be little time or computer storage available for sophisticated real-time feature extraction and pattern recignition. Clearly there is little alternative but to use the CDC 6600 for this purpose.

In situations when real-time processing is not required, a quite different picture emerges. The mini-computer has considerable capability to perform sophisticated processing when processing time is not a significant factor. When the V77-600 computer is not occuppied with data collection and display generation functions, it will be capable of performing most of the required signal processing. Thus the same signal processing capability will be required by both computers.

Similarly, when limited data collection is required as might be the case when setting up tests the mini-computer can be used for both the data collection and real-time data processing. This will avoid the problem of scheduling the CDC 6600 for abortive tests. All pretesting and preliminary signal processing should be possible using only the mini-computer.

5.0 Structured Software Development

Since most signal processing algorithms will be implemented in software, software developemnt will require considerable effort and support. One of the problems which must be faced from the very beginning is the need for software which is accessible to users with limited knowledge of the computer operating systems. This means that a highly structured approach to data processing software development must be established and adhered to. Each program must be well documented and written in such a way that unsophisticated users can learn to use it quickly and with confidence. Standardizing the structure of data processing programs is essential to the development of a readily useable data processing library. An example of a workable type of structure follows.

First there should be a program which lists all of the signal processing programs available. This listing should include a description of each program and its current status, such as: available, unavailable, restricted use, etc. For each program listed in the library, the user should by simple instruction be able to obtain a description of the program and information on how to use it.

Next, each program should be structured so it can easily be loaded and run by an unsophisticated user. In running any data processing program, input and output is an important consideration. Usually, if a lot of data is to be processed and extensive data is produced as a result of the processing, the most convenient method of input and output is by using files having a specified format.

For example, a program designed to plot data on a video display monitor will accept a file of data to be plotted and then plot the data on the screen. The file must, however, be in the proper format expected by the video software. In other cases a program such as a fast Fourier transform accepts a file of data samples and produces a file of computed frequency samples. If the spectrum is next to be plotted on a video

screen or an X-Y plotter, the output file of the fast Fourier transform program must be in a format acceptable to the video display terminal or X-Y plotter. In general these different display devices will require files having different formats. Clearly, the appropriate file formats must be established and made known to everyone who develops new software if a large amount of wasted programming effort and time is to be avoided. It is necessary to decide on basic file formats from the very beginning, and to insist that all users follow these standards. Moreover, simple programs can be written, by those familiar with the operating systems of the computers, to assist programmers, without such knowledge, in the automatic formatting of their input and output files.

In many programs it will be desirable to provide a variety of options. Before running a plot program, the user must decide how many points he wants to plot, what scaling factor he wants to use, etc. To avoid the need for the user to learn complicated input data formats in order to use the program in a specific way, all programs can be written in an interactive mode so that the program asks appropriate questions about the required inputs, such as: How many points? What scaling factor? etc. Following this approach, a sequence of questions is asked, and the user simply supplies the necessary answers to obtain his desired configuration. The appropriate input files are then entered and the output file computed.

By following a procedure such as this an unsophisticated user can easily use very complicated programs.

5.1 Data Base Management

As already indicated, on-line data processing will require the use of the 6600 computer. However, off-line processing may be performed using the mini-computer. In fact, due to economic considerations, software development will also be performed using the mini-computer whenever

possible. Many off-line processing algorithms will be the same as the on-line algorithms. Thus both computers will require data bases (in this case programs) which are the same. This requires careful maintenance of the files in both data bases. When a higher level language such as Fortran is used, this is less of a problem than when incompatible languages are used. When a commonly supported higher level language such as Fortran is used, the software development can be carried out on the V77-600 mini-computer. The source files which are stored in ASC II charcters can then be transmitted in a file format compatible with the 6600 source files. These files can then be compiled on the 6600 creating compatible object files and hence a common data base. This procedure will require the development of a simple program to convert the mini-computer source files into the correct 6600 source file format. The files can then be passed back and forth between the 6600 and the minicomputer via the new data communication links. The communication links which will be installed between the two computers and the centrifuge provide for many options so there should be no problem in obtaining the necessary data links for this purpose.

One problem still remains which must be overcome. Normally the creation of new programs on the V77-600 computer and their subsequent transmission to the 6600 computer should, for economic reasons, be done while the 6600 is operating in the batch mode. Unfortunately the batch files and the real-time operating files have different formats. Thus additional 6600 software must be developed to convert from one format to another, otherwise the 6600 will have to be operated in real-time whenever new software must be introduced or whenever existing software must be modified. This would clearly be uneconomical. Another consideration is the maintenance of both data bases created in this way. A procedure must be established to check that any software modifications to one data base have been made to the other.

When programs which have been developed to perform the same functions on both machines are written using different languages, the data base management problem becomes far more difficult. Unfortunately, the need for such programs will likely arise. The reason for this is due to the significant difference between the operating speeds of the two computers as well as to the large difference in their word lengths. Real-time mini-computer software which operates on the raw data, will likely have to be written in assembly language or it may even have to be microprogrammed. Similarly, mini-computer programs requiring a high degree of iterative processing and hence accuracy may require double precision arithmetic in the mini-computer. In general this will not be required in the 6600 due to its much longer word length, i.e., single-precision arithmetic in the 6600 is more accurate than the double precision mini-computer arithmetic. Finally, higher level languages such as APL or PASCAL may be supported on the 6600 but may not be supported on the mini-computer. The equivalent programs must therefore be developed in a different mini-computer language. The difficulty which arises is that it is difficult to determine if all modifications made to the data base of one computer have also been made to the other. It is even difficult to always determine if they truly perform the same functions which they were designed to perform.

The file management problem can not be overstated. Failure to deal with it effectively will result in continual difficulty and frustration with the operation of the system.

5.2 Test Preparation Using Transfer Files

If all the programs are developed using standard input/output file formats, and the programs are structured in an interactive mode as suggested in Section 5.0, it might appear to be difficult to use a sequence of such programs in a real-time test environment, when there is not time to respond to the questions.

This problem is easily overcome by creating Transfer Files and Answer Files. A test is constructed by stringing together different data processing programs. The individual programs are called in the proper order by constructing a Transfer File which calls the programs in the order required. When the Transfer File runs a program, the answers to the questions, which set the basic parameters, are entered from a prepared Answer File. The data processing programs accept the files created by previous programs and produce new files which can be further processed by subsequent programs.

Preparation of tests then reduces to the preparation of transfer and answer files. Tests prepared in this way can be directed to perform any sequence of data processing operations predetermined by the experimenters. This procedure can be further extended to provide for branching which depends on the outcomes of the real-time data processing. A simple example of a transfer file and answer file for an HP1000 minicomputer is given in the Appendix.

6.0 Data And Signal Processing Outputs

The V77-600 mini-computer and available software make it particularly well suited for displaying data processing results in graphic form. This form of output can be useful to the experimenter. Generally, however, only critical signals can be so analyzed and displayed in real-time. For example, signals which might indicate the need to abort a test. Since the graphic display is not a convenient way of producing accurate hard copy, a good X-Y plotter output is necessary if test reporting is to be automated and the results produced quickly. Of course, printed output is necessary and will likely be most common form of output. It should also be noted that the mini-computer can be used to produce reports if a good printed output is available. Using the computer CRT terminal and Vortex operating system editor, reports can be typed into files where they can be edited, stored on tape or disc for future reference, and printed out when desired. Similarly, standard reporting forms can be stored and simply reproduced with the results of tests included. Finally, it will likely be necessary to reconstruct analog signals from the processed data. These signals can be displayed in real time using an oscillascope type of display. Scopes having short and long persistence are needed for this purpose.

7.0 Summary and Conclusions

The basic data processing system presently under development has the potential to greatly expand the use of the DFS. There are, however, some problems which must be overcome before this potential is realized. Though these problems, which are summarized below, are all solvable, they are not trivial. If, however, proper attention is given to their solution, the basic system now being implemented should provide a foundation upon which a powerful signal processing system may be developed. A system which meets the signal processing requirements for the DFS for many years to come.

In Section 3.0 the potential problems associated with data collection and storage are assessed. The greatest potential problem is that the A/D conversion capacity being installed (and probably required) is greater than the data storage capacity which will be available. As discussed in Section 3.0, this problem can be overcome by providing data compression software and possibly some additional mini or micro computer capability or by providing additional data storage capacity. It is premature to estimate which if any of these solutions will be required. Only operational experience will provide the answer. However, it is more than likely that at the very least some form of data compression software will have to be developed.

In Section 4.0 the types of data processing likely to be required in the reduction of measured data is investigated. It appears safe to conclude that with proper software development, the CDC 6600 will be able to provide a very powerful real-time signal processing capability. Similarly, the mini-computer will be able to provide the same type of off-line capability. Of course the necessary software has not been developed nor could such a development be expected at this time. It will only be developed from the cumulative efforts of many test programs, providing it is developed properly.

Section 5.0 considers the important aspects of proper software development. Since such development is the key to the development of an expanding signal processing capability, this problem must be carefully considered. Some of the potential difficulties which may arise are considered in this section and a possible solution is proposed.

Of all the probelms which might arise, it is likely that software development and maintenance will be the most difficult to solve. For a facility of the type considered in this study, it is essential that an economical and effective solution to this problem be found. Unfortunately, there is far less understanding of the form of the best solution to this problem as compared to hardware problems. One fact is however quite clear. It will be necessary to designate one or more people to be responsible for the operation of the system and for ensuring that all system users abide by the basic rules of software development. These individuals will have to be intimately familiar with the mini-computer and its operating system. Moreover, they will also have to ensure that the system is regularly backed-up so that when the system "crashes" (and it will) no significant software losses occur. Additionally, since there will be many users who are not familiar with the details of the mini-computer operation, they will frequently find new and imaginative ways to "hang-up" the system. Only if individuals are available, who are knowledgeable about the system, will this problem be overcome with a minimum of frustration and wasted effort.

REFERENCES

1. A.J. Jerri, "The Shannon Sampling Theorem-Its Various Extensions and Applications: A Tutorial Review", Proceedings of the IEEE, V. 65, N. 11, pp 1565-1596, November 1977.

2. J.W. Bayless, S.J. Campanella, and A.J. Goldberg, "Voice Signals: bit-by-bit", IEEE Spectrum, October 1973.

3. N.S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers", Proceedings IEEE, May 1974.

4. L.D. Davisson and R.M. Gray (Editors), "Data Compression", Benchmark Papers in Electrical Engineering and Computer Science, Vol. 14, Dowden Hutchinson & Ross, Inc., 1976.

5. B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Waveform", J. Acoust. Soc. Amer., 50(2), 637-650, 655(1971).

6. P.A. Bello, R.N. Lincoln, and H. Gish, "Statistical Delta Modulation", Proc, IEEE, 55(3), 308-316, 319(1967).

7. J.E. Abate, "Linear and Adaptive Delta Modulation", Proc. IEEE, 55(3), 298-308(1967).

8. F. Jelinek and J.B. Anderson, "Instrumentable Tree Encoding of Information Sources", IEEE Trans. Inform. Theory, IT-17, 118-119(1971).

9. R.J. Dick, T. Berger, and F. Jelinek, "Tree Encoding of Gaussian Sources", IEEE Trans. Inform. Theory, IT-20(3), 332-336(1974).

10. A.J. Viterbi and J.K. Omura, "Trellis Encoding of Memoryless Discrete-Time Sources with a Fidelity Criterion", IEEE Trans. Inform. Theory, IT-20(3), 325-332(1974).

11. J.B. Anderson and J.B. Bodie, "Tree Encoding for Speech", IEEE Trans. Inform. Theory, IT-21(4), 379-382(1975).

12. R.G. Gallager, "Tree Encoding for Sources with a Distortion Measure", IEEE Trans. Inform. Theory, IT-20, pp 65-76, January 1974.

13. L.N. Kanal, "Patterns in Pattern Recognition: 1968-1974,"IEEE Transactions on Information Theory, V. IT-20, N. 6, pp 697-722, November 1974.

* References 5-11 can be found in reference 4.

APPENDIX

The following is an example of the use of transfer and answer files to set-up and run a series of data processing programs as predetermined in the design of a test.

TRANSFER FILE

```
:RU,SIGEN,1G,2G
:RU,PLOT,1G,2G
:TR,3G
:RU,PLOT,1G,2G
:TR,3G
:RU,FFT,1G,2G
:RU,PLOT,1G,2G
:TR,3G
:RU,SIGEN,1G,2G
:RU,PLOT,1G,2G
:TR,3G
:RU,FFT,1G,2G
:RU,PLOT,1G,2G
:TR,3G
:RU,PLOT,1G,2G
```

The above transfer file runs a sequence of programs. Each program requires the answers to a basic set of questions to set its parameters. The following file provides those answers.

ANSWER FILE

```
SINE
256
MODULATED SINEWAVE
50% MODULATION
SI
.5
64
8
NO
NO
9
ER
PL,SINE
NO
EX
9
PL,SINE
YE
LA
SAMPLE NUMBER
MODULATED SINEWAVE
```