INSTRUMENT SCAN, PERFORMANCE, AND MENTAL WORKLOAD IN AIRCRAFT PILOTS

by

ALBERT THOMAS STEPHENS

B.S.A.E., University of Alabama
(1979)

SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1981

Signature of Author ....
Department of Aeronautics and Astronautics
September 11, 1981

Certified by ....
John R. Tole
Thesis Supervisor

Accepted by ....
Harold Y. Wachman
Chairman, Departmental Graduate Committee

To My Wife, Sandy

for teaching me the meaning of love

To John Tole

for teaching me the meaning of friendship

To My Parents

for teaching me the Way, the Truth, and the Life

INSTRUMENT SCAN, PERFORMANCE, AND MENTAL WORKLOAD
IN AIRCRAFT PILOTS

by

ALBERT THOMAS STEPHENS

Submitted to the Department of Aeronautics and Astronautics
on September 14, 1981 in partial fulfillment of the
requirements for the Degree of Master of Science in
Aeronautics and Astronautics

ABSTRACT

An experimental study was conducted to determine the relationship between level of performance on a skilled man-machine control task, the skill of the operator, the level of mental workload induced by an additional task imposed on the basic control task, and visual scanning behavior. The basic control task involved maintaining a general aviation flight simulator on a straight and level, constant sensitivity, Instrument Landing System (ILS) course with a low level of turbulence. A task employing an algorithm based on relative magnitudes of a sequence of numbers was used to increment the subject's mental workload. This level of loading for various conditions was also estimated using a side task. The subject's lookpoint on the instrument panel during each ten minute run was computed via a TV oculometer and stored. A total of thirteen pilots of varying skill participated in two sets of experiments.

The results indicate an increase in fixation dwell times, especially on the primary instrument, with increased mental loading task. Analysis methods included examination of sequences of fixations. The percentage occurrence of the subject's most used sequences decreased with increased task difficulty for novice subjects but not for highly skilled subjects. Piloting and number task performance measures were recorded and a combined performance measure was computed. This was used in developing a model relating performance, skill, and mental workload. Entropy rate (bits/sec) of the sequence of fixations was used to quantify the scan pattern. It consistently decreased for most subjects over the four loading levels used. An exponential equation in task difficulty was found to be a good predictor of entropy rate. When solved for task difficulty, the equation provided perceived task difficulty of a subject which was related to mental workload. This relation was also employed in the performance/skill/mental workload model development. The resulting exponential model fit the data fairly well. The purpose of the model would be for performance prediction from scanning behavior. Analysis of the periodicity of the subject's instrument scan was accomplished using autocorrelation. Skilled pilots were found to scan their primary instrument in a periodic fashion, and the period was related to the interval between number task presentation. A similar result was not observed in novice pilots. This finding suggests that skilled pilots may handle the additional loading task in a much more systematic fashion that do novice pilots.

Thesis Supervisor: Dr. John Tole

Title: Principal Research Scientist in Health
Sciences and Technology

ACKNOWLEDGEMENTS

The work reported here was supported by NASA Co-operative aagreement NCCI-23. The guidance and friendship of Dr. John Tole of the Biomedical Engineering Center for Clinical Instrumentation was invaluable in the composition of this thesis. The personnel at NASA/Langley Research Center provided much needed assistance during the experiments. Special appreciation is extended to Dr. Randy Harris of the Automation Research Branch, NASA/Langley Research Center and provided assistance in data analysis. Dr. Arye Ephrath of the Department of Aeronautics and Astronautics provided a great deal of assistance in the theoretical analysis and much needed encouragement throughout the project. Dr. Alan Natapoff first suggested the use of information theory which greatly enhanced the data analysis, and Nerville Moray first suggested the verbal loading task which proved to be very effective. A special word of thanks is extended to all of my friends at the Man-Vehicle Laboratory and the Biomedical Engineering Center for Clinical Instrumentation who were always willing to lend a hand or word of encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

There was a time when flying was considered to be an "art" and pilots knew what to do from an information center located conviently in "the seat of their pants". Though these ideas still exist for the enthusiast, commercial and military avaition have dictated a more practical approach to this means of transportation. With the desire to be able to fly in low visibility conditions came the need to provide the pilot with information about the aircraft's attitude and position, and the desire to land in these conditions required even more accurate position information. Thus the age of instrument flight began. Since that time, the complexity and size of the aircraft have increased accompanied by an increase in the amount of information provided to the pilot. The introduction of highly reliable electronic components into the cockpit opened the door to the automation of many manual tasks previously performed by the pilot. A gradual shift was thus taking place in the piloting task and the type of loading to which the pilot was subjected.

There are two types of workload, physical and mental, which sum up to a total workload for an individual. In the early stages, physical work had been the primary workload factor for the pilot, but the events described above have shifted the emphasis from physical to mental workload as the primary source of loading for the pilot (Sheridan 1978, Hay et al 1978). The question arises as to the significance of these events. Why should one be concerned about the amount of mental workload a pilot, or anyone for that matter, has to cope with? The answer to this is straightfoward and well documented. It has been shown by numerous investigators (see Albanse 1977, Wempe & Baty 1968, White & Ware 1969, Klein and Cassidy 1972, Kennedy 1975, Wickens 1974 and many more) that a decrease in performance on a primary task will eventually occur with a sufficient increase in workload. For highly skilled pilots, performance tends to remain constant with increased mental workload until a "critical" level is reached, at which time a significant decrease in performance is usually observed. The other end of the spectrum is also a problem. In tasks that require less participation by the operator, for example monitoring tasks such as sonar, highly sophisticated autopilots, and nuclear reactors, it has been shown that the lower the mental workload, the higher the probability of missing an important event(ie a decrease in performance). Therefore, as demonstrated in Figure 1, there appears to be an optimal loading region which minimizes the probability of error. If this theoretical plot is, in fact, correct, mental workload may provide a useful metric for the probability of error or performance of a task.

Given the importance of keeping mental workload within some "optimal range" and the fact that mental workload in the cockpit has been on the increase, it would appear that one would want to take this factor into consideration when designing new instruments or crew procedures for the cockpit. A quantitative measure of mental workload would provide a means of designing the procedures, displays, tasks, etc for the pilot in such a way that severe under- or over-loading would not occur during any anticipated circumstances.

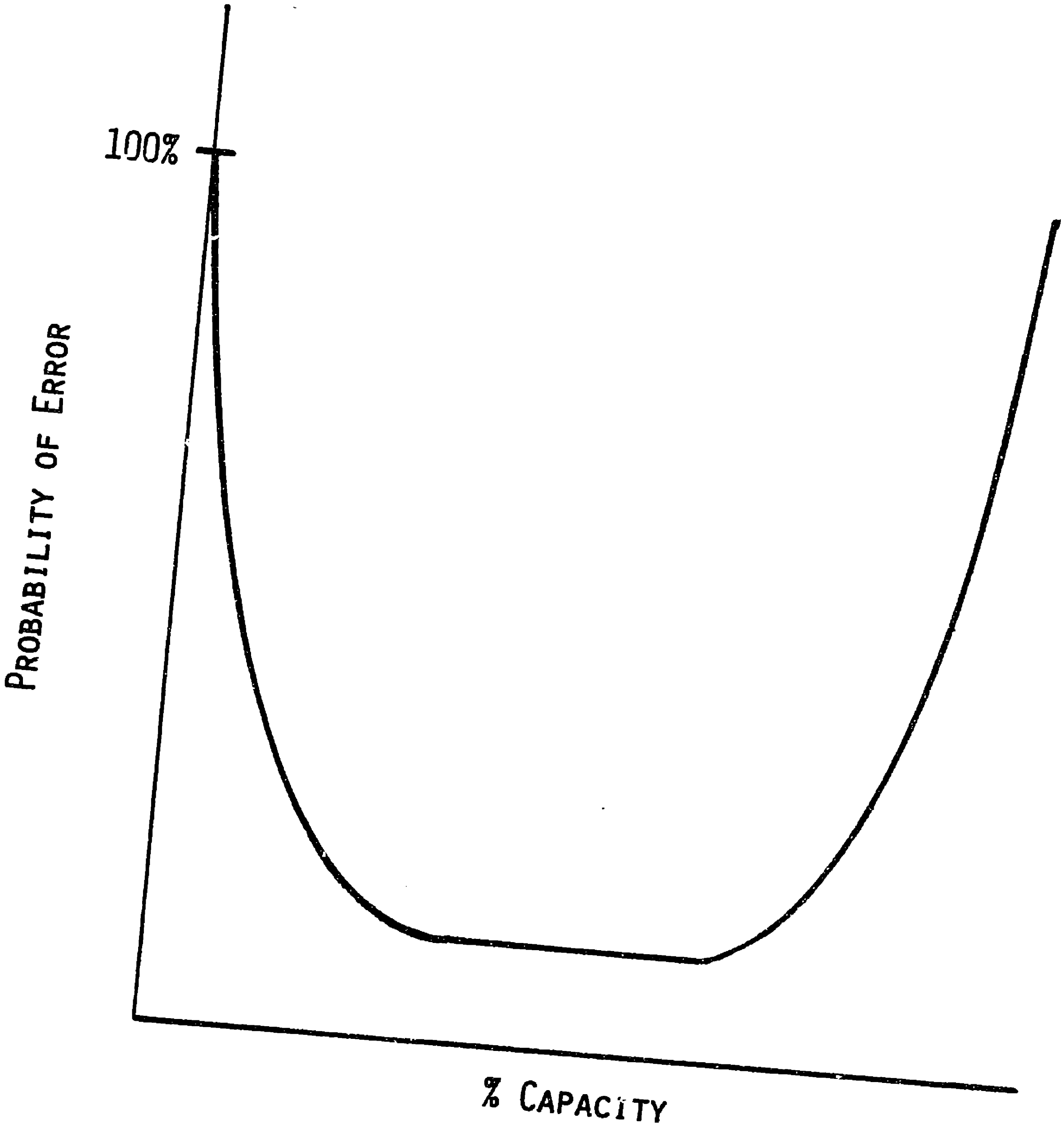


Figure 1 Probability of Error vs % Capacity

Before a measure can be developed, the relationship of the components of mental workload must be defined. Because of their interrelationships, workload, skill, and performance cannot be divorced from one another but must be studied together when attempting to understand the actions of any one of them. In most tasks, performance is a function of skill level, workload, inherent difficulty of the task or tasks being performed, and random noise. Skill is a function of recency of experience and total experience on a task as well as the innate skill of the operator. Sources of physical loading in the cockpit could be the amplitude and frequency of control wheel inputs, the amount of g-forces encountered in high performance aircraft, the amplitude and frequency of throttle inputs, ambient noise, cabin temperature,etc. As mentioned previously, the physical workload of the pilot is much less than the mental workload and will therefore be neglected. Sources of mental workload in the cockpit are instrument scanning, radio and cockpit communications, navigational computations, emergencies, etc. Mental workload is, therefore, some function of the operator`s skill, psychological/emotional state, and the amount of information retrieval, processing, and storage required. A functional definition relating mental workload to the skill of the operator and to his performance on some task will provide us with the information we need. The actual functional relation and other details will be discussed in a later chapter.

Fundamental to the work in this thesis is the hypothesis that some change takes place in the pilot's instrument scan as his mental workload varies. Before proceeding, it is important to explain the reasoning behind this hypothesis. Since the claim has been made that mental workload is dependent upon the amount of information processing required, a knowledge of the changes in information processing may lead to a measure of mental workload. The pilot has many sources of information input but the most important one during instrument flight is probably the visual pathway. During instrument flight, some sensory inputs may even provide false information such as the condition of vertigo which results from conflicting visual and vestibular information. Therefore, the pilot is trained to rely completely upon his instruments designating the visual pathway as the sole source of information. The pilot obtains information about the state of the aircraft by cross-checking or scanning the flight instruments. The exact method of scanning the instrument panel varies from pilot to pilot but there are some basic features common to many pilots. Since the pilot's information retrieval depends upon his scanning pattern, and his mental workload is related to his information retrieval and processing, it is hypothesized that there will be a change in some features of the pilot's instrument scan as his mental workload is increased.

The pilot may be considered an information receiver. As such, he will have a fixed channel capacity above which no more information may be processed. This maximum channel capacity may coincide with the "critical" workload level mentioned earlier. Above this level, should the pilot attempt to process any more information, a precipitous decrease in his task performance may result.

In response to the needs stated above, this thesis explores the possibilities of using a measure of the pilot's scan pattern to determine the level of his mental workload. A hypothesis for the relationship of performance, skill, and mental workload was developed, and a graphaical representation of this hypothesis is shown in Figure 2. This figure is discussed in detail in Chapter 6. Experiments designed to explore the interrelationships between skill, performance, and mental workload were carried out. The procedures, results, and model developed will be presented in this work. The next two chapters discuss the many methods of workload measurement which have been utilized by investigators in the past and the theory on which a majority of the analysis used in this thesis is based.

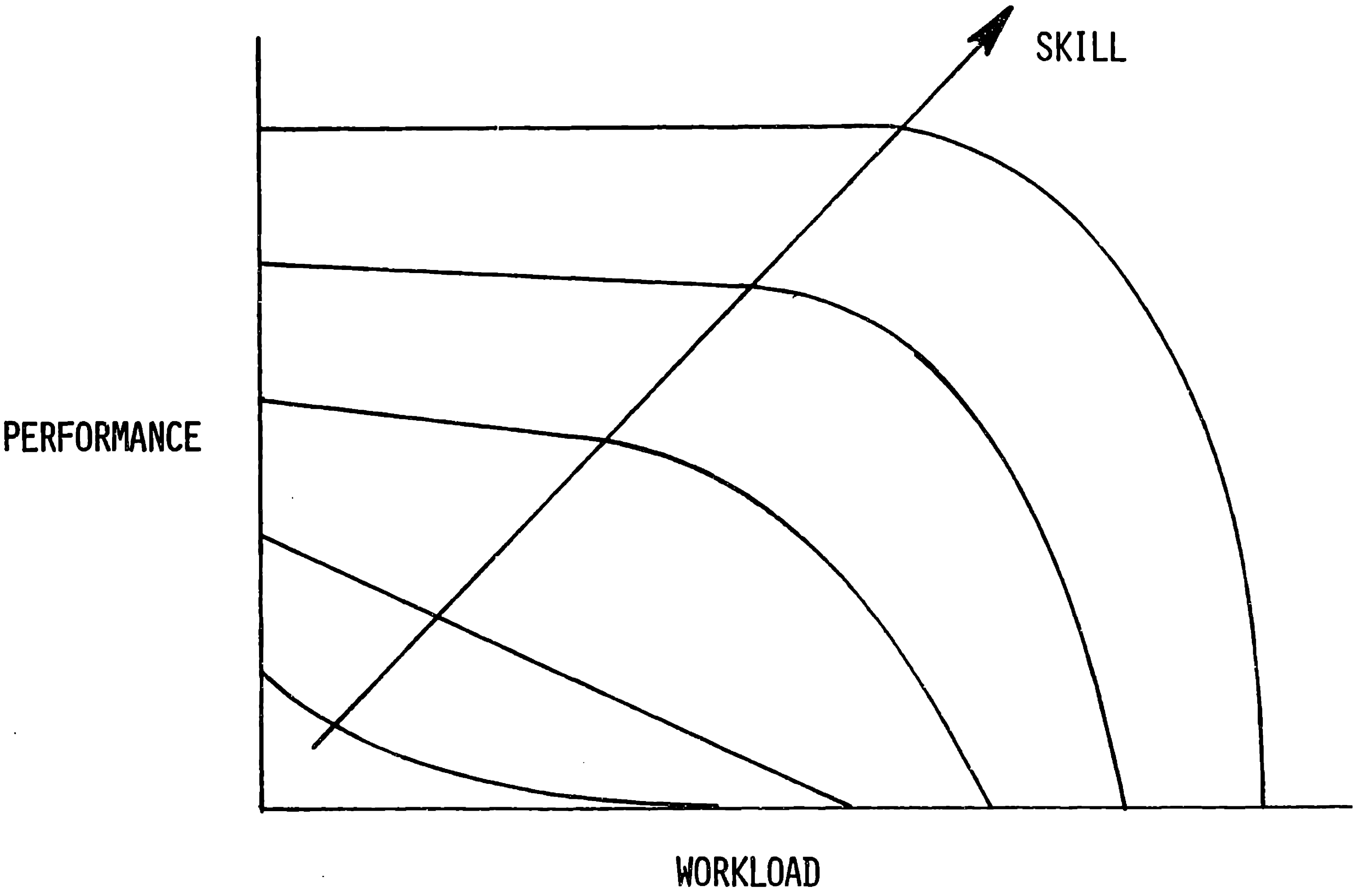


FIGURE 2 GRAPHICAL REPRESENTATION OF HYPOTHETICAL RELATION BETWEEN TASK PERFORMANCE AND MENTAL WORKLOAD

CHAPTER 2

METHODS OF MEASURING MENTAL WORKLOAD

A multitude of methods have been employed by numerous investigators over the past 20 years in attempts to quantify mental workload. A detailed discussion of all or part of these methods would be difficult, lengthy and is unwarranted in this work. However, an overview of the methods of measuring mental workload in use today will give the reader a better perspective for the method to be presented in this thesis. A good collection of methods for measuring mental workload in the aircraft/simulator environment may be found in Wierwille and Williges (1978). They have suggested four main categories into which mental workload measurements may be divided. These categories are 1) Subjective opinions 2) Spare mental capacity 3) Primary task measures and 4) Physiological measures. There are many different methods within each category, each having its own advantages and disadvantages for use in a given situation.

Subjective Opinion

Subjective opinion methods have been a widely used measure of mental workload. They usually take the form of rating scales, questionnaires or interviews. Of these, the rating scales have been the most popular. Many scales have been the workload equivalent to the Cooper-Harper aircraft handling scale (Katz, 1980). In many cases, workload rating scales are used in conjunction with other measurement methods. Some success has been reported using rating scales but most measures tend to be situation specific. Advantages of subjective studies are their non- invasiveness and their acceptability by pilots (Katz, 1980, Wierwille, 1978, Sheridan and Simpson, 1979) with disadvantages being confusion of mental and physical workload, variability with emotional state, experience and learning, and the difficulty of properly calibrating a subject to estimate his own mental workload.

Spare Mental Capacity

The concept of spare mental capacity has also been widely used and is based on the notion of a pilot having a limit on the amount of information he can process at any given time. This concept assumes that the pilot obtains information in a single-channel sampling fashion. Any time not devoted to the primary task(s) is thus theoretically available or spare. Spare mental capacity methods may be divided into two categories, task analytic and secondary task.

Task analytic methods take all of the tasks that a pilot will perform and assign a length of time required to complete each of the tasks. If the sum of all the times for all the tasks is less than the time available then the pilot has spare capacity. Another approach is to measure the sampling rate required by each instrument and assume that the dwell time on that instrument is a function of its information content. These times are then summed and used as discussed above. Both of these methods are lacking in validation experiments to confirm their ability to generalize in-flight workload measurement.(see Hay et al, 1978)

The most widely used spare capacity measurement method is the so-called side (or secondary) task. The usual instruction to the pilot using this method is to perform the side task only when he has time free from the primary task, which is usually flying a flight simulator. Assuming the pilot will be working at his capacity on the two tasks, the percent of work done on the side task is proportional to the amount of free time from the primary task. It is assumed that the primary task performance remains constant while the secondary task performance varies with spare capacity. Disadvantages of the side task include interference with the primary task, the assumption that the operator is working under maximum loading conditions, and the assumption of a linear relationship between the primary and secondary tasks.

Many types of secondary tasks have been used. Examples include light cancelling tasks, in which a pilot is asked to turn off two randomly occuring lights using some type of switch (Ephrath, 1975) and dual tracking tasks. Tasks in which a pilot is required to perform some mathematical/logical computation have also been used (Tole et al, 1981). Since the pilot is a manual controller in the loop, non-adaptive and adaptive tracking tasks have been employed to provide secondary tasks that are in some ways similar to the primary task. The light cancelling and mental math side tasks were used in the experiments in this thesis and will be discussed in more detail in a later chapter.

Primary Task Measures

Measures of performance on the primary task, without any additional information, have, for some time, been considered poor indicators of pilot mental workload. This is due to the observation that the skilled pilots, who are more often used in these experiments, tend to compensate for higher loading by working harder and thus maintaining a constant performance level. This observation appears to be true up to a certain point at which the performance drops off rapidly with any further increase in workload. Though this behavior does exist in aircraft pilots (Cooper and Harper, 1969 [illegible] is obvious that some part of their strategy must be changing [illegible] the constant performance. Both single and multiple measures of the primary task have been used but more success has been achieved with the multiple measures. However, even for multiple measures, conflicting results have been found. Measures such as RMS error from intended path, shift in frequency of tracking, RMS accelerations, and number of control reversals are a few examples of the many measures that have been tried. The major limitation of this method is that for highly skilled pilots, performance decrements on primary tasks occur only near and above the pilot's overload or "critical" level. This would provide information on high workload levels but none for low or moderate levels.

Physiological Measures

It appears as though any non-stationary physiological process of the human body that can be measured has been tested for correlations with mental workload. This is not surprising since the center that controls all mental processing, the brain, also controls all of the bodys physiological functions. The largest dilemma for these approaches have been the sorting out and understanding of the complex interactions taking place between and among all of the physiological processes when attempting to "prove" a cause and effect relation between a change in mental loading and a variation in a physiological parameter. Those measures showing promise for use in determining mental loading in aircraft pilots will be discussed briefly. One caveat should be expressed before proceeding. Until now, the methods discussed had little to do with actual physical contact with the subjects. Obviously, some of the measures about to be discussed will require the instrumentation of the subjects. The effects of this instrumentation on a subject`s mental state varies, even though they have agreed to allow it, and should be determined to the best extent possible to aviod contamination of the data with non-task related stresses.

One of the most researched areas of physiological measures involves the heart and circulatory system. This is due to the fact that the heart is controlled by the autonomic nervous system. Consequently, the sympathetic system, which is responsible for preparing the body for stress situations, will, theoretically, cause changes in the heart`s function when workload is changed. Measures of this system include electrocardiogram (EKG) analysis, heart rate, heart rate variability, and blood pressure.

Heart rate, heart rate variability, and blood pressure have been extensively examined as possibly measures of mental workload (Jex and Allen, 1970; Spyker et al, 1971; Stackhouse, 1973; Mulder, 1973; Sayers, 1973). To date no conclusive evidence has been found to indicate that any of these measures correlate with changes in mental workload. This is probably due to the fact that these measures are influenced by many other factors besides mental workload. Hence, these measures show little promise as mental workload indicators.

Feature extraction from the electrocardiogram may provide a useful measure of workload. Features such as T-wave amplitude, the variance of the T-wave amplitude, R-R intervals, and "turning points" or changes in the direction of heart rate have been studied. Though feature extraction methods have come a long way, little systematic work has been done to correlate them with mental workload (Wierwille and Williges, 1978).

Another physiological measure which shows promise as a workload measure is the evoked cortical potential (ECP). This measure is based on the hypothesis that the neural activity associated with information processing in the brain may be singled out and recorded. The data is obtained from an electroencephalograph signal. ECP peaks are labeled by their direction (positive or negative) and their latency (time between triggering event and peak occurrence). The magnitude and latency of the

P300 (P = positive; 300 = latency in milliseconds) peak have been shown to vary as mental tasks become more difficult. Problems of validating the results and application of the procedure still exist, but the possibility of a measure that reflects mental activity directly is needed (see Spyker et al, 1971; Wickens et al, 1977).

Examination of a pilot's instrument scanning behavior may provide a measure of mental workload. As discussed in the introduction, this is due to the relation between scanning behavior and information retrieval for a pilot during instrument flight. Therefore, changes in the amount of information processing, which infer changes in mental workload, should produce changes in the scanning behavior of the pilot.

Many different methods have been used in the aquisition of the pilot's lookpoint on the instrument panel. The earliest methods employed photographic techniques which were slow and difficult to analyze. More recently, electrooculography (EOG) methods have been used in conjunction with head position monitors to determine the pilot's lookpoint. The most significant development in eye movement data came with the advent of the TV oculometer, a device which uses infrared reflections from the eye to compute lookpoint. This method was utilized in this work and will be discussed in detail in Chapter 4 (see Young and Sheens, 1975 for a survey of eye movement recording methods). Eye scanning analysis of a pilot's instrument scanning pattern has veen studied by many investigators. The pioneering work in the field was done by Jones, Milton, and Jones, et al (1946) who employed a sequential photographic method to examine the frequency, duration, and sequence of eye fixations during instrument flight for United States Air Force pilots. They reported finding differences in the frequency and dwell time of fixations on instruments, and among scanning behaviors of pilots. Senders, et al (1966) presented a model describing the sampling behavior of the human visual system. They found that the frequency of observation of an instrument is a function of the bandwidth of the signal. Weir and Klein (1970) examined the differences in scanning patterns for pilots flying a manual ILS approach and those flying an ILS approach using a flight director system. They showed that the different configurations produced changes in the dwell time on primary instruments and in the overall scanning rate for experiments done in a fixed-base flight simulator. Later investigators (Corkindale, 1974, Strother, 1973) have used eye movements to study the performance and usage of secondary tasks. A concept of visual free time similar to spare mental capacity was employed in these experiments. DeMaio, et al (1976) measured reaction times to situations presented in slides of an aircraft instrument panel. It was found that experienced pilots did not use a standardized scanning pattern and that when presented with a novel scanning task, could develop an efficient scanning pattern faster than non- experienced pilots. Krebs and Wingert (1976) varied vehicle stability and wind gusts in a fixed base flight simulator and examined scanning pattern changes in pilots. Using an oculometer they found high variability in the average percent of time spent on various instruments. However, they found small, but systematic, changes in blink rate, pupil diameter, fixation duration, and saccade length with changes in task difficulty. Waller (1976) found a relationship between measures of instrument scanning behavior and Cooper-Harper rating scale scores using stepwise regress in analysis. The measures included total time on instruments and the number of transitions between certain instruments on the panel. Dick and Bailey (1976), also using the oculometer, found that actual scan patterns and pilot's opinions of their scan pattern differed. Dick (1980) demonstrates that pilots scan instrument "clusters" providing categories of information as opposed to individual instruments.

In addition to eye-scanning behavior, many investigators have examined other visual phenomenon such as blink rate and pupil diameter. Holland and Tarlow (1972) showed that blink rate was inversly proportional to auditory mental loading. Pupil dilation, with increase in mental workload, has been observed by numerous investigators (Beatly, 1976).

Summary

This chapter describes some of the methods which have been used to study mental workload. These methods fall into four main categories which are 1) subjective opinions 2) spare mental capacity 3) primary task measures and 4) physiological measures. Subjective opinion methods usually take the form of rating scales, questionnaires, or interviews and appear to be acceptable to pilots. Spare mental capacity expresses a pilot's workload as a percentage of the time available after performing his primary task. It is studied using task analytic methods which compute the difference between the time required and the time available for a task and by using side task methods which measure the amount of time available to perform a secondary task. Primary task measures use performance measures of the primary task to estimate mental workload of pilots. This, unfortunately, does not work well for highly skilled pilots who tend to compensate for higher workload levels by working harder. The final area discussed is that of physiological measures. A large number of physiological processes have been used including heart rate, heart rate variability, blood pressure, EKG feature extraction, evoked cortical potentials (ECP), and eye movements, each of which has shown some promise as a workload measure.

CHAPTER 3

EXPERIMENT FUNDAMENTALS

This thesis had two primary objectives:the hypothesis that a change in scanning behavior accompanies increased workload and the development of a model relating performance, skill, and workload. A set of experiments was designed with these objectives in mind. The experiments in this thesis were both performed at the NASA/Langley Research Center, Flight Management Branch, in Hampton, Virginia, making use of their flight simulator and oculometer facilities. Three factors were basic to the experiments: 1) a piloting task, 2) a mental loading task, and 3) a workload calibration side task. Each of these basic components will be discussed below.

In the past, investigators have usually chosen some relatively complicated piloting task, for example an ILS approach, for their primary task in workload experiments. It was assumed that this would more closely represent a real life situation than a more basic maneuver. However, the ILS approach represents a constantly changing task difficulty as touchdown in approached. This change in the primary task makes it difficult to accurately control the amount of mental workload on a pilot. Therefore, the decision was made to use a constant sensitivity glide slope and localizer in the primary task and vary the mental workload using a separate, constant loading task the difficulty of which could be varied. Such conditions of constant mental loading over a period of time were deemed necessary in order to provide enough data to determine whether a steady state scanning pattern exists.

Our experiments are thus concerned with the relationships between "steady-state" levels of the various independent parameters: piloting performance, skill, and workload. Mental workload may vary from instant to instant and while such variations are potentially of great importance we decided that we were not in a position to measure rapid changes at the present time. We sought a representative constant piloting maneuver which might be realistically expected to occur for periods of up to 10 minutes in actual flight. This run length was chosen as an estimate of the minimum amount of time required to provide a sufficient number of fixations to satisfy the assumption of steady state conditions. The piloting task required the pilot to fly a precision straight and level course with zero degree glide slope and live localizer while maintaining a constant heading and airspeed. In order to force some pilot vigilance on this task, a low level of turbulence was also introduced in each run. A desktop general aviation instrument flight simulator (Analog Training Computers ATC-510) was used to simulate these flight manuevers. Pilot lookpoint on seven instruments (Attitude Indicator `ATT`, Directional Gyro `DG`, Altimeter `ALT`, Vertical Speed Indicator `VSI`, Airspeed `AS`, Turn and Bank `T*B`, and Glide Slope/Localizer `GSL`) was recorded as indicated below. Performance as measured by course deviation was recorded.

The mental loading task was chosen so as not to directly interfere with the visual scanning of the pilot (i.e. the task would not require the pilot to look away from the instruments) while providing constant loading during the maneuver. The purpose of the mental loading task was not that of a side task, but as a second primary task. Assuming the piloting task was constant, the amount of mental workload was controlled by the difficulty of the mental loading task. Katz (1980) demonstrated that ATC communications increased the mental workload of the pilot. This suggested that a verbally presented task requiring some degree of mental processing could be used to control the amount of mental workload of the pilot.

The task used required the pilots to respond verbally to a series of evenly spaced three-number sequences (Wittenborn, 1943) presented to them over a speaker. The pilot was told that he must respond to each three-number sequence by saying either "plus" or "minus" according to the algorithm : first number largest, second number smallest = "plus" (e.g. 5-2-4), last number largest, first number smallest = "plus" (e.g. 1-2-3), otherwise, "minus" (e.g. 9-5-1). The interval between the three-number sequences provided the means of varying the mental loading. The larger the interval, the easier the task and, therefore, the lower the mental workload. The pilot was instructed to give the number task priority equal to that of the piloting task as if the verbal questions represented a constant rate of radio communication. The actual score on this task was not obtained in the first set of experiments described in Chapter 5. It was decided that if the pilot conscientiously attempted the numbers, the actual score was not important since the mental effort was still being applied. However, the performance, as monitored by the experimenter, was very good. The performance was recorded for the second set of experiments (see Chapter 7) by having the pilot press a 3-position rocker switch up for plus and down for minus.

The amount of mental loading imposed on the pilot by the number task was calibrated using a side task. The runs made with the side task were not used in the scanning analysis, however, due to the alteration of normal scanning caused by the task. The side task employed a CRT which could display an asterisk appearing in the upper half or in the lower half of the screen. The display was mounted to the left of the simulator just outside the pilot's peripheral view. The asterisk appeared at random intervals between one and three seconds and remained on for one second (Ephrath, 1975). The pilot was told to turn the symbols off by using a three position rocker switch on the control grip. Moving the switch up turned the upper asterisk off, down turned the lower asterisk off. This task was done only when the pilot had time left from performing the primary tasks of flying the airplane and answering the number task. Thus the number of correct responses on the side task gave a measure of the residual capacity of the pilot from which a workload index could be calculated. The expression used to calculate the workload is given below (Ephrath, 1975). The constants were obtained using the best least squares fit weighting coefficients.

$$WLX = \frac{(.780)(RT) + (.626)(MISS)}{(.780 + .626)(NSTIM)} \times 100 \text{ percent} \qquad (1)$$

where:

WLX = workload index

RT = cumulative response time (seconds)

MISS = number of incorrect responses

NSTIM = total number of stimuli (symbols) presented

Summary

There are three main features of the experiments performed in this thesis. They are a piloting task, a mental loading number, and a workload calibrating side task. The pilots were required to fly a precision straight and level path for ten minutes using a desk-top, general aviation flight simulator. The mental loading number task, used to control the amount of mental workload the pilots are under, consisted of a verbally presented mental arithmetic task. The calibrating side task involved the cancelation of one of two lights that came on at random intervals.

CHAPTER 4

EXPERIMENTAL APPARATUS

This chapter describes the equipment used in the two experiments performed for this work.

Oculometer

The pilot's eye was tracked in all experiments with a Honeywell oculometer which had been highly modified by NASA/Langley Research Center to determine pilot lookpoint on the instrument panel. Four subsystems are involved in the determination of pilot lookpoint; 1) the oculometer 2) the manual control station 3) the head position monitor and 4) the video pointer assembly.

The oculometer consists of an electro-optical (E-O) sensing unit and a NOVA 1220 minicomputer. The E-O sensing unit generates an infrared beam which is reflected by the subject's cornea and retina. The reflection from the corneal surface produces a point while the retinal reflection backlights the pupil. The beam tracks the eye within a 1-foot cube with the aid of servo-driven, computer controlled mirrors which also return the reflections to a television vidicon tube. This picture is then scanned to determine the location of the two reflections. The relative position of the two reflections may then be used to calculate the subjects' lookpoint. The calibration of subjects is based on data from a test eye which was carefully set up and evaluated. Each subject would be asked to look at a series of pre-selected points on the instrument panel and values for these locations would be stored. The subject's values would then be compaired with the "ideal" values from the test eye, and a set of coefficients for a linearization model would be computed that best fit the subjects data with the data from the test eye. This provided an accuracy of better than 13 mm on the instrument panel. The oculometer is capable of tracking in a +- 30 degree horizontal and 30 degree to -10 degree vertical eye angle with respect to the tracking mirror position (Harris, 1979).

While the oculometer provides the basic functions of the system, the other three subsystems provide monitoring support. The manual control station provides the control capability to the operator to regain track of the pilot's eye in the event that it is lost. The operator has a pencil stick controller to adjust the mirror angles and a control to adjust the focus in case of a forward or backward shift in the pilot's head position. The station also includes an eye position TV monitor which displays the raw video output of the oculometer.

The head position monitor shows the operator a view of the pilot's head to assist in reaquiring the pupil image in the event of head movement rapid enough to escape from the oculometer monitor. The subsystem includes an IR vidicon camera which will pick up both the subject's head and the infrared beam. It also contains a head position TV monitor which displays the image of the pilot's head and shows a bright spot indicating the position of the IR beam.

The video pointer assembly combines the instrument panel image with a superimposed symbol to indicate the center of gaze of the pilot. This image may be monitored during a run and stored on videotape for later reference. This subsystem requires a video camera mounted over the pilot's shoulder, a video pointer to generate the lookpoint symbol, a TV monitor to display the combination of signals, and a videotape unit to provide a permanent record of the information.

Simulator

An Analog Training Computers' ATC-510 desk-top flight simulator was used in both sets of experiments. The ATC-510, pictured in Figure 3, is a procedures trainer for light, single engine, fixed pitch prop, fixed gear, IFR equipped aircraft. The simulator was modified to allow for constant sensitivity glide slope and localizer beams to provide a constant difficulty task for a period of ten minutes. This was accomplished by commands input to the simulator via the transponder and NAV/COM settings. The transponder was set to 2002, the COM frequency was 123.6 and the NAV frequency (which determined the course heading) was 108.3 for Phase I and 109.1 for the Phase II experiments. Signals from the instruments were obtained and fed into a 14-channel FM magnetic tape recorder to provide performance data for the primary task. The simulator was equipped with a turbulence level control which was set to the first level above calm conditions during the experiments.

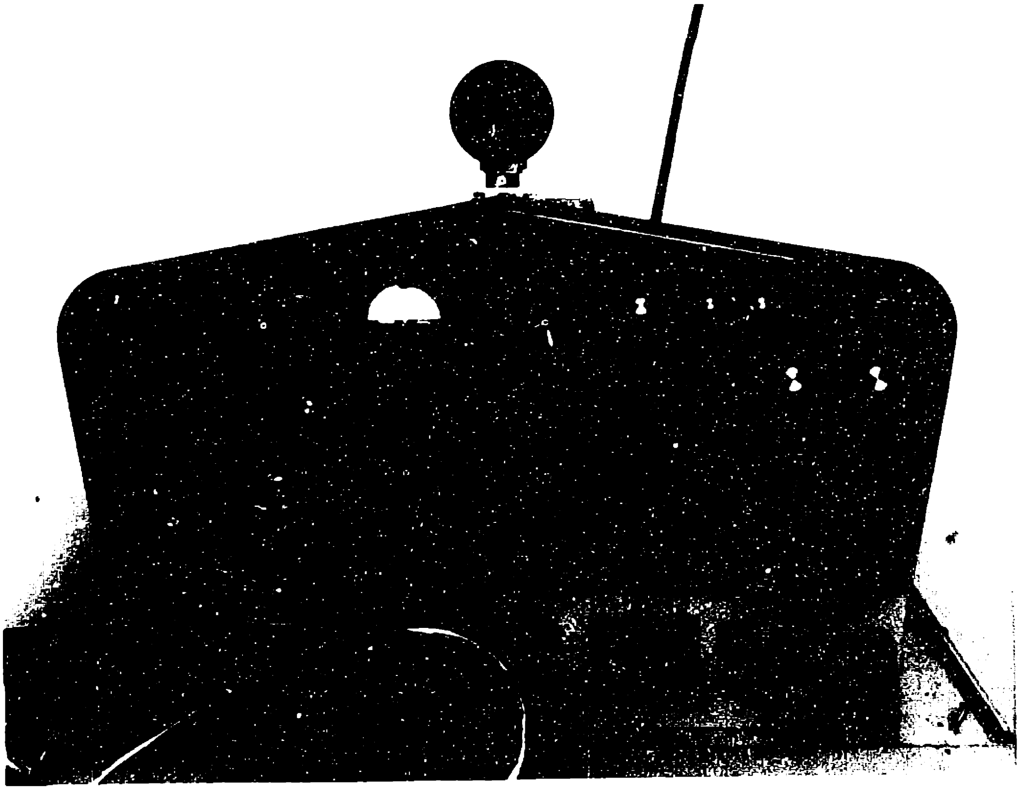


FIGURE 3 ATC-510 FLIGHT SIMULATOR

Microprocessor Matrix

A microprocessor matrix system based on the Intel 8085 was utilized for the stimulus presentation and data aquisition during the experiments. The single bus system supported 64 K of memory (RAM), two magnetic cassette tape drives for mass storage, an A/D converter for signal aquisition, and an electro-static printer. The operating system/higher level language available on the system was STOIC (Stack Oriented Interactive Compiler; similar to FORTH). The basic software package for controlling the experiments was developed by Dr. John Tole and may be found in Appendix A. Corrections and additions to the package were made by the author as familiarity with the system and language were gained.

Digital Synthesized Voice Unit

A Texas Instruments SPEAK & SPELL digital synthesized voice module was integrated into the microprocessor system to provide automated number task presentation during the Phase II experiments. The chip has a set vocabulary of approximately two hundred words as well as the numbers zero through nine which are accessed from the module, when modified to operate on the microprocessor system, by sending the corresponding address to a special buffer. The module, which plugged into the matrix system, provides a jack for a speaker used in presenting the number task to the pilots.

Summary

This chapter describes the equipment used in the experiments performed in this thesis. The pilot's lookpoint was obtained using a TV oculometer system located at NASA/Langley Research Center. The piloting task flown in the experiments used an ATC-510 desk-top, general aviation flight simulator also located at NASA/Langley Research Center. The data aquisition and analysis and the mental loading task presentation were controlled by a 8085 based microprocessor matrix system. In the second phase of experiments, the mental loading number task was produced using a Texas Instruments Speak and Spell speech synthesis voice module.

CHAPTER 5

PHASE I EXPERIMENTS

Procedures

The first phase of experiments were designed to 1) demonstrate the existance of changes in the scanning behavior of subjects with changes in mental workload and 2) evaluate the effectiveness of the computer controlled data acquisition system and the systems that interface with it. The experiments were carried out during the summer of 1980 at the NASA/Langley Research Center. The equipment, discussed in the chapter on experimental apparatus , was arranged as shown in Figure 4. The ATC-510 simulator was placed on a table located in front of an adjustable seat. These were then mounted on a platform to assure stability of the pilot and to aid the oculometer in tracking the pilot's eye. The oculometer's electro-optical (E-O) head was mounted independent of the simulator/chair and was located just below the turn and bank indicator on the lower left side of the instrument panel. It was sufficiently removed from the instruments so as not to interfere with the scanning of any instrument. The subject head position camera was located on top of the simulator instrument panel and aimed at the subject's face. The video pattern camera was mounted on an independent stand over the pilot's left shoulder to provide a view of the instrument panel. The control wheel on the ATC-510 was extended and rotated 180 degrees so that it would not block the infrared beam from the E-O head or the instruments from the video pointer camera's view. The CRT for the light-cancelling side task was located to the left of the simulator/chair assembly just outside the pilot's peripheral view. The

LABORATORY EQUIPMENT

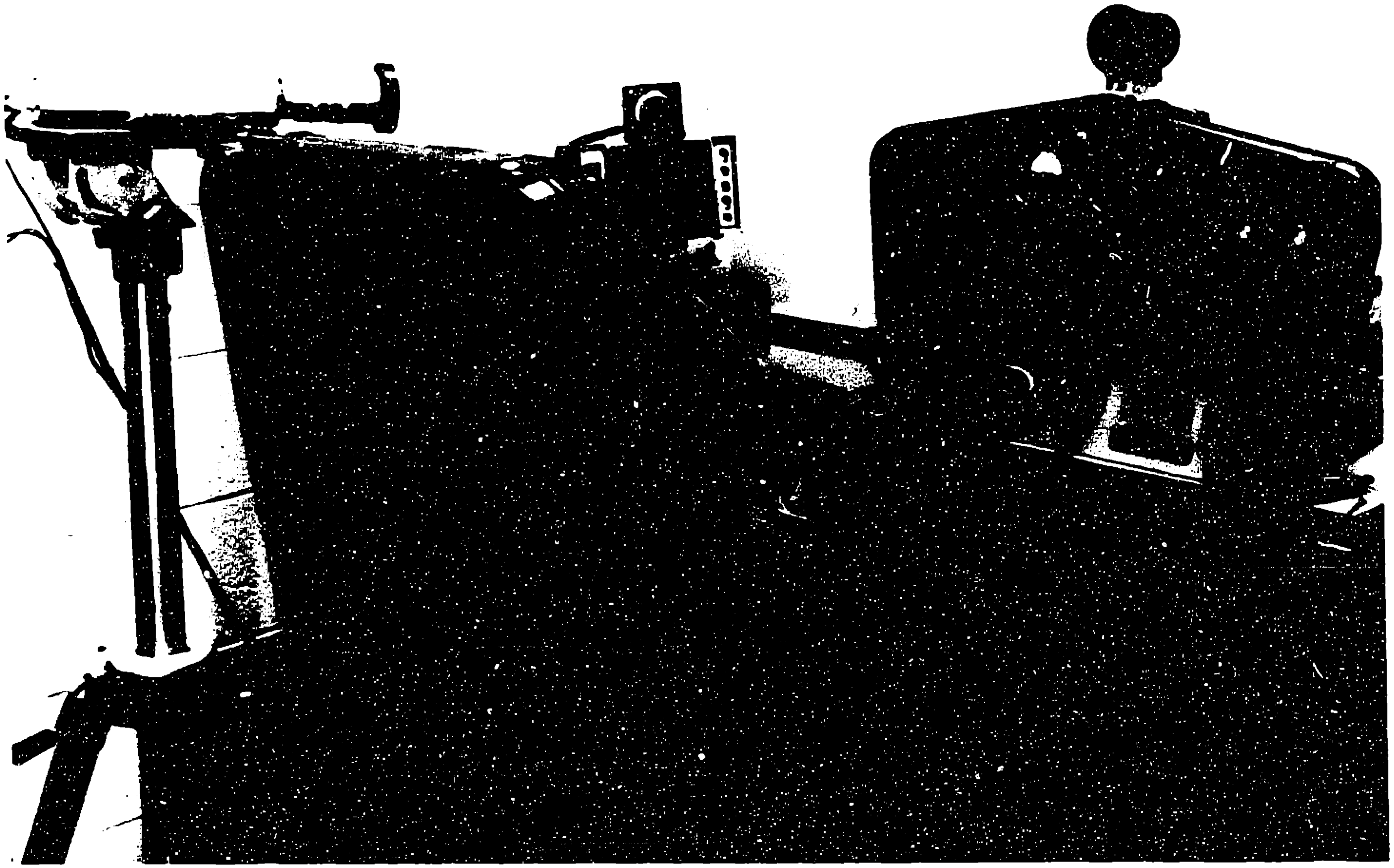


FIGURE 4 ARRANGEMENT OF EXPERIMENTAL APPARATUS

fourteen-channel FM tape recorder was located to the right of the simulator/chair assembly. The data acquisition microprocessor and its peripherals were located on a cart behind the pilot so that the operator could monitor the experiments without interfering. The other sub-systems of the oculometer described earlier were located to the right of the FM tape recorder. The location of all the experimental equipment in one room was somewhat crowded but greatly aided the calibration and monitoring process.

Calibration of the oculometer was required at the beginning of each session for each pilot. It required varying lengths of time to complete depending on the quality of the signal a pilot's eye produced. A brief description of the procedure follows. After the pilot was seated and had made himself comfortable, his eye was captured by the oculometer operator. The operator would then have the pilot look at three calibration points, X, Y, and null. These points were then used in a calibration routine to produce output scale factor coefficients. The pilot was then instructed to look at a series of preselected points on the instrument panel to provide data for the linearization routine. This provided coefficients for the model relating the pilot's coordinates for lookpoints with the "ideal" coordinates. The linearization procedure was repeated until there was good agreement between the pilot's intended lookpoint and the instrument obtained from the microprocessor. A more detailed discussion of the oculometer calibration procedure may be found in Merchant and Morrissette (1973).

The final result from the oculometer consisted of a voltage proportional to the X and Y coordinates of the lookpoint. The oculometer's processor converted these coordinates into an estimate of the instrument being viewed. This estimate was available as an analog voltage with twelve possible discrete levels. This voltage was sampled by an A/D card in the microprocessor. The value was sent to a routine which compared this value with voltage intervals associated with each instrument of interest. This routine returned the instrument name associated with the pilot's lookpoint.

Six subjects, varying in skill-level from non-pilot to a highly experienced NASA test pilot participated in these experiments. The subject numbers and approximate skill levels are as follows:

SUBJECT #	SKILL LEVEL
4	Highly skilled NASA test pilot
11	Highly skilled general aviation instructor pilot, NASA employee experienced in simulation experiments
9	General aviation pilot, current only in simulators
5	Current VFR rated general aviation pilot
10	Student pilot
7	Non Pilot

The pilot numbers are a result of the numbering system used in the pilot pool at NASA/Langley. Pilots 4-11 were the only pilots tested in these experiments and all resulting data is presented in this paper. Subjects 5, 7, 9 and 10 are referred to collectively as "novice" pilots in this paper. Subjects 9 and 4 were the only pilots with any previous experience in this particular simulator. The subjects were allowed to practice the flying maneuver and verbal task until they felt comfortable with the situation.

The mental loading task utilized in this set of experiments was presented to the pilots via a cassette tape recorder as described earlier. The levels of mental workload were no numbers for the lowest loading case, four second intervals between number groups for the medium level of loading, and two second intervals between number groups for the highest level of loading. These intervals were chosen, based on preliminary experiments, to provide no, moderate, and severe additional mental loading respectively. The pilots responded verbally to the numbers but the score was not recorded. Some of the pilots were also asked to complete runs which included the light-cancelling side task to provide confirmation of the effectiveness of the mental loading number task.

A typical session ran as follows. The first step was to teach the pilot the mental loading number task. The algorithm for responding plus or minus was described to the pilot and was written down if necessary. The pilot was then allowed to practice on a sample set of numbers until he felt comfortable with the task and could perform it with few or no mistakes. The pilot would then adjust the seat to the position he desired for flying the simulator. A practice run flying the simulator alone would be conducted for those pilots who had never flown the simulator before. This would last as long as the pilot requested in order to familiarize himself with the handling characteristics of the simulator. The instructions to the pilot were given as each part of the experiment was demonstrated. They were repeated as many times as necessary for clairity and were summarized at the end of the introduction. The next step of the introduction was to provide a practice run flying the piloting task and simultaneously responding to the mental loading number task. When the pilot was satisfied with his ability to handle these tasks, the light-cancelling side task would be added to the other two tasks. Once practiced with all three tasks, the pilot would be ready to begin the data runs. Before each data run the pilot would be told which tasks he was to perform and the interval size on the mental loading number task. When the pilot had the simulator under control and on course, the data collection and number task, if any, would begin. Each run lasted ten minutes. The pilot was given a 5 to 10 minute break between runs during which information about the data just obtained was printed out and was stored on magnetic tape. Usually, all six runs (three with side task, three without side task) would be made by a pilot in one session.

Results and Discussion

Workload Index

Ten minute runs with the side task were performed with three of the pilots. The workload index defined above was determined for each pilot for all loading levels (Table I). The index increased monotonically for all subjects with increased rate of presentation of the number task. The average workload index varied from 80 percent for no mental loading task to 92 percent at the 4 second interval and 96 percent at the 2 sec intervals. Although we were not able to evaluate the workload index with all pilots, the results with these three pilots did allow us to confirm quantitatively that the mental loading is increased as the interval between number presentations decreases.

Dwell Time Histograms

Perhaps the most striking effect observed in these experiments is the effect of the verbal loading task on the dwell time histograms of individual instruments for a given maneuver. In the four novice subjects, the dwell time on the primary instrument (the Attitude Indicator in all but the non-pilot who used Glide Slope/ Localizer) became progressively weighted toward extremely long dwells as the verbal task difficulty increased. Figure 5 shows the dwell time histograms for all pilots on the Attitude Indicator, Directional Gyro, Glide Slope/Localizer and Vertical Speed Indicator. First consider the plots for subject #5 who has intermediate skills. Note that for the no loading case, the dwell histogram on the Attitude Indicator of subjects #5, #9 and #10 has a fairly standard shape (7). When numbers are added to the piloting task, the dwell becomes longer and the mode of the histogram at 1/2 second begins to disappear. The effect is even more dramatic for 2-second interval case; the entire distribution is skewed toward extremely long dwells on Attitude as the pilot apparently begins to "stare" more and more at this instrument. Similar effects are seen for pilots 9 and 10.

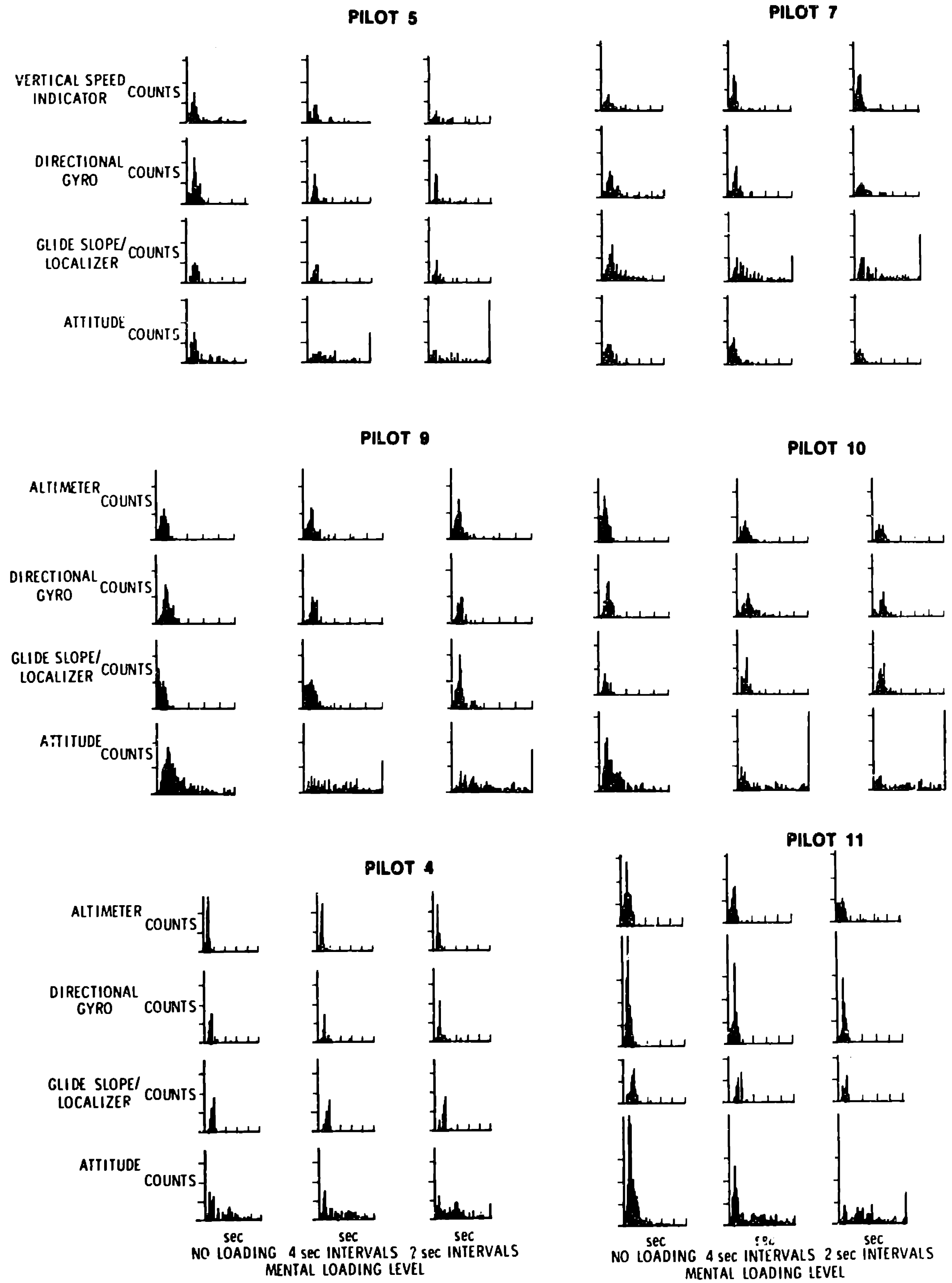


FIGURE 5

DWELL HISTOGRAMS

An interesting difference occurs for subject # 7, the non-pilot, however. This subject had no previous piloting experience and was only given enough practice to allow him to stay nominally on course during the precision straight and level maneuver. Note that this subject adopted the Glide Slope/Localizer as the primary instrument apparently in an effort to accomplish the precision task by keeping the needles centered. Even though the subject adopts the inappropriate instrument to accomplish the piloting task, the dwells on this instrument are affected in a manner similar to those on Attitude for the more experienced subjects.

The visual scanning behavior of the two subjects with higher levels of skill was also affected by the verbal loading (subjects 4 & 11 in Figure 5). However, the effect was much less than seen in the novice pilots. Figure 5 also shows the dwell time histograms for the NASA test pilot, subject #4. Note that he develops a slight stare on the Attitude Indicator for the highest loading condition but his histograms are otherwise unaffected. Subject #11, who had the next highest skill level, was somewhat more affected, especially at the highest loading level, as indicated by the histograms for the Attitude Indicator (Figure 5). Subject #11 uses a large number of short dwells on the Attitude Indicator under the no loading case. When the mental loading task is introduced at 4-second intervals, his distribution is shifted to somewhat longer dwells. However, there is still a very significant peak at around 1/2 second. The actual shift in dwell times is not as large as that seen in the novice pilot's histograms, even though there appears to be a large change due to the reduction in magnitude of the histogram peak.

The shift to longer dwells may also be demonstrated by looking at the percentage change from the no loading case in the number of dwells on the primary instrument that are 5 seconds or longer in duration as the mental workload is changed. The raw counts of such dwells are shown as the last element in the histograms. Table II shows the percentage change from the no loading case for each pilot. The percentage of dwells is seen to increase with decreasing skill level. This holds for all subjects except subject #7, the non-pilot. It should be pointed out, however, that subject #7 used a different primary instrument from the rest of the pilots and therefore had a completely different basic scan pattern from the other pilots. This fact may not allow direct comparison of the results from subject #7 with the other subjects.

The dwell time characteristics on secondary instruments were most affected in the novice subjects. The secondary instrument dwells are seen to change in a different manner than the primary instrument dwells. As opposed to the shift to longer dwells, as in the case for the primary instruments, the effect of loading in the secondary instruments is to decrease the number of looks at that instrument, perhaps an example of a phenomenon known as load shedding. The shape of some of the histograms changes under varying loading conditions. Subject #4 was the only subject whose dwell time histograms on secondary instruments were not affected by loading. Subject #11 appears to exhibit some load shedding, primarily on the Altimeter and Vertical Speed Indicator.

Fixation Sequences

It was also of interest to examine whether pilots develop a scan pattern or patterns during the constant flying maneuver in our experimental paradigm. Assuming that such patterns might exist, it appeared of interest to determine whether they might be altered by the addition of mental loading. The results from one method of studying this question are presented below.

As mentioned earlier, the oculometer provides as indication of instrument dwells as a function of time. If the dwell times on individual instruments are ignored, an ordered list of instrument fixations may be developed for each pilot for the various loading cases. These lists may be broken up into smaller segments (or sequences) of various lengths for easier analysis. Each different sequence may be considered a component of the overall scan pattern. One may hypothesize that those sequences which occur most frequently during the maneuver are those of most importance to the pilot and ones which might indicate an ordered scan pattern.

Examination of the results indicated that sequences of four instrument fixations were the longest for which there was a significant amount of repetition during a run, hence sequences of length four were chosen for analysis. The number of times each four instrument sequence occurred during a 10 minute run was obtained as was the total number of sequences of length four in the run. From these data, the percentage of occurrence was calculated for each observed sequence. For example there might be 800 sequences of length four in 10 minutes. If the sequence, ATT-DG-ALT-DG, occurs 40 times during the run, its percentage of occurrence would be 40/800 X 100 percent = 5 percent. In this fashion, the percentage of occurrence of all length four sequences in the no loading case was determined for each pilot. The 10 sequences which occurred most frequently for each pilot were arbitrarily chosen as indicators of the scan patterns normally used by the various pilots. The manner in which the percentage occurrences for these 10 sequences change for each subject as a function of mental loading is shown in Figure 6. Figure 7 plots the sum of these percentages across loading for all the subjects. It is important to note that the sequences used as the basis for calculation for all conditions are the 10 most frequent for the no loading case. Each line beginning at the no loading case and ending at the 2-sec interval case represents the same sequence.

Several interesting observations may be made by comparing the plots of the skilled pilots (Figure 6) with those of the novice subjects (Figure 6). A difference may be seen between the two groups in the percentage of occurrence of the most often used sequences. The first ten sequences used by the skilled pilots comprise over 50 percent of their scan pattern (see sum in Figure 7). The usage of these ten sequences is relatively constant with changes in loading, suggesting that the patterns are not disturbed by the verbal number task. This finding is certainly in keeping with the intuitive development presented in the introduction which suggested that it should be difficult to interfere with a skilled task. The novice pilots` results differ in several respects from those of the skilled subjects however. The ten

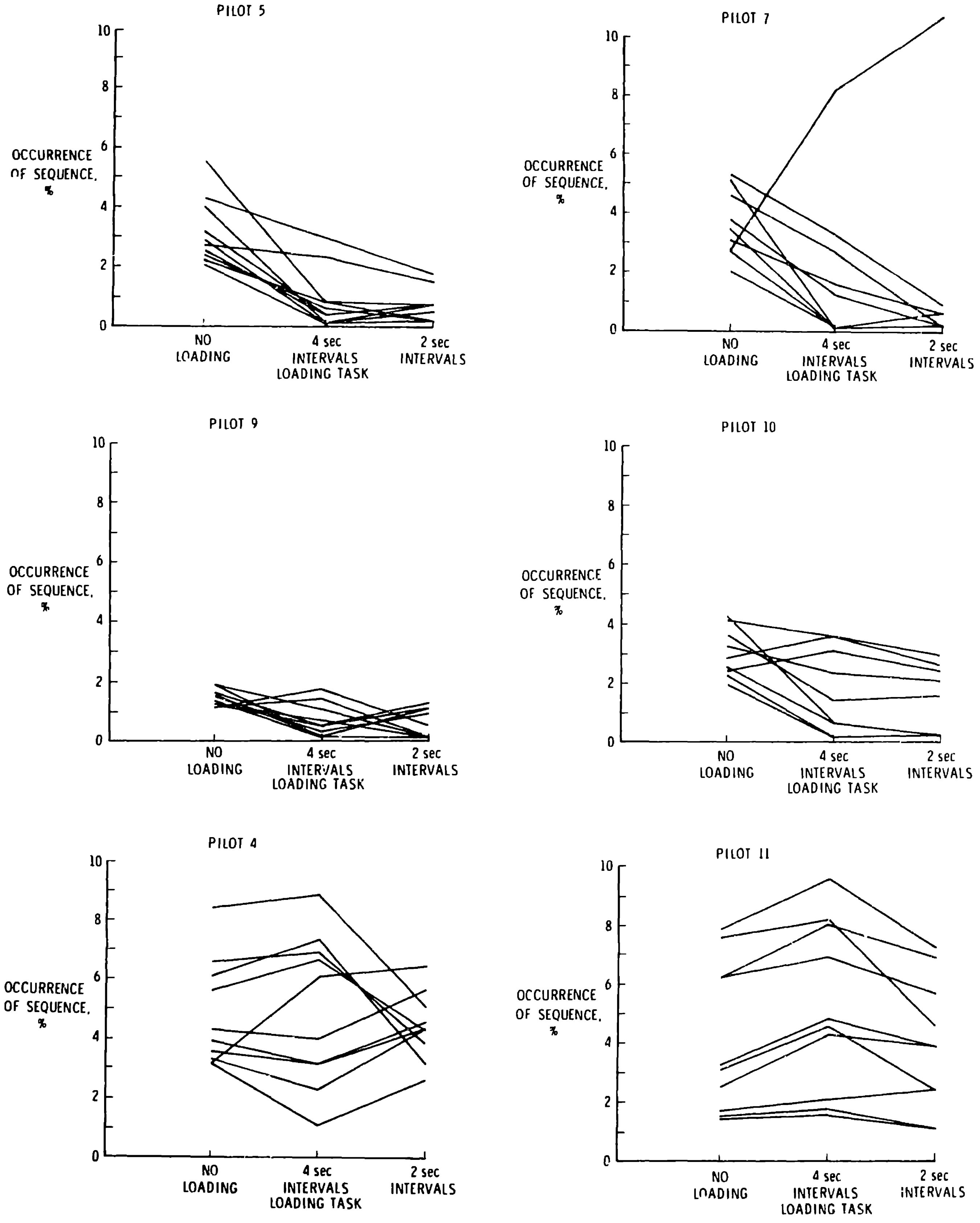


FIGURE 6

PERCENT OCCURRENCE OF SEQUENCE VERSUS LOADING TASK

PERCENT OCCURRENCE OF SEQUENCE VERSUS LOADING TASK

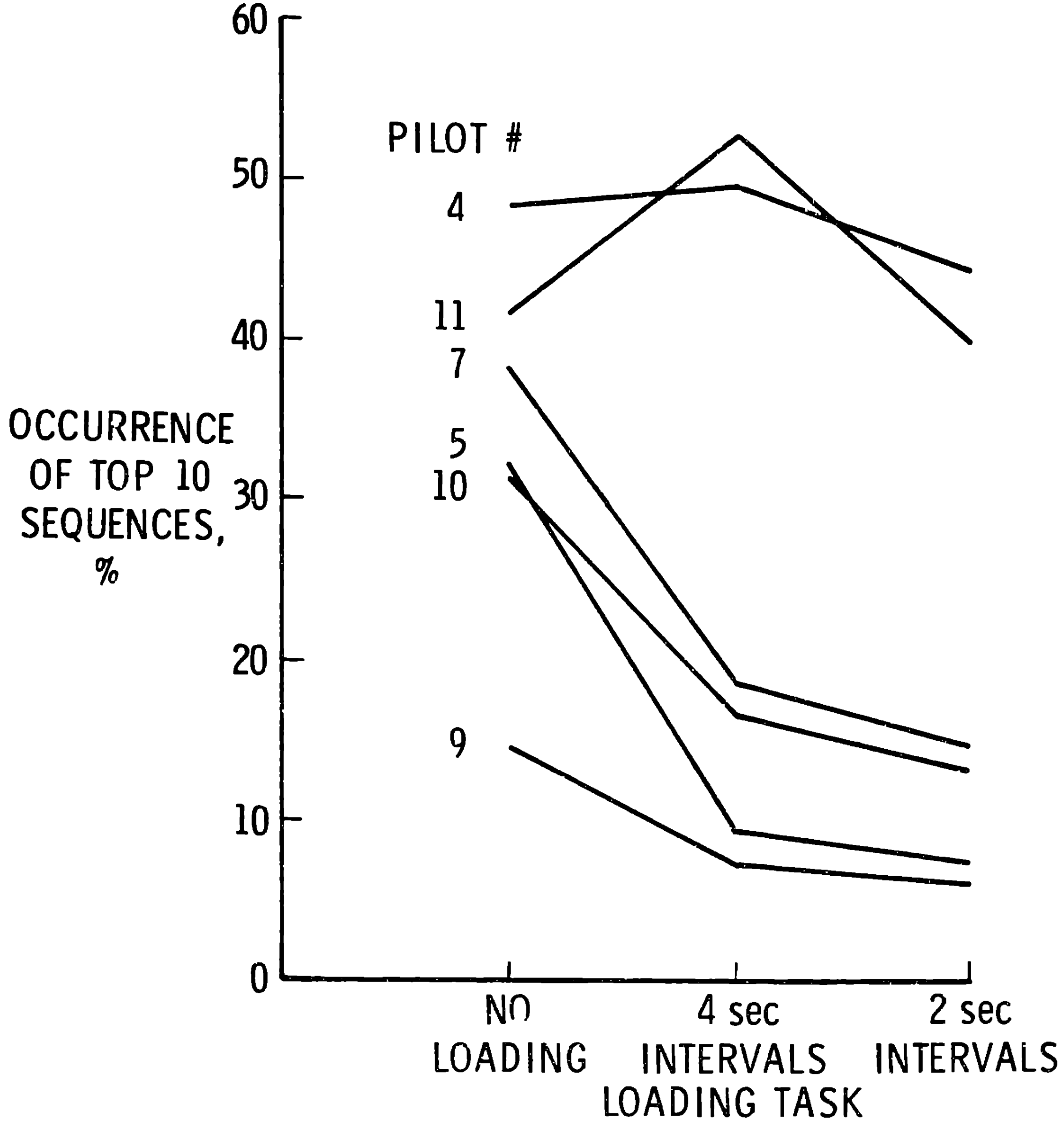


FIGURE 7

most frequently used sequences in the no loading run occupy much smaller percentages of the total scan than do those of the skilled pilots. This suggests the novices' scans are more random than those of the skilled subjects, even without the imposition of an additional task.

The novice subjects also show a consistent decrease in the percentage occurrence of the ten sequences as the workload is increased. This decrease may be the result of either the equalization of the number of occurrences of each sequence in the run (i.e. a trend to randomization) or a change to a different set of sequences from those used in the no loading case. Subjects 5, 9 and 10 were found to shift their fixations to other patterns more or less equally as the loading increased while subject 7 (the non-pilot) was found to use the latter of the two methods mentioned above (note that one of the sequences that became more important was within the top ten sequences for the no loading case). This group of results supports our original hypothesis of a change in the basic scan pattern as workload is increased, but indicates the effect is much more evident in pilots of moderate skill.

Summary

The procedures and results from the first phase of experiments are presented in this chapter. Six subjects, of varying skill levels, flew three ten-minute experimental runs. Three of the subjects also flew runs including a side task. Every run included the precision straight and level piloting task on the flight simulator and one of three levels of loading from the mental loading number task presented to the pilots via a cassette tape recorder. The pilot's lookpoint was obtained using an oculometer system and, data was stored using the microprocessor matrix system. The runs employing a side task were not analyzed for lookpoint data due to the change in scanning caused by this task.

The data was analyzed and the results are presented from the workload index, dwell time histograms, and fixation sequences. The side task was performed by three of the pilots and the results are presented in Table I. The results indicate that mental workload was increased by the mental loading number task. The dwell time histograms provide the most interesting results. Figure 5 shows the results for each of the pilots. A definite increase in dwell time with increase in mental loading may be observed. This may also be demonstrated by the results in Table II. The novice subjects were seen to change dwell times more than the skilled pilots. The fixation sequences of the pilot's instrument scans were analyzed, and the percentage occurrence of the ten most frequently occurring sequences are plotted in Figure 6. The sum of the percentages in each of these plots is shown in Figure 7. These results indicate that: 1) skilled pilots use a higher percentage of their ten most frequently occurring sequences than do novice pilots and 2) the scan pattern of the novice subjects were affected more by the increase in mental loading than were the patterns of the highly skilled pilots.

CHAPTER 6

DEVELOPMENT OF THEORETICAL ANALYSIS

A hypothetical relationship between performance, skill, and workload, and a possible relation between workload and scanning behavior will be developed in this chapter. A mathematical construct will be provided for the concepts presented. The equations developed here will be used in the analysis of the results from the experiments performed.

Performance, Skill, Workload Model

The model relating performance, skill, and workload was constructed using results from several sources including other investigators (Sheridan and Simpson, 1981; Albanese, 1977), experiments performed during this work (Tole et al, 1981), and intuition. Many investigators have shown that for highly skilled pilots, there is no decrease in performance until and overload condition occurs. This may be true for pilots of all skill levels, but those of lower skill may reach their overload condition before the higher skilled pilots under conditions of increasing mental workload. A graphical representation of the behavior described above is presented in Figure 2. In this plot, each line represents the postulated behavior of a particular pilot as his mental workload is increased. The increase in mental workload for some initial condition is plotted along the abscissa while the performance of the task in question is plotted along the ordinate. The higher the initial value of performance, the higher the skill level. (This should be approximately true in most tasks requiring skilled operators.) In order to test this hypothetical relationship, a mathematical expression for these curves is needed. The equation developed for this model is given below:

$$P = P_0 - (\mathrm{EXP}(\mathrm{TD}/\mathrm{Skill})^2) \qquad (2)$$

where

P = combined performance measure

P_0 = combined performance measure with no mental task

TD = perceived task difficulty

Skill = pilot skill level

In this equation, the combined performance measure is made up of performance on the piloting task and performance on the mental loading number task to be described in Chapter 7. The perceived task difficulty may be directly related to the mental workload of a pilot if all other functional variables of mental workload are held constant. The skill term mentioned above employed an equation from Hollister, et al (1973) that scored a pilot on the basis of recency of experience, total flight time, years since certification, time in particular type of aircraft, the age of the pilot, and on the error term. Details of this analysis may be found in Chapter 7.

Entropy

The fact that a pilot looks at one of the instruments on the panel does not guarantee that he acquires any information from that instrument. In the experiments used in this thesis (see Chapters 5 and 7) a situation is created which necessitates that the pilot look at some combination of instruments at some minimum rate in order to accomplish the task of flying the airplane. Intuition suggests that the pilot must therefore look at least at some of the instruments some of the time and that because of the repetitive nature of the task, a regular pattern of fixations is expected to emerge. These notions are strengthened by preliminary results on percentage use of different sequences in the scan and by the accompanying dwell time histograms as discussed earlier. The problem comes in developing a consistent numerical measure of the pattern(s) which are present.

Traditionally, much of the quantitative analysis of scanning patterns has employed Markov transition probability matrices. Such matrices do describe the predominant patterns in the scan via the relative sizes of transition probabilities but it is either extremely unwieldy or impossible to compare two of these matrices for different experimental conditions. Considerable effort was expended therefore, to develop an alternate method for quantifying the instrument scan. One of the promising methods resulting from this effort is the calculation of the entropy and entropy rate of the scan.

As shall be seen shortly, entropy is based on the relative probalities of different states of a system. Before defining entropy, calculation of the number of states that the visual scanning system can occupy in these experiments is first needed. In the most general case, M instruments may be arranged in some arbitrary fashion on the cockpit panel. Log2(M) bits are required in order to uniquely encode these instruments. An association of a certain number of bits with sequences of instrument fixations can be made. The sequences may be of arbitrary length, N. For a given number of instruments, M, and sequence length N, the maximum number of different sequences is given by:

Q = M(M-1)**N-1 = maximum number of sequences of length N

The calculation of Q allows an estimate of the number of bits necessary to encode spatial fixations. The number of bits required to uniquely encode all Q possible sequences is log2 Q. In the experimental paradigm, used in this thesis, 7 instruments are used and sequences up to length 4 were considered. Corresponding values of Q and log2 Q are:

Length	Q	log2 Q
1	8	3
2	56	5.8
3	392	8.6
4	2744	11.4

An important goal of this research is to develop a model of the instrument scan under varying mental loading conditions of the pilot. To accomplish this goal, a quantitative method for the comparison of two scanning patterns is needed. This method should be independent of the number and arrangment of instruments on the panel to allow for generalization of the procedure. One of the features of a scanning pattern that is likely to change is the randomness of the order in which the instruments are viewed. The concept of entropy from information theory, provides a measure of the randomness of a probabilistic sequence. The entropy of a sequence is defined as :

$$Ho = -\sum_{i=1}^{D} p_i \log_2 p_i \qquad (3)$$

where

Ho = observed average entropy

D = number of different sequences

p_i = probability of sequence i occurring

L = R-N+1 = number of sequences in a run

R = number of fixations in a run

N = sequence length (N=1,2,3 or 4)

The term entropy has been associated with Ho due to its similarity with certain formulations of entropy in statistical mechanics (Shannon and Weaver, 1949). The observed entropy is in the units of bits/sequence and provides a measure of the randomness of the scanning pattern.

Since the entropy is a function of L, comparison of entropies for different pilots for different run lengths required the entropies to be referenced to the same value. The value choosen was the maximum entropy for a given sequence length, log Q, which is a constant for a given sequence length. Therefore,

$$\frac{Hcorr}{\log 2\ L} = \frac{Ho}{\log 2\ Q}$$

$$Hcorr = Ho \frac{\log 2\ Q}{\log 2\ L} \qquad (4)$$

where

Hcorr = entropy corrected for sequence length

Q = M(M-1)**N-1 = maximum number of sequences of length N

M = number of different instruments

The entropy provides a useful measure for numerically quantifying the scan pattern obtained from the oculometer for a pilot. The characteristics of Ho show that for a completely deterministic scan (ie staring at one instrument) $p_i = p = 1$ and therefore Ho = 0. For a completely random scan (ie no two sequences of instruments are the same during the entire run) the entropy would take on its largest value, log2 Q. This provides us with a relative measure of the randomness of the pilot's scan based only on the order in which he scanned the instruments.

Entropy Rate

While entropy should help to explain the orderliness of the scanning pattern, the development presented up to this point does not include the fact that the dwell time for each fixation is different and may influence the information content of the pilot's scan pattern to the observer. This observation is closely related to the calculation of the channel capacity of the visual scanning system. Channel capacity is the concept used in communication theory to describe the maximum rate at which information may be transmitted over a noiseless channel. The channel capacity is defined as (Schwartz, 1959):

CC = Havg/t

where Havg is the average entropy for the system and t = smallest interval in which a transition may occur. Thus channel capacity is related to the probabilities of different system states and to the time occupied in each state. These are exactly the parameters available as raw data from the instrument scan.

Channel capacity is usually defined for the worst case of a system with equiprobable states running at its maximum rate. When a system is running at below its theoretical channel capacity, it may be more appropriate to talk in terms of entropy rate, Hrate where Hrate < or = CC. Hrate is defined in the same fashion as channel capacity except that the actual entropy of the system rather than its maximum possible entropy is used in the calculation.

The channel capacity of the visual scan may be calculated using the maximum values for entropy which were determined above together with dwell time statistics for the various instrument sequences in the scan. The minimum duration (interval) of a fixation is related to the properties of the oculomotor system. It is possible to estimate an upper bound on fixation rate based on physiological considerations. As is well known, saccadic eye movements can occur no more rapidly than about 5/sec due to the approximate latency of 200 ms in the oculomoter system. If we assume that a fixation is a steady eye position between saccades, then the maximum fixation rate is about 5 fixations/sec. Isolated brief looks may be as short as 100 msec however. These short looks may be particularly characteristic of at least some of a pilot's scan of secondary instruments (Harris and Christhilf, 1980). These rates are clearly higher than what can be sustained in a long term steady-state condition, however. A more realistic average value is probably about 2 fixations/sec for a long period of instrument scan (say any interval > 10 sec).

The maximum channel capacity for visual scanning can now be calculated under these various assumptions. The maximum entropy for each look (a length 1 sequence) is log2 Q = 3. Using the 0.5 sec/look (2 fixations/sec) as the most likely average interval, the maximum channel capacity is calculated to be

CC = 3/0.5 = 6 bits/sec

For the more instantaneous cases:

CC = 3/0.2 = 15 bits/sec for 200 msec looks

CC = 3/0.1 = 30 bits/sec for 100 ms looks

It is important to point out that the three values for channel capacity represent upper bounds under average, short term, and spurious scans, respectively. Since Q is used in their calculation, they also represent situations in which every possible state of the system is equally probable. Since we suspect that the pilot must have some regularity in his or her scan, the numbers we would expect to obtain for entropy rate under these three types of scanning will be lower than the maximes presented above.

In order to include time in the scan measure, the entropy for each sequence was divided by the average dwell time on that sequence which resulted in an average entropy rate, HR.

$$HR = \sum_{i=1}^{D} Hcorr_i / DT_i \qquad (5)$$

where

DT_i = average dwell time for ith sequence

Note that self-entropy rates for each of the i sequences appear as intermediate results in the calculation of entropy rate. Entropy rate is a measure of the amount of the visual scanning system's channel capacity which is in use by the pilot in a given instrument flight maneuver. As the mental workload of the pilot is increased, the rate at which he can scan for visual information should probably decrease as the other input(s) vie for his available input channels. Therefore a decrease in entropy rate of the scan with increase in mental workload is predicted.

Summary

The hypothetical relationship relationship between performance, skill, and mental workload, and the theoretical relations between mental workload and scanning behavior have been developed in this chapter. The hypothetical model relating performance, workload, and skill is shown in Figure 2 and its mathematical expression is given in equation 1. The concept of entropy from information theory is introduced, and a form to be used in the data analysis is presented in equation 3. The concepts of entropy rate and channel capacity are then introduced and discussed in relation to their application to this work.

CHAPTER 7

PHASE II EXPERIMENTS

Procedures

The experiments described in Phase I, provided us with much useful information on basic relationships of scanning, skill, and mental workload, and pointed out several areas needing futher investigation. After the initial analysis of the data, a new set of experiments following the same basic design as the previous set was developed. The major changes were:

1) A change in the number presentation from cassette tape recording to microprocessor controlled synthesized voice. This change was initially implemented by interfacing a Texas Instruments Speak & Spell module to our microprocessor system.

2) Automated scoring of response to the mental loading task to allow computation of an overall performance score. This was accomplished by changing the task from a verbal answer to pressing a 3-position rocker switch on the control wheel. Details may be found in the section on performance measures.

3) An increase in the number of loading levels to provide better resolution of changes taking place with increased loading.

4) Aircraft performance data was digitized and made available for use. Details may be found in the section on performance measures.

The addition of microprocessor controlled synthesized voice for the presentation of the mental loading task allows several things not possible with the tape recorded numbers. The interval between numbers can be very precisly controlled. Convenient, automated scoring of task accuracy is now available. Reaction time of response can also be easily computed in the revised scheme. And, finally, the knowledge of the precise presentation time with respect to the lookpoint data permits an investigation of the effect, if any, of the number task on the scan at the time of presentation of the numbers.

The rate of presentation of numbers in the mental loading task during the previous experiments was found to be sufficient to load the pilot but more resolution in changes with mental loading was desired. The intervals between number sequences was therefore changed from no numbers, 4-sec intervals, and 2-sec intervals to no numbers, 10-sec intervals, 5-sec intervals, and 2-sec intervals. The 2-sec interval numbers were chosen empirically in order to force all pilots into the region of rapidly decreasing performance as shown on the hypothesized performance vs workload curve (see Figure 2).

The modifications discussed above were implemented and, another set of experiments were carried out. The simulator, location, and equipment arrangement used were the same as in the previous set of experiments. The piloting task and side task were also the same as in the earlier experiments. Each session consisted of four 10-minute runs with a 5-minute break between each run. The difficulty of the mental loading task would start at no numbers for the first run and increase to 2-sec intervals by the fourth run. Some subjects participated in two sessions, one without and one with the side task. Each subject was allowed to practice all three tasks until he felt comfortable with them. Eleven subjects ranging in skill from NASA test pilots to non-pilots participated in the experiments.

Results and Discussion

Determination of Relative Skill Levels

In order to assess the effects of skill on performance and mental workload, a quantitative measure was needed. This measure would provide an independent, relative ordering of the pilots by skill level. The measure chosen came from Hollister, et al (1973) and was given as

$$\begin{aligned}\text{Skill} = 1.42 &+ 0.25(\text{recency}) + 0.73(\log(\text{total time})) \\ &- 0.030(\text{years certified}) + 0.15(\log(\text{time in type})) \\ &- 0.0088(\text{age}) + e\end{aligned} \qquad (6)$$

Skill = score reflecting relative piloting performance

recency = number of flight hours in past 30 days

total time = total number of flight hours

years certified = time in years since last certificate or rating

age = subjects's age in years

e = residual variance not explained by the model

This model was developed in order to predict the current level of skill of a variety of pilots flying light, single engine aircraft. Approximately fifty-five pilots were used to obtain the coefficients using stepwise, multiple linear regression. The pilot's skill had been evaluated by experienced instructor pilots using a five point scale. While not verified experimentally for the subjects used in this thesis, the model did offer a relatively objective means of rank ordering the current subjects.

A raw skill score was calculated for each of the pilot subjects using the above model. The pilot with the highest resulting skill score was then used to normalize all of the scores so that skill levels would range between 0% and 100%. Though great care must be taken when applying an equation such as this in a different set a experimental conditions, the overall rank ordering of the pilots by this method is probablly accurate as it generally agreed with subjective rating of the pilot's skills by experienced observers at the NASA/Langley Research Center. The Langley Research Center pilot code number, the symbol to be used in the following graphs, and skill score for each pilot may be found in Table III. This information was obtained from a questionnaire filled out by each pilot. A sample questionnaire pilot may be found in Appendix B.

Imposed Task Difficulty

The relationship between intervals used on the mental loading number task and the mental workload experienced by the pilot is probably not linear. Consequently, a method of estimating the relative increase in difficulty of the number task was needed. From observations, an estimate was made that related task difficulty with the inverse of the intervals between numbers on the mental loading number task. This relationship is given by the following equation:

$$TD = 1/\text{interval between \#task} \qquad (7)$$

where TD is equal to imposed task difficulty. Therefore, the four loading levels which use no-numbers (infinite intervals), ten second intervals, five second intervals, and two second intervals will have task difficulties of 0.0, 0.1, 0.2, and 0.5, respectively.

Flight Performance Measures

Several variables were obtained from each of the two tasks in order to allow the computation of performance scores. The scores developed ran between 0 percent and 100 percent with 100 percent being obtained if the pilot never deviated from the intended path in space on the piloting task, and if all number task sequences were answered correctly for the mental loading number task. The scores from the piloting and the mental loading tasks were then combined to provide a performance measure to be used in the validation of the proposed performance/skill/workload model.

The variables obtained for scoring performance on the piloting task were the errors from the intended track for the glide slope and localizer courses. Those signals were obtained from the GSL instrument and recorded on FM magnetic tape. The signal was digitalized at the NASA/Langley Research Center and the mean and variance were computed. This information was used to compute the RMS error from the intended path for both the glide-slope and localizer.

Discussions with several of the highly skilled pilots revealed that accuracy of tracking the glide slope and localizer might not provide the whole performance picture. It was discovered that these pilots were willing to trade off "smoothness" when the loading task became more difficult. That is to say, the pilot may perform the piloting task to the same level of accuracy, as far as deviations from a designated path are concerned, on two different runs but produce two very different ride qualities for these runs. One possible measure for "smoothness" could be the frequency of oscillation around the intended path. The higher this frequency is, the less "smooth" the ride becomes. In order to examine this performance measure, the power-spectral density of the course deviations was computed. This provided a breakdown of the frequencies appearing in the tracking. In order to provide a single measure for this analysis, the power-spectral densities were integrated to obtain a plot of the cumulative power density function from which values were obtained at a selected frequency of 0.1 Hertz for each run. The percentage of power greater than 0.1 Hertz would change if the frequency content of the signal changed. This was computed for both the glide slope and the localizer and combined with the two RMS measures to provide four candidate variables to be included in a performance score for the piloting task. Each of these variables is plotted versus imposed task difficulty in Figure 8 through Figure 11.

There are four pilots missing from these plots: pilots #4(B), #12(G), #8(I), #16(K). Pilot #4(B) is not shown because there were major flaws in his performance data stage due to equipment malfunction. Pilots #12(G), #8(I), and #16(K) are not shown since they were selected for use in verifying the model parameters that resulted from the data of the seven remaining pilots. Pilots #12(G) and #8(I) were chosen for this because their entropy rates did not follow the same pattern as the other nine pilots. The purpose of this was to see if the model would account for the abnormalities in their data. Observation of these two pilots during the experiments indicated unaccounted for influences such as motivational problems contaminating the data. For this reason, their data has been omitted from all calculations except the ones for model verificationn. The results from pilots #4(B) and #16(K), however, are included

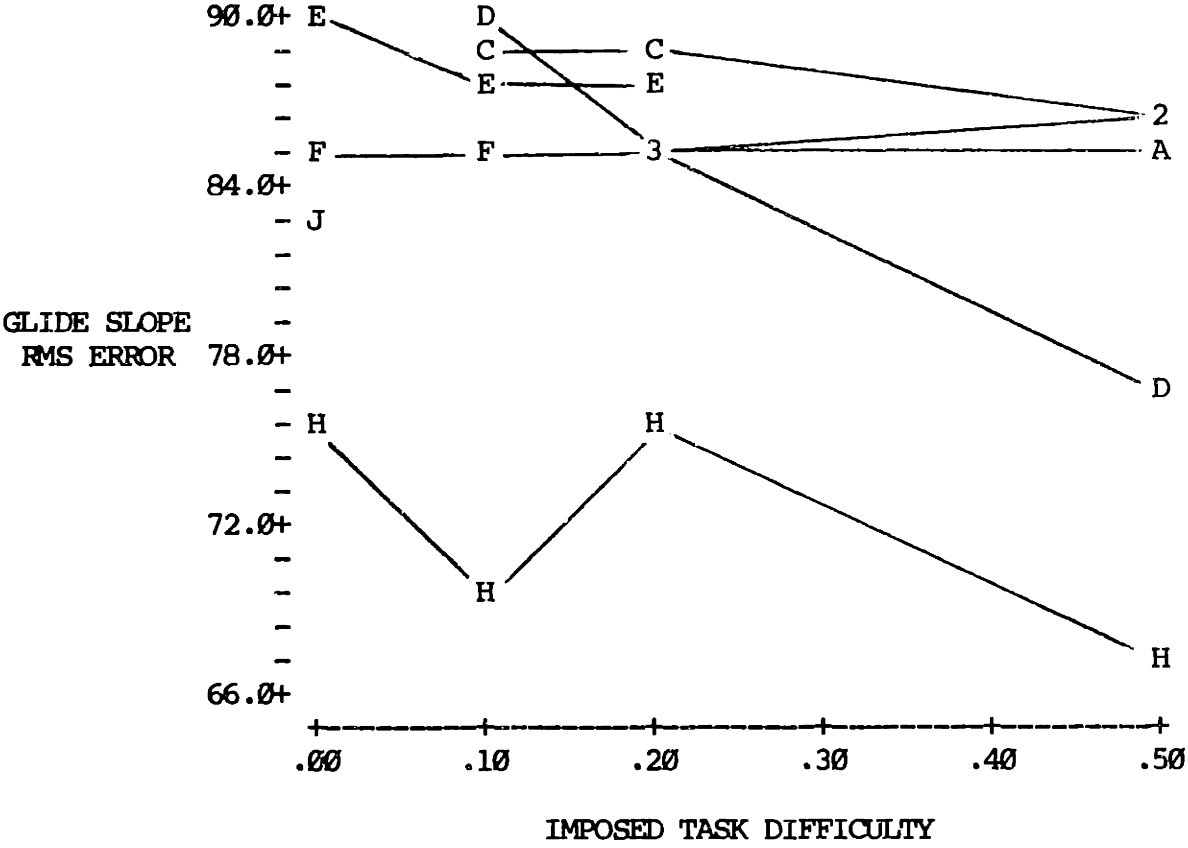


Figure 8 RMS Error on Glide Slope Indicator vs Imposed Task Difficulty

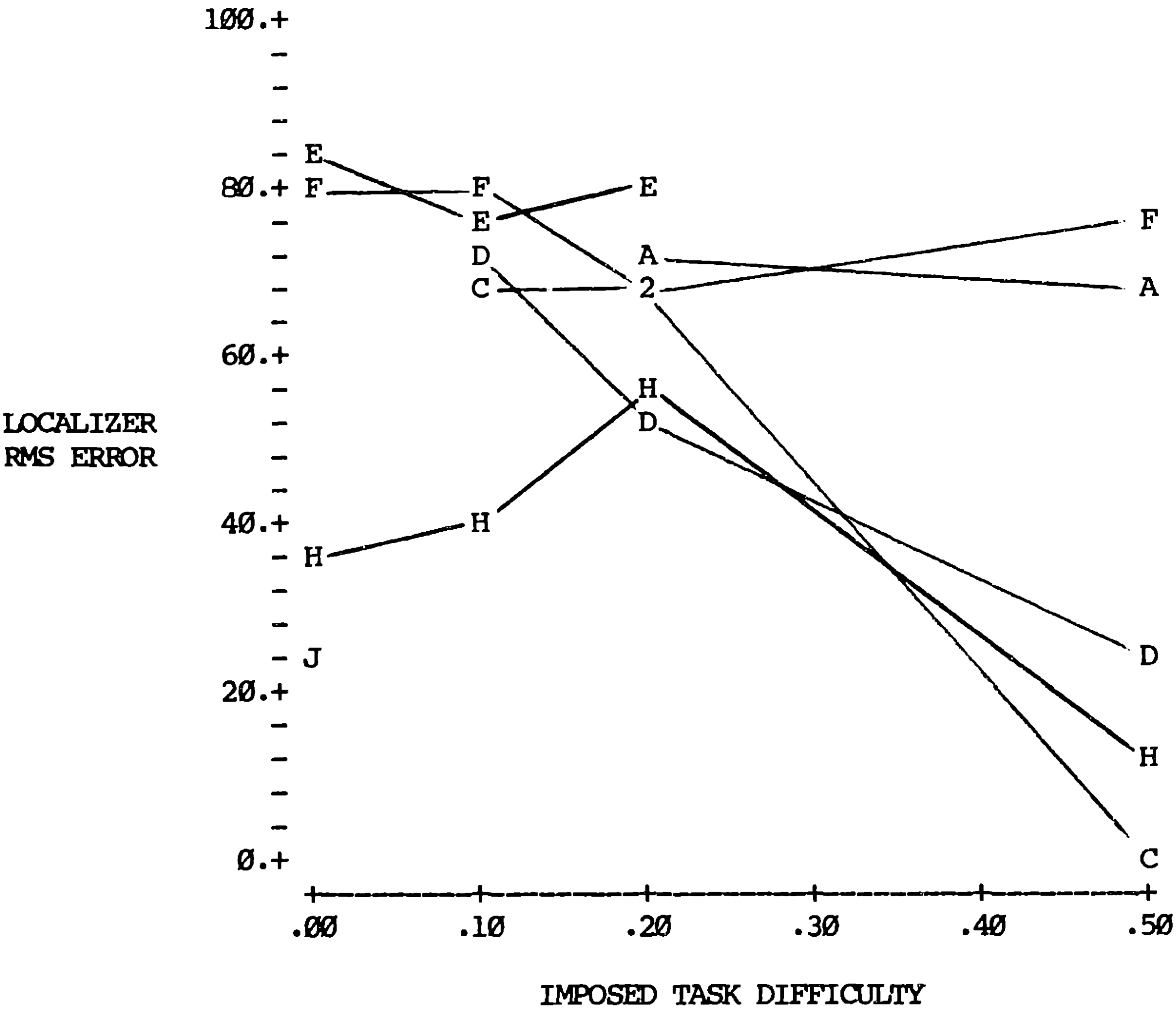


Figure 9 RMS Error on Localizer Indicator vs Imposed Task Difficulty

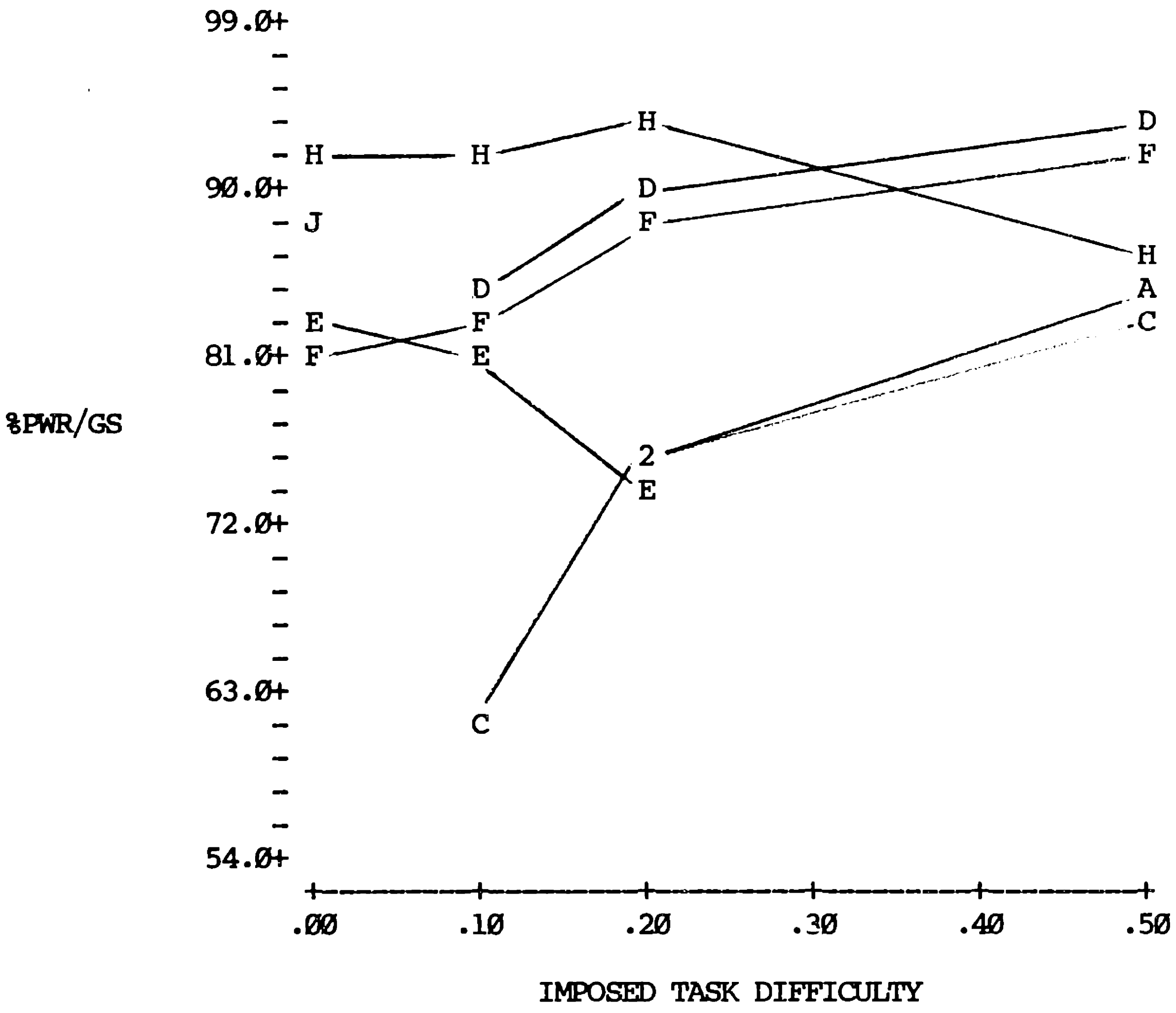


Figure 10 Percent Change in Power of Frequencies Greater Than 0.1 Hz for Glide Slope

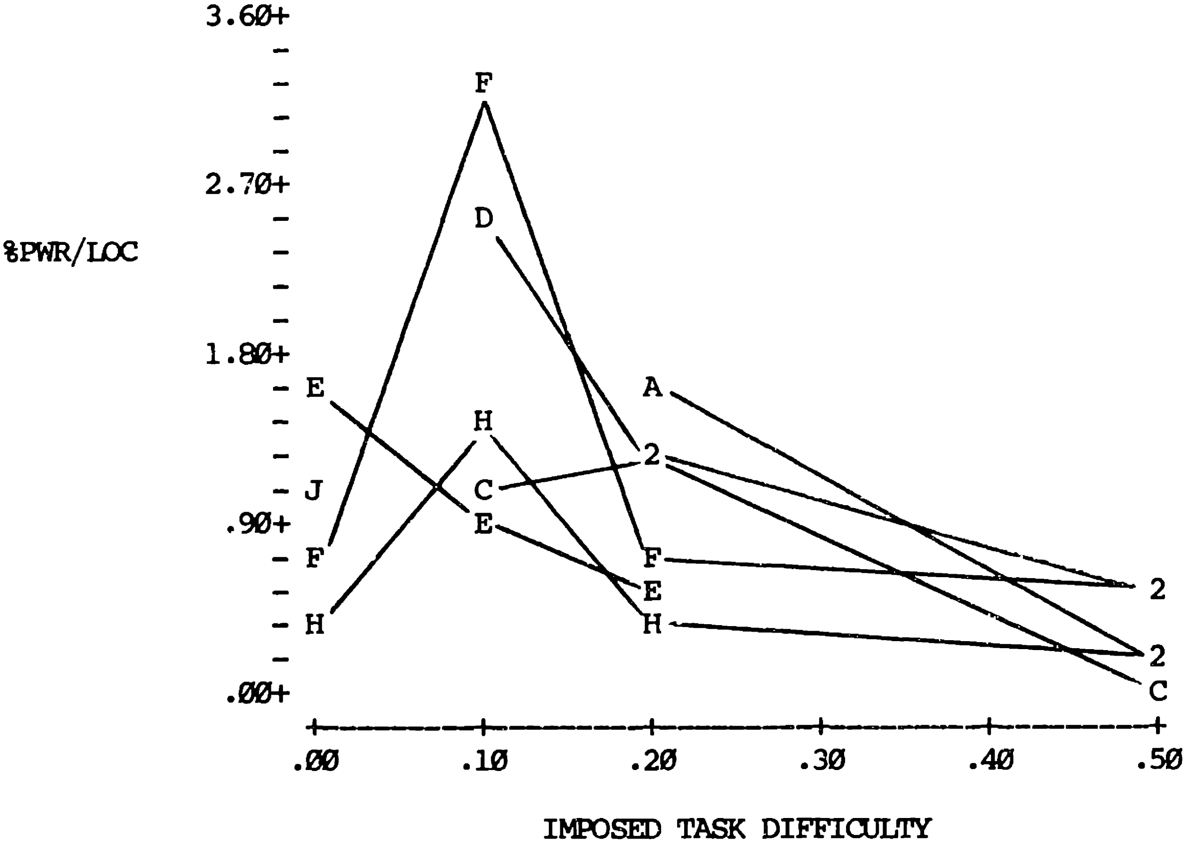


Figure 11 Percent Change in Power of Frequencies Greater Than 0.1 Hz for Localizer

whenever available. The data from Pilot #16(K) appeared to follow the majority of pilots and he was selected to prevent all the test cases from being abnormal. The final devision of pilot subjects resulted in seven pilots to be used for the model development, and three pilots to be used for the model verification. One pilot's performance data was discarded due to equipment malfunction.

Performance on Verbal Loading Task

The mental loading number task, which was supposed to be given equal priority with the piloting task during the experiments, was scored as follows. The response to the three number sequences was given by the pilot by pressing up for plus and down for minus on a three-position rocker switch located on the left handle of the control wheel. The responses and reaction time of the pilots were stored on magnetic tape via the microprocessor matrix system. The scoring measure for this task was computed as given below.

$$\#TP = \frac{(\#TOT - \#WRO - \#MIS)}{\#TOT} \times 100\% \qquad (8)$$

where

#TP = mental loading number task performance

#TOT = total number of stimuli presented

#WRO = number of incorrect responses

#MIS = number of missed responses

This score was 100 percent if the pilot answered every sequence correctly and zero percent if a pilot either answered incorrectly or missed all of the stimuli presented.

The score on the number task is plotted versus task difficulty in Figure 12 for the seven pilots used in the model verification.

Since the pilots were instructed to give equal priority to the piloting task and the mental loading number task, both were included in the development of a combined performance score. While a weighting of 0.5 might have been assigned to each task, it was decided to leave the weighting free to allow the model fitting procedure to determine the relative weights. A linear relationship between all of the terms was assumed and the form of the equation became:

$$P = CONST + a(\#TP) + b(RMS/GS) + c(RMS/LOC) + d(\%PWR/GS) + e(\%PWR/LOC) \qquad (9)$$

where

P = combined performance measure

CONST = constant term

#TP = mental loading number task performance

RMS/GS = RMS error from glide slope track

RMS/Loc = RMS error from localizer track

%PWR/GS = percent of power from the power-spectral density for the glide slope greater than 0.1 Hertz

%PWR/LOC = percent of power from the power-spectral density for the localizer greater than 0.1 Hertz

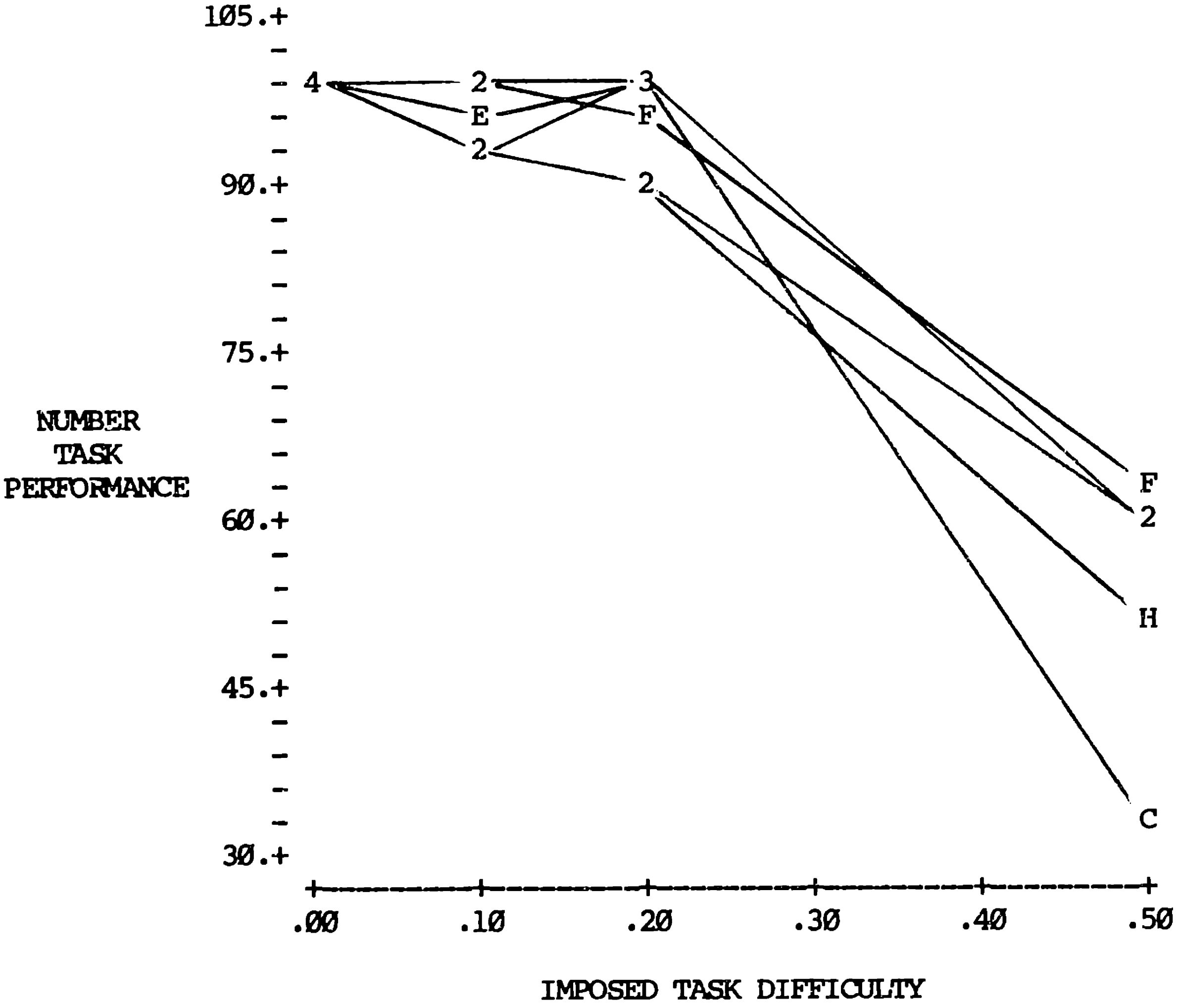


Figure 12 Mental Loading Number Task Performance vs Imposed Task Difficulty

Entropy and Entropy Rate

To implement the entropy computation, each of the instruments to be examined was given a number. Then a sequence of these numbers was stored as the pilot scanned the instrument panel. The dwell time for each fixation was also stored. From this data, the entropy was computed for sequence lengths of 1, 2, 3, and 4 instruments. The analysis incorporates a window the width of the sequence length which was moved along the resulting sequence of instrument numbers. The sequence in the window, along with its dwell time, would be recorded and the window would advance one instrument. This process was continued until the last fixation in the run was reached. The number of occurrences of each sequence was then divided by L, the total number of sequences in the run, to obtain the probabilities, p_i , used to calculate the entropy. The sequence with the most counts would then be stored along with its respective cumulative dwell times and the occurrences of this sequence would be removed from the larger sequence for that run. This would leave "holes" in the larger sequence which would not be utilized on the next pass. The window would then be positioned at the first remaining sequence and the above process was repeated. When a missing instrument was reached, the window would jump to the next string of instrument numbers equal to or longer than the sequence length being examined. This entire process was repeated until all remaining strings of instrument numbers were less than the sequence length being examined. The result of this computation would be a list sequences one, two, three, or four instruments long depending on what was specified and then corresponding cummulative dwell times and number of occurrences. To illustrate briefly, suppose the resulting sequence of instrument numbers from a run was

1 2 1 3 1 2 3 2 1 2 1 3 1 4 1 2 5

and we want to examine length two sequences. For 16 instrument numbers there are [illegible] - 1 = 17 - 1 = 16 length two sequences.

Sequence	Number of Counts
12	4
21	3
13	2
31	2
14	1
23	1
25	1
32	1
41	+1
	16

Therefore sequence 12 would be stored and removed from the run as shown below.

- - 1 3 - - 3 2 - - 1 3 1 4 - - 5

The process would be repeated with the following results:

Sequence	Number Of Counts
13	2
14	1
32	1

So that sequence 13 would be stored and removed as shown:

- - - - - - 3 2 - - - - 1 4 - - 5

This leaves sequences 14 and 32 which will be stored and removed during the next two passes and the single instrument number 5 which will not be utilized in the entropy calculation.

The next step in the entropy calculation involves the determination of the probabilities for each of the stored sequences from the previous step. The probability is computed by dividing the actual number of counts on a sequence by the maximum possible number of sequences. The maximum possible number of sequences is found by dividing the number of fixations in the run by the length of sequence to be examined. This quantity is used as opposed to dividing by the actual number of length- N sequences that occcur in order to differentiate between those runs with several unused instrument numbers and those runs with few instrument numbers not utilized by the analysis. An example will help clarify the last statement. Suppose the following two sequences were obtained from two runs and a comparison of their entropies is desired.

I. (1 2 3 4) 1 2 (1 2 3 4) 5 (1 2 3 4) 1

II. (1 2 3 4)(1 2 3 4)(1 2 3 4)(1 2 3 4)

Dividing by the actual number of lenth-N sequences would result in the same probability for the length-N sequence in each case and, therefore, the same entropy would result for both. Sequences I and II are different, however, and should be differentiated. Dividing by the maximum possible number of length-N sequences (= L/N = 16/4 = 4) will result in different probabilities for the two sequences and, therefore, different entropies will result.

Once the probabilities were determined, the component of entropy for each sequence was computed. This was then corrected for sequence length as described in the theory section and divided by the average dwell time for the sequence to obtain the entropy rate. The components for the entropy, corrected entropy, and entropy rate would be summed, respectively, to obtain their average values for the run. This was done for sequences of length one, two, three, and four. Sequences greater than length-four were not examined due to the large memory requirements. It is not likely that analysis of sequence lengths greater than four instruments will show any significant results not found in the first four sequence lengths. However, if this work were transferred to a larger machine, the effects of longer sequence lengths should be examined.

No prior assumption was made as to the length of the sequences to be examined for changes in entropy rate. Therefore, the entropy rate for sequences of length one to four were computed and examined. Sequences of length two were found to produce the largest and most consistent changes with task difficulty out of the four lengths examined. Fortunately, these sequences require less memory to compute than the longer length sequences and, therefore, reduce computational problems. The data for all pilots except #12(G) and #8(I) are plotted in Figure 13. A trend toward lower entropy rate with higher task difficulty may be seen. A two-way analysis of variance was performed for the entropy rate data from nine pilots on levels of task difficulty and between subjects. The F-ratio for the task difficulty factor was determined to be 15.37. The critical value from the F-table for $p = 0.01$ and degrees of freedom of 3 and 24, respectively, is 4.72. The null hypothesis of equal means between the four loading levels is thus rejected. The F-ratio for the intersubject differences was 6.57 with a critical value of 3.36 for $p = 0.01$. Therefore, the null hypothesis of equal means between each subject is also rejected. Though the changes are small, null hypotheses of no differences between levels of task difficulty could be rejected (T-test $p < 0.05$) for all six combinations of level differences. The details of the tests described above may be found in Appendix C.

On the basis of this finding, it was decided to develop a functional relationship between HR and TD. The relationship using the data from the seven model development pilots may be given by the following equation

$$HR = 0.9419 \, EXP(-TD) \qquad (10)$$

where HR is the entropy rate and TD is the task difficulty. This equation was obtained using regression analysis with R-squared = 97.06%. R-squared is called the coefficient of determintation and has several interpretations. The first is that it is the

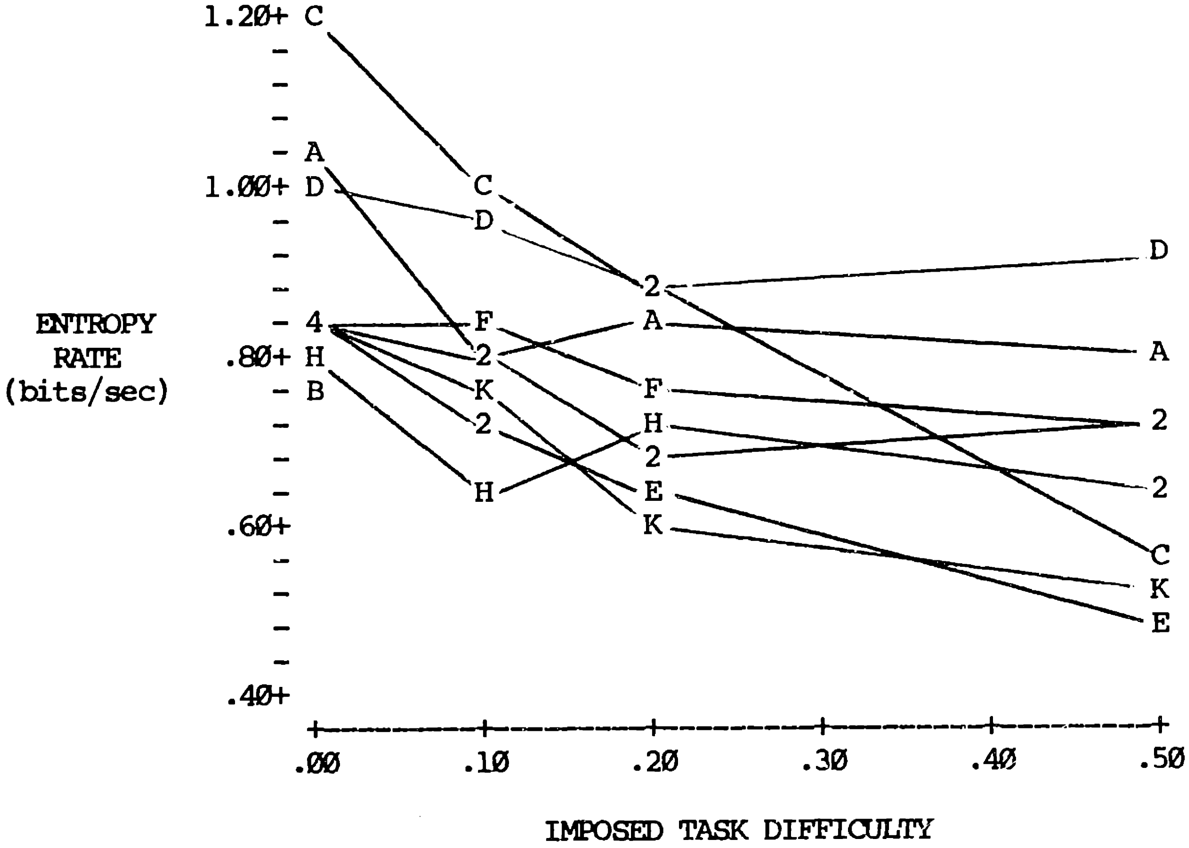


Figure 13 Entropy Rate on Length 2 Sequences vs Imposed Task Difficulty

square of the correlation coefficient, rho, between HR and EXP(-TD) making rho = 0.985. The R-squared value may also be obtained by dividing the residual sum of squares by the regression sum of squares, which means that R-squared is the fraction of variation explained by the straight line regression model. Equation 10 may be solved for task difficulty with the following results:

$$TD = -(0.059856 + \ln HR). \qquad (11)$$

The regression of HR on TD was performed again including the results from pilots #4(B) and #16(K) who had been left out for reasons discussed earlier and the resulting equation was

$$HR = 0.9279\ EXP(-TD) \qquad (12)$$

with an R-squared value of 97.30%. The results indicate a decrease in entropy rate with increased task difficulty. Though some increases in entropy rate exist for the intermediate loading levels, there is an overall decrease between the lowest and highest loading level for all subjects. In conclusion, an important result was presented in this section. This result is that perceived task difficulty is proportional to entropy rate, on the average. For this reason, entropy rate was chosen to map from scanning behavior into perceived task difficulty (i.e. workload).

Model Development and Verification

One of the major goals of this work was the development of a model relating performance, skill, and mental workload. The form of this model was described in the chapter on theory and the procedures used in obtaining and verifying it will now be discussed. The ultimate goal is the prediction of performance given estimates for skill and scanning parameters.

The first step is the determination of mental workload from the entropy rate of a pilot's scan pattern. Assuming all other functional variables related to mental workload are held constant, mental workload should be proportional to perceived taskk difficulty. Therefore, the equation relating entropy rate to perceived task difficulty may be solved for perceived task difficulty and used to determine workload level from the measured value of entropy rate (see equation 11).

The equation relating performance, skill, and mental workload presented earlier is stated again for reference.

$$P = P_0 - \mathrm{EXP}((\mathrm{TD}/\mathrm{Skill})^2) \qquad (13)$$

This equation may be rearranged as follows:

$$\mathrm{EXP}((\mathrm{TD}/\mathrm{Skill})^2) = P_0 - P \qquad (14)$$

which states that the exponential term is equal to the difference in the performance at the no-loading level (P_0) and the performance at the present level of mental loading (P). Using the values for the level of skill and task difficulty calculated in equations 7 and 11 respectively, the left hand side of the equation may be computed. The right hand side of the equation must be expressed in terms of measurable performance indicators. This is done by expressing P and P as given in equation 9. Hence, the right had side of the equation becomes:

$$\begin{aligned} P_0 - P = {} & [a(\#TP_0) + b(RMS/GS_0) + c(RMS/LOC_0) + d(\%PWR/GS_0) \\ & + e(\%PWR/LOC_0)] \\ & -[a(\#TP)\ b(RMS/GS) + c(RMS/LOC) + d(\%PWR/GS) + e(\%PWR/LOC)] \end{aligned} \tag{15}$$

Combining terms results in:

$$\begin{aligned} P_0 - P = {} & a(\#TP_0 - \#TP) + b(RMS/GS_0 - RMS/GS) \\ & + c(RMS/LOC_0 - RMS/LOC) + d(\%PWR/GS_0 - \%PWR/GS) \\ & + e(\%PWR/LOC_0 - \%PWR/LOC) \end{aligned} \tag{16}$$

The values for each of these measures were recorded during the experiments. However the coefficients of each term are unknown. Therefore a multiple regression analysis was performed on the equation in the form given below

$$\begin{aligned} EXP((TD/Skill)^2) = {} & a(\#TP_0 - \#TP) + b(RMS/GS_0 - RMS/GS) \\ & + c(RMS/LOC_0 - RMS/LOC) + d(\%PWR/GS_0 - \%PWR/GS) \\ & + e(\%PWR/LOC_0 - \%PWR/LOC) \end{aligned} \tag{17}$$

where the subscript 0 indicates a no-loading value. The variables a through e are the weighting coefficients to be found by the regression analysis.

The results of the first attempt at regression indicated that the coefficient of the %PWR/LOC term could not be differentiated from zero based on a Student's T-test. This variable was then eliminated from the analysis and the term on the left hand side of equation 17 was regressed on the four remaining variables on the right hand side of the equation. This regression produced values for the coefficients a through d, with e equal to zero and included a constant term. The resulting equation was:

$$\begin{aligned} \mathrm{EXP}((\mathrm{TD/Skill})^2) &= 1.4483 + 0.0351(\#\mathrm{TP}_0 - \#\mathrm{TP}) \\ &+ 0.1765(\mathrm{RMS/GS}_0 - \mathrm{RMS/GS}) - 0.0366(\mathrm{RMS/LOC}_0 - \mathrm{RMS/LOC}) \\ &+ 0.0377(\%\mathrm{PWR/GS}_0 - \%\mathrm{PWR/GS}) \end{aligned} \qquad (18)$$

This analysis had an R squared value of 76.6 percent and an F-ratio of 12.28 with a critical value of 4.89 for $p = 0.01$. A 95 percent confidance interval for the predicted y-values based on the population mean at the value of perceived task difficulty equal to zero is given by:

$$1.011 \pm 2.131(0.206)$$

or

$$(0.572,\ 1.450)$$

The coefficients determined above may now be used in equation 9 which becomes

$$\begin{aligned} P = {} & 1.4483 + 0.0351(\#\mathrm{TP}) + 0.1765(\mathrm{RMS/GS}) \\ & - 0.0366(\mathrm{RMS/LOC}) + 0.0377(\%\mathrm{PWR/GS}). \end{aligned} \qquad (19)$$

These coefficients provide the relative weightings for each of the performance terms but they need to be scaled in order to provide the proper characteristics for the equation. If each of the terms were at their maximum value, that is 100 percent, then the combined performance measure should also equal 100 percent. However, using the coefficients as given above, the value of P with all terms at 100 percent is 22.72 percent. In order to make this 100 percent, each coefficient must be multiplied by 100./22.72 = 4.40. The modified performance equation becomes:

$$P = 6.3750 + 0.1545(\#TP) + 0.7769(RMS/GS) - 0.1611(RMS/LOC) + 0.1659(\%PWR/GS) \quad (20)$$

A plot of this fuction versus the perceived task difficulty, obtained from the relation between entropy rate and task difficulty, equation 11, is provided in Figure 14. These curves should resemble those given in the hypothetical plot in Figure 2 and for some of the pilots, a general overall downward trend is present. Even though the curves do not match the hypothetical ones exactly, there are some common features between them. First of all, the curve for the lowest skilled pilot (J = #7) is seen to decrease much more rapidly than the curves for the more highly skilled pilots (A = #3, C = #11; the two points of A are for the third and highest levels of mental loading respectively).

To test this model's value as a predictive tool, the data from three subjects not included in the model determination, were substituted into equation 20 and plotted versus perceived task difficulty in Figure 15. Pilots #12(G), #8(I), and #16(K) produce some interesting, if not encouraging curves in this plot. All three pilots show a net decrease in performance between their lowest and highest task difficulties even though they

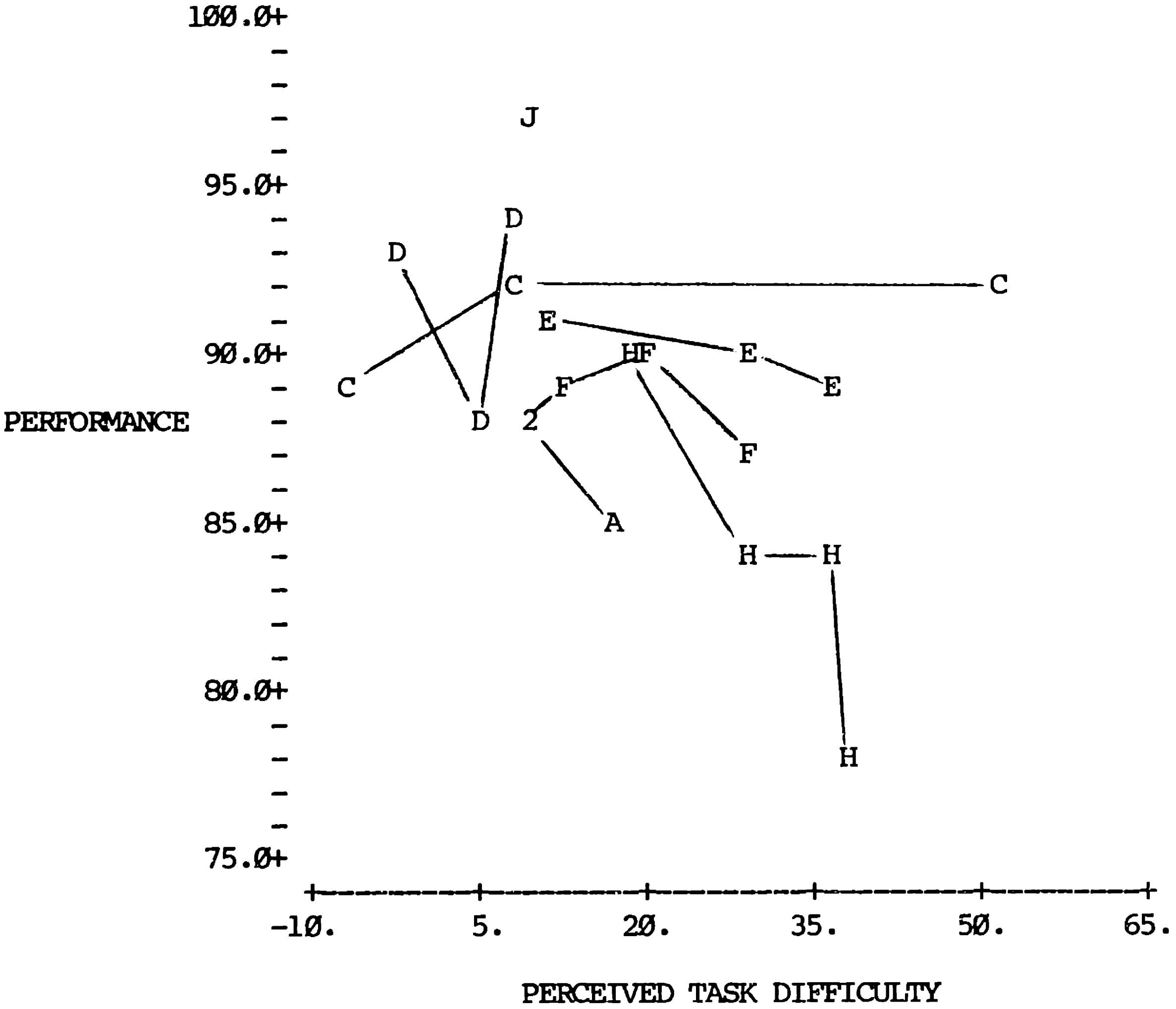


Figure 14 Combined Performance (from model) vs Perceived Task Difficulty for 7 pilots used in model

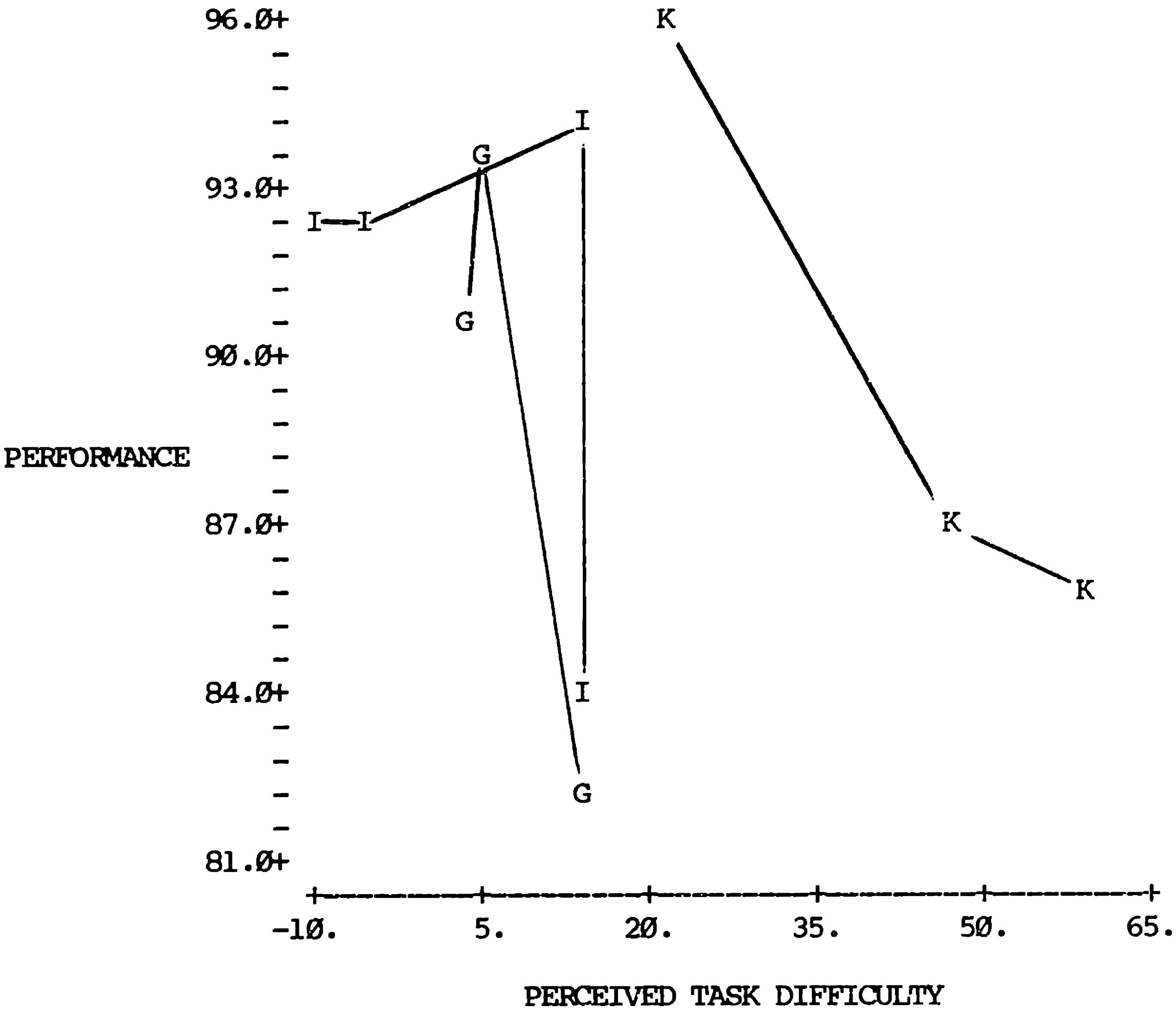


Figure 15 Combined Performance vs Perceived Task Difficulty for 3 test cases of model

accomplished this decrease in very different ways. Pilot #8(I) appears to be the closest to the theoretical model with his sharp decrease in performance over a very small task difficulty increase. Pilot #16(K), on the other hand, appears to be decreasing at an exponentially decreasing rate as opposed to the model which predicts decreasing performance at an exponentially increasing rate. The three points of pilot #12(G), and pilot #16(K) are for the second, third, and highest loading levels. Pilot #12(G) increases performance sharply between his second and third runs and then decreases just as sharply between the third and fourth runs. These curves do not fit the hypothetical curves very well but they each show some encouraging trends. The results from these three subjects is inconclusive but the trends in the results may indicate that the general form of the model may be correct.

Since the choice of the exponential model was, in part, an educated guess,it seemed reasonable that another model of similar form could possibly fit the results better than the exponential. For this reason, two other forms for the performance/skill/workload model were also examined. A model which assumed the drop in performance with increased mental workload followed a circular path was investigated. The equation

$$\mathrm{TD}^2 + \mathrm{P}^2 = \mathrm{Skill}^2 \qquad (21)$$

was used for this model. Even though the regression analysis resulted in a slightly higher R-squared value (R-squared = 77.9%)

it only utilized two of the five terms included in the analysis. This was considered to be less desirable than the analysis of the exponential which employed four of the five terms included in the analysis.

The third model to be examined was a linear relation between performance and task difficulty. This model was expressed as

$$P = P_0 - K/\mathrm{Skill} \ast TD \qquad (22)$$

where K is an arbitrary constant. This model produced an R-squared value of only 51.5% and used only three of the five performance values given. On the basis of these results, this model was also rejected in favor of the exponential.

Dwell Time Histograms

The dwell time histograms were computed in the same manner as those presented in Phase I. They are presented for each pilot in Figure 16 - Figure 26. The shift to longer dwell times is once again seen to occur with increase in mental loading. Part of this information is now included in the entropy rate measure but useful information may still be obtained by examining these results.

FIGURE 16 DWELL TIME HISTOGRAMS FOR PILOT #3 (A)
(DOTTED LINES INDICATE 1 SEC INTERVALS)

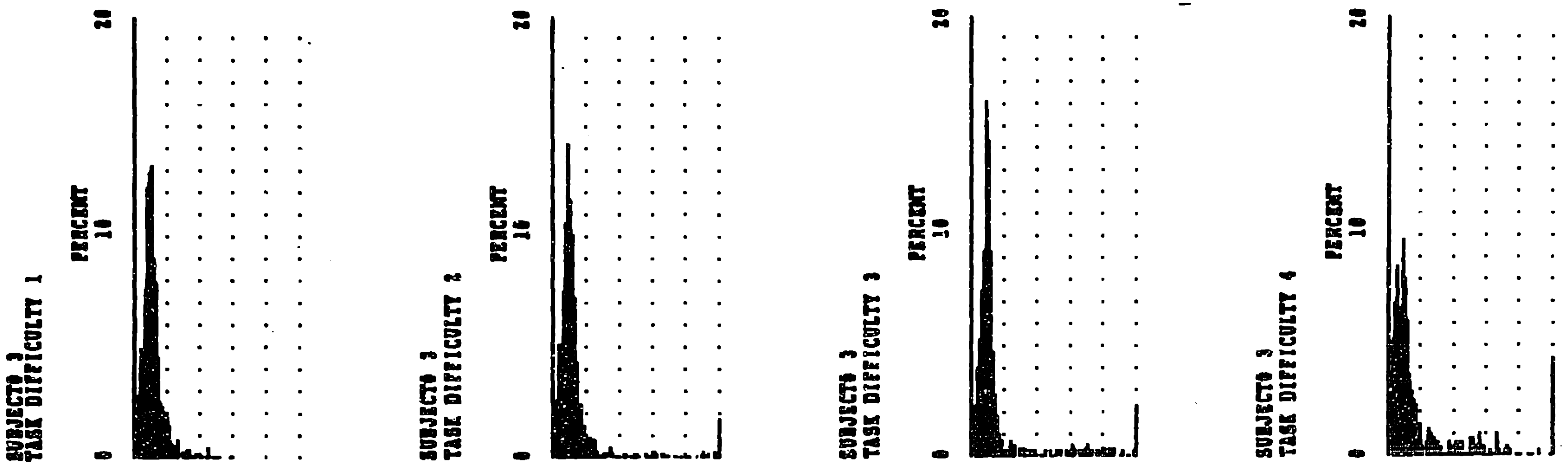


FIGURE 17 DWELL TIME HISTOGRAMS FOR PILOT #4 (B)

(DOTTED LINES INDICATE 1 SEC INTERVALS)

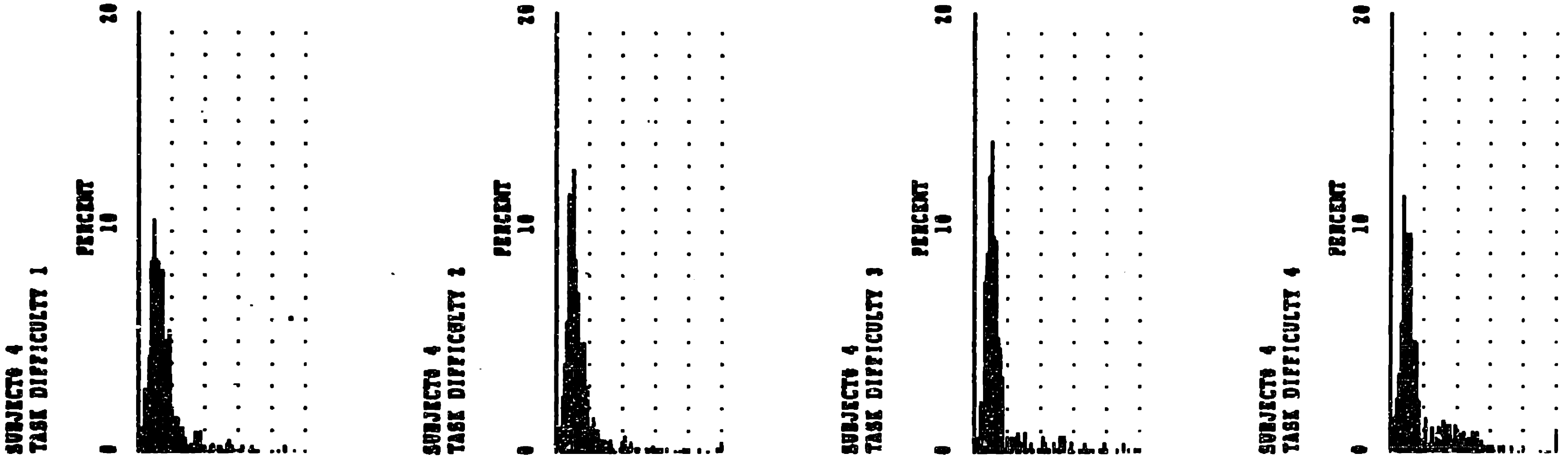


FIGURE 18 DWELL TIME HISTOGRAMS FOR PILOT #11 (C)
(DOTTED LINES INDICATE 1 SEC INTERVALS)

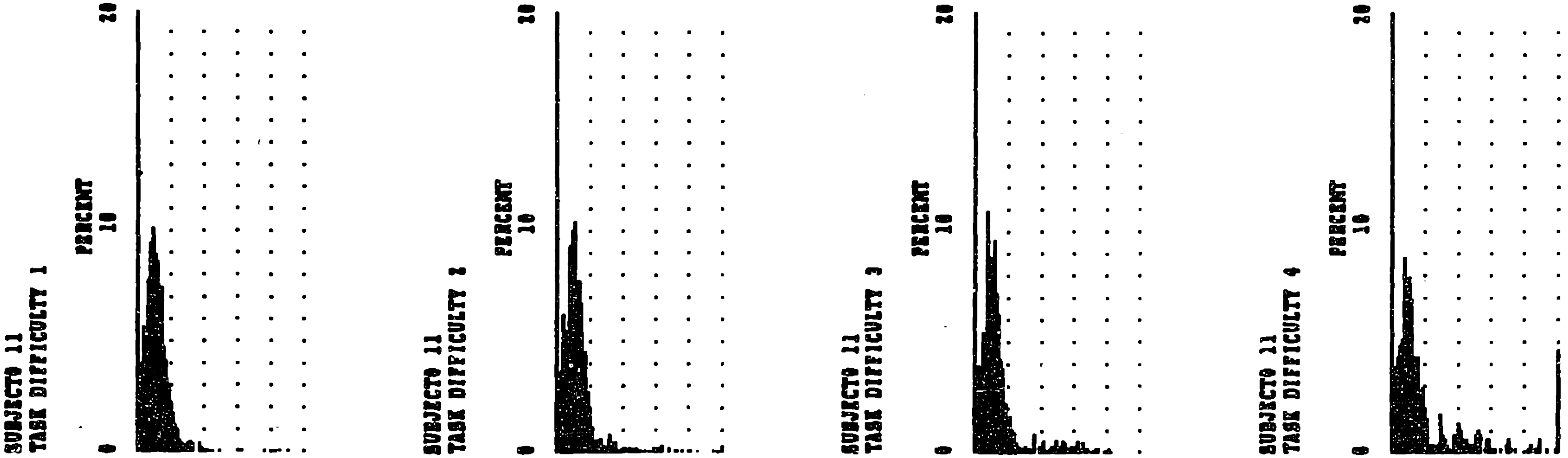


FIGURE 19 DWELL TIME HISTOGRAMS FOR PILOT #13 (D)
(DOTTED LINES INDICATE 1 SEC INTERVALS)

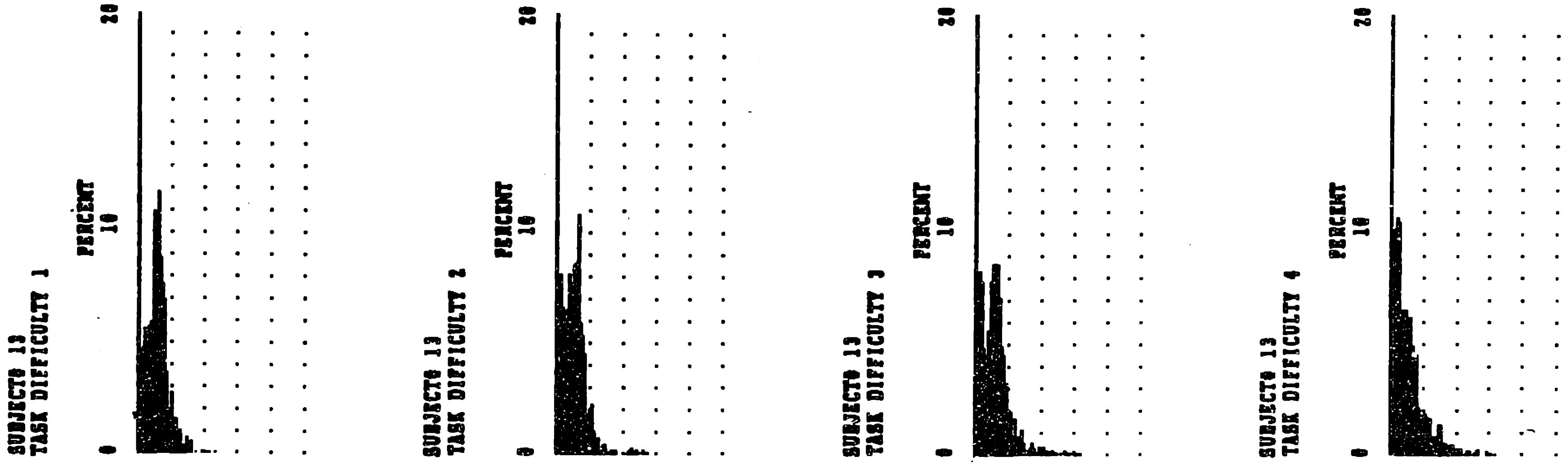


FIGURE 20 DWELL TIME HISTOGRAMS FOR PILOT #15 (E)

(DOTTED LINES INDICATE 1 SEC INTERVALS)

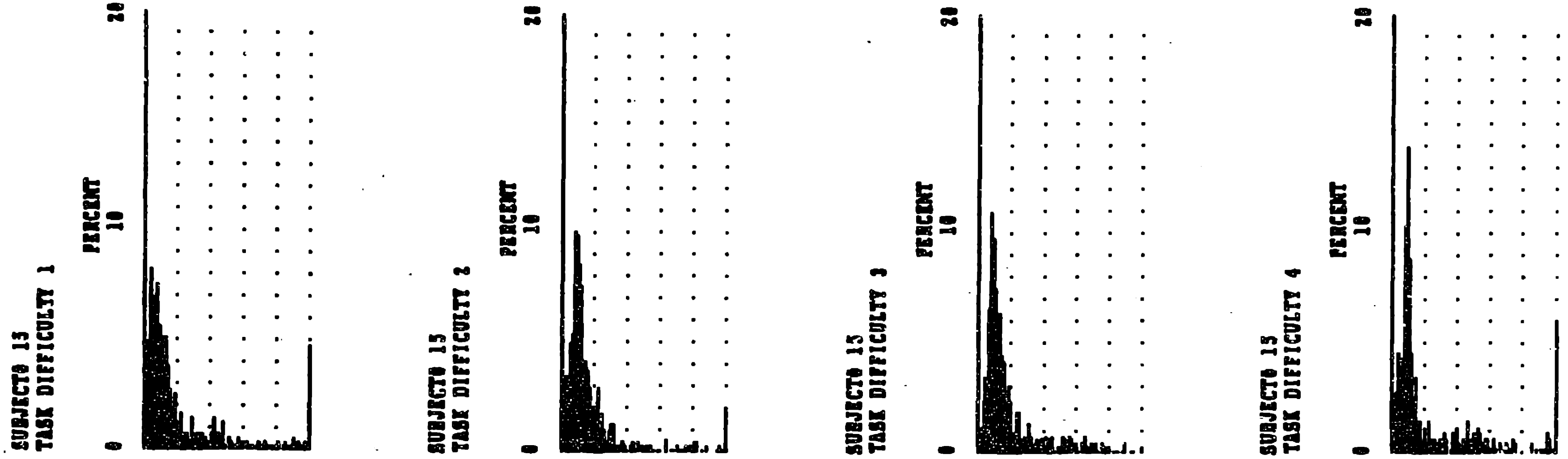


FIGURE 21 DWELL TIME HISTOGRAMS FOR PILOT #6 (F)
(DOTTED LINES INDICATE 1 SEC INTERVALS)

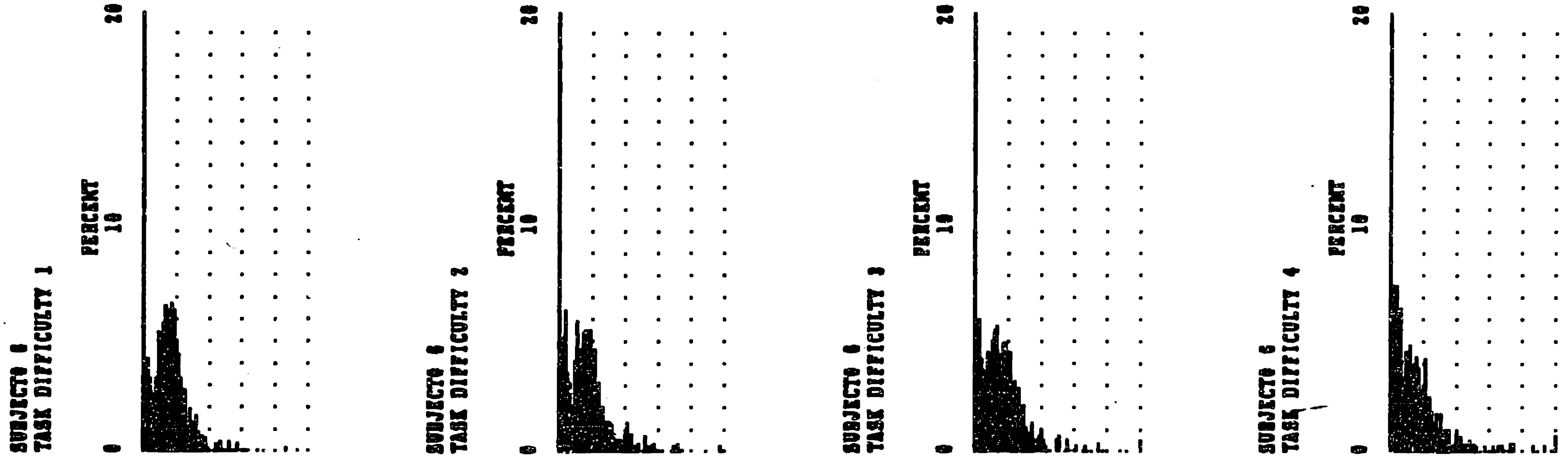


FIGURE 22 DWELL TIME HISTOGRAMS FOR PILOT #12 (G)

(DOTTED LINES INDICATE 1 SEC INTERVALS)

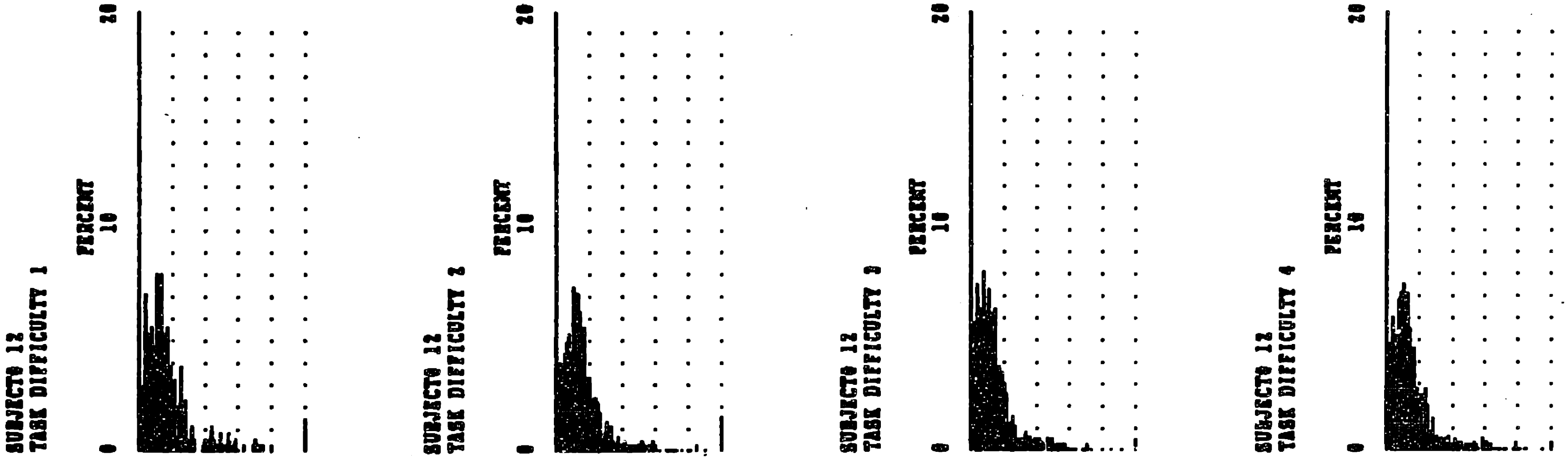


FIGURE 23 DWELL TIME HISTOGRAMS FOR PILOT #14 (H)

(DOTTED LINES INDICATE 1 SEC INTERVALS)

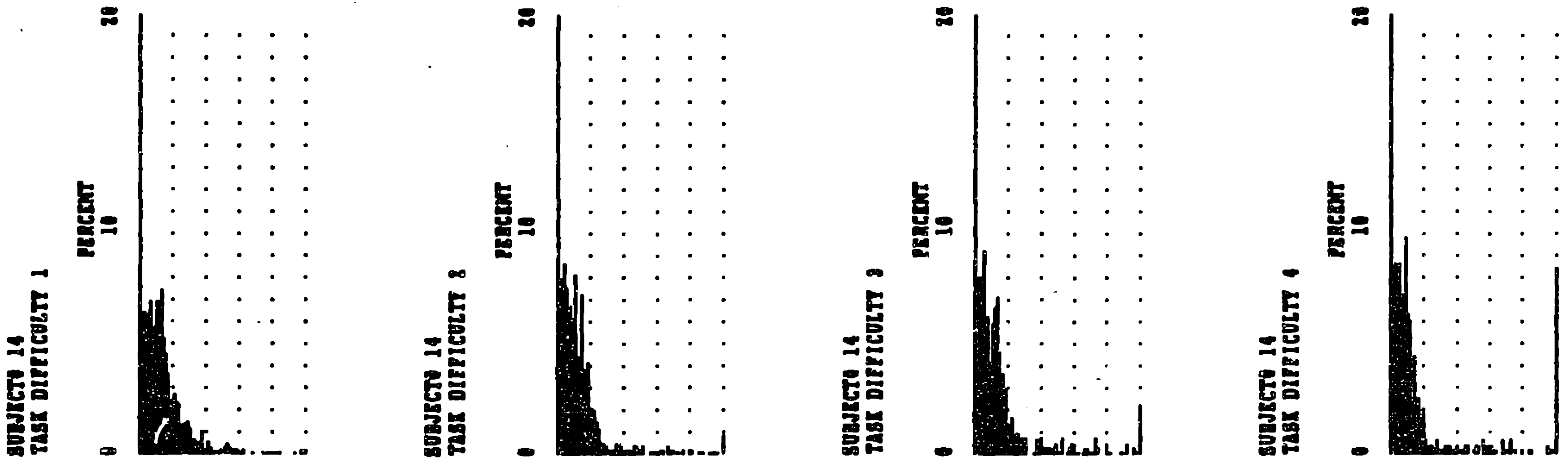


FIGURE 24 DWELL TIME HISTOGRAMS FOR PILOT #8 (I)
(DOTTED LINES INDICATE 1 SEC INTERVALS)

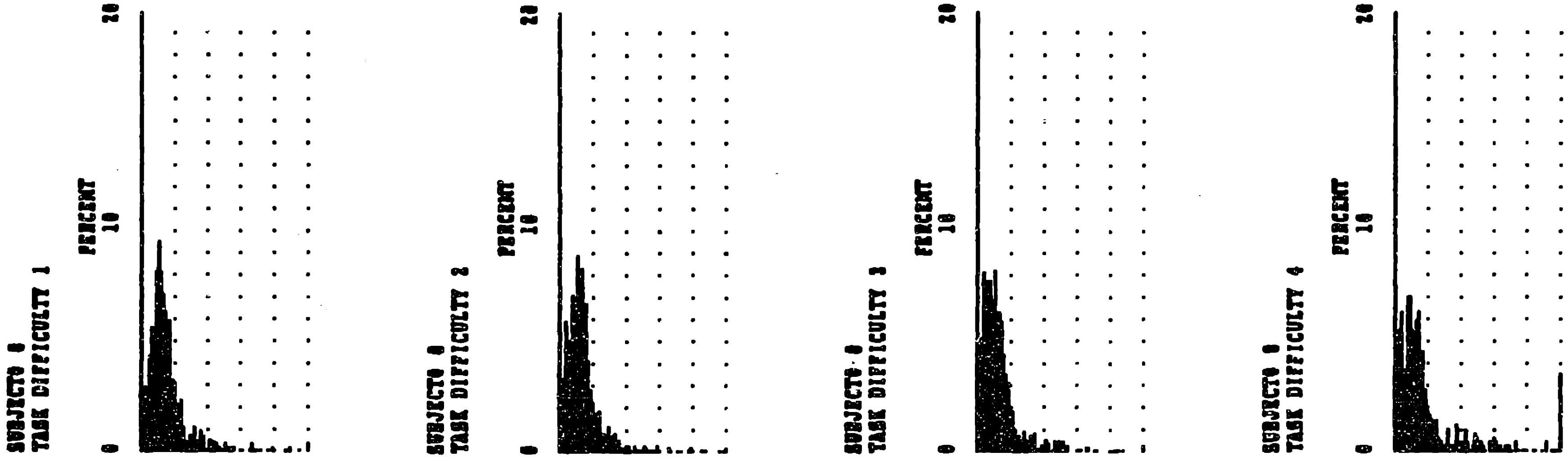


Figure 25 Dwell Time Histograms For Pilot #7 (J)
(dotted lines indicate 1 sec intervals)

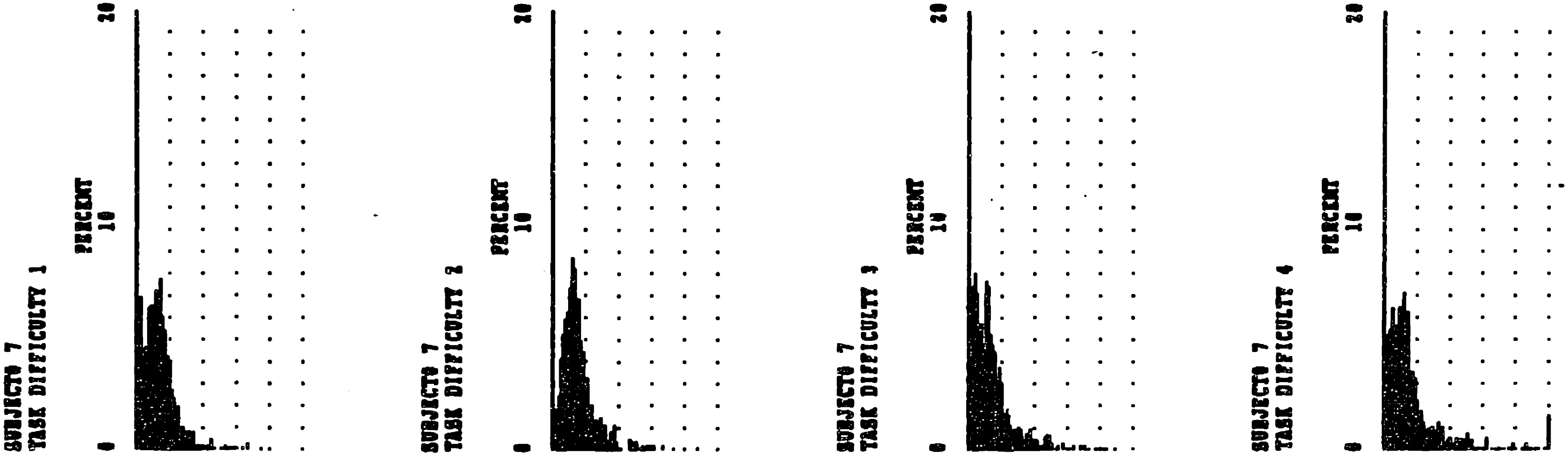
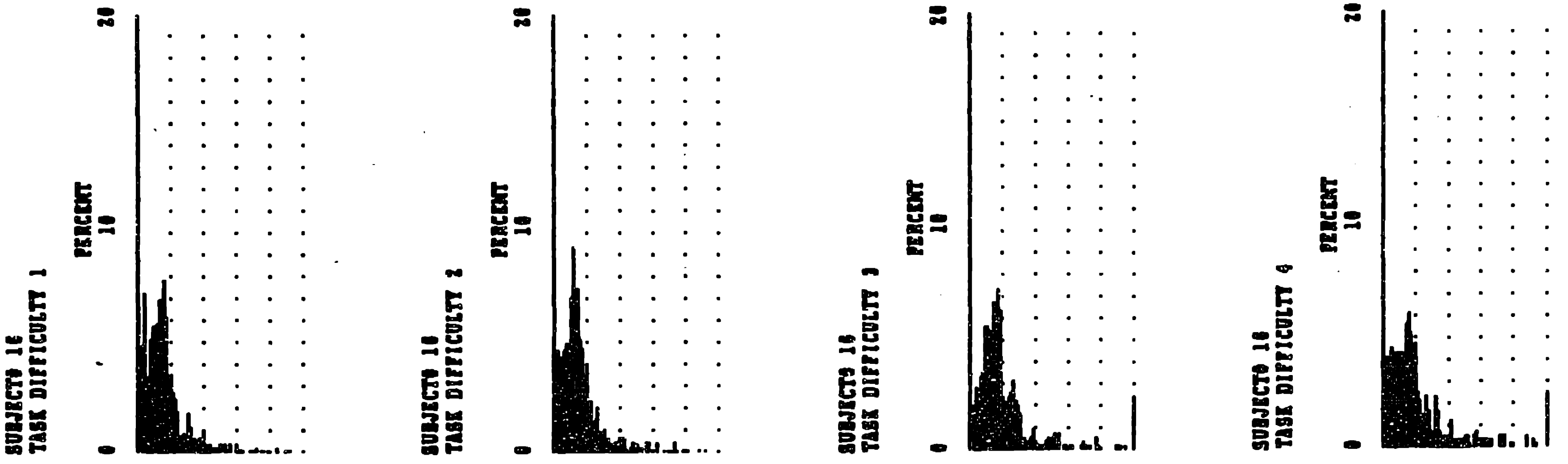


FIGURE 26 DWELL TIME HISTOGRAMS FOR PILOT #16 (K)
(DOTTED LINES INDICATE 1 SEC INTERVALS)



The percentage of fixations that were equal to and longer than four seconds are presented for each case and for each pilot in Table IV. These data are also plotted and may be found in Figure 27. There seems to be only a slight increase in the average percentage over the first three loading levels, but a marked increase occurs during the highest loading level.

The majority of subjects tended to increase the time spent on any one fixation as mental workload increased. Notice should be taken of the data from pilot #15(E) and pilot #12(G), however. Pilot #15(E) has unusually long dwell times for the no-loading case as compared with the other pilots. The amount of his time spent on long dwells then decreases over the next two loading levels indicating that his value for the no-loading case is something more than a possible bad data point. However, what is even more unusual is the sudden turn taken between the third and fourth loading levels to become the highest percentage of all pilots for the highest level of mental loading. Possible explanations for this behavior could be that learning effects might have been large enough to effect his scan until the highest loading case exceeded his critical mental workload limit, or he may have adopted different scanning stratedgies on each of the runs. Pilot #12(G) shows a consistent decrease in percentage with increase in loading level. This, also being in the opposite direction of the other nine pilots, is interesting since her entropy rate increased as opposed to decreasing with increased number task difficulty. An observation of this pilot's behavior

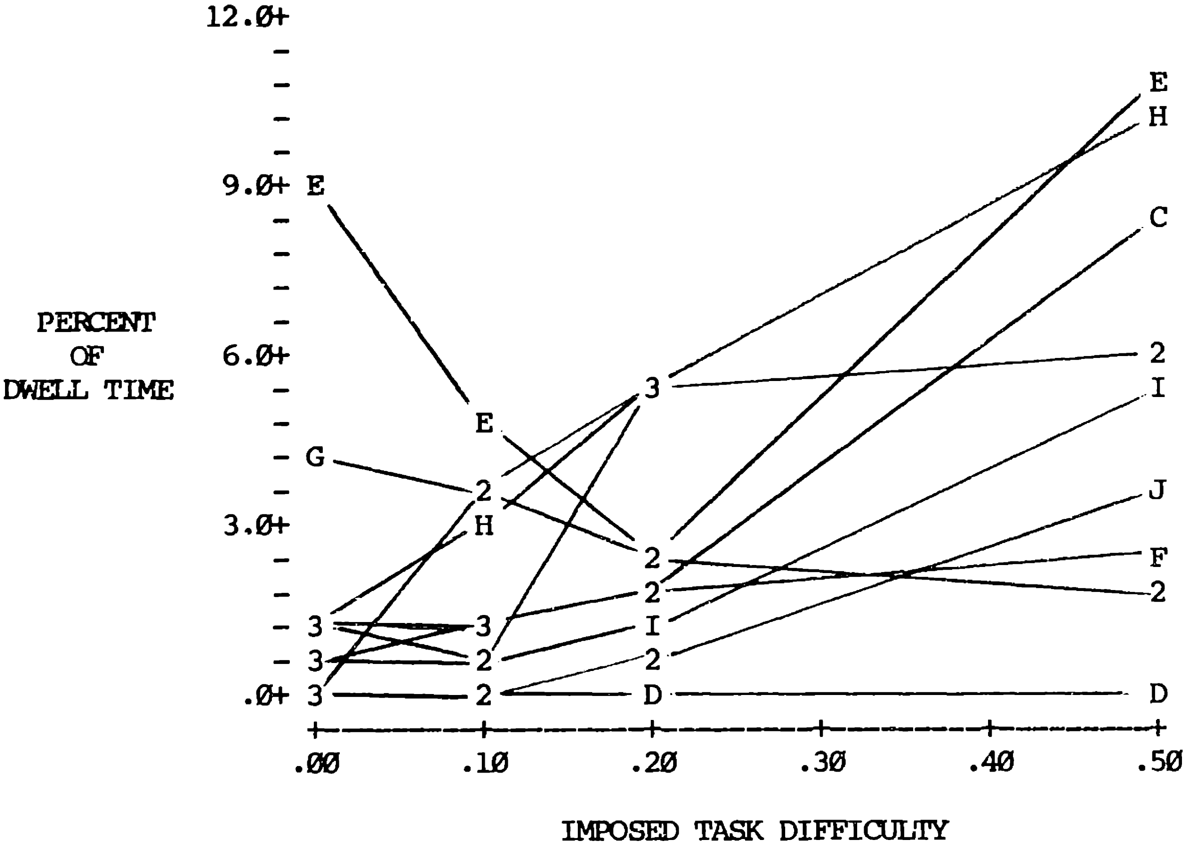


Figure 27 Percent of Dwell Times ≥ 4 sec
vs Imposed Task Difficulty
for all pilots

suggested a lack of motivation to perform the experiment well which may have some implications for the results but these are difficult to assess. If the data from these two subjects are removed, the remaining data takes on a much more consistant trend. This may be seen in Figure 28.

Autocorrelation and Power-Spectral Density

Another analysis method which produced some interesting results was the autocorrelation of the instrument scan pattern. The purpose of this particular method of analysis was to determine whether or not the pilot's scan was altered by the mental loading number task in a periodic fashion. Gopher (1973) found that subjects presented with an auditory task would 1) reduce the number of spontaneous eye movements and 2) would make eye movements in the direction of the ear to which the task was presented. The fundamental result, to be applied here, is that the subjects' eye movements were altered by auditory tasks. One possible alteration that might be encountered is that the frequency at which an instrument is sampled may change as the auditory task changes. In fact a change in frequency proportional to the frequency of presentation of the mental loading number task would provide some evidence that the task directly affects the scanning pattern.

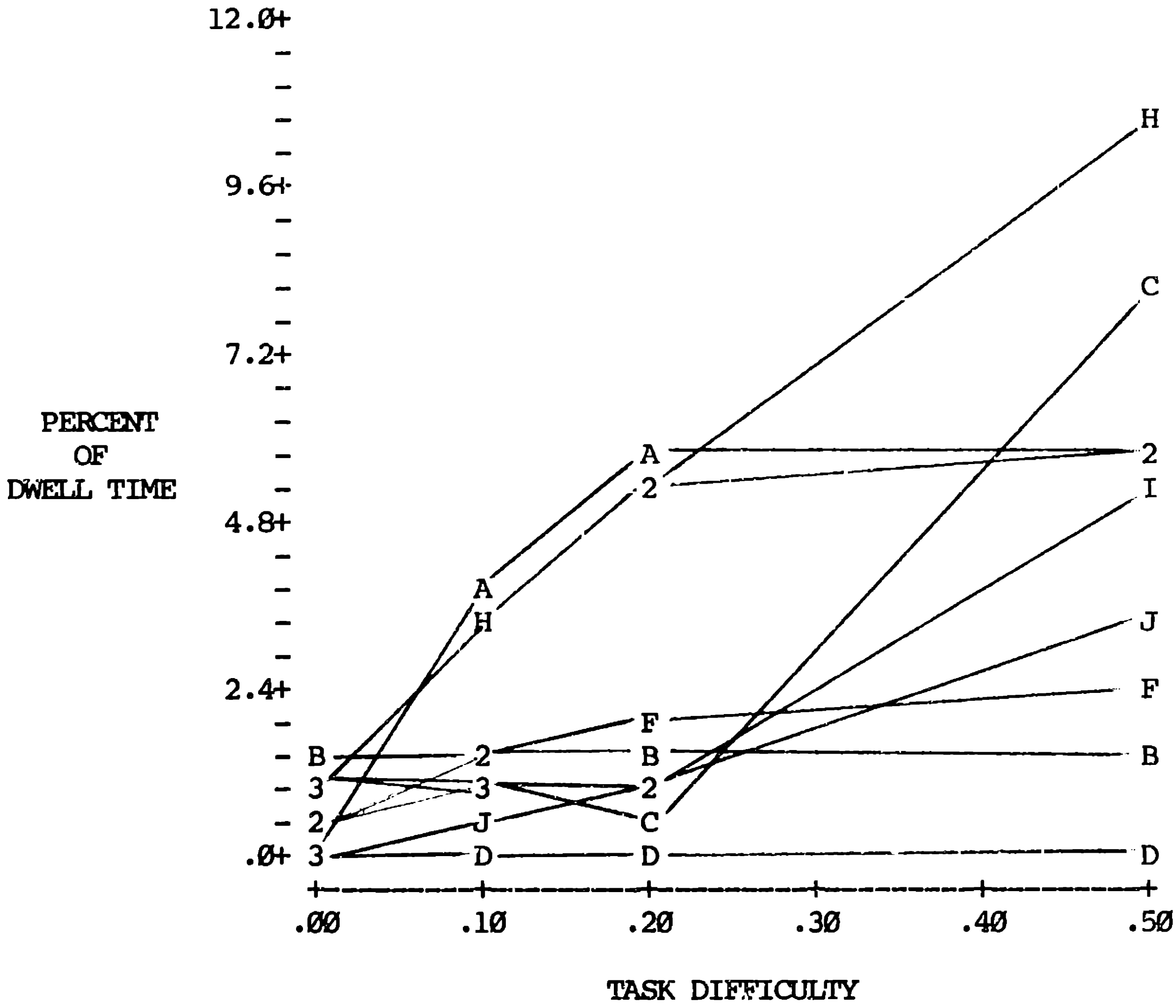


Figure 28 Percent of Dwell Times ≥ 4 sec vs Imposed Task Difficulty excluding pilots #12(G) & #15(E)

The autocorrelation was performed on the data as described below. A sequence of instrument numbers versus time was developed from the data and stored on a disk. Due to the arbitrary nature of the assignment of instrument numbers, the autocorrelation of the signal containing all instrument numbers would not necessarily produce meaningful results. For this reason each of the seven instruments were examined by replacing the time sequence of all instruments with a sequence where the value is 1 for the instrument being studied and 0 for all other instruments. This provided a time sequence of ones and zeros on which a valid autocorrelation could be performed and meaningful results could be obtained. The form of the sample autocorrelation used is given below.

$$R = 1/n \quad f(t) \times f(t + \tau) \qquad (23)$$

where

R = autocorrelation function

n = number of samples at equal time intervals

f(t) = 1 if specified instrument is being fixated and
0 otherwise

$f(t + \tau)$ = f(t) shifted by

Before the autocorrelation was performed, the values of the sequence were summed and divided by the total number of samples. This average value of the sequence was then subtracted from each value in the sequence. The autocorrelation was performed on the resulting sequence. This computation was performed for each of the seven instruments for each loading case on each pilot.

Autocorrelation was chosen since one of its characteristics is that it provides a convenient method of examining the periodic nature of a sequence. An area which may be examined using this method is that of the effects of the mental loading number task on the pilot's scanning behavior in the vicinity of the number presentation. That is, if a pilot were to return to the same instrument each time a number sequence was presented, a periodic feature would develop in his scan due to the periodic nature of the mental loading number task. This information would be useful in determining to what extent pilots use task multiplexing in this particular experimental arrangement. If a pilot's scan was found to be periodic, and the period of his scanning pattern corresponded with that of the number task presentation, it is likely that the pilot's scanning pattern is being driven by the number task. This result could have possible implications for ATC communications in the cockpit.

In order to determine the frequency of any periodicity in the scan, the Fourier transform of the autocorrelation was taken. This produces the power-spectral density for the sequence of instrument numbers. Large peaks in the density function over a given frequence range will indicate an increase in the power or dominance of that frequency range. From this a value for the dominant frequency may be obtained. The power-spectral density was computed on selected autocorrelations and will be discussed later. The power-spectral density was obtained by using a Fast Fourier Transform (FFT) package available on the microprocessor system.

Some interesting results emerged from this analysis the first of which may be seen in Figures 29-32. This shows the autocorrelations for pilot #4 (second highest skill level) for his attitude indicator on each of the four different mental loading cases. A change in the dominant frequency may be seen as the loading is increased. The power-spectral densities shown in Figures 33-36 show the dominant frequencies for the low (ten second intervals), medium (five second intervals), and high (two second intervals) levels of mental workload to be 0.0928 Hertz, 0.1709 Hertz, and 0.3175 Hertz respectively. These frequencies correspond to periods of 10.78 seconds for the low, 5.84 seconds for the medium, and 3.15 seconds for the high level of mental workload. These periods correspond to some degree with the intervals between stimulus presentation in the mental loading number task. Particularly striking is the fact that the periodicity appears for large values of (>100 sec) in the autocorrelation as shown in Figure 31. This implies, at least for this pilot, that the loading task directly influences his scan pattern.

FIGURE 29 AUTOCORRELATION FOR PILOT #4 (B) USING ATTITUDE INDICATOR
(DOTTED LINES INDICATE 10 SEC INTERVALS)
TASK DIFFICULTY .0

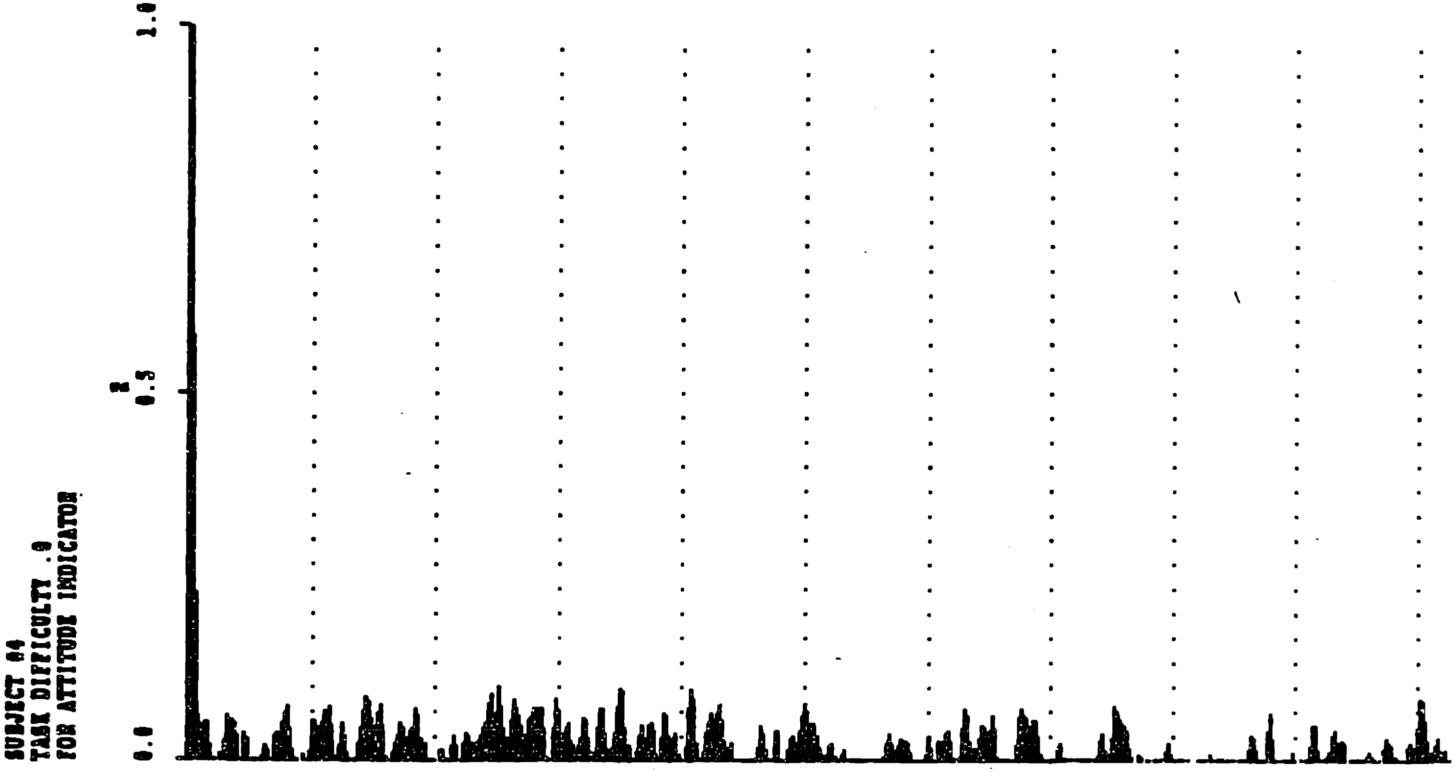


FIGURE 30 AUTOCORRELATION FOR PILOT #4 (B) USING ATTITUDE INDICATOR
(DOTTED LINES INDICATE 10 SEC INTERVALS)
TASK DIFFICULTY .1

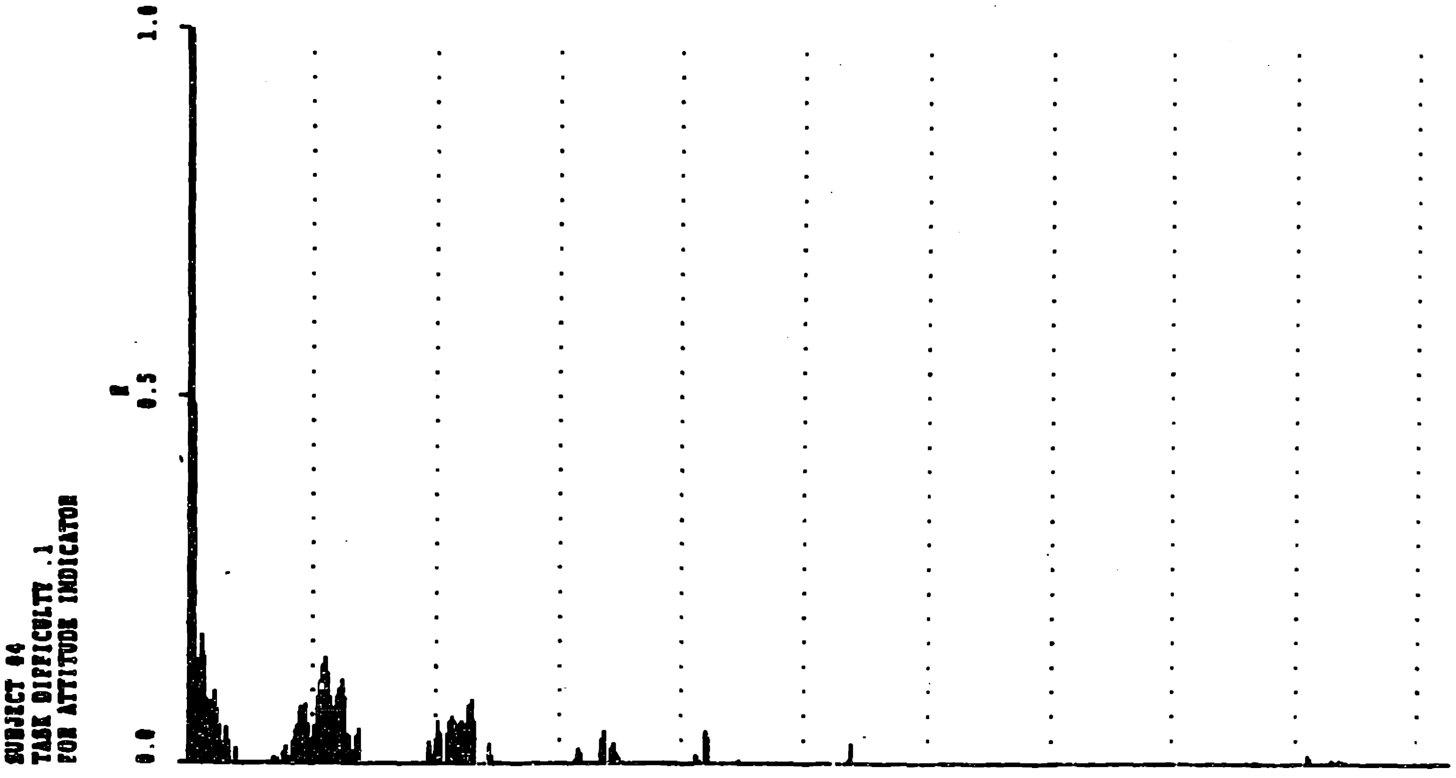


Figure 31 Autocorrelation For Pilot #4 (B) Using Attitude Indicator
(dotted lines indicate 10 sec intervals)
Task Difficulty .2

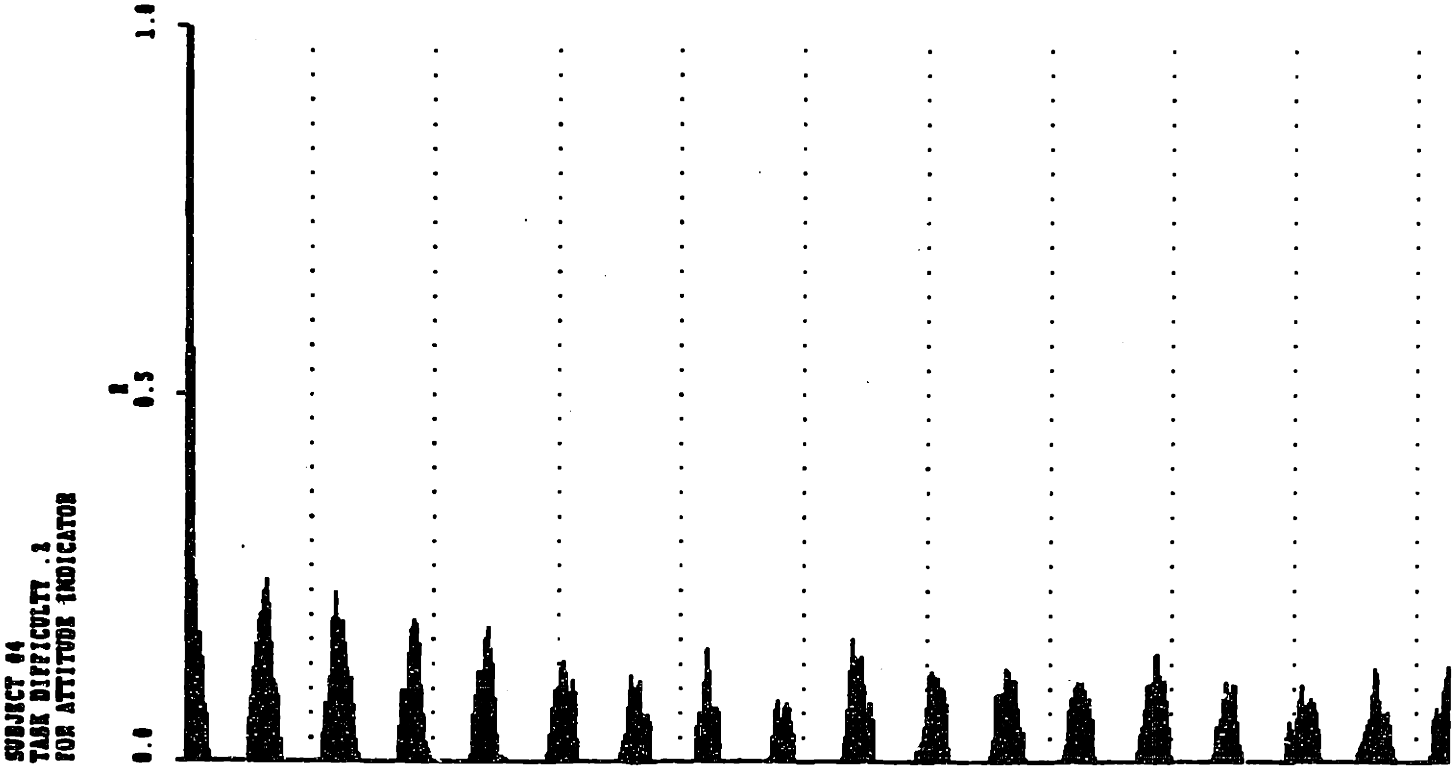


FIGURE 32 AUTOCORRELATION FOR PILOT #4 (B) USING ATTITUDE INDICATOR
(DOTTED LINES INDICATE 10 SEC INTERVALS)
TASK DIFFICULTY .5

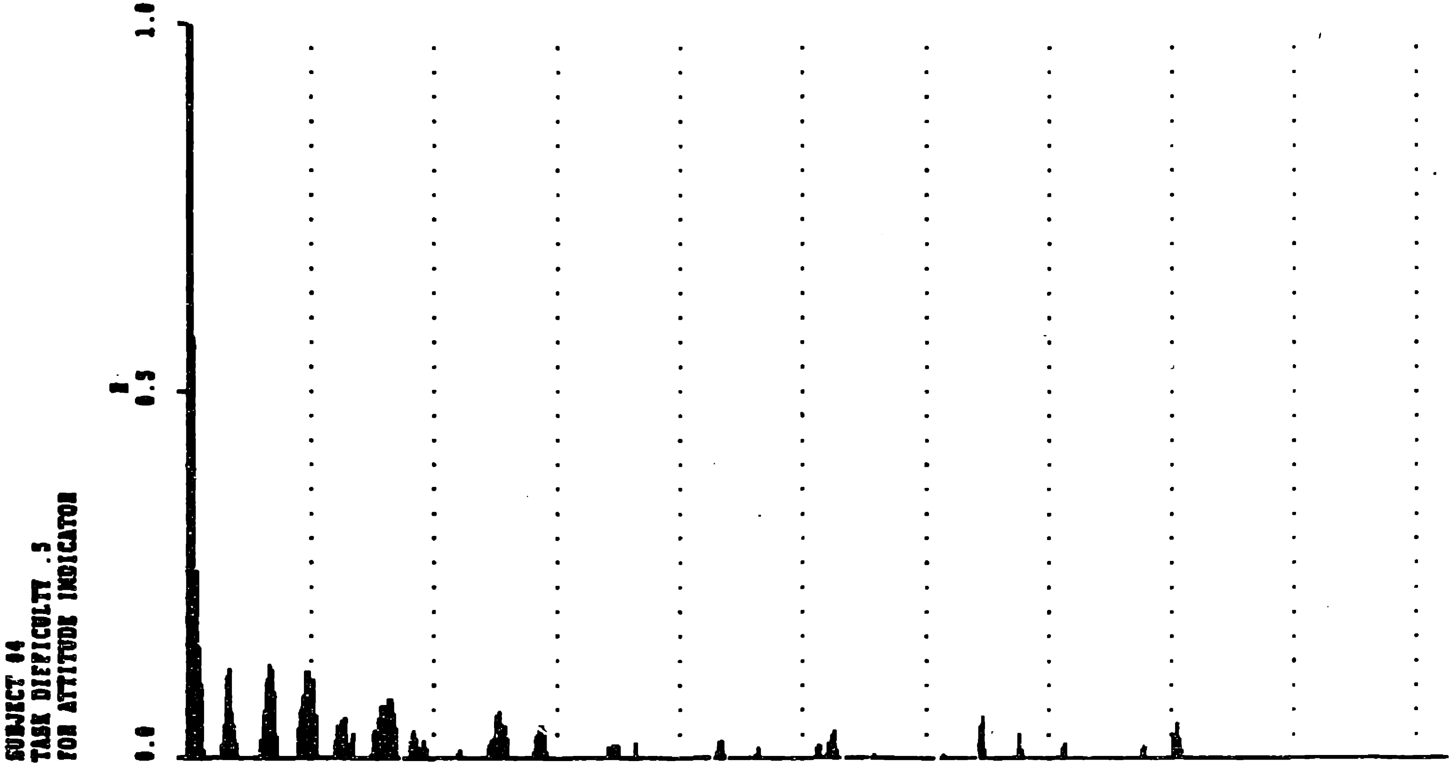


FIGURE 33 POWER-SPECTRAL DENSITY FOR PILOT #4 (B) USING ATTITUDE INDICATOR (DOTTED LINES CORRESPOND TO PERIODS OF 10, 5, AND 2, RESPECTIVELY) TASK DIFFICULTY .0

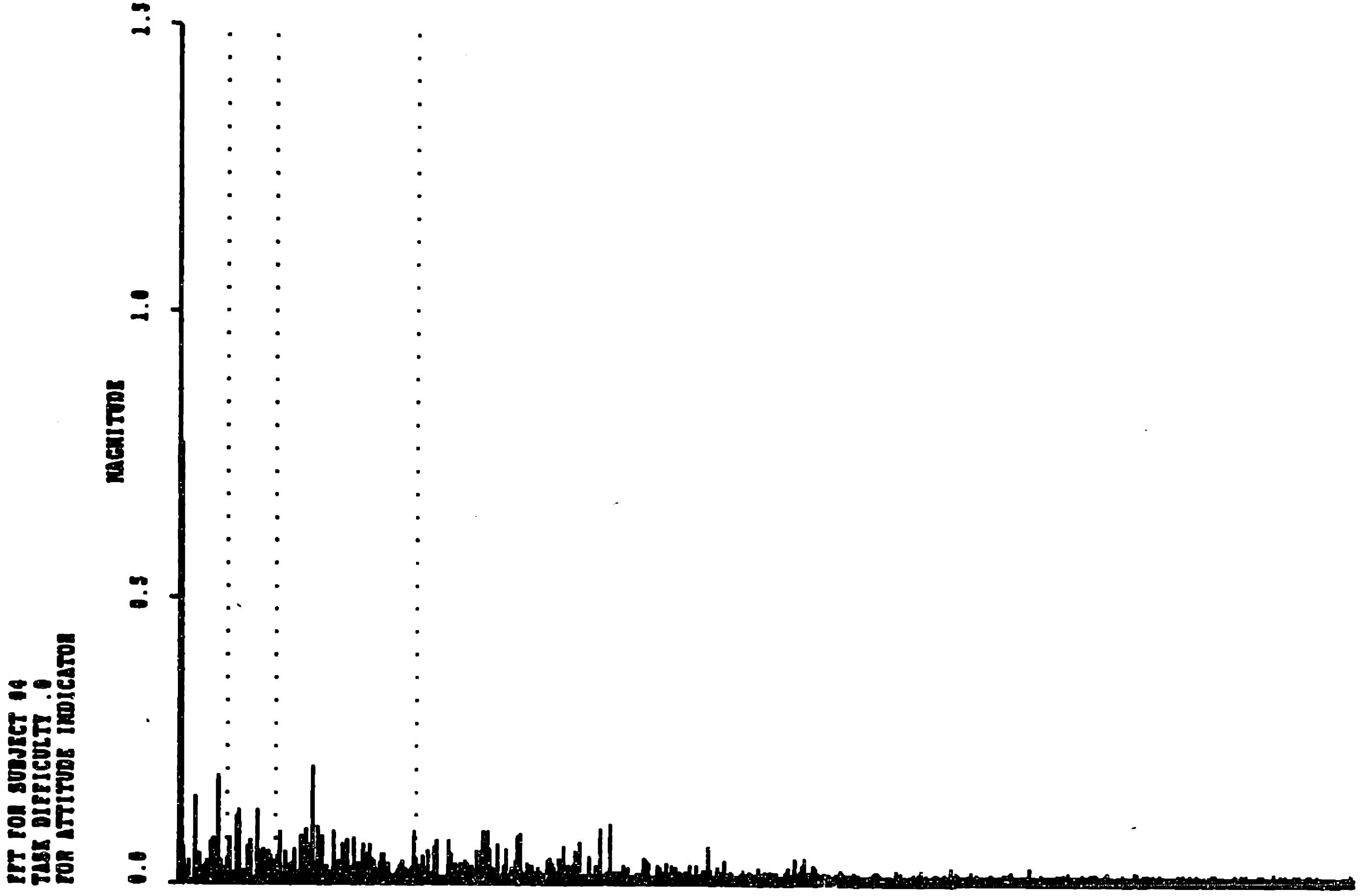


FIGURE 34 POWER-SPECTRAL DENSITY FOR PILOT #4 (B) USING ATTITUDE INDICATOR (DOTTED LINES CORRESPOND TO PERIODS OF 10, 5, AND 2 SEC, RESPECTIVELY) TASK DIFFICULTY .1

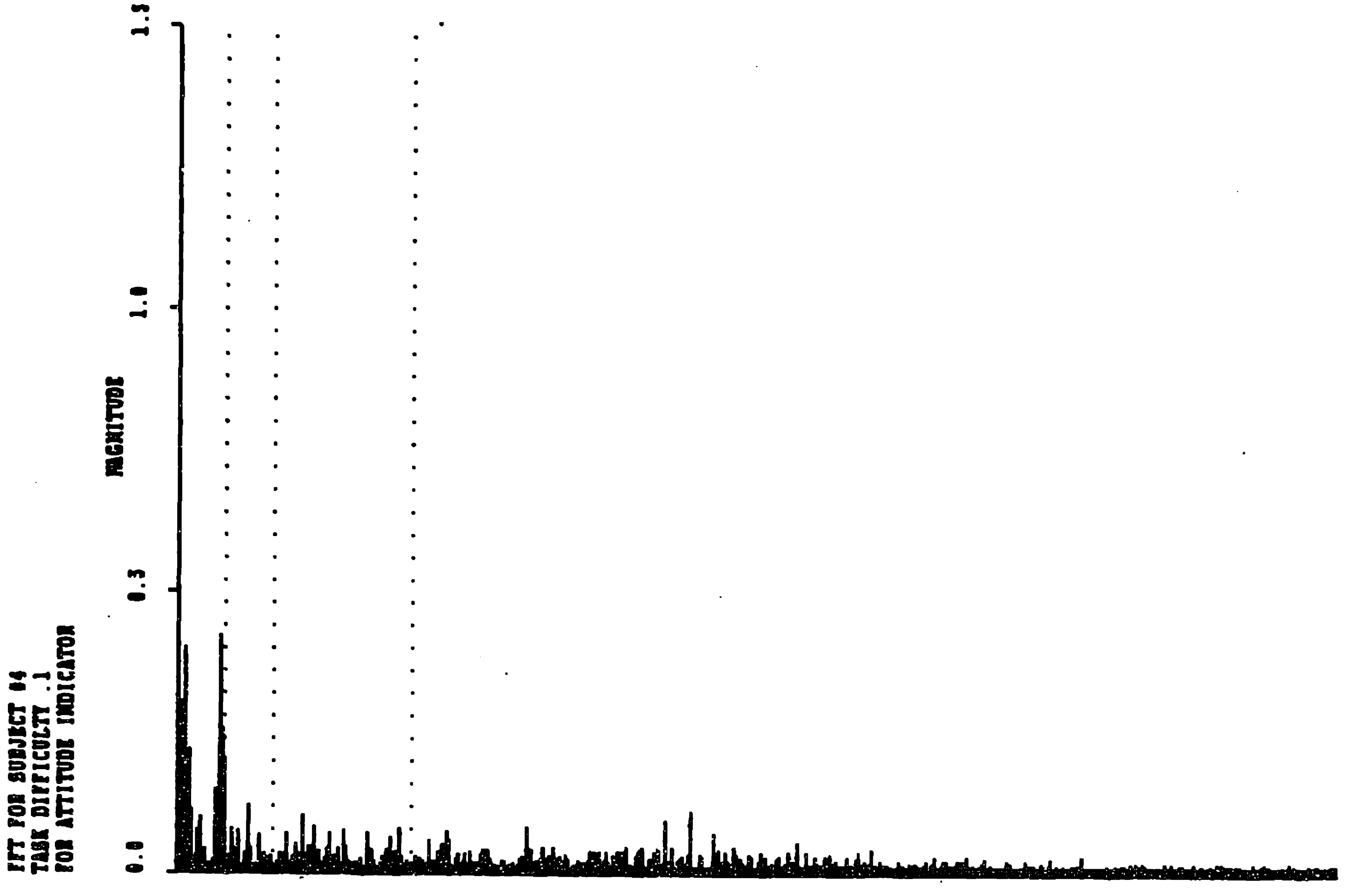


FIGURE 35 POWER-SPECTRAL DENSITY FOR PILOT #4 (B) USING ATTITUDE INDICATOR (DOTTED LINES CORRESPOND TO PERIODS OF 10, 5, AND 2 SEC, RESPECTIVELY) TASK DIFFICULTY .2

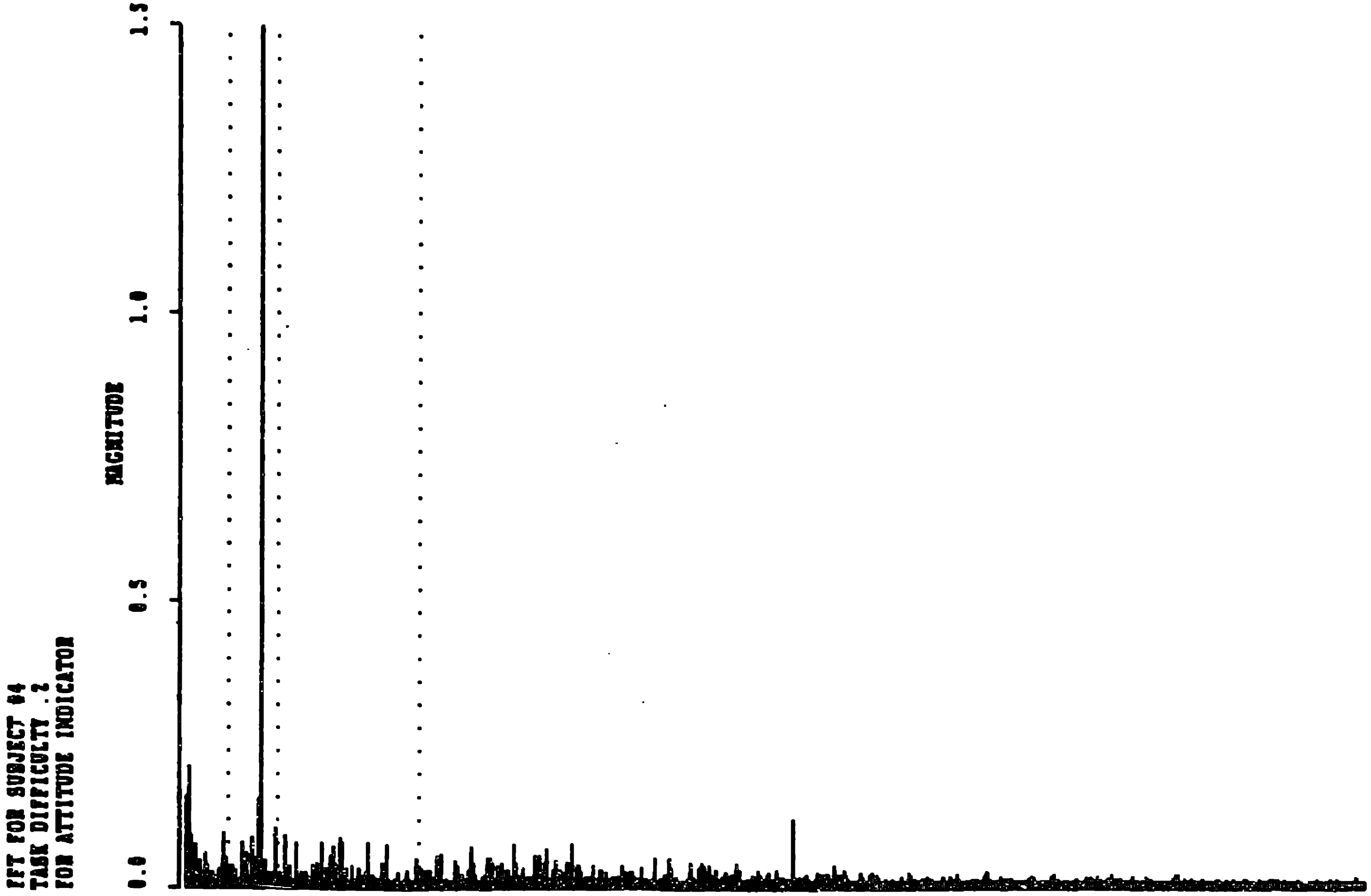
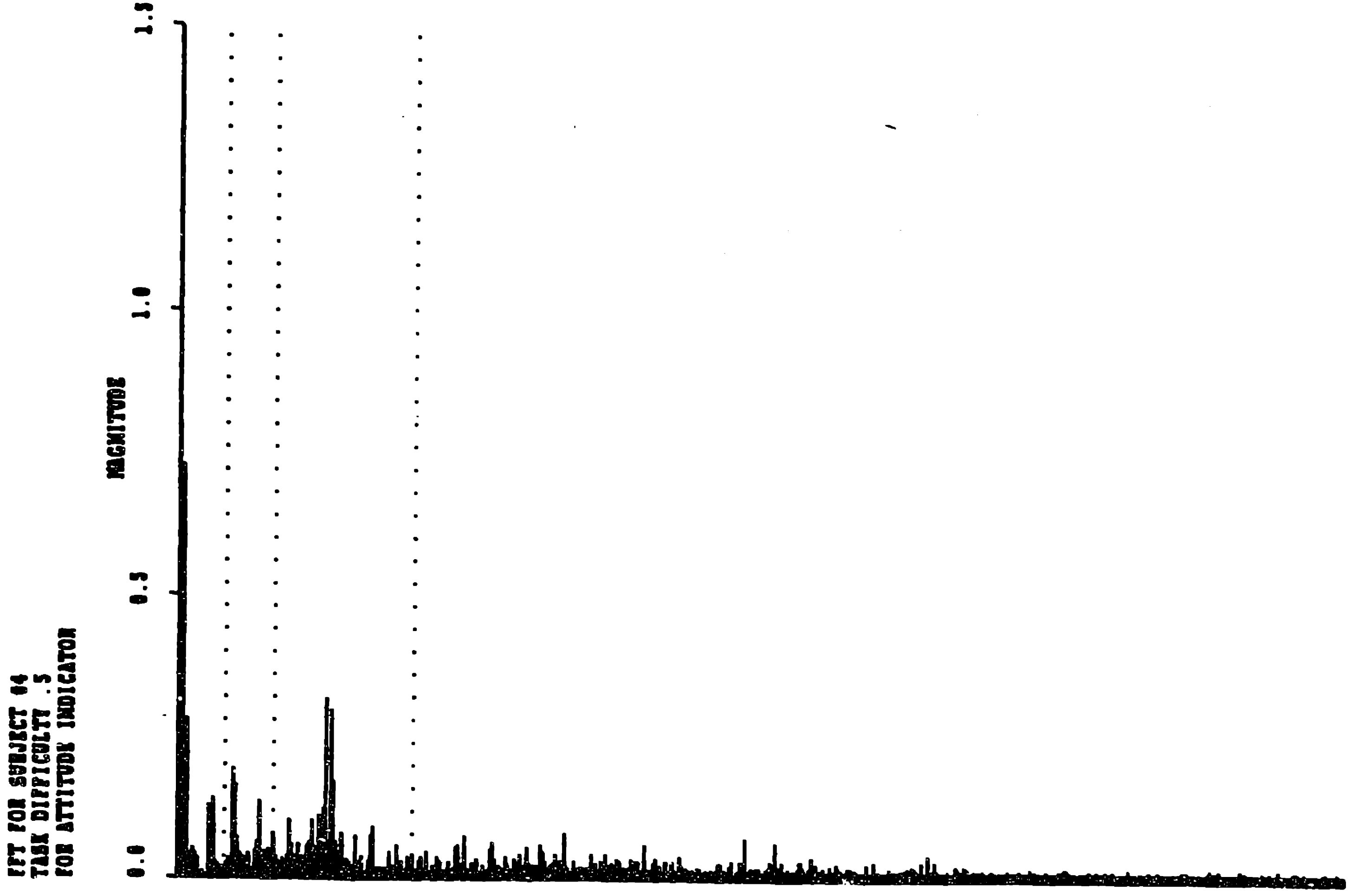


FIGURE 36 POWER-SPECTRAL DENSITY FOR PILOT #4 (B) USING ATTITUDE INDICATOR (DOTTED LINES CORRESPOND TO PERIODS OF 10, 5, AND 2 SEC, RESPECTIVELY) TASK DIFFICULTY .5



In fact, this appears to be the result for all of the higher skilled pilots. This is demonstrated in Table V which presents the period corresponding to the frequency of the largest peak from the power-spectral density for each run. The pilots are arranged from highest to lowest skill level. The data was obtained from the attitude indicator in every case except pilot #13(D). Examination of the autocorrelations for each of this pilot`s instruments revealed the lack of a frequency shift on the glide slope/localizer was found. This is interesting since pilot #13(D) used this instrument more than any of the others. Examination of the rest of the autocorrelations indicated that this was the only case in which looking at a different instrument would improve the matching of the periods to those presented to the pilot via the mental loading number task.

The periods of oscillation for the five pilots of highest skill appear to match those presented to them by the number task very closely. However, the other six pilots do not seem to have any consistent pattern in their autocorrelation of sequences. Most of the pilots showed little or no periodicity in the no-loading case. One possible explanation of these results may be that the higher skilled pilots adapted their scanning to the task much faster and better than the lower skilled subjects. DeMaio, et al(1976) found that skilled pilots evidently developed optimum scanning strategies when presented novel tasks much faster than unskilled pilots. Another explanation may be that skilled pilots have a better developed ability to time multiplex several simultaneous tasks.

Summary

This chapter presents the majority of the analytical work performed for this thesis. The second phase of experiments are presented along with the analysis of their results. There are four major changes in the experiments from Phase I. These are 1) the mental loading number task was presented by a microprocessor controlled voice synthesizer as opposed to a cassette tape recorder 2) the response to the number task was stored to provide a performance measure 3) the number of loading levels was increased from three to four and 4) aircraft performance data was digitilized and made available for use.

Results are presented from many analysis methods. The skill of each pilot was rank ordered using an equation developed for pilots by another investigator. The performance measures measures utilized in this work are obtained from 1) the piloting task and 2) the mental loading number task. Four indicators of piloting task performance are 1) glide slope RMS error 2) Localizer RMS error and 3) percent of power above a specified frequency for the glide slope and localizer. Each of these are plotted versus task difficulty in Figure 8 through Figure 11. The mental loading number task is scored by computing the percentage of correct responses. These results are used to form a combined performance measure which is used in the model development. The method of entropy and entropy rate computation are discussed. The results of the entropy rate for length two sequences fit an exponential model with task difficulty with a high loading level based on results of a Student`s T-test. The exponential model relating performance, skill, and mental workload was then developed using data from seven of the pilots. This model was then applied to the results of three other subjects to partially evaluate model performance. The results were inconclusive but encouraging trends appeared in the curves. The last analysis method was autocorrelation of the pilot`s instrument scan pattern. The results from individual instruments (usually the attitude indicator) demonstrate a pronounced effect of the number task on the scanning patterns of the higher skilled pilots but not for pilots of lower skill. This may result from faster development of optimum scanning stratedgies by the higher skilled pilots and from the fact that higher skilled pilots may have a better developed ability to time mutiplex several simultaneous tasks.

CHAPTER 8

DISCUSSION AND CONCLUSIONS

Discussion

The primary purpose of this work has been the examination of the relationship between skill, performance, and mental workloaad using pilot scanning behavior as an analysis aid. The results presented in Chapters 5 and 7 provide some evidence that such a relationship does exist and that scanning behavior may provide a convenient method of mental workload prediction. The model presented describes performance as a function of skill and mental workload. The results of this work are not conclusive enough to warrant immediate application but they do strongly support the validity of the method. An important point to make is that the generality of these results has not been examined. Therefore, application of this model to different situations should be done with great care. Further validation and model development in this area are needed.

In addition to demonstrating the usefulness of this method as a performance prediction, many basic facts concerning scanning behavior have been observed and examined. The increase in dwell time on the instruments as mental workload increases has been observed by other investigators and has once again been clearly demonstrated in this work. It may, in fact, be possible to obtain a mental workload measurement from this result.

The use of entropy rate to estimate relative task difficulty appears to have a great deal of potential. Though the changes with mental loading were small, they were significant and fairly consistant. More importantly, they appear to be independent of skill level. This characterisitic is desirable in the evaluation of highly skilled pilots. The entropy rate is useful since it combines both spatial and temporal characteristics into one measure.

One point to make clear is that the entropy rate presented here is not the information processing rate of the pilot. A much more detailed accounting of how much information is obtained from each instrument will be required in order to make a definitive statement as to the actual information processing rate of the pilot. Even so, the good fit of the entropy rate data to the exponential equation with task difficulty demonstrates the accuracy and usefulness of this measure as a predictive tool for mental workload assessment.

The autocorrelation of the pilot's instrument/dwell time sequence demonstrated several interesting characteristics. The most interesting result was the influence of the number task on the skilled pilots' scanning patterns. This result is evidenced by the correspondence between the dominant frequencies in the pilot's scan and the frequency of presentation of the mental loading number task. For many pilots, their most frequently used instrument was the attitude indicator, for which the autocorrelation produced the most interesting results. As seen in Table VI, only the first five pilots show a correspondance between their dominant scanning frequency and the frequency of the number task presentation. The reasons for this are not fully understood but may depend on the skilled pilot's ability to adopt to new scanning situations more readily than unskilled pilots, and the fact that skilled pilots may be better task multiplexers when performing simulaneous tasks. One other possible explanation for this may be that the skilled pilots are interrupting the piloting task completely, answering the mental loading number task, and then returning to the piloting task once again. This is simply one mechanism for task multiplexing mentioned above. This could explain the observed behavior to to a certain extent since the lower skilled pilots would not be able to leave the piloting task for the same length of time as the skilled pilots and still be able to control the piloting task. Therefore, the periodicty in the scanning patterns of the less skilled pilots would not be as evident and their performances would tend to drop off faster than for higher skilled pilots.

Conclusions

Probably the most significant result of this work is the relation between entropy rate and imposed task difficulty. This allows the computation of perceived task difficulty (mental workload) from a scanning behavior parameter. Therefore, a non-invasive method for the determination of mental workload is an aircraft pilot has been demonstrated for this particular experimental paradigm.

During the development of this method, several factors were found to be a function of the skill level of the pilot. The percent occurrence of the two most used sequences of the pilot and the autocorrelation of his instrument scan provided the best indicators of skill. Further research in these two areas may produce a much needed method of quantifying skill in aircraft pilots. Another result of this work is a model relating performance, skill, and workload in aircraft pilots for this experimental design. This model provides a means of performance prediction from skill level and perceived task difficulty. The perceived task difficulty may be obtained from the entropy rate of the instrument scan pattern, and the relative skill level may be estimated using equation 6 in the thesis. This model produced fair results but is probably not complete. It is possible that there are some terms that are not considered which are important in the prediction of performance. However, the model does make a first step toware the development of a method of performance prediction.

The applications of these results may be useful in the evaluation of many tasks requiring skilled operators. Optimum display design is an area that would find a method of mental workload and performance prediction such as this invaluable. Though this model is not yet ready for such an ambitious application as this, the general work has been laid down on which to continue experimentation. One possible model might compare the change in entropy rates and performance between two candidate display formats to determine which allowed the maximum performance while requiring the minimum mental workload.

Another area of possible application for this work is in the field of flight training. Monitoring of progress and determination of skill are two possible outcomes from this work that would greatly assist the training of new instrument pilots. A properly designed system could aid in the detection and correction of problem areas for students. This method of scanning pattern analysis will be especially important with the introduction of CRT based instrumentation into the aircraft. New types of information in many possibly formats will be available to many pilots, both student and professional, in the near future , and little is known of the optimum method of presentation, scanning, or training pilots to scan these new instruments. This work may provide the basis of the accomplishment of these tasks.

This thesis has provided the ground work for the development of a very powerful, non-invasive mental workload measurement method. In the process, a great deal of information on the fundamentals of instrument scanning by pilots was acquired. This information was employed in the development of a model relating performance, skill, and mental workload of the aircraft pilot. Application of the results and analysis method of this work may provide a useful predictive method for pilot performance.

Recommendations for Further Research

There are a number of areas in which further research would possibly provide interesting and useful results. Several of them are listed below for those interested in pursuing this research area.

1) model verification - application of the model developed above to a design problem with known results to test the model's validity. More subjects are needed.

2) information theory - analytical work is needed to develop the actual relationship between the pilot's scanning pattern and the information actually being processed.
A determination of the information obtained from each instrument is needed.

3) skill determination - both analytical and experimental work are needed in the area of skill determination to provide a generally applicable method of skill assessment. Percent occurrence of the ten most used sequences, autocorrelation of the scan pattern, or dwell time histograms may provide the model needed.

4) specific experiments - several specific experimental situations that should be examined are:

    i) repeat basic experiments using a varying interval mental loading task.
    This will help eliminate task specific results obtained thus far and allow examination of effects of a randomly applied task.
    Examination of the use of cross correlation to analyze scanning data from this type of experiment is also needed.

    ii) change mental loading task to a more realistic task of ATC communications making use of the VOTRAX speech module.

    iii) perform experiments of varying lengths to help determine the minimum time required to obtain good results.
    Try ILS approaches to determine the effects of changing task difficulty during the run.

iv) perform experiments in more complex simulations to begin extensions to the procedures to allow eventual "real world" applications.

References

Albanese, R.A. Mathematical analysis and computer simulation in military mission workload assessment. Proceedings of the AGARD-CPP-216, April, 1977, A13-1 - A13-6.

Allen, R.W., Clement, W.F., and Jex, H.R., "Research on Display Scanning, and Reconstruction using Separate Main and Secondary Tracking Tasks", NASA CR-1569, July, 1970.

Atteneave, F., Application of Information Theory in Pyschology, Holt, Rinehart, and Winston, 1959.

Beatty, J. Pupillometric measurement of cognitive workload. Proceedings of the 12th Annual NASA-University Conference on Manual Control, University of Illinois, May, 1976, 135-143. (NASA TM X-73 70).

Bergeron, H.P. "Single Pilot IFR Autopilot Complexity/Benefit Tradeoff Study", AIAA Aircraft Systems and Technology Meeting, Anaheim, California, August, 1980, AIAA Paper #80-1869.

Cooper, G.E. and Harper, R.P., Jr. The use of pilot rating in the evaluation of aircraft handling qualities. Moffett Field, California: National Aeronautics and Space Administration, Ames Research Center, NASA TN-D-5153, April, 1969.

Corkindale, M.G.G. A flight simulator study of missile control performance as a function of concurrent workload. Proceedings of the AGARD Conference on Simulation and Study of High Workload Operations, AGARD-CP-146, April, 1974, A5-1 - A5-6.

DeMaio, J., Parkinson, S., Leshowitz, B., Crosby, J., and Thorpe, J., "Visual Scanning: Comparisons between Student and Instructor Pilots", Air Force Human Factors Resources Laboratory Report # AFHRL-TR-76-10.

Dick, A.O. "Instrument Scanning and Controlling: Using Eye Movement Data to Understand Pilot Behavior and Strategies", NASA CR-3306, September, 1980.

Dick, A.O. and Bailey, G. A comparison between oculometer data and pilot opinion on the usefulness of instruments during landing. Rochester, New York: University of Rochester, Center for Visual Science, Technical Report 4-76, 1976.

Ellis, S.R. and Stark, L. "Eye Movements during the Viewing of Necker Cubes", Perception, 7:575-581, 1978.

Ephrath, A.E., "Pilot Performance in Zero-Visibility Precision Approach", Ph.D. Thesis, Dept. of Aero. and Astro., M.I.T., June, 1975.

Gafni, H. and Zeevi, Y.Y., "A Model for Separation of Spatial and Temporal Information in the Visual System", Biol. Cybernetics, 28:

73-82, 1977.

Gopher, D. Eye movement patterns in selective listening tasks of focused attention. Perception and Psychophysics, 1973, 14, 259-264.

Harris, R.L., Preliminary nvestigation of Pilot Scanning Techniques of Dial Pointing Instruments. NASA Technical Memorandum 80079, November, 1979.

Harris, R.L., Sr., and Christhilf, D., "What do Pilots See in Displays?", presented at the Human Factors Society Meeting, Los Angeles, October, 1980.

Harris, R.L., Sr. and Mixon, R., "Advanced Transport Operation Effects on Pilot Scan Patterns", proceeding of the Human Factors Society, 23rd annual meeting, Boston, Massachusetts, 1979, pp 347-351.

Hay, G.C., House, C.D. and Sulzer, R.L. Summary Report of 1977-1978 Task Force on Crew Workload. Federal Aviation Administration Report No. FAA-EM-78-5, December, 1978.

Holland, M.K. and Tarlow, G. Blinking and mental load. Psychological Reports, 1972, 31, 119-127.

Hollister, W., LaPointe, A., Oman, C. and Tole, J. "Identifying and Determining Skill Degradations of Private and Commercial Pilots", FAA Report FAA-RD-73-91, June, 1973.

Jex, H.R. and Allen, R.W. Research on a new human dynamic response test battery. Part I. Test development and validation. Proceedings of the 6th Annual NASA-University Conference on Manual Control, Wright-Patterson AFB, Ohio, April, 1970, (a).

Jex, H.R. and Allen, R.W. Research on a new human dynamic response test battery. Part II. Psychophysiological correlates. Proceedings of the 6th Annual NASA—University Conference on Manual Control, Wright-Patterson AFB, Ohio, April, 1970, (b).

Jones, R.E., Milton, J.L., and Fitts, P.M., "Eye Fixations of Aircraft Pilots: A Review of Prior Eye Movement Studies and a Description of a Technique for Recording the Frequency, Duration, and Sequence of Eye Fixations during Instrument Flight", USAF Tech Report 5837 (AT I 65996), 1946.

Katz, J.G. Pilot Workload in the Air Transport Environment: Measurement, Theory, and the Influence of Air Traffic Control, Master's Thesis, Massachusetts Institute of Technology, June, 1979.

Kennedy, J.P. Time-sharing effects on pilot tracking performance. Master's thesis, Naval Postgraduate School, Monterey, California, September, 1975. (AD A016 378).

Klein, T.J. and Cassidy, W.B. Relating operator capabilities to system demands. Dallas, Texas: LTV Aerospace, Vought Systems Division, 1972.

Krebs, M.J., and Wingert, J.W., "Use of the Oculometer for Pilot Workload Measurements" NASA CR-144951, Feb, 1976.

Merchant, J. and Morrissette, R. A Remote Oculometer Permitting Head Movement. AMRL-TR-73-69, U.S. Air Force, November, 1973.

Middleton, D.B., Hurt, G.J. Jr., Wise, M.A., and Holt, J.D. "Description and Flight Tests of an Oculometer" NASA Technical Note NASA TN D-8419, June, 1977.

Miller, G.A., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", Psychological Review 63(2):81-97, March, 1956.

Moray, N., Ed., Mental Workload, Theory and Measurement, Plenum, 1979.

Mulder, G. and Mulder-Hajonides van der Meulen, W.R.E.H. Mental load and the measurement of heart rate variability. *Ergonoomics*, 1973, 16, 69-83.

Noton D. and Stark, L., "Eye Movements and Visual Perception", Scientific American, 244(6):35-43, June, 1971.

Sayers, B.M. Analysis of heart rate variability. *Ergonomics*, 1973, 16, 17-32.

Schwartz, M., Information Transmission, Modulation, and Noise, McGraw Hill, New York, 1959.

Senders, J.W., Elkind J.L., Grignetti, M.C., and Smallwood, R. An Investigation of the Visual Sampling Behaviour of Human Observers. NASA CR-434, 1966.

Shannon, C.E. and Weaver, W. THE MATHEMATICAL THEORY OF COMMUNICATION, The University of Illinois Press, 1949.

Sheridan, T. "The Changing Role of the Pilot from Manual Controller to Computer Supervisor". Proceedings of the Symposium on Man-System Interface: Advances in Workload Study, Washington, D.C., 1978.

Sheridan, T. and Simpson, R.W. Toward the Definition and Measurement of the Mental Workload of Transport Pilots. Technical Report, Massachusetts Institute of Technology, 1979.

Spady, A.A., "Airline Pilot Scanning Behavior during Approaches and Landing in a Boeing 737 Simulator", Presented at AGARD 25th Guidance and Control Panel Meeting/Symposium on "Guidance and Control Design Considerations for Low Altitude and Terminal Area Flight", Dayton, Ohio October 17-20, 1977.

Spady, A.A., "Airline Pilot Scan Patterns During Simulated ILS Approaches", NASA Technical Paper 1250, October 1978.

Spyker, D.A., Stackhouse, S.P., Khalafalla, A.S. and McLane, R.C. Development of techniques for measuring pilot workload. Washington, D.C. National Aeronautics and Space Administration, Contractors' Report NASA CR-1888, November, 1971.

Stackhouse, S. Workload evaluation of LLNO display. Minneapolis, Minnesota: Honeywell, 7201-3408, October, 1973.

Strother, D.D. Aircrew performance in army aviation. Proceedings of Conference 27-29 November, 1973, U.S. Army Aviation Center, Fort Rucker, Alabama, November, 1973, 188-192.

Tole, J.R., Stephens, A.T., Harris, R.L.,Sr., and Ephrath, A., "Visual Scanning Behavior and Mental Workload in Aircraft Pilots", accepted for publication in Aviation Space and Environmental Medicine, 1981.

Waller, M.C. An investigation of correlation between pilot scanning behavior and workload using stepwise regression analysis. Hampton, Virginia: NASA Langley Research Center, NASA TM X-3344, March, 1976.

Weir, D.H. and Klein, R.H. Measurement and analysis of pilot scanning behavior during simulated instrument approaches. Proceedings of the 6th Annual NASA-University Conference on Manual Controol, Wright-Patterson AFB, April, 1970, 83-108.

Weirwille, W.W. and Williges, R.C. Survey and Analysis of Operator Workload Assessment Techniques. Final TEchnical Report, Naval Air Test Center, September, 1978.

Wempe, T.E. and Baty, D.L. Human information processing rates during certain multiaxis tracking tasks with a concurrent auditory task. IEEE Transactions on Man-Machine Systems, 1968, MMS-9, 129-138.

White, R.T. and Ware, C.T. Prediction of human operator performance in the design of command and control systems. Paper presented at the Annual Meeting of the Western Psychological Association, Vancouver, British Columbia, June 21, 1969. (Douglas Paper 5539).

Wickens, C.D. The effect of time sharing on the performance of information processing tasks: A feedback control analysis. Ann Arbor, Michigan: The University of Michigan, Human Performance Center, Technical Report No. 51, August, 1974.

Wittenborn, J.R., "Factorial Equations for Tests of Attention", Psychometrika, 8(1):19-35, March, 1943.

TABLES

	WORKLOAD INDEX		
Pilot #	No Loading	4-sec Intervals	2-sec Intervals
9	87	93	95
5	82	94	97
7	70	89	**
Average	80	92	96

** Data not available

TABLE I

Workload Index

Pilot #	4-sec Intervals	2-sec Intervals
4	0.63	3.69
11	1.95	7.33
9	6.80	8.46
5	8.59	20.08
10	19.80	23.39
7	6.90	13.21

TABLE II

Percent Dwell Times $\geq$ 5 sec for Each Loading Case

NASA Pilot#	Pilot Symbol	Skill Score(%)
3	A	100.00%
4	B	85.31%
11	C	76.64%
13	D	53.96%
15	E	38.81%
6	F	37.47%
12	G	33.23%
14	H	31.71%
8	I	22.74%
7	J	15.28%
16	K	12.83%

TABLE III

Relative Skill Score

	Loading Level			
Pilot #	1	2	3	4
3(A)	0.098	3.820	5.657	5.916
4(B)	1.223	1.261	1.621	1.674
11(C)	0.338	0.903	0.660	8.293
13(D)	0.000	0.118	0.000	0.132
15(E)	9.010	4.915	2.631	11.025
6(F)	0.583	1.302	1.944	2.321
12(G)	3.927	3.790	2.134	2.000
14(H)	0.897	3.150	5.328	10.476
8(I)	0.938	0.861	0.925	5.458
7(J)	0.128	0.282	0.766	3.535
16(K)	0.992	0.858	5.364	5.777
Average	1.649%	1.933%	2.457%	5.146%
Average minus #15 and #12	0.578%	1.395%	2.474%	4.842%

Table IV

Percent of Dwell Times ≥ 4 sec

	Period (sec)		
Loading Level	2	3	4
Pilot			
3(A)	9.75	5.69	4.18
4(B)	10.78	5.85	3.15
11(C)	9.75	6.40	6.02
13 (D)	9.31	5.25	2.84
15(E)	9.75	6.40	2.93
6(F)		5.25	34.13
12(G)		7.59	12.80
14(H)	5.25	5.69	6.61
8(I)	9.31	12.80	3.79
7(J)	1.32	7.88	13.65
16(K)	17.07	20.48	7.88

"Data from GSL, all other data from ATT

Table V

Period corresponding to frequency of largest peak in power-spectral density

APPENDIX A

Computer Programs

This Appendix contains a listing of the programs used in controlling the experiments, and collecting and analyzing the data. The first ten files were used during the experiments to control the number task, collect the data, and compute and output preliminary results. These files are loaded via LDSWL and require ISTOIC, COMROM, CASE, FP, and PT460 (control program for Paper Tiger) to be loaded into memory. Once these are loaded, loading LDSWL will load each file necessary for running the experiments. To run experiments:

1) Load ISTOIC, COMROM, CASE, FP, and PT460 (or AX820)
2) Load LDSWL
3) Set values of Subject # and Man/Trial #
4) Before experiments, oculometer output may be checked using I-POLL
5) Type (number task interval (sec))(run length (min))RUNWL

The other five programs listed were used in the data anlysis. Their description and operating procedures are given at the beginning of the file. Note that there is an HHIST1 and an HHIST2. HHIST1 is used for running the experiments. HHIST2 when loaded with the other experiment controlling files was utilized in several of the data anlysis programs. In this version the work H/DT-SORT requires a file name preceeding it in which the sequence of instrument numbers and dwell time counts (1/30 sec) are stored in a byte and a word, respectively.

```
% FILE : ST.SI

RADIX @ HEX

% CHARACTER DISPLAY REGISTERS

0DB00 `DPC CONSTANT
DPC 1 + `DSIZE CONSTANT
DPC 4 + `DCH CONSTANT
DPC 16 + `DXLLD CONSTANT
DPC 42 + `DYLLD CONSTANT
DPC 80 + `DZ CONSTANT

% J GOULD A/D BOARD REGISTERS

0E500 `ADC CONSTANT
ADC 8 URSHIFT `A/DRST CONSTANT
ADC 1 + `A/DCSR CONSTANT
ADC 2 + `A/DLDAT CONSTANT
ADC 3 + `A/DHDAT CONSTANT
ADC 4 + `A/DCHAN CONSTANT
ADC 5 + `A/DXBAL CONSTANT
ADC 6 + `A/DYBAL CONSTANT

        % ADDRESSING CONSTANTS FOR A/D2

0E100 `A-D CONSTANT
A-D 1 + `CSR/AD CONSTANT
A-D 2 + `CHGN/AD CONSTANT
A-D 8 + `DLO/AD CONSTANT
A-D 9 + `DHI/AD CONSTANT
A-D 8 URSHIFT `RST/AD CONSTANT

0 `ADCLFREQ VARIABLE

`EN-ADINT : ADCLFREQ B@
        40 OR
        CSR/AD B! ;

`DIS-ADINT : 0 CSR/AD B! ;

DECIMAL

1300 `IBUFLEN CONSTANT
IBUFLEN `INSTBUF ARRAY          % DEFINE STORAGE ARRAY FOR INSTRUMENT
                                %       FIXATION AND DWELL TIMES
INSTBUF IBUFLEN 2* + MINUS `-IBUFEND CONSTANT
                                % DEFINE END OF INSTBUF
0 `IOABORT VARIABLE             % IO ERROR FLAG
                                % 1 = RBUF FULL
                                % 2 = INSTBUF FULL
0 `IBUFPTR VARIABLE             % PTR INTO INST BUF
0 `RUNCOUNT VARIABLE            % COUNT OF 1/30 SEC IN RUN
0 `RUNMAX VARIABLE              % MAX TIME OF RUN
0 `ICOUNT VARIABLE              % COUNT OF 1/30 SEC WITHIN FIXATION
```

```
0 `#FIX VARIABLE                % TOTAL # OF FIXATIONS IN RUN
0 `#HITS VARIABLE               % TOTAL # OF FIXATIONS BEFORE REMOVAL
                                %       OF OUT OF TRACKS
8 `#INSTR VARIABLE              % TOTAL # OF INSTRUMENTS USED
0 `NINSTR# VARIABLE             % NEWEST INSTR# FROM OCULOMETER
0 `IFLAG VARIABLE
0 `TINSTR# VARIABLE             % TEMP INSTR# FOR FIXATION ESTIMATE
0 `TCOUNT VARIABLE              % TEMP COUNT FOR FIX ESTIMATE
0 `INSTR# VARIABLE              % CURRENT INSTR #

0 `DFLAG VARIABLE               % FLAG INDICATING CURRENT MODE OF SIDE TASK
                                %       0 = DRAW `*`; 1 = RAD PER BETWEEN `*`
32741 `SEED VARIABLE
0 `DCOUNT VARIABLE              % COUNT OF DELAY FOR SIDE TASK
0 `RDELAY VARIABLE              % REACTION TIME TO CURRENT `*`
0 `RSIGN VARIABLE               % STORES CURRENT SUBJECT RESPONSE, IF ANY
                                % 80(H) = UP
                                % 40(H) = DOWN
0 `CUMRESTIM VARIABLE           % STORAGE OF CUMM RES TIME TO SIDE TASK FOR RUN
0 `#STIM VARIABLE               % TOTAL # OF STIM PRESENTED BY SIDE TASK
0 `MISS# VARIABLE               % TOTAL # OF MISSED RESPONSES TO SIDE TASK
0 `YPOS VARIABLE                % POSITION OF `*` IN SIDE TASK DISPLAY
0 `240HZ/COUNT VARIABLE         % USED TO SUBDIVIDE 240 HZ CLOCK

300 `PARBUFLEN CONSTANT         % LENGTH, IN WORDS, OF PARAMETER BUFFERS
PARBUFLEN `PUPBUF ARRAY         % DEFINE PUPIL DIAMETER BUFFER
PUPBUF PARBUFLEN 2* + MINUS `-PBEND CONSTANT
                                % END OF PUPIL BUFFER
0 `PUPBIPTR VARIABLE            % POINTER INTO PUPIL DIAM BUFFER
PARBUFLEN `RESPBUF ARRAY        % DEFINE RESP BUFFER
RESPBUF PARBUFLEN 2* + MINUS `-RESBEND CONSTANT
                                % END OF RESP BUFFER
0 `RESPBIPTR VARIABLE   % POINTER INTO RESPIRATION BUFFER
400 `RBUFLEN CONSTANT           % DEFINE LENGTH OF RBUF
RBUFLEN `RBUF ARRAY             % DEFINE RR INT BUFFER
RBUF RBUFLEN 2* + MINUS `-RBUFEND CONSTANT
                                % DEFINE END OF RBUF
0 `RBUFIPTR VARIABLE            % POINTER INTO RR INT BUFFER

        % DRAW AN `*` AT COORDINATES (0,YPOS)

. ASSEMBLER<
        0 H LXI,
        DXLLD SHLD,             % ZERO X POS
        YPOS LHLD,
        DYLLD SHLD,             % YPOS = CUR SPECIFIED Y POS
        42 A MVI,               % SPECIFY AN `*` TO BE DRAWN
        DCH STA,
        .
          DPC LDA,              % WAIT IF CHAR BUSY
          2 ANI,
        JNZ,
        RET, >
`(DRAW*) CONSTANT
```

```
    % CHECK FOR RESPONSE FROM SIDE TASK SWITCH

. ASSEMBLER<
    A/DCHAN LDA,
    192 ANI,
    IFZ,
     RSIGN STA,        % SAVE RESPONSE IN RSIGN
     2 A MVI,
     A/DCSR STA,       % CLEAR RESPONSE FLAG
     DCOUNT LDA,
     A E MOV,
     30 A MVI,
     E SUB,
     RDELAY STA,       % RDELAY = 30 - DCOUNT
     0 A MVI,
     A/DCSR STA,       % CLEAR GOULD A/D CSR
    THEN,
    RET, >
`(MRKTIM) CONSTANT

. ASSEMBLER<
      A ORA,           % YES; IS OLD INSTR# NOT = -1 ?
      IFM,
       #FIX LHLD,      % INCR TOTAL # OF FIXATIONS
       H INX,
       #FIX SHLD,
       ICOUNT LHLD,    % YES, UPDATE INSTR TRANS TABLE
       H DAD,
       H DAD,
       H DAD,
       H DAD,
       L ORA,          % `OR` IN OLD INSTR#
       A L MOV,
       XCHG,           % D,E = ICOUNT(BITS4-15); INSTR#(BITS0-3)
       IBUFPTR LHLD,   % GET NEXT ADDRESS FOR STORAGE
       E M MOV,
       H INX,
       D M MOV,
       H INX,
       IBUFPTR SHLD,   % UPDATE PTR FOR NEXT PASS
       -IBUFEND B LXI, % TEST FOR INSTBUF FULL
       B DAD,
       IFNC,
        2 A MVI,       % YES, FLAG INSTBUF FULL
        IOABORT STA,
       THEN,
      THEN,
      RET, >
`(I-STORE) CONSTANT

    % INSTRUMENT SORTING AND DATA PACKING ROUTINE

. ASSEMBLER<
    DLO/AD LDA,        % GET LOW BYTE OF DATA
    CMA,
```

```
A E MOV,
DHI/AD LDA,             % GET HIGH BYTE OF INSTR DATA
CMA,
A D MOV,
20 H LXI,               % ADD 20(10) TO SAMPLE
D DAD,
XCHG,
0 H LXI,
128 B LXI,              % DIVIDE INSTR VOLT BY 128 = 2048*12/16/12 + 1
DIV CALL,               % AFTER DIV D = 0, E = INSTR #
E A MOV,                % LIMIT INSTR# TO 12
12 SUI,
IFM,
  12 E MVI,
THEN,
E A MOV,
NINSTR# STA,
TINSTR# LDA,
E CMP,                  % [E] = NEW INSTR = INSTR# ?
IFNZ,
  TCOUNT LDA,
  A ORA,
  IFNZ,
    ICOUNT LHLD,
    H INX,
    ICOUNT SHLD,
  ELSE,
    A INR,              % INCR TCOUNT
    TCOUNT STA,
    3 CPI,              % TCOUNT > 3 ?
    IFNZ,
      INSTR# LDA,
      (I-STORE) CALL,
      TINSTR# LDA,
      INSTR# STA,
      TCOUNT LHLD,
      ICOUNT SHLD,
      0 A MVI,
      TCOUNT STA,
      IFLAG STA,
    THEN,
  THEN,
ELSE,
  TCOUNT LDA,
  A ORA,
  IFNZ,
    A INR,
    TCOUNT STA,
    NINSTR# LDA,
    TINSTR# STA,
  ELSE,
    INSTR# LDA,
    A E MOV,
    NINSTR# LDA,
    E CMP,
```

```
            IFNZ,
              TINSTR# STA,
              ICOUNT LHLD,
              XCHG,
              TCOUNT LHLD,
              D DAD,
              H INX,                         % ICOUNT = ICOUNT + TCOUNT + 1
              ICOUNT SHLD,
              0 A MVI,
              TCOUNT STA,
            ELSE,
              TCOUNT LDA,
              A INR,
              TCOUNT STA,
              3 CPI,
              IFNZ,
                IFLAG LDA,
                A ORA,
                IFNZ,
                  1 A MVI,
                  IFLAG STA,
                  INSTR# LDA,
                  (I-STORE) CALL,
                  0 H LXI,
                  ICOUNT SHLD,
                THEN,
                ICOUNT LHLD,
                XCHG,
                TCOUNT LHLD,
                H DCX,
                D DAD,
                ICOUNT SHLD,
                12 A MVI,
                INSTR# STA,
                1 A MVI,
                TCOUNT STA,
              THEN,
              NINSTR# LDA,
              TINSTR# STA,
            THEN,
          THEN,
        THEN,
        RET, >
`(INSTRUMENT-SORT) CONSTANT

        % RANDOM DELAY UPDATE

. ASSEMBLER<
        SEED LHLD,
        16345 D LXI,
        MUL CALL,
        XCHG,
        SEED SHLD,
        0 D LXI,
        XCHG,
```

```
            MISS# SHLD,
            #STIM LHLD,         % INCR COUNT OF STIM
            H INX,
            #STIM SHLD,
            (RANDOM-DELAY) CALL, % UPDATE RANDOM DELAY
            DCOUNT STA,
            1 A MVI,            % SPECIFY WAIT BETWEEN `*`
            DFLAG STA,
          THEN,
        THEN,
        RET, >
`(RESPONSE-UPDATE) CONSTANT

        % SIDE TASK/INSTRUMENT SORT SUBROUTINE

. ASSEMBLER<
          DFLAG LDA,            % CHECK FOR MODE OF SIDE TASK
          A ORA,
          IFZ,
            DCOUNT LDA,         % BETWEEN `*`
            A DCR,
            IFNZ,
              (HI/LO-RANDOM) CALL,      % UPDATE YPOS FOR NEXT `*`
              2 A MVI,          % CLEAR ANY SUBJECT RESPONSE
              A/DCSR STA,
              0 A MVI,
              A/DCSR STA,
              DFLAG STA,        % SPECIFY `*` DRAW
              30 A MVI,         %       FOR 1 SEC
            THEN,
            DCOUNT STA,
          ELSE,
            (DRAW*) CALL,       % DRAW AN *
            (MRKTIM) CALL,      % CHECK FOR SUBJECT RESPONSE
            (RESPONSE-UPDATE) CALL,     % UPDATE RESPONSE IF NECESSARY
          THEN,
          (INSTRUMENT-SORT) CALL,
          RUNCOUNT LHLD,        % INCR TOTAL COUNT OF 1/30 SEC IN RUN
          H INX,
          RUNCOUNT SHLD,
          RET, >
`(ST-IS) CONSTANT

RADIX !
;F
```

```
RADIX @ HEX

% FILE : SPEAK
% Stoic Software to drive the Speak and Spell/Speak2Me2 Package

% 11/25/80


0 `DATABYTE VARIABLE
0 `TEMPBYTE VARIABLE
0 `WORDCODE VARIABLE

0E901 `SP/SP/STATUS CONSTANT
0E902 `SSOUT CONSTANT

`SS : BEGIN
        SP/SP/STATUS B@
        80 AND
        EQZ
      END ;

`STROBE CODE< DATABYTE LDA,         % TOP = DATABYTE
        CMA,
        SSOUT STA,
        10 XRI,                     % COMPLEMENT STROBE BIT (BIT 4)
        DATABYTE STA,
        SSOUT STA,                  % TO SPEAK AND SPELL
        10 XRI,                     % COMPLEMENT STROBE BIT
        SSOUT STA,
        DATABYTE STA,
        NEXT JMP, >


`RESET : 5 0
        DO
          0 DATABYTE B!
          STROBE
        LOOP
        8 DATABYTE B!
        STROBE ;

`OUTCHR : 02 DATABYTE B!
        STROBE
        TEMPBYTE B@
        DATABYTE B!
        STROBE ;


        % INTERCHANGE HIGH AND LOW BYTES OF WORD ADDRESS
`WCS CODE< WORDCODE LHLD,
        L E MOV,
        H L MOV,
        E H MOV,
        WORDCODE SHLD,
```

```
        NEXT JMP, >

`WCS1 : WORDCODE @              % GET NEXT 4 BIT NIBBLE FROM WORDCODE
          0F AND
          TEMPBYTE B!           %       AND STORE IN TEMPBYTE
          WORDCODE @            % SHIFT WORDCODE RIGHT 4 PLACES
          4 URSHIFT
          WORDCODE ! ;
`OUTADD : WCS                   % INTERCHANGE HIGH AND LOW BYTES OF ADDRESS
        5 0
        DO
          WCS1
          OUTCHR                % OUTPUT THIS CODE
        LOOP
        WORDCODE 0<-
        08 DATABYTE B!          % FINISH UP
        STROBE
        0A DATABYTE B!
        STROBE
        0E DATABYTE B!
        STROBE
        0 DATABYTE B!
        STROBE
        SS ;                    % WAIT IF SPEAK AND SPELL BUSY

DECIMAL

`TALK : WORDCODE !              % POP CODE FOR NEXT WORD FROM TOP
        SS                      % WAIT IF SPEAK AND SPELL BUSY
        0 DATABYTE B!
        STROBE
        OUTADD ;

`T : TALK ;

RADIX !
;F
```

```
% Speak a number on top of stack.
% N S=
%
% JMT 11/26/80

RADIX @ DECIMAL

-29429 `ZERO CONSTANT
-1781 `ONE CONSTANT
17420 `TWO CONSTANT
-30196 `THREE CONSTANT
-6644 `FOUR CONSTANT
13069 `FIVE CONSTANT
-27891 `SIX CONSTANT
-9715 `SEVEN CONSTANT
6670 `EIGHT CONSTANT
18958 `NINE CONSTANT
-20210 `TEN CONSTANT


-19703 `YOU CONSTANT
-248 `ARE CONSTANT
-1484 `WEIRD CONSTANT

28970 `FIRE CONSTANT

22 `NUMS ARRAY


`SINIT :
        ZERO NUMS !      ONE NUMS 2 + !  TWO NUMS 4 + ! THREE NUMS 6 + !
        FOUR NUMS 8 + ! FIVE NUMS 10 + ! SIX NUMS 12 + ! SEVEN NUMS 14 + !
        EIGHT NUMS 16 + ! NINE NUMS 18 + ! 0 NUMS 20 + ! ;

`S= : 10 SWAP
        BEGIN
          DUP
          RADIX @
          MOD
          SWAP
          10 /
          DUP
          EQZ
        END
        DROP
        BEGIN
          DUP
          2*
          NUMS +
          @
          DUP
          NEZ
          IF
            TALK
          ELSE
```

```
            DROP
          THEN
         10 EQ
        END ;

`ROCKET : TEN T NINE T EIGHT T SEVEN T SIX T FIVE T FOUR T THREE T
        TWO T ONE T FIRE T ;


`YR : YOU T ARE T WEIRD T ;


RADIX !
;F
```

```
% RANDOM TRIPLET GENERATOR
%   by W. A. Morrison

%   November 1, 1979
% Modified by A.T. Stephens on January 24, 1981

RADIX @ DECIMAL

400 `RSULT ARRAY                        % NUMBER TASK RESPONSE STORAGE ARRAY

0 `RAPNTR VARIABLE

% 32741 `SEED VARIABLE (ALREADY DEFINED)
3 `TRIPLET ARRAY
0 `INDEX VARIABLE

% DELAY LOOP

% `0.1SEC CODE<
%       6687 H LXI,                     % SET LOOP COUNT
%       .
%         H DCX,                        % DECREMENT COUNT
%         H A MOV,
%         L ORA,                        % REACHED ZERO?
%       JNZ,                            %  NO, LOOP -
%       NEXT JMP, >                     %  YES, EXIT -
%
% `DELAY : ( 0.1SEC ) ;

`RAND/3 CODE< 4 A MVI,                  % INITIALIZE TRIPLET ARRAY
            INDEX STA,                  %  POINTER/LOOP COUNTER
            .
              SEED LHLD,                % `SEED` -> H,L
              16345 D LXI,              % PRIME# -> D,E
              MUL CALL,                 % H,L * D,E -> HLDE
              XCHG,                     % LOW ORDER PART -> H,L
              SEED SHLD,                %  AND -> `SEED`
              0 D LXI,                  % SET UP FOR
              XCHG,                     %  DIVIDE
              6554 B LXI,               %   TO SINGLE
              DIV CALL,                 %    DIGIT
              E A MOV,                  % DIGIT -> A
              INDEX LHLD,               % TRIPLET POINTER
              XCHG,                     %  -> D,E
              TRIPLET H LXI,            % SET POINTER
              D DAD,                    %  TO NEXT ENTRY
              A M MOV,                  %   AND INSERT DIGIT
              INDEX H LXI,              % DECREMENT
              M DCR, M DCR,             %  INDEX AND
            JP,                         %   LOOP IF NON-NEGATIVE
            NEXT JMP, >

0 `SIGN VARIABLE
0 `MFLAG VARIABLE
```

```
% TRIPLET FORMATTER
%   by W. A. Morrison

%   November 1, 1979

`TRPLET : BEGIN                                  % GENERATE A TRIPLET:
            RAND/3                               % ARE ALL
            TRIPLET @ TRIPLET 2 + @ NE           %  THREE DIGITS
            TRIPLET 2 + @ TRIPLET 4 + @ NE       %   UNIQUE?
            TRIPLET @ TRIPLET 4 + @ NE
            AND
            AND
            IF                                   % YES,
              TRIPLET 2 + @ TRIPLET 4 + @ LT     %  SPECIFY TRIPLET
              TRIPLET 4 + @ TRIPLET @ LT         %   AS `PLUS`,
              AND                                %    OTHERWISE `MINUS`
              TRIPLET @ TRIPLET 2 + @ LT
              TRIPLET 2 + @ TRIPLET 4 + @ LT
              AND
              OR
              IF                                 % `PLUS`:
                -1 -1                            %   RETAIN, AND CONTINUE
              ELSE                               % `MINUS`:
                MFLAG B@                         %   WAS LAST ONE ALSO `MINUS`?
                IF                               %   YES,
                  MFLAG 0<-                      %    RESET MFLAG
                  0 0                            %     AND LOOP TO TRY AGAIN
                ELSE                             %   NO,
                  1 MFLAG B!                     %    BUT FLAG THIS ONE
                  0 -1                           %     AND RETAIN
                THEN
              THEN
              IF
                IF
                  64 SIGN !                      % `+` = 100(8)
                ELSE
                  128 SIGN !                     % `-` = 200(8)
                THEN
                TRIPLET @ 100 *                  % BUILD
                TRIPLET 2 + @ 10 *               %  TEXT
                TRIPLET 4 + @ + +                %   STRING
                <# # # # DROP
                -1
              THEN
            ELSE                                 % NO,
              0                                  %  LOOP AND TRY AGAIN
            THEN
          END ;
```

```
% RADIX @ HEX
                                                        % `(A/DCONV)`
% A/D BOARD REGISTER DEFINITIONS

% 0E600 `ADC CONSTANT           % SLOT ASSIGNMENT: E8
% ADC 8 URSHIFT `A/DRST CONSTANT        % RESTART WORD
% ADC 1 + `A/DCSR CONSTANT      % CONTROL STATUS REGISTER
% ADC 2 + `A/DLDAT CONSTANT     % LOW BYTE OF DATA
% ADC 3 + `A/DHDAT CONSTANT     % HIGH BYTE OF DATA
% ADC 4 + `A/DCHAN CONSTANT     % CHANNEL SELECT FOR INPUT
% ADC 5 + `A/DXBAL CONSTANT     % X-CHANNEL BALANCE (OFFSET)
% ADC 6 + `A/DYBAL CONSTANT     % Y-CHANNEL BALANCE (OFFSET)


% DECIMAL

% 0 `CHAN VARIABLE
% 0 `A/DVAL VARIABLE


`3N :                           % PUSH THE THREE RANDOM NUMBERS STORED IN
        TRIPLET 4 + @           % TRIPLET ONTO THE TOP OF THE STACK
        TRIPLET 2 + @
        TRIPLET @
        ;


0 `RESTIM VARIABLE
0 `RESSIGN VARIABLE
0 `RESDELAY VARIABLE

% RESPONSE ACQUISITION WORD
%       by W. A. Morrison

%       November 19,1979
% Modified by A.T. Stephens on January 24, 1981

. ASSEMBLER<
        A/DCHAN LDA,                    % HAS THERE
        192 ANI,                        %  BEEN A RESPONSE?
        IFZ,                            % YES,
          RESSIGN STA,                  %  SAVE THE RESPONSE
          2 A MVI,                      %   AND ZERO
          A/DCSR STA,                   %    THE RESPONSE BITS
          RESTIM LHLD,                  %  ALSO SAVE
          RESDELAY SHLD,                %   THE LATENCY
          0 A MVI,
          A/DCSR STA,
        THEN,                           % NO,
        RET, >                          %  RETURN
`(MT) CONSTANT                          % MT = MARK TIME
```

```
0 `COR VARIABLE
0 `WRO VARIABLE
0 `MIS VARIABLE
0 `CWM VARIABLE


`INIT/Z CODE<
        0 H LXI,                          % INITIALIZE
        RESSIGN SHLD,                     %   RESPONSE SIGN,
        RESDELAY SHLD,                    %    RESPONSE LATENCY,
        RESTIM SHLD,                      %     AND RESPONSE TIME CLOCK
        NEXT JMP, >

HEX

`RESPON CODE<
        RAPNTR LHLD,                      % LOAD POINTER LOCATION
        XCHG,
        RESSIGN LDA,
        A ORA,                            % OR TO SET STATUS BITS FOR IFZ
        IFZ,                              % TEST FOR RESPONSE
          SIGN LHLD,                      % TEST
          L A MOV,                        %    FOR
          RESSIGN LHLD,                   %      CORRECT/WRONG
          L CMP,                          %                  RESPONSE
          IFN,
            43 A MVI,                     % IF CORRECT RESPONSE, STORE "C"
            CWM STA,                      % IN CWM
            COR LHLD,                     % INCREMENT
            H INX,                        %       CORRECT
            COR SHLD,                     %               COUNT
            RESDELAY LHLD,
            H A MOV,                      % OR IN RESPONSE TYPE
            20 ORI,                       % 20(HEX) = CORRECT
          ELSE,
            57 A MVI,                     % IF WRONG RESPONSE, STORE "W"
            CWM STA,                      % IN CWM
            WRO LHLD,                     % INCREMENT
            H INX,                        %       WRONG
            WRO SHLD,                     %               COUNT
            RESDELAY LHLD,
            H A MOV,                      % OR IN RESPONSE TYPE
            40 ORI,                       % 40(HEX) = WRONG
          THEN,
          D STAX,                         % STORE HIGH BYTE OF DATA
          D INX,                          % INCREMENT POINTER
          L A MOV,
          D STAX,                         % STORE LOW BYTE OF DATA
          D INX,                          % INCREMENT POINTER
          XCHG,
```

```
 RAPNTR SHLD,                      % STORE POINTER LOCATION
       ELSE,
         4D A MVI,                         % STORE "M" FOR MISS
         CWM STA,                          % IN CWM
         MIS LHLD,                         % INCREMENT
         H INX,                            %       MISS
         MIS SHLD,                         %         COUNT
         60 A MVI,                         % 60(HEX) = MISS
         D STAX,                           % STORE HIGH BYTE OF DATA
         D INX,                            % INCREMENT POINTER
         D INX,                            %       TWICE
         XCHG,
         RAPNTR SHLD;                      % STORE POINTER LOCATION
       THEN,
       NEXT JMP, >


% SORT RESPONSE AND DELAY TIME ON NUMBER TASK

`R/D-SORT CODE<
               RAPNTR LHLD,                % STORE
               XCHG,                       %       POINTER
               D LDAX,                     % LOAD HI BYTE OF DATA
               60 ANI,                     % AND TO GET RESPONSE
               CWM STA,                    % STORE RESPONSE
               D LDAX,                     % RELOAD HI BYTE OF DATA
               D INX,                      % INCREMENT POINTER
               1F ANI,                     % AND TO GET DELAY TIME
               A H MOV,
               D LDAX,                     % LOAD LO BYTE OF DATA
               A L MOV,
               RESDELAY SHLD,              % STORE DELAY TIME
               D INX,                      % MOVE POINTER TO HI BYTE OF NEXT WORD
               XCHG,
               RAPNTR SHLD,                % STORE POINTER LOCATION
               NEXT JMP, >


DECIMAL

0 `RCNT VARIABLE

% PRINT OUT NUMBER TASK RESULTS

`PRRSULT :
       RSULT RAPNTR !                      % INITIALIZE POINTER
       400 0
       DO
         R/D-SORT                          % UNPACK RESPONSE DATA
         CWM @ NEZ                         % TEST FOR END OF RESPONSES
```

```
          IF
            CWM @ 96 - NEZ                  % TEST FOR MISS
            IF
              CWM @ 32 - EQZ                % TEST FOR CORRECT
              IF                            % RESPONSE
                `C MSG
              ELSE
                `W MSG
              THEN
              RESDELAY @ 24 / = 1 TAB       % DISPLAY DELAY TIME
              RCNT 1+!                      % INCREMENT RESPONSE COUNTS
              RCNT @ 16 GE                  % PRINT 8 RESPONSES
              IF                            % PER LINE
                CR
                0 RCNT !
              THEN
            ELSE
              `M MSG
              RESDELAY @ 24 / = 1 TAB
              RCNT 1+!
              RCNT @ 16 GE
              IF
                CR
                0 RCNT !
              THEN
            THEN
          THEN
        LOOP
        CR CR
        "#CORRECT=" MSG
        COR @ = CR
        "#WRONG=" MSG
        WRO @ = CR
        "#MISS=" MSG
        MIS @ = CR
        CR
        ;


RADIX !
;F
```

```
% FILE : WLCALC

RADIX @ HEX
0DF02 `SWITCHES CONSTANT

DECIMAL

0 `#DIFSEQ VARIABLE      % NUMBER OF DIFFERENT SEQUENCES OF LENGTH SEQLENGTH
0 `SEQLENGTH VARIABLE
0 `MAXCNT VARIABLE       % USED IN SEARCH FOR MAX FREQ OF OCCUR OF SEQ
0 `MAXSEQ VARIABLE       % TEMP STORAGE FOR CURRENT SEQ WITH GREATEST FREQ
0 `#TRIES VARIABLE       % # OF TRIES BEFORE LOOP EXIT IN H CALC
0 `#SEQ VARIABLE         % # OF POSS-SEQ OF LENGTH SEQLENGTH
0.0 `HT FVARIABLE        % TOTAL ENTROPY FOR SEQUENCE OF LENGTH N
0.0 `RF FVARIABLE        % REDUNDANCY FACTOR

#INSTR @ 75 * 1+ 2/ `DTBUFLEN CONSTANT
DTBUFLEN `DTBUF ARRAY    % ARRAY FOR STORAGE OF DWELL TIME HISTOGRAMS

        % PRINT FLOATING PT NUMBER ON TOP,TOP-1 AS A DECIMAL QUANTITY

 `PDEC : DDUP
        INTEGER =
        `. MSG
        FRAC
        1000.0 F* INTEGER
        DUP 10 LT
        IF
          `0 MSG
        THEN
        DUP 100 LT
        IF
          `0 MSG
        THEN
        = ;

        % MAP FROM LANGLEY INSTR CODE TO 8 INSTR CODE

%       MIT CODE                LANGLEY CODE
. ASSEMBLER<
        0 B,                    % 0     NON SPECIFIC INSTR
        0 B,                    % 1     MAG COMPASS
        0 B,                    % 2     ADF
        1 B,                    % 3     AIRSPEED
        2 B,                    % 4     ATTITUDE INDICATOR
        3 B,                    % 5     ALTITUDE
        4 B,                    % 6     GLIDE SLOPE/LOCALIZER
        0 B,                    % 7     CLOCK
        5 B,                    % 8     TURN * BANK
        6 B,                    % 9     GYRO COMP
        7 B,                    % 10    VSI
        0 B,                    % 11    %POWER INDICATOR
        12 B, >                 % 12    OUT OF TRACK
`ICODE CONSTANT
```

```
        % LANGLEY INSTR #, IMAP

`IMAP : ICODE + B@ INSTR# B! ;

`MI0 : " Nsp " ;
`MI1 : " AS  " ;
`MI2 : " ATT " ;
`MI3 : " ALT " ;
`MI4 : " GSL " ;
`MI5 : " T*B " ;
`MI6 : " DG  " ;
`MI7 : " VSI " ;
`MI8 : " X8  " ;
`MI9 : " X9  " ;
`MI10 : " X10 " ;
`MI11 : " X11 " ;
`MI12 : " OoT " ;

        % RETURNS INSTR TAG FOR THIS INSTR#

`MAP : CASE
          MI0 MI1 MI2 MI3 MI4 MI5 MI6 MI7 MI8 MI9 MI10 MI11 MI12
        THEN
        MSG ;

        % INSERT VERTICAL DOTTED LINE IN PLOTTER OUTPUT

`DOTLINE : 400 0
          DO
            I 10 /MOD UNDER EQZ IF I -1 AXDOT THEN
          LOOP ;

        % DRAW VERTICAL LINE EVERY 5 SECONDS IN INSTRUMENT PLOT

`5SECLINE : RUNCOUNT 1+!
        RUNCOUNT B@
        25 GT
        IF
          DOTLINE                        % DRAW VERTICAL DOTTED LINNE
          RUNCOUNT 0<-
        THEN
        AXLNCNT @                        % DRAW INSTRUMENT REFERENCE LINES
        7 AND
        EQZ
        IF
          #INSTR @ 0
          DO
            I 34 * 10 + -1 AXDOT
          LOOP
        THEN ;
```

```
`IUNPACK : I 2* INSTBUF + @
    DUP
     15 AND             % AND TO GET INSTR#
     IMAP               % MAP FROM LANGLEY CODE TO MIT INSTR CODE
     INSTR# B@ ;        % LEAVE INSTR# ON TOP, PACKED DW AT TOP-1

    % LIST ENTRIES IN INSTRUMENT BUFFER

`ILIST : #HITS @ 0
    DO
     IUNPACK            % UNPACK INSTR# AND DW TIME
     MAP                % MAP INSTR# INTO INSTR CODE
     4 URSHIFT
     =                  % UNPACK AND PRINT DW TIME
     I 7 AND            % ALLOW 8 ENTRIES PER LINE
     IF
      I 8 /MOD 10 * TAB DROP
     ELSE
      CR
     THEN
    LOOP
    CR CR
    SWITCHES B@
    1 AND
    IF                  % PLOT FIXATIONS VS TIME IF SW REG 0 = 1
     #INSTR @ 0         % PRINT INSTR TAGS AS HEADER FOR PLOT
     DO
      I MAP
     LOOP CR
     RUNCOUNT 0<-       % CLEAR RUNCOUNT
     <AXLN              % INIT PLOT
     0
     #HITS @ 0
     DO
      IUNPACK           % UNPACK NEXT ENTRY IN INSTRUMENT BUFFER
      34 * 10 +
      DUP <L            % SAVE A COPY ON LOOP STACK
      2SWAP             % TOP = SCALED INSTR#, TOP-1 = OLD INSTR#
      DDUP
      LT
      IF
       SWAP
      THEN
      DO
       I -1 AXDOT       % DRAW LINE BETWEEN SUCCESSIVE INSTRS
      LOOP
      L>                % POP SCALED INSTR# FROM LOOP STACK
      SWAP
      4 URSHIFT 6 /MOD  % DIVIDE DWELL TIME BY 6
      3 GE              % INCR QUOT IF REMAINDER >= 3
      IF
```

```
        1+
      THEN
     ( DUP -1 AXDOT
       5SECLINE               % DRAW TIME GRID EVERY 5 SECONDS
      AXLN )
    LOOP
    DROP
    AXLN>
  THEN ;
```

```
    % LIST PARAMETER BUFFERS

`PP : SWITCHES B@
    2 AND           % PRINT OTHER PARAMETERS IF SW REG 1 = 1
    IF
     AX-TER             % ENABLE AX820
     CR
     `RESP MSG
     20 TAB
     `RR-INT MSG
     40 TAB
     `P-D MSG
     CR CR
     PARBUFLEN 2* 0
     DO
      RESPBUF I + B@    % GET NEXT RESPIRATION
      8 LSHIFT          %         CORRECT BYTE FOR SIGN
      8 RSHIFT
      =                 %         AND OUTPUT
      20 TAB            % TAB TO COLUMN 20
      RBUF I + B@ =         % GET NEXT RR INTERVAL AND PRINT
      40 TAB
      PUPBUF I + B@         % GET PUPIL DIAMETER AND PRINT
      4 * FLOAT
      80.6 F/
      PDEC
      50 TAB
      `MM MSG
      CR
     LOOP
     T-TER
    THEN ;

    % PLOT DWELL TIME HISTOGRAM OF INSTRUMENT WHOSE CODE # IS ON TOP

`DTPLOT : DUP           % TOP = INSTR#
    MAP                 % PRINT INSTR TAG
    25 TAB
    `COUNTS MSG CR
    14 TAB
    `10 MSG
    29 TAB
    `20 MSG
    45 TAB
    `30 MSG
    CR
    <AXLN               % INIT AX820
    500 0
    DO                  % PRINT ORDINATE AXIS
     I -1 AXDOT
    LOOP
```

```
        AXLN
        75 *                            % MULT INSTR# BY 75
        DTBUF +                         % TOP = DTBUF(INSTR#*75)
        INSTR# !                        %        = INSTR#
        75 0
        DO
          I 1+ 15 /MOD UNDER
          EQZ                           % DRAW 1 SEC LINES IN HIST
          IF
            DOTLINE
          THEN
          2 0                           % PRINT TWO LINES FOR EACH BIN
          DO
            INSTR# @ J + B@ 10 * 0      % DRAW CONTENTS OF CURRENT BIN
            DO
              I -1 AXDOT
            LOOP
            AXLN                        % OUTPUT A LINE
          LOOP
        LOOP
        AXLN> ;

        % PLOT DWELL TIME HISTOGRAMS OF ALL INSTRUMENTS

`DT-HIST : 0
        BEGIN
          DUP                           % INSTR# ON TOP
          DTPLOT
          CR CR
          1+
          DUP
          #INSTR @
          GE
        END ;

        % CALCULATE WORKLOAD INDEX
        %    (0.780*CUMRESTIM/30+0.626*MISS#)
        %       --------------------------   * 100 %
        %       [#STIM * (0.780 + 0.626)]

`WLINDEX-CALC : CUMRESTIM @ FLOAT       % GET CUM RES TIME AND FLOAT
        0.026 F*                        % 0.026 = 0.780/30
        MISS# @ FLOAT
        0.626 F*
        F+
        #STIM @ FLOAT
        F/
        71.12 F*                        % MULT BY 100.0 * ( 1/[0.780 + 0.626] )
        "WLINDEX " MSG
        PDEC
        " %" MSG CR ;
```

```
0.0 `QS FVARIABLE
0.0 `Q FVARIABLE
0.0 `LOGSL FVARIABLE
0.0 `KH FVARIABLE

`Q-CALC : #INSTR @ FLOAT
        DDUP
        1.0 F-                    % ( #INSTR - 1 )
        FLN                       % LN (#INSTR -1)
        SEQLENGTH @
        DUP                       % SAVE A COPY OF SEQ LENGTH
        #DIFSEQ @
        MIN FLOAT                 % TAKE LOG2 OF MIN (SEQLENGTH,#DIFSEQ)
        FLOG2
        LOGSL F!                  % LOGSL = LOG2(SEQLENGTH)
        FLOAT                     % FLOAT SEQ LENGTH
        1.0 F-                    % SEQLENGTH - 1
        F*                        % LN(#INSTR-1)*(SEQLENGTH-1)
        FEXP
        F*                        % Q =#INSTR *(#INSTR-1)^SEQLEN-1
        DDUP
        Q F!
        "Q " MSG                  % PRINT Q
        PDEC CR
        #FIX @ FLOAT              % CALCULATE KH
        Q F@
        F/                        % KH = MIN (#FIX/Q,1.0)
        DDUP
        1.0                       % KH > 1 ?
        FGE
        IF
          2DROP 1.0
        THEN
        DDUP
        KH F!
        "K " MSG KH F@            % PRINT KH
        PDEC CR
        Q F@
        F*                        % KH*Q
        FLOG2                     % LOG2(KH*Q)
        LOGSL F@
        F-
        QS F! ;                   % QS = LOG2(KH*Q)-LOGSL

        % CALCULATE NORMALIZED QUALITY FACTOR, A(H)

`A(H)-CALC : Q-CALC               % CALC Q AND ASSOCIATED PARAMETERS
        HT F@
        LOGSL F@
        F-                        % TOP = HT - LOGSL
        QS F@
```

```
F/                      % W =( HT - LOGSL)/(LOG Q - LOGSL)
FMINUS
1.0
F+                      % TOP = A(H) = 1 - W
DDUP
RF F!                   % STORE REDUNDANCY FACTOR
"RF " MSG
PDEC CR
;
```

```
% FILE : HHIST1

RADIX @ DECIMAL

        % THIS FILE CALCULATES ENTROPY,H AND GENERATES DWELL TIME HISTOGRAMS

`RAD-#INS : #INSTR @ RADIX ! ;   % SET RADIX = #INSTR

RAD-#INS                    % SET RADIX = # OF INSTR

10 `C10 CONSTANT
100 `C100 CONSTANT
1000 `C1000 CONSTANT

DECIMAL

2 `ITBUF ARRAY              % TEMP BUFFER FOR SORTING OF SEQS FOR H CALC
#INSTR @ 2 + 2/ `HBUF1 ARRAY     % TOP OF BUF FOR SEQ LENGTH 1
#INSTR @ 1+ C10 * 2/ `HBUF2 ARRAY        % TOP OF BUF FOR SEQ LENGTH 2
#INSTR @ 1+ C100 * 2/ `HBUF3 ARRAY       % TOP OF BUF FOR SEQ LENGTH 3
#INSTR @ 1+ C1000 * 2/ `HBUF4 ARRAY      % TOP OF BUF FOR SEQ LENGTH 4
0 `VHBUF VARIABLE                % CONTAINS TOP ADDR OF HBUF(X)

`Z-HBUF : HBUF1 #INSTR @ 2 + 2/ 0FILL
HBUF2 #INSTR @ 1+ C10 * 2/ 0FILL
HBUF3 #INSTR @ 1+ C100 * 2/ 0FILL
HBUF4 #INSTR @ 1+ C1000 * 2/ 0FILL ;

        % INCR APPROPRIATE DWELL TIME BIN FOR LAST INST/FIX

. ASSEMBLER<
        ITBUF 3 + LDA,
        A ORA,                   % OMIT IF INSTR# = -1
        RM,
        A E MOV,
        0 D MVI,                 % GET INSTR# AT JUST PREVIOUS FIXATION
        0 H LXI,
        75 A MVI,
        .
          D DAD,
          A DCR,
        JNZ,
        XCHG,                    % D,E = 75 * INSTR# -1
        ICOUNT LHLD,
        H A MOV,
        RAL,
        H A MOV,
        RAR,
```

```
        A H MOV,
        L A MOV,
        RAR,
        A L MOV,                % PUSH ICOUNT/2
          H PUSH,
        -75 B LXI,
        B DAD,                  % ICOUNT/2 > 75 ?
          H POP,                % POP ICOUNT/2
        IFNC,
          74 H LXI,             % YES, LIMIT ICOUNT/2 TO 74
        THEN,
        D DAD,                  % H,L = (75*INSTR#) + ICOUNT/2
        DTBUF D LXI,
        D DAD,                  % H,L = DTBUF(75*INSTR# + ICOUNT/2)
        M INR,                  % INCR [DTBUF(75*INSTR# + ICOUNT/2)]
        RET, >
`(DT-SORT) CONSTANT

`H-SORT CODE<
        ITBUF 3 + LDA,
        A E MOV,
        INSTR# LDA,             % LAST INSTR = CURRENT INSTR ?
        E CMP,
        IFNZ,
          #FIX LHLD,            % YES, DCR # OF FIXATIONS
          H DCX,
          XCHG,
          -1 H LXI,
          D DAD,
          IFNC,                 % # OF FIX CAN BE NO LESS THAN 1
            XCHG,
          ELSE,
            1 H LXI,
          THEN,
          #FIX SHLD,
          ICOUNT LHLD,          % ICOUNT = ICOUNT + TCOUNT
          XCHG,
          TCOUNT LHLD,
          D DAD,
          ICOUNT SHLD,
        ELSE,
          (DT-SORT) CALL,       % SORT FOR DWELL TIME HIST
          TCOUNT LHLD,          % LET TCOUNT = ICOUNT
          ICOUNT SHLD,
                        % ITBUF(3) -> ITBUF(2) -> ITBUF(1) -> ITBUF(0) -> LOST
          3 H MVI,
          ITBUF D LXI,
          ITBUF 1 + B LXI,
          .
            B LDAX,
            D STAX,
```

```
  B INX,
  D INX,
  H DCR,
JNZ,
INSTR# LDA,            % NEWEST INSTR# IN ITBUF(3)
D STAX,
      % DETERMINE NEXT SEQ OF LENGTH 1 AND INCR CORRESPONDING COUNTER
ITBUF 3 + H LXI,
M E MOV,
0 D MVI,               % D,E = [ITBUF(3)]
  D PUSH,              % PUSH [ITBUF(3)]
HBUF1 H LXI,
D DAD,
M INR,                 % INCR [HBUF1([ITBUF(3)])]
ITBUF 2 + H LXI,
M E MOV,
0 D MVI,               % D,E = [ITBUF(2)]
E A MOV,
A ORA,
IFM,
  0 H LXI,

  C10 A MVI,           % MULT BY C10
  .
    D DAD,
    A DCR,
  JNZ,                 % H,L = C10 * [ITBUF(2)]
    D POP,             % POP [ITBUF(3)]
  D DAD,               % H,L = Q2 = C10*[ITBUF(2)] + [ITBUF(3)]
    H PUSH,            % PUSH THIS RESULT
  HBUF2 D LXI,
  D DAD,               % H,L = HBUF2(Q2)
  M INR,               % INCR [HBUF2(Q2)]
  ITBUF 1 + H LXI,
  M E MOV,
  0 D MVI,             % D,E = [ITBUF(1)]
  E A MOV,             % TEST FOR -1 ENTRY
  A ORA,
  IFM,
    0 H LXI,
    C100 A MVI,        % MULT BY C100
    .
      D DAD,
      A DCR,
    JNZ,
      D POP,           % POP Q2
    D DAD,             % H,L = Q2 + C100*[ITBUF(1)] = Q3
      H PUSH,          % PUSH Q3
    HBUF3 D LXI,
    D DAD,             % H,L = HBUF3(Q3)
    M INR,             % INCR [HBUF3(Q3)]
```

```
          ITBUF H LXI,
          M E MOV,
          0 D MVI,
          E A MOV,              % TEST FOR -1 ENTRY
          A ORA,
          IFM,
            0 H LXI,
            C1000 B LXI,        % MULT BY C1000
            .
              D DAD,
              B DCX,
              B A MOV,
              C ORA,
            JNZ,
              D POP,            % POP Q3
            D DAD,              % H,L = Q3 + C1000*[ITBUF(0)]
            HBUF4 D LXI,
            D DAD,              % H,L = HBUF4(Q4)
            M INR,              % INCR [HBUF4(Q4)]
          ELSE,
            H POP,
          THEN,
        ELSE,
          H POP,
        THEN,
      ELSE,
        H POP,
      THEN,
    THEN,
    NEXT JMP, >

`H/DT-SORT : ICOUNT 0<-         % ZERO ICOUNT
    Z-HBUF                      % ZERO HBUF
    ITBUF 2 -1 FILL             % FILL ITBUF WITH -1
    DTBUF DTBUFLEN 0FILL        % CLEAR DTBUF
    #HITS @ #FIX !              % SET #FIX = #HITS
    #HITS @ 0
    DO
      IUNPACK                   % UNPACK DWELL TIME AND INSTR#
      SWAP                      % TOP = PACKED DWELL, TOP-1 = INSTR#
      4 URSHIFT
      TCOUNT !                  % SHIFT RIGHT 4 PLACES TO GET TCOUNT
      #INSTR @                  % ONLY SORT IF INSTR# <= #INSTR
      LE
      IF
        H-SORT                  % UPDATE H SEQUENCE COUNTS ( AND DT HIST )
      ELSE
        TCOUNT @                % ADD DWELL TIME TO LAST INSTR
        ICOUNT @ +
        ICOUNT !
        #FIX 1-!                % IF ILLEGAL INSTR#, DCR # OF FIXATIONS
```

```
          THEN
        LOOP ;

RAD-#INS        % SET RADIX = # OF INSTR

        % MAP FROM SEQUENCE # IN MAXSEQ TO STRING OF INSTR TAGS

`SEQ-MAP : MAXSEQ @
        SEQLENGTH @
        4 GE
        IF
          1000 /MOD
          SWAP
          MAP                   % DECODE AND PRINT INSTR TAG
        THEN
        SEQLENGTH @
        3 GE
        IF
          100 /MOD
          SWAP
          MAP
        THEN
        SEQLENGTH @
        2 GE
        IF
          10 /MOD
          SWAP
          MAP
        THEN
        MAP ;

DECIMAL

`P-EN/AUX : MAXCNT @            % GET MAXCNT ON TOP
        DUP
        =                       % LIST FREQ OF OCCURENCE OF THIS SEQ
        36 TAB
        FLOAT                   % FLOAT MAXCNT
        #SEQ @
        FLOAT                   % FLOAT #SEQ
        F/                      % P(i) = MAXCNT(i)/#SEQ
        DDUP
        PDEC                    % OUTPUT P(i)
        47 TAB
        DDUP
        FLOG2                   % CALC LOG2[P(i)]
        F*                      % TOP = P(i)*LCG2[P(i)]
        FMINUS                  % NEGATE RESULT SO THAT H IS ALWAYS +
        DDUP
        PDEC                    % OUTPUT H(i)
        CR ;
```

```
`SUP-MSG : RADIX @ <L
        DECIMAL
        =                               % OUTPUT # OF REMAINING SEQUENCES
        " Other Seq(s) w freq" MSG
        L> RADIX ! ;

HEX

`SUPPRESS-H1 : 0DF02 B@                 % CHECK SW REQ BIT 7
        80 AND
        IF                              % IF SET, SUPPRESS DETAILED PRINTING OF RESULTS
          CR                            %       FOR SEQ OF FREQ 1
          #SEQ @                        % #REMAINING SEQ = #SEQ - #TRIES
          #TRIES @ -
          DUP
          DUP #DIFSEQ @ +               % ADD TO NUMBER OF DIF SEQ
          #DIFSEQ !
          SUP-MSG                       % PRINT SUPPRESSION MSG
          FLOAT                         % FLOAT #REMAINING SEQS
          1B TAB
          P-EN/AUX                      % PRINT ENTROPY FOR 1 SEQ OF FREQ 1
          F*                            % MULT SELF-H FOR FREQ 1 * #REMAINING SEQ
          #SEQ @
          #TRIES !                      % SET #TRIES = #SEQ TO ALLOW EXIT FROM
        ELSE
          SEQ-MAP                       % NO SUPPRESION DESIRED, PRINT SEQ OF FREQ 1
          1B TAB                        %       AS USUAL
          P-EN/AUX
        THEN ;

DECIMAL

`P-ENTROPY : RADIX @ <L
        DECIMAL
        MAXCNT @                        % OPTIONALLY DEFEAT PRINT OF SEQ FREQ OF 1
        1 EQ
        IF
          SUPPRESS-H1
        ELSE
          SEQ-MAP
          27 TAB
          P-EN/AUX                      % PRINT SEQ, ENT, ETC.
        THEN
        HT F@
        F+
        HT F!                           % ADD SELF ENTROPY FOR THIS SEQ TO HT
        L> RADIX ! ;

        % SEARCH VHBUF FOR LARGEST ELEMENT
        % LEAVE COUNT OF THIS ELEMENT IN MAXCNT, SEQ CODE IN MAXSEQ
```

```
`H-C-AUX CODE< 0 B MVI, % ZERO B FOR USE BELOW
        0 H LXI,
        MAXCNT SHLD,            % ZERO MAXCNT
        MAXSEQ SHLD,            % ZERO MAXSEQ
        XCHG,                   % D,E = 0
          H POP,                % POP UPPER LIMIT ON SEARCH
        -HL CALL,
        RUNCOUNT SHLD,
        . <L
          VHBUF LHLD,
          D DAD,                % H,L = VHBUF(SEQ)
          M C MOV,              % C = [VHBUF(SEQ)]
          MAXCNT LDA,
          C CMP,                % [VHBUF(SEQ)] > MAXCNT ?
          IFNC,
            XCHG,               % H,L = SEQ CODE; D,E = VHBUF(SEQ)
            MAXSEQ SHLD,
            XCHG,               % D,E = SEQCODE
            C A MOV,            % STORE COUNT FOR THIS SEQ IN MAXCNT
            MAXCNT STA,
          THEN,
            D INX,              % DCR POINTER INTO VHBUF
            RUNCOUNT LHLD,
            D DAD,
            NEXT JC,
          L> JMP, >

`H-CALC : H-C-AUX               % SEARCH CUR VHBUF FOR SEQ WITH LARGEST COUNT
        MAXCNT @
        IF
          P-ENTROPY             % PRINT, SEQ, P(i),H, ETC.
          MAXCNT @              % INCR # OF SEQ FOUND BY MAXCNT
          #TRIES @
          +
          #TRIES !
          #DIFSEQ 1+!           % INCR # DIF SEQ
          0 VHBUF @ MAXSEQ @ + B!        % CLEAR ENTRY IN HBUF(X)
        THEN
        #TRIES @
        #SEQ @                  % EXIT IF ALL POSSIBLE SEQUENCES FOUND
        GE
        IF
          EXIT
        THEN ;
```

```
`SEQMSG : "LENGTH = " MSG
        SEQLENGTH ? CR
        "#SEQ = " MSG
        #SEQ ? CR CR
        "SEQ" MSG
        26 TAB
        "FREQ" MSG
        36 TAB
        "PROB" MSG
        47 TAB
        "SELF H" MSG
        58 TAB
        "H RATE" MSG CR CR ;

`ENT-INIT : RADIX @ <L
        DECIMAL
        DUP
        SEQLENGTH !                     % SET SEQLENGTH = TOP
        1-                              % SELECT BUFFER FOR THIS SEQ LENGTH
        CASE
          HBUF1 HBUF2 HBUF3 HBUF4
        THEN
        VHBUF !
        #DIFSEQ 0<-                     % ZERO COUNT OF # OF DIF SEQS
        #FIX @
        SEQLENGTH @ 1-
        -
        #SEQ !                  % #SEQ = #FIX - (SEQLENGTH-1)
        SEQMSG
        L> RADIX !
        0.0 HT F!                       % CLEAR TOTAL ENTROPY
        #TRIES 0<- ;

RAD-#INS                % SET RADIX = # OF INSTR

`ENTROPY : 1 ENT-INIT                   % INIT ENT CALC FOR SEQ LENGTH 1
        10 ( 10 H-CALC )
        EN-AUX                          % OUTPUT TOTAL ENTROPY FOR SEQ LENG 1
        2 ENT-INIT                      % INIT ENT CALC FOR SEQ LENGTH 2
        100 ( 100 H-CALC )
        EN-AUX                          % OUTPUT TOTAL ENTROPY FOR SEQ LENG 2
        3 ENT-INIT                      % INIT ENT CALC FOR SEQ LENGTH 3
        1000 ( 1000 H-CALC )
        EN-AUX                          % OUTPUT TOTAL ENTROPY FOR SEQ LENG 3
        4 ENT-INIT                      % INIT ENT CALC FOR SEQ LENGTH 4
        10000 ( 10000 H-CALC )
        EN-AUX ;                        % OUTPUT TOTAL ENTROPY FOR SEQ LENG 4

RADIX !
;F
```

```
RADIX @ DECIMAL

       % FILE TO COLLECT SINGLE CHANNEL OF EKG DATA
       % AND TO DETECT AND ANNOTATE R WAVES

600 `BUFLEN CONSTANT            % DEFINE LENGTH OF DATA BUFFER
BUFLEN MINUS `-BUFLEN CONSTANT  % DEFINE -BUFLEN
-BUFLEN 2* `-2BUFLEN CONSTANT
BUFLEN `BUFFER ARRAY            % DEFINE BUFFER `BUFLEN` WORDS LONG
BUFFER BUFLEN 2* + MINUS `MBUFEND CONSTANT     % DEF - END OF BUFFER

0 `IBUFIPTR VARIABLE            % INPUT BUFFER INPUT PTR
0 `IBUFOPTR VARIABLE            % IN BUF OUT PTR
0 `BUFICOUNT VARIABLE           % COUNT OF UNPROCESSED PTS IN INPUT BUF

0 `NEWVAL VARIABLE
100 `ZTH VARIABLE               % THRESHOLD FOR ZERO VELOCITY
-500 `-RTHRESH VARIABLE % THRESHOLD FOR R WAVE DETECTION
0 `RRCOUNT VARIABLE             % COUNTER FOR RR INTERVAL

7 `TBUF ARRAY                   % DEFINE TEMP BUFFER FOR FIR FILTER

OCTAL

40 `PBUFLEN CONSTANT            % DEFINE LENGTH,IN WORDS OF PARAMETER BUFFERS
PBUFLEN 2* 1- `MODPBUFLEN CONSTANT      % DEFINE MOD OF PBUFLEN
PBUFLEN `PBUF ARRAY
PBUFLEN `VBUF ARRAY
0 `I1 VARIABLE                  % POINTER TO UPDATED, BUT UNPROCESSED DATA
0 `I2 VARIABLE                  % POINTER TO NEXT DATA POINT TO BE PROCESSED
0 `TS VARIABLE

HEX

0DF03 `LAMPS CONSTANT   % DEFINE LAMP REGISTER ADDRESSING CONSTANT

. ASSEMBLER<
       .
         IOABORT LDA,           % EXIT IF IOABORT IS SET ELSEWHERE
         A ORA,
         RNZ,
         BUFICOUNT LHLD,        % WAIT IF NO POINTS IN INPUT BUFFER
         H A MOV,
         L ORA,
       JZ,
       H DCX,
       BUFICOUNT SHLD,
       IBUFOPTR LHLD,           % H,L = [IBUFOPTR]
       M E MOV,
       H INX,
       M D MOV,
       H INX,
       XCHG,
       NEWVAL SHLD,
       MBUFEND H LXI,
```

```
         D DAD,
         IFNC,
           BUFFER D LXI,
         THEN,
         XCHG,
         IBUFOPTR SHLD,
         RET, >
 `(GETNEXTVAL) CONSTANT

         % COMPLETION ROUTINE FOR EKG SAMPLING

 . ASSEMBLER<
         BUFICOUNT LHLD,
         XCHG,
         -BUFLEN H LXI,
         D DAD,
         IFNC,
           BUFLEN D LXI,
         ELSE,
           D INX,
         THEN,
         XCHG,
         BUFICOUNT SHLD,

         DLO/AD LDA,
         CMA,
         A E MOV,
         DHI/AD LDA,
         CMA,
         A D MOV,

         IBUFIPTR LHLD,
         E M MOV,
         H INX,
         D M MOV,
         H INX,
         XCHG,
         MBUFEND H LXI,
         D DAD,
         IFNC,
           BUFFER D LXI,
         THEN,
         XCHG,
         IBUFIPTR SHLD,
         RET, >
`(EKG) CONSTANT

DECIMAL

`ENOUGH CODE<
         I2 LHLD,
         I1 LDA,
         L SUB,
         MODPBUFLEN ANI,                      % A = I1-I2, MODPBUFLEN
         20 SUI,                              % {I1-I2,MODPBUFLEN} > 20 ?
```

```
        IFM,
          -1 H LXI,                     % YES, FLAG ENOUGH DATA
        ELSE,
          0 H LXI,                      % NO, FLAG NOT ENOUGH DATA
        THEN,
        PUSH JMP, >

`PARAMETER-UPDATE CODE<
        (GETNEXTVAL) CALL,

        12 H MVI,
        TBUF D LXI,
        TBUF 2 + B LXI,
        .
          B LDAX,
          D STAX,
          B INX,
          D INX,
          H DCR,
        JNZ,
        NEWVAL LHLD,
        XCHG,
        E M MOV,
        H INX,
        D M MOV,

        TBUF 6 + LHLD,
        XCHG,
        I1 LHLD,
        PBUF B LXI,
        B DAD,
        E M MOV,
        H INX,
        D M MOV,

                % CALC VELOCITY OF EKG

        TBUF LHLD,
        L E MOV,
        H D MOV,
        D DAD,
        D DAD,
        XCHG,
        TBUF 2 + LHLD,
        XCHG,
        D DAD,
        D DAD,
        XCHG,
        TBUF 4 + LHLD,
        D DAD,
        -HL CALL,
        XCHG,
        TBUF 12 + LHLD,
        XCHG,
        D DAD,
```

```
        D DAD,
        D DAD,
        XCHG,
        TBUF 10 + LHLD,
        XCHG,
        D DAD,
        D DAD,
        XCHG,
        TBUF 8 + LHLD,
        D DAD,
        XCHG,
        I1 LHLD,
        VBUF B LXI,
        B DAD,
        E M MOV,
        H INX,
        D M MOV,

        I1 LDA,                         % INCR I1
        2 ADI,
        MODPBUFLEN ANI,
        I1 STA,
        NEXT JMP, >

        % DEFINE (<RR>), ROUTINE TO DET BOUNDS ON RR INTERVAL AFTER DETECT

. ASSEMBLER<
        64 A MVI,
        LAMPS STA,
        700 H LXI,
        .
          H DCX,
          H A MOV,
          L ORA,
        JNZ,
        6 A MVI,        % ALLOW 6 STEPS BACK TO LOOK FOR START OF R WAVE
        TS STA,
        I2 LHLD,
          H PUSH,
        . <L
          RRCOUNT LHLD,                 % DECR RR INTERVAL TIMER
          H DCX,
          RRCOUNT SHLD,
            H POP,                      % POP POINTER
          L A MOV,
          2 SUI,
          MODPBUFLEN ANI,
          A L MOV,
            H PUSH,                     % SAVE UPDATED POINTER FOR ANOTHER
          VBUF B LXI,
          B DAD,
          M E MOV,
          H INX,
          M D MOV,
          D A MOV,
```

```
  A ORA,
  -DE CP,
  ZTH LHLD,                     % ABSVAL [VBUF(POINTER)] < ZTH ?
  D DAD,
  IFC,
    TS LDA,                     % NO, DCR # OF CHANCES AND LOOK AGAIN
    A DCR,
    TS STA,
L> JNZ,
  THEN,

  H POP,                        % POP POINTER TO START OF R WAVE
RRCOUNT LHLD,                   % H,L = LAST RR INTERVAL
XCHG,
-255 H LXI,                     % RR INT > 255 COUNTS ?
D DAD,
IFNC,
  255 D LXI,                    % YES, LIMIT TO 255 COUNTS
THEN,
RBUFIPTR LHLD,
E M MOV,
H INX,
RBUFIPTR SHLD,
-RBUFEND B LXI,                 % END OF BUFFER ?
B DAD,
IFNC,
  1 A MVI,                      % YES FLAG RBUF FILLED
  IOABORT STA,
THEN,
0 H LXI,                        % ZERO RR INT COUNTER
RRCOUNT SHLD,

I2 LDA,
6 ADI,
MODPBUFLEN ANI,
A L MOV,
0 H MVI,
  H PUSH,
8 A MVI,                        % ALLOW 8 CHANCES TO FIND END OF RWAVE
TS STA,
. <L
  RRCOUNT LHLD,                 % INCR RR INT COUNTER
  H INX,
  RRCOUNT SHLD,
    H POP,                      % POP POINTER
  L A MOV,
  2 ADI,
  MODPBUFLEN ANI,
  A L MOV,
    H PUSH,                     % PUSH POINTER FOR ANOTHER TRY
  VBUF B LXI,
  B DAD,
  M E MOV,
  H INX,
  M D MOV,
```

```
          D A MOV,
          A ORA,
          -DE CP,
          ZTH LHLD,                       % ABSVAL [VBUF(POINTER)] > ZTH ?
          D DAD,
          IFNC,
            TS LDA,                       % YES, DCR COUNT AND LOOK AGAIN
            A DCR,
            TS STA,
        L> JNZ,
          THEN,
          H POP,                          % POP POINTER AND STORE IN I2
        I2 SHLD,
        0 A MVI,
        LAMPS STA,
        RET, >
`(<RR>) CONSTANT

        % DEFINE R WAVE DETECTOR ROUTINE

`R-DETECTOR CODE<
        I2 LHLD,
        VBUF B LXI,
        B DAD,                            % H,L = VBUF(I2)
        M E MOV,
        H INX,
        M D MOV,
        D A MOV,                          % SGN [VBUF(I2)] + ?
        A ORA,
        IFM,
          -DE CALL,                       % D,E = -[VBUF(I2)]
              D PUSH,
          I2 LDA,
          6 ADI,
          MODPBUFLEN ANI,
          0 H MVI,
          A L MOV,                        % H,L = I2+6, MODPBUFLEN
          VBUF B LXI,
          B DAD,
          M E MOV,
          H INX,
          M D MOV,                        % D,E = [VBUF(I2+6)]
            H POP,
          D DAD,                          % H,L = [VBUF(I2+6)] - [VBUF(I2)]
          XCHG,
          -RTHRESH LHLD,
          D DAD,                                  % DEL > RTHRESH ?
          H A MOV,
          A ORA,
          IFM,
            (<RR>) CALL,                          % YES, FINISH UP
          ELSE,
            I2 LDA,                       % NO, R WAVE NOT DETECTED
            2 ADI,
            MODPBUFLEN ANI,               % INCR I2,MODPBUFLEN
```

```
          I2 STA,
          RRCOUNT LHLD,                 % INCR RR INT COUNTER
          H INX,
          RRCOUNT SHLD,
        THEN,
      ELSE,
        I2 LDA,                         % NO, R WAVE NOT DETECTED
        2 ADI,
        MODPBUFLEN ANI,                 % INCR I2,MODPBUFLEN
        I2 STA,
        RRCOUNT LHLD,                   % INCR RR INT COUNTER
        H INX,
        RRCOUNT SHLD,
      THEN,
      NEXT JMP, >

RADIX !
;F
```

```
% FILE : WLINPT

RADIX @ DECIMAL

    % DIVIDE D,E BY 2^X

. ASSEMBLER<
    % C CONTAINS PWR OF 2 FOR SHIFT
    .
     D A MOV,
     RAL,
     D A MOV,
     RAR,
     A D MOV,
     E A MOV,
     RAR,
     A E MOV,
     C DCR,
    JNZ,
    RET, >
`(/X) CONSTANT

    % COMPLETION ROUTINE FOR RESPIRATION SAMPLING

. ASSEMBLER<
    DLO/AD LDA,
    CMA,
    A E MOV,
    DHI/AD LDA,
    CMA,
    A D MOV,
    4 C MVI,
    (/X) CALL,
    RESPBIPTR LHLD,
    E M MOV,
    H INX,
    RESPBIPTR SHLD,             % UPDATE PTR FOR NEXT PASS
    -RESBEND B LXI,             % TEST FOR END OF BUFFER
    B DAD,
    IFNC,
     3 A MVI,                   % BUFFER FULL, SET IOABORT = 3
     IOABORT STA,
    THEN,
    RET, >
`(RESP) CONSTANT

    % COMPLETION ROUTINE FOR PUPIL DIAMETER
    % PUPIL DIA SCALING : 10 V = 25.4 MM
    %       FOR 12 BIT CONVERSION 2048 CTS/25.4 MM

. ASSEMBLER<
    DLO/ ~ LDA,
    CMA,
    A E MOV,
    DHI/AD LDA,
```

```
        CMA,
        A D MOV,
        2 C MVI,                        % SET UP TO DIVIDE PUPIL BY 2^2
        (/X) CALL,
        PUPBIPTR LHLD,
        E M MOV,
        H INX,
        PUPBIPTR SHLD,
        -PBEND B LXI,                   % TEST FOR BUFFER FULL
        B DAD,
        IFNC,
          4 A MVI,                      % BUFFER OVERFLOW, SET IOABORT = 4
          IOABORT STA,
        THEN,
        RET, >
`(PUPIL) CONSTANT

1 `SIDE/NUM VARIABLE

. ASSEMBLER<
        IOABORT LDA,                    % DO NOT ACCEPT SAMPLE IF IOABORT NONZERO
        A ORA,
        IFZ,
          DHI/AD LDA,                   % CLEAR DONE BIT
          DISMISS JMP,                  % EXIT
        THEN,
        240HZ/COUNT LDA,                % INCR 240 HZ COUNT
        A INR,
        240 CPI,                        % RESET IF COUNT = 240
        IFNZ,
          0 A MVI,
        THEN,
        240HZ/COUNT STA,
        SIDE/NUM LDA,                   % SPECIFY TASK FOR SWITCH RESPONSE
        A ORA,                          % 1 = NUMBER TASK    0 = SIDE TASK
        (MT) CNZ,                       % CHECK FOR RESPONSE IF SIDE/NUM = 0
        RESTIM LHLD,                    % INCREMENT
        H INX,                          %         RESPONSE
        RESTIM SHLD,                    %                TIME
        240HZ/COUNT LDA,
        1 ANI,                          % TEST IF EKG INT
        IFZ,
          240HZ/COUNT LDA,
          1 CPI,                        % RESPIRATION SAMPLE ?
          IFNZ,
            (RESP) CALL,                % YES READ AND STORE RESP SAMPLE
          ELSE,
            3 CPI,                      % PUPIL DIAMETER SAMPLE ?
            IFNZ,
              (PUPIL) CALL,             % YES, READ AND STORE PUPIL DATA
            ELSE,
              7 ANI,                    % SIDE TASK/INST SORT
              7 CPI,
              IFNZ,
                (ST-IS) CALL,
```

```
              ELSE,
                DHI/AD LDA,     % UNUSED SAMPLE, CLEAR DONE BIT
              THEN,
            THEN,
          THEN,
          4 A MVI,              % SPECIFY EKG SAMPLE NEXT
        ELSE,
          (EKG) CALL,           % READ EKG SAMPLE
          240HZ/COUNT LDA,
          A ORA,
          IFNZ,
            6 A MVI,            % SPECIFY RESPIRATION SAMPLE NEXT
          ELSE,
            2 CPI,
            IFNZ,
              5 A MVI,          % SPECIFY PUPIL DIAM NEXT
            ELSE,
              A INR,
              7 ANI,
              7 CPI,
              IFNZ,
                7 A MVI,        % SPECIFY INST/SIDE TASK NEXT
              ELSE,
                4 A MVI,        % SPECIFY EKG NEXT
              THEN,
            THEN,
          THEN,
        THEN,
        CHGN/AD STA,            % STORE CODE OF NEXT CHAN IN CHGN REG

        ADCLFREQ LDA,
        A E MOV,
        64 A MVI,               % 64(10) = 40(H)
        E ORA,
        CSR/ ⁿ STA,
        DISMISS JMP, >
`A/AD2-IH CONSTANT

        % ENABLE SYSTEM INTERRUPTS

`EI CODE< EI, NEXT JMP, >

        % INIT INT HANDLER FOR A/D-2 BOARD

`IH-INIT : A/AD2-IH RST/AD INTERRUPT ;

        % INIT SYSTEM FOR RUN

`INIT : EI                      % INIT SYSTEM INTS
        IH-INIT                 % INIT A/D2 INTS
        0 DSIZE B!                      % SET DISPLAY SIZE
        4 CHGN/AD B!            % SPECIFY EKG AS FIRST SAMPLE
        RUNCOUNT 0<-
        ICOUNT 0<-
        #FIX 0<-
```

```
        MISS# 0<-
        #STIM 0<-
        CUMRESTIM 0<-
        240HZ/COUNT 0<-
        30 DCOUNT !
        1 DFLAG B!
        -1 TINSTR# B!
        -1 INSTR# B!
        IFLAG 0<-
        TCOUNT 0<-
        COR 0<-
        WRO 0<-
        MIS 0<-
        RSULT 400 0FILL
        RSULT RAPNTR !
        INIT/Z
        INSTBUF IBUFLEN 0FILL
        INSTBUF IBUFPTR !
        IOABORT 0<-

        RRCOUNT 0<-
        I1 0<-
        I2 0<-
        BUFFER BUFLEN 0FILL
        BUFICOUNT 0<-
        TBUF 7 0FILL
        BUFFER BUFLEN 2/ + IBUFOPTR !
        BUFFER IBUFIPTR !
        PUPBUF PUPBIPTR !
        RESPBUF RESPBIPTR !
        RBUF RBUFIPTR !
        RBUF RBUFLEN 0FILL
        RESPBUF PARBUFLEN 0FILL
        PUPBUF PARBUFLEN 0FILL
        32741 SEED ! ;

0 `SUBJECT# VARIABLE
0 `MAN/TRIAL# VARIABLE

`WORKLOAD : AX-TER             % SPECIFY OUTPUT TO AX820
        RADIX @ <L
        DECIMAL                % SET RADIX TO DECIMAL
        "DATE" MSG CR
        "SUBJ# " MSG
        SUBJECT# ?
        CR
        "MANEUVER " MSG
        MAN/TRIAL# ?
        CR CR
        WLINDEX-CALC           % CALCULATE AND PRINT WLINDEX FOR RUN
        ILIST                  % LIST INSTBUF AND PLOT FIXATIONS
        SIDE/NUM @ NEZ
        IF
          PRRSULT              % PRINT NUMBER TASK RESULTS
        THEN
```

```
        H/DT-SORT               % SET UP H AND DT TABLES
        ENTROPY                 % PRINT ENTROPY RESULTS
        DT-HIST                 % PLOT DT HISTOGRAMS
        PR                      % PRINT PARAMETER BUFFERS
        L> RADIX ! ;            % RESET RADIX

0 `DT VARIABLE

`RUNWL : T-TER
        1800 U* RUNMAX !        % RUNMAX = TOP * 60 SEC * 30 CNTS/SEC
        240 U* DT !             % DELAY TIME = TOP * 240 HZ
        RESET                   % INITIALIZE S&S CHIP
        SINIT                   % ASSIGN S&S ADDRESSES TO NUMBERS
        10 ADCLFREQ B!          % SET CL FREQ = 240HZ
        INIT                    % INIT SYSTEM
        EN-ADINT                % ENABLE A/D2 INTS
        TRPLET                  % COMPUTE RANDOM THREE-NUMBER SEQUENCE
        3N                      % PUSH THESE THREE NUMBERS ON TOP OF STACK
        S= S=                   % SPEAK NUMBERS ON TOP OF STACK
        INIT/Z                  % INITIALIZE RESPONSE VARIABLES
        S=
        BEGIN
          RESTIM @ DT @ GE
          IF
            RESPON              % STORE RESPONSE AND DELAY TIME ON NUMBER TASK
            SIDE/NUM @ NEZ
            IF
              CR
              CWM @ TYO CR      % SHOW RESPONSE
              RESDELAY @ =      % SHOW DELAY TIME
              CR
            THEN
            TRPLET              % COMPUTE RANDOM THREE-NUMBER SEQUENCE
            3N                  % PUSH THESE THREE NUMBERS ON TOP OF STACK
            S= S=               % SPEAK NUMBERS ON TOP OF STACK
            INIT/Z
            S=
          THEN
          ENOUGH                % ENOUGH DATA TO SEARCH FOR R WAVE
          IF
            R-DETECTOR          % CHECK FOR R WAVE
          ELSE
            PARAMETER-UPDATE    % UPDATE EKG PARAMETERS
          THEN
          IOABORT B@            % HALT IF BUFFER FULL
          IF
            IOABORT B@
            DUP
            1 EQ
            IF
              `RWAVE MSG
              DROP
            ELSE
              2 EQ
              IF
```

```
              `I MSG
            ELSE
              "P OR RES" MSG
            THEN
          THEN
          " BUF FULL" MSG CR
          -1
        ELSE
          RUNCOUNT @
          RUNMAX @
          UGE
        THEN
      END
      DIS-ADINT                   % DISABLE A/D2 INTS
      #FIX @
      #HITS !                     % SET #HITS = #FIX
      WORKLOAD ;                  % CALCULATE WORKLOAD ETC,

HEX

      % POLL A/D2 CHANNEL GIVEN BY CHGN/AD

`POLL : 20 CSR/AD B!
      BEGIN
        CSR/AD B@
        80 AND
        NEZ
      END
      DLO/AD @
      MINUS
      0 CSR/AD B! ;

DECIMAL

      % POLL INSTRUMENT CHANNEL FROM OCULOMETER

`I-POLL : 7 CHGN/AD B!
      BEGIN
        POLL
        128 /
        DUP =                     % DISPLAY LANGLEY CODE #
        DUP
        12 LE
        IF
          IMAP                    % MAP INTO MIT INSTR#
          INSTR# B@
          MAP
        ELSE
          DROP
        THEN
        CR
      0
      END ;

      % POLL PUPIL DIAMETER CHANNEL
```

```
        % OUTPUT GIVEN IN MILLIMETERS

`PD-POLL : 5 CHGN/AD B!
        BEGIN
          POLL
          FLOAT
          80.6 F/
          PDEC
          CR
        0
        END ;

        % POLL RESPIRATION CHANNEL

`RESP-POLL : 6 CHGN/AD B!
        BEGIN
          POLL
          = CR
        0
        END ;

RADIX !
;F
```

```
% FILE: NLRWDATA

RADIX @ HEX

% WORD PUTWORD

`PUTWORD : DUP                      % COPY WORD
        00FF AND                    % MASK FOR LOW ORDER BYTE
        PUTBYTE                     % WRITE OUT LOW ORDER BYTE
        100 /                       % MASK FOR HIGH ORDER BYTE
        PUTBYTE ;                   % WRITE OUT HIGH ORDER BYTE

DECIMAL

`WRDATA : OFILE                     % OPEN FILE FOR WRITING
        WOPEN
        SUBJECT# @                  % WRITE OUT SUBJECT #
        PUTWORD
        MAN/TRIAL# @                % WRITE OUT MAN/TRIAL #
        PUTWORD
        CUMRESTIM @                 % WRITE OUT SIDE TASK RESULTS
        PUTWORD
        #STIM @
        PUTWORD
        MISS# @
        PUTWORD
        #HITS @
        PUTWORD
        9 ( -1 PUTWORD )            % WRITE OUT 10 SPARE WORDS
        IBUFLEN 0
        DO                          % WRITE OUT INSTBUF
          INSTBUF I 2* + @
          PUTWORD
        LOOP
        COR @
        PUTWORD
        WRO @
        PUTWORD
        MIS @
        PUTWORD
        400 0
        DO                          % WRITE OUT RSULT
          RSULT I 2* + @
          PUTWORD
        LOOP
        PARBUFLEN 2* 0
        DO
          RESPBUF I + B@
          PUTBYTE
          PUPBUT I + B@
          PUTBYTE
        LOOP
        RBUFLEN 2* 0
        DO
          RBUF I + B@
```

```
          PUTBYTE
        LOOP
        5 ( 0 PUTBYTE )             % PUT 5 TRAILING 0`S ON FILE
        SHRINK
        CLOSE
        FLUSH ;

HEX

% GETWORD WORD

`GETWORD : GETBYTE                  % RETRIEVE LOW ORDER BYTE
        GETBYTE                     % RETRIEVE HIGH ORDER BYTE
        100 *                       % SCALE HIGH ORDER BYTE
        OR ;

DECIMAL

`RDDATA : OFILE                     % OPEN FILE
        OPEN
        GETWORD
        SUBJECT# !                  % READ IN SUBJECT #
        GETWORD
        MAN/TRIAL# !                % READ IN MAN/TRIAL #
        GETWORD
        CUMRESTIM !                 % READ IN SIDE TASK RESULTS
        GETWORD
        #STIM !
        GETWORD
        MISS# !
        GETWORD
        #HITS !
        9 ( GETWORD DROP )
        IBUFLEN 0
        DO                          % READ IN INSTBUF
          GETWORD
          INSTBUF I 2* + !
        LOOP
        GETWORD
        COR !
        GETWORD
        WRO !
        GETWORD
        MIS !
        400 0
        DO                          % READ IN RSULT
          GETWORD
          RSULT I 2* + !
        LOOP
        PARBUFLEN 2* 0
        DO
          GETBYTE
          RESPBUF I + B!
          GETBYTE
          PUPBUT I + B!
```

```
        LOOP
        RBUFLEN 2* 0
        DO
          GETBYTE
          RBUF I + B!
        LOOP
        CLOSE
        FLUSH ;

RADIX !
;F
```

```
% FILE: TOPTEN

% AUTHOR: Albert T. Stephens

% TOPTEN contains definitions which compute the % occurrence of the
% top-N most frequently occurring sequences (RUNTOP), produce a plot
% of % occurrence vs task difficulty (PLOTOP), and prints out a summary
% of variables for each run (PS).  This file assumes that words in the
% experiment controlling files are defined.  The form of the data
% should correspond to that produced by the experiment controlling
% files.
% To Run This File:
%    1. load experiment controlling files
%    2. load RUNTOP
%    3. type `FILNAM RUNTOP PLOTOP PS <CR>
%       FILNAM is the input data file name
%
% The number of sequences, N, may be specified by setting the variable
% TOPN to the desired number.


RADIX @ DECIMAL

15 `TOPN VARIABLE                       % # OF SEQS INCLUDED IN CALCULATIONS
TOPN @ `TTEN ARRAY                      % CONTAINS TOP TEN SEQS
TOPN @ 2 * `TTCNTS ARRAY                % CONTAINS FREQ OF TOP TEN SEQS
0 `OFFSET VARIABLE                      % OFFSET INTO TTCNTS FOR EACH RUN
4 `#SEQS ARRAY                          % CONTAINS THE TOTAL COUNTS FOR EACH RUN

% RUNTOP FINDS THE `TOPN` MOST FREQUENTLY OCCURRING SEQUENCES IN THE NO-LOADING
% CASE AND STORES THEIR FREQUENCIES.  THESE SEQUENCES ARE THEN FOUND IN EACH
% OF THE FOLLOWING LOADING CASES AND THEIR FREQUENCIES ARE STORED.


`RUNTOP :
        RDDATA                          % READ IN DATA
        0 OFFSET !                      % ZERO OFFSET
        H/DT-SORT                       % FILL HBUF4
        16 AX820                        % SELECT PROPORTIONAL CHAR SPACING
        12-PRT/DEN                      % SET PRINT DENSITY TO 12 CPI
        T-TER                           % DISABLE PRINTER
        4 ENT-INIT                      % INIT ENT CALC FOR SEQ LENGTH 4
        #SEQ @ #SEQS !                  % STORE TOTAL COUNTS
        AX-TER                          % ENABLE PRINTER
        ENHANCE/PRINT
        "SUBJ# " MSG
        SUBJECT# ? CR CR
        NORM/PRINT
        T-TER
        TOPN @ 0
        DO
          4096 H-C-AUX                  % FIND LARGEST SEQ
          MAXSEQ @ TTEN I 2* + !        % STORE MAX SEQ
```

```
          MAXCNT @ TTCNTS I + B!         % STORE MAX COUNTS
          #SEQ @ #SEQS !                 % STORE TOTAL COUNTS
          0 VHBUF @ MAXSEQ @ + B!        % CLEAR ENTRY IN HBUF4
        LOOP
        TOPN @ OFFSET +!                         % SET OFFSET FOR SECOND RUN
        3 0
        DO
          RDDATA                         % READ IN NEXT SET OF DATA
          H/DT-SORT                      % PUT COUNTS INTO HBUF4
          T-TER                          % DISABLE PRINTER
          4 ENT-INIT                     % INIT ENT CALC FOR SEQ LENGTH 4
          TOPN @ 0
          DO
            TTEN I 2* + @                % PUSH SEQ# ON STACK
            VHBUF @ + B@                 % ADD TO ADDR AND RETRIEVE COUNTS
            TTCNTS OFFSET @ + I + B!     % STORE COUNTS IN TTCNTS
          LOOP
          TOPN @ OFFSET +!                       % INCR OFFSET FOR NEXT RUN
          #SEQ @ #SEQS I 2* + 2 + !      % STORE TOTAL COUNTS FOR RUN
        LOOP
        ;

% PRINTOP PRINTS OUT THE PERCENTAGE OCCURRENCES FOR THE TOP TEN SEQUENCES
% FROM THE FREQUENCIES OBTAINED IN RUNTOP

`PRINTOP :
        AX-TER
        15 TAB `1100 MSG
        30 TAB `1200 MSG
        45 TAB `1300 MSG
        60 TAB `1400 MSG
        CR
        TOPN @ 0
        DO
          4 0
          DO
            TTCNTS J + TOPN @ I * + B@  % PUSH NUMBER OF COUNTS
            FLOAT
            #SEQS I 2* + @ FLOAT F/     % DIVIDE BY TOTAL COUNTS
            100. F*                     % CHANGE TO PERCENT
            15 I 1 + * TAB              % TAB OVER FOR NEXT RESULT
            PDEC                        % PRINT RESULT
            " %" MSG
          LOOP
          CR CR
        LOOP
        ;


15 `DLL VARIABLE                        % DASHED LINE LENGTH FOR TOTAL % PLOT
5 `SPL VARIABLE                         % LENGTH OF SPACES FOR DASHED LINE
0 `DLC VARIABLE                         % CURRENT LENGTH OF LINE
0 `SPC VARIABLE                         % CURRENT NUMBER OF SPACES
4 `TOT% ARRAY                           % CONTAINS SUM OF % FOR TOP TEN SEQS
TOPN @ 4 * `HEIGHT ARRAY                % CONTAINS Y VALUE OF POINTS
0 `POINT VARIABLE                       % CONTAINS CURRENT Y VALUE
```

```
% PLOTOP PLOTS THE PERCENTAGE OCCURRENCE VS LOADING LEVEL
% FOR TOPN SEQUENCES

`PLOTOP :
        0 DLC !
        0 SPC !
        0 POINT !
        HEIGHT TOPN @ 4 * 0FILL
        TOT% 4 0FILL
        AX-TER                  % ENABLE PRINTER
        34 TAB
        `PERCENT MSG CR CR
        37 TAB
        `5 MSG
        74 TAB
        `10 MSG CR
        <AXLN
        500 0
        DO
          I -1 AXDOT
        LOOP
        AXLN
        4 0
        DO
          TOPN @ 0
          DO
            TTCNTS I + TOPN @ J * + B@     % PUSH # OF COUNTS
            FLOAT
            #SEQS J 2* + @ FLOAT F/        % DIVIDE BY TOTAL COUNTS
            400. F*                        % SCALE FOR PERCENT PLOT
            10. F*                         % TOP OF SCALE = 10 %
            INTEGER DUP
            HEIGHT I 2* + TOPN @ 2* J * + !        % STORE Y VALUE
            5 /                            % TOP OF SCALE = 50% FOR TOT%
            TOT% J 2* + +!                 % INCR TOTAL %
          LOOP
        LOOP
        3 0
        DO
          112 0
          DO
            DLC @ DLL @ LT                 % TEST FOR DESIRED LINE LENGTH
            IF
              TOT% J 2* + 2 + @            % PUSH Y VAL @ X+1
              TOT% J 2* + @                % PUSH Y VAL @ X
              DUP POINT !                  % STORE Y VAL @ X
              - FLOAT                      % FIND DIFF AND FLOAT RESULT
              112. F/                      % DETERMINE SLOPE OF LINE
              I 1 + FLOAT F* INTEGER       % DETERMINE NEW Y VAL
              POINT +!                     % STORE NEW Y VAL
              POINT @ -1 AXDOT             % STORE NEW Y VAL FOR PLOT
              DLC 1+!                      % INCR COUNTER
            ELSE
              SPC @ SPL @ LT               % TEST FOR DESIRED # OF SPACES
```

```
       IF
        SPC 1+!                 % INCR COUNTER
       ELSE
        0 SPC !                 % ZERO SPACE COUNTER
        0 DLC !                 % ZERO LINE COUNTER
       THEN
      THEN
      TOPN @ 0
      DO
       HEIGHT I 2* + TOPN @ 2* K 1 + * + @     % RETRIEVE Y VAL @ X+1
       HEIGHT I 2* + TOPN @ 2* K * + @         % RETRIEVE Y VAL @ X
       POINT !                  % STORE Y VAL @ X
       POINT @ - FLOAT          % SUBT X FROM X+1 (Y VAL)
       112. F/                  % DETERMINE SLOPE OF LINE
       J 1 + FLOAT F* INTEGER   % DETERMINE NEW Y VAL
       POINT +!                 % STORE NEW Y VAL
       POINT @ -1 AXDOT         % STORE NEW Y VAL FOR PLOT
      LOOP
      AXLN
     LOOP
     DOTLINE
    LOOP
    AXLN>
    4 ( CR )
    06 AX820                    % SELECT FIXED CHAR SPACING
    TOPN @ 0
    DO
     I 2* TTEN + @              % PUSH SEQ#
     MAXSEQ !                   % STORE IN MAXSEQ
     SEQ-MAP                    % PRINT CORRESPONDING INST NAME
     CR
    LOOP
    ;

`SP : ( WAIT 32 AX820 ) ;       % PRINT N SPACES ON PT460

`PTTAB : ( WAIT 09 AX820 ) ;    % MOVE TO Nth TAB


% SUMMARY PRINTS OUT SEQUENCE PARAMETERS

`SUMMARY :
    RDDATA
    H/DT-SORT                   % SET UP ENTROPY ARRAYS
    T-TER                       % DISABLE PRINTER
    4 ENT-INIT                  % INIT ENT CALC FOR SEQ LEN 4
    4096 ( 4096 H-CALC )        % ORDER SEQS BY FREQ
    EN-AUX                      % COMPUTE ENTROPY INFO
    AX-TER                      % ENABLE PRINTER
    MAN/TRIAL# @ = 2 PTTAB      % PRINT RUN #
    #SEQ @ = 5 SP               % PRINT #SEQ
    HT F@ PDEC 5 SP             % PRINT ENTROPY
    #DIFSEQ @ = 5 SP            % PRINT # OF DIFFERENT SEQS
    Q F@ PDEC 5 SP              % PRINT MAX # OF SEQS OF LEN N
    KH F@ PDEC 5 SP             % PRINT FRACTIONAL LEN OF RUN
```

```
        RF F@ PDEC 5 SP                         % PRINT REDUNDANCY FACTOR
        COR @ = 5 SP                            % PRINT # CORRECT ON # TASK
        WRO @ = 5 SP                            % PRINT # WRONG ON # TASK
        MIS @ = 5 SP                            % PRINT # MISSED ON # TASK
        CR
        ;

% PS PRINTS OUT SUMMARY FOR EACH LOADING LEVEL

`PS :
        AX-TER
        06 AX820                                % SELECT FIXED CHAR SPACING
        16.8-PRT/DEN                            % SET PRINT DENSITY TO 16.8 CPI
        CR CR
        "RUN#" MSG
        10 SP
        "#SEQ     H(OBS)    #DIFSEQ       Q             K          RF" MSG
        "       #COR   #WRO   #MIS" MSG CR CR
        4 0
        DO
          SUMMARY
        LOOP
        9 ( CR )
        ;

RADIX !
;F
```

```
% FILE : GETDAT

% Author : A.T. Stephens
% Date : 6/81
% Date of Last Revision : 7/11/81


% GETDAT contains programs that collect, store and partially analyze the
% ASCII aircraft performance data from the February 1981 pilot mental
% workload experiments.  The data was transfered from NASA/Langley,
% where it had been digitized from 14-channel FM mag tape, to MIT in
% four files named MIT__ and was stored on floppy disks.  The data
% included means and standard deviations from six channels, # of control
% inputs, and a frequency analysis of the glide slope and localizer
% channels.  The data format is given below.
%
%       SUBJ/RUN#      # DATA POINTS
%         MEAN             STD DEV      GLIDE SLOPE
%          .                  .         LOCALIZER
%          .                  .         ELEVATOR
%          .                  .         AILERONS
%          .                  .         PITCH ATTITUDE
%         MEAN             STD DEV      ROLL ATTITUDE
%        CONTROL           CONTROL
%        INPUTS            INPUTS       ELEVATOR, AILERON
%          1            POWER AMPL GS
%          2            POWER AMPL GS
%          .                 .
%          .                 .
%          .                 .
%          256          POWER AMPL GS
%          1            MAX PERIOD GS
%          1            POWER AMPL LOC
%          2            POWER AMPL LOC
%          .                 .
%          .                 .
%          .                 .
%          256          POWER AMPL LOC
%          2            MAX PERIOD LOC
%
%
% To run the programs, open the file to be analyzed and position the
% pointer to the byte preceding the first byte of the desired run.
% Results of the analysis include RMS values for the glide slope and
% localizer data, percent shift in frequency of the glide slope and
% localizer data, and computation of an aircraft performance measure.

RADIX @ DECIMAL

`LEXI LOAD
`RETR LOAD
```

```
0 `FACTOR VARIABLE                                % USED IN BYTE COMBINATION
0 `SUBJ VARIABLE                                  % CONTAINS CURRENT SUBJECT#
0 `RUN VARIABLE                                   % CONTAINS CURRENT RUN#
48 `MEAN ARRAY                                    % ARRAY FOR STORAGE OF MEANS
48 `STDEV ARRAY                                   % ARRAY FOR STORAGE OF VARIANC
0 `MVPTR VARIABLE                                 % POINTER FOR MEAN & STDEV
1028 `CDF ARRAY                                   % PARTIAL SUM ARRAY
0 `CDFPTR VARIABLE                                % POINTER FOR CDF ARRAY
32 `PEAK ARRAY                                    % CONTAINS PERIOD OF MAX AMPL
0 `OS VARIABLE                                    % OFFSET POINTER
4 `SUM ARRAY                                      % CONTAINS CURRENT AMPL SUM
0 `FLAG VARIABLE                                  % INDICATES BAD LINE
0 `FLAG1 VARIABLE                                 % INDICATES NO FREQ DATA
0 `FLAG2 VARIABLE                                 % INDICATES NO FREQ DATA, 1 RL
0 `PPTR VARIABLE                                  % PEAK ARRAY POINTER
32 `AMPCOM ARRAY                                  % AMPL COMPARISON ARRAY
0 `1NUM VARIABLE                                  % CURRENT SAMPLE #
0.0 `2NUM FVARIABLE                               % CURRENT AMPLITUDE
0 `ACOS VARIABLE                                  % AMPCOM OFFSET POINTER
8 `PF ARRAY                                       % PERFORMANCE FACTOR ARRAY


`Z-RUN :
        CDF 1028 0FILL
        SUM 4 0FILL
        0 SUBJ !
        0 RUN !
        0 CDFPTR !
        10 OS !
        0 FLAG2 !
        ;

`Z-PILOT :
        Z-RUN
        MEAN 48 0FILL
        STDEV 48 0FILL
        PEAK 32 0FILL
        AMPCOM 32 0FILL
        0 MVPTR !
        0 PPTR !
        0 ACOS !
        0 FLAG1 !
        ;


`1ST : 6 PTR ! ;

`2ND : 16 PTR ! ;
```

```
`GETDAT :
        10 FACTOR !                                 % SET FACTOR FOR SUBJ#
        2 0
        DO
          RDBUF I OS @ + + B@                       % PUSH BYTE FOR SUBJ#
          48 -                                      % CONVERT FROM ASCII
          FACTOR @ *
          DUP
          0 GT                                      % TEST FOR VALID #
          IF
            SUBJ +!                                 % COMBINE BYTES FOR SUBJ#
          ELSE
            DROP
          THEN
          FACTOR @ 10 / FACTOR !                    % DECR FACTOR FOR NEXT BYTE
        LOOP
        1000 FACTOR !
        6 2
        DO
          RDBUF I OS @ + + B@                       % PUSH BYTE FOR RUN#
          48 -                                      % CONVERT FROM ASCII
          FACTOR @ *                                %
          RUN +!                                    % COMBINE BYTES FOR RUN#
          FACTOR @ 10 /                             % DECR FACTOR FOR NEXT BYTE
        LOOP
        "SUBJECT " MSG
        SUBJ @ = CR
        "RUN " MSG
        RUN @ = CR                                  % PRINT RUN#
        2ND                                         % SET POINTER FOR 2ND #
        RIL                                         % PUSH # OF SAMPLES
        "# OF SAMPLES " MSG
        = CR
        6 0                                         % THE 6 MEANS & STDEVS ARE FOR
        DO                                          % GSL, LOC, ELE, AIL, PIT, BANK
          1 GL                                      % GET NEXT LINE
          1ST                                       % SET POINTER FOR 1ST #
          RFL                                       % PUSH MEAN
          MEAN MVPTR @ + F!                         % STORE MEAN
          2ND                                       % SET PTR FOR 2ND #
          RFL                                       % PUSH STDEV
          STDEV MVPTR @ + F!                        % STORE STDEV
          4 MVPTR +!                                % INCR POINTER
        LOOP
        1 GL
        1ST
        RIL                                         % PUSH # OF ELEVATOR INPUTS
        "# OF ELEVATOR INPUTS " MSG
        = CR
        2ND
```

```
RIL                                 % PUSH AILERON INPUTS
"# OF AILERON INPUTS " MSG
= CR
CR CR
1 GL
1ST
BEGIN                               % THIS LOOP PUSHES 1 OR "
  PTR @ RDBUF + B@                  % GET ASCII #
  PTR 1+!                           % INCR POINTER
  DUP
  32 GT                             % TEST FOR 1 OR "
  IF
    -1                              % IF 1 OR ", END LOOP
  ELSE
    DROP
    0                               % OTHERWISE, CONTINUE LOOP
  THEN
END
CR
34 NE                               % TEST FOR "
IF
  0 CDFPTR !
  GPOS                              % GET CURRENT POINTER POSITION
  SWAP
  26 -                              % COMPUTE ADDR OF PREVIOUS LINE
  SWAP
  SPOS                              % MOVE POINTER BACK 1 LINE
  2 0
  DO
    260 0
    DO
      1 GL
      27 0
      DO
        RDBUF I + B@                % TEST FOR FORM FEED IN DATA
        DUP
        12 EQ
        SWAP
        57 GT
        OR
        IF
          1 FLAG !                  % IF FORM FEED, SET FLAG
        THEN
      LOOP
      FLAG @ 1 NE                   % TEST FLAG NOT SET
      IF
        1ST
        RIL                         % PUSH SAMPLE #
        1NUM !                      % STORE CURRENT SAMPLE #
        2ND
        RFL                         % PUSH AMPL OF FREQ PLOT
```

```
              2NUM F!                                   % STORE CURRENT AMPLITUDE
              2NUM F@
              SUM J 4 * + F+!                           % INCR TOTAL
              SUM J 4 * + F@                            % PUSH CORRECT TOTAL
              CDF CDFPTR @ + F!                         % STORE CURRENT TOTAL
            ELSE
              SUM J 4 * + F@
              CDF CDFPTR @ + F!                         % IF BAD LINE, STORE LAST VALUE
            THEN
            0 FLAG !                                    % RESET FLAG
            1NUM @ 256 EQ
            IF
              EXIT                                      % EXIT LOOP IF SAMP# = 256
            THEN
            4 CDFPTR +!                                 % INCR POINTER
          LOOP
          1 GL
          2ND
          RFL                                           % PUSH PERIOD OF MAX AMPLI
          PEAK PPTR @ + F!                              % STORE PERIOD
          4 PPTR +!                                     % INCR POINTER
        LOOP
      ELSE
        "FREQUENCY DATA NOT AVAILABLE" MSG CR
        1 FLAG1 !
        1 FLAG2 !
        CR CR CR
      THEN
      ;

0 `H1 VARIABLE                                          % HEIGHT FOR CDF PLOT
0 `SHIFT VARIABLE                                       % PROVIDES OFFSET FOR LINES
                                                        % TO BE PLOTTED


`SUMPLOT :
      06 AXB20                                          % SELECT FIXED SPACING
      10-PRT/DEN                                        % SET PRINT DENSITY TO 10 CPI
      14 TAB
      `CDF MSG CR
      `0. MSG
      14 TAB
      `.5 MSG
      29 TAB
      `1.0 MSG CR
      <AXLN                                             % ENTER GRAPHICS MODE
      256 0
      DO
        I -1 AXDOT                                      % DRAW AXIS LINE
      LOOP
      256 0
```

```
          DO
            I 128 LT
            IF
              CDF CDFPTR @ + F@                  % PUSH CUMULATIVE AMPL
              SUM OS @ + F@                      % PUSH TOTAL AMPL
              F/                                 % COMPUTE % OF TOTAL AMPL
              256. F*                            % SCALE FOR PLOT
              INTEGER H1 !
              2 0
              DO
                I 2 /MOD
                SWAP DROP
                IF
                  0 SHIFT !
                ELSE
                  1 SHIFT !
                THEN
                H1 @ 0
                DO
                  I SHIFT @ + -1 AXDOT
                  2
                +LOOP
                AXLN
              LOOP
            THEN
            I 13 EQ                              % TEST SAMPLE #
            I 26 EQ                              % TEST SAMPLE #
            OR
            IF
              CDF CDFPTR @ + F@                  % PUSH CUMULATIVE AMPL
              SUM OS @ + F@                      % PUSH TOTAL AMPL
              F/                                 % COMPUTE % OF TOTAL AMPL
              100. F*                            % SCALE TO # BETWEEN 0 & 100
              AMPCOM ACOS @ + F!                 % STORE PERCENTAGE
              4 ACOS +!                          % INCR AMPCOM POINTER
            THEN
            4 CDFPTR +!                          % INCR CDF POINTER
          LOOP
%         AXLN>                                  % EXIT GRAPHICS MODE
          CR CR
          ;


32 `RMS ARRAY                                    % CONTAINS RMS FOR GSL & LOC

`RMSCOMP :
          0 OS !
          4 0
          DO
            2 0
```

```
     DO
      MEAN 24 J * + 4 I * + F@            % PUSH MEAN
      DDUP
      F*                                  % SQUARE MEAN
      STDEV 24 J * + 4 I * + F@           % PUSH ST DEV
      DDUP
      F*                                  % SQUARE ST DEV
      F+                                  % ADD TO MEAN SQUARED
      FSQRT
      RMS 8 J * + 4 I * + F!              % STORE RMS
     LOOP                                 % GS1,LOC1, ... GS4,LOC4
    LOOP
    ;

3 `H ARRAY                                    % BAR HEIGHT ARRAY

`LIGHT :
    7 0
    DO
     I 2 /MOD                             % ALTERNATE DOT OFFSET
     SWAP DROP
     IF
      H 0 + @ 0
      DO
       I 1 + -1 AXDOT
       2
      +LOOP
     ELSE
      H 0 + @ 0
      DO
       I -1 AXDOT
       2
      +LOOP
     THEN
     AXLN
    LOOP
    ;

`SOLID :
    7 0                                   % START SOLID BAR
    DO
     H 2 + @ 0
     DO
      I -1 AXDOT
     LOOP
     AXLN
    LOOP
    ;

`LINED :
    7 0
```

```
        DO
          H 4 + @ 0
          DO
            I -1 AXDOT
            2
          +LOOP
          AXLN
        LOOP
        ;


`1BAR : <AXLN SOLID AXLN> ;                                  % PRINT 1 BAR WHOSE
                                                             % HEIGHT IS IN H 2 +

`2BAR : <AXLN LIGHT SOLID AXLN> ;                            % PRINT 2 BARS WHOSE
                                                             % HEIGHTS ARE IN H 0 +
                                                             % AND H 2 +

`3BAR : <AXLN LIGHT SOLID LINED AXLN> ;                      % PRINT 3 BARS WHOSE
                                                             % HEIGHTS ARE IN H 0 +
                                                             % H 2 +, AND H 4 +
0 `#GL VARIABLE

`COMPUTE :
        "Are you starting a new pilot? Type Y or N" MSG CR CR
        TYI `Y 1+ B@ EQ                               % TEST RESPONSE
        IF
          Z-PILOT
        THEN
        "Type # of continuous files for this pilot (1-4)" MSG CR
        TYI 48 - 0
        DO
          AX-TER
          06 AX820
          12-PRT/DEN
          Z-RUN
          1 GL
          GETDAT
          #GL @ GL
          FLAG2 @ 0 EQ                                % TEST FOR NO FREQ DATA
          IF
            "GS SPECTRUM CDF" MSG CR CR
            0 CDFPTR !
            0 OS !
            SUMPLOT
            CR CR
            "LOC SPECTRUM CDF" MSG CR CR
            1024 CDFPTR !
            4 OS !
            SUMPLOT
            CR CR
```

```
          ELSE
            16 ACOS +!
          THEN
        LOOP
        RMSCOMP
        CR
        ;

`RMSPRINT :
        "RMS VALUES FOR GS AND LOC" MSG CR CR
        2 0
        DO
          I 0 EQ
          IF
            "GS " MSG
          ELSE
            "LOC " MSG
          THEN
          4 0
          DO
            RMS I 8 * + J 4 * + F@
            F=
          LOOP
          CR
        LOOP
        CR CR
        4 0
        DO
          2 0
          DO
            RMS J 8 * + I 4 * + F@                          % PUSH RMS
            5. F*                                           % SCALE FOR PLOT
            INTEGER H I 2* + !                              % STORE BAR HEIGHT
          LOOP
          2BAR                                              % PRINT DOUBLE BAR
          CR
        LOOP
        CR CR
        ;

`MAXFREQ :
        "MAX FREQUENCY FOR GS AND LOC" MSG CR
        CR
        FLAG1 @ 0 EQ                                        % TEST FOR NO FREQ DATA
        IF
          2 0
          DO
            I 0 EQ
            IF
              "GS " MSG
            ELSE
```

```
              "LOC " MSG
            THEN
            4 0
            DO
              1.
              PEAK I 8 * + J 4 * + F@
              F/
              F=
            LOOP
            CR
          LOOP
          CR
          4 0
          DO
            2 0
            DO
              1.
              PEAK J 8 * + I 4 * + F@
              F/
              5000. F*                                 % SCALE FOR PLOT
              INTEGER
              H I 2* + !
            LOOP
            2BAR
            CR
          LOOP
        ELSE
          "Unable to plot due to missing data" MSG CR CR CR
        THEN
        ;

`%POWER :
        0 ACOS !
        2 0
        DO
          I 0 EQ
          IF
            CR CR
            "GS % OF POWER  >.1 Hz : 1ST BAR    >.2 Hz : 2ND BAR" MSG CR
            CR
            2 0
            DO                         % PRINT % OF POWER VALUES
              I 0 EQ
              IF
                0 ACOS !
                ">.1 Hz  " MSG
              ELSE
                4 ACOS !
                ">.2 Hz  " MSG
              THEN
              4 0
```

```
    DO
      AMPCOM ACOS @ + F@
      FMINUS
      100. F+
      F=
      16 ACOS +!
    LOOP
    CR
  LOOP
  0 ACOS !
 ELSE
  CR CR
  "LOC % OF POWER  >.1 Hz : 1ST BAR    >.2 Hz : 2ND BAR" MSG CR
  CR
  2 0
  DO                                   % PRINT % OF POWER VALUES
    I 0 EQ
    IF
      8 ACOS !
      ">.1 Hz  " MSG
    ELSE
      12 ACOS !
      ">.2 Hz  " MSG
    THEN
    4 0
    DO
      AMPCOM ACOS @ + F@
      FMINUS
      100. F+
      F=
      16 ACOS +!
    LOOP
    CR
  LOOP
  8 ACOS !
THEN
4 0
DO
  AMPCOM ACOS @ + F@
  FMINUS
  100. F+
  5. F*
  INTEGER
  H 0 + !
  4 ACOS +!
  AMPCOM ACOS @ + F@
  DDUP
  FMINUS
  100. F+
  5. F*
  INTEGER
```

```
            H 2 + !
            FGTZ
            IF
              2BAR
              CR
            ELSE
              "??????????" MSG CR
              CR CR CR
            THEN
            12 ACOS +!
          LOOP
        LOOP
        CR CR
        ;

PERFACT :
        0 ACOS !
        4 0
        DO
          2 0
          DO
            AMPCOM ACOS @ + F@
            4 ACOS +!
            AMPCOM ACOS @ + F@
            4 ACOS +!
            2DROP
            FMINUS
            100. F+
            .1 F*
          LOOP
          F+
          2 0
          DO
            RMS J 8 * + I 4 * + F@
            .15 F*
          LOOP
          F+
          F+
          DDUP
          PF I 4 * + F!
          "PERFORMANCE FACTOR " MSG
          F= CR
        LOOP
        CR CR
        4 0
        DO
          PF I 4 * + F@
          5. F*
          INTEGER H 2 + !
          1BAR
          CR
```

```
          LOOP
          ;

ANALYZE : COMPUTE RMSPRINT MAXFREQ %POWER PERFACT T-TER  ;



RADIX !
;F
```

```
% FILE: NHIST

% AUTHOR: A.T. Stephens


% THIS FILE CONTAINS PROGRAMS FOR PLOTTING THE DWELL TIME HISTOGRAMS
% FOR THE RESPONSE TIME TO THE NUMBER TASK AND FOR FIXATIONS ON ALL
% INSTRUMENTS COMBINED.  THIS PROGRAM REQUIRES THAT WORDS FROM THE
% EXPERIMENT CONTROLLING FILES BE DEFINED (HHIST1).  TO RUN THIS
% PROGRAM, TYPE FOUR DATA FILE NAMES (FORMAT IN RWDATA) AND THE
% WORD G.  THE RESULTS OF THIS PROGRAM MAYBE SEEN IN FIGURES 16
% THROUGH 26.


RADIX @ DECIMAL

100 `BINS ARRAY                    % RESPONSE TIME COUNTS (EACH BIN = 1/20 SEC)
0 `RTOTAL VARIABLE                 % TOTAL NUMBER OF RESPONSES TO NUMBER TASK
0 `DTMAX VARIABLE                  % DELAY TIME BETWEEN NUMBERS


% SORT RESPONSE TIMES INTO BINS ( 20 PER SEC )

`PUTBIN :
        20 * 1 - DTMAX !                  % (20 * TOP OF STACK) - 1 = DTMAX
        RSULT RAPNTR !                    % INITIALIZE POINTER
        COR @ WRO @ + MIS @ + RTOTAL !    % RTOTAL = TOTAL # OF RESPONSES
        RTOTAL @ 0
        DO
          R/D-SORT                        % UNPACK RESPONSE/RESPONSE TIME
          CWM @ 96 - NEZ                  % JUMP IF MISS
          IF
            RESDELAY @ 1200 GE            % TEST FOR DELAY >= 5 SEC
            IF
              DTMAX @ 2* BINS + 1+!       % IF SO, STORE IN LAST BIN
            ELSE
              RESDELAY @ 12 / 2* BINS + 1+!  % DIVIDE COUNTS BY 12 COUNTS/BIN
            THEN                             % TO DETERMINE OFFSET INTO ARRAY
          ELSE
            DTMAX @ 2* BINS + 1+!         % COUNT MISS AS MAX DELAY TIME
          THEN
        LOOP
        ;


% PRINT CONTENTS OF "BINS"

`TEST :
        100 0
        DO
          I 2* BINS + @ =
```

```
        LOOP
        ;

% PLOT HISTOGRAM OF RESPONSE TIMES

`RTHIST :
        COR @ WRO @ + MIS @ + RTOTAL !
        "SUBJECT# " MSG
        SUBJECT# @ = CR
        "MANEUVER " MSG
        MAN/TRIAL# @ = CR
        CR
        34 TAB
        `PERCENT MSG CR CR
        14 TAB
        `25 MSG
        30 TAB
        `50 MSG
        45 TAB
        `75 MSG
        60 TAB
        `100 MSG
        CR
        <AXLN
        400 0
        DO
          I -1 AXDOT
        LOOP
        AXLN
        100 0
        DO
          I 1+ 20 /MOD UNDER            % PRINT DOTTED LINE EVERY SECOND
          EQZ
          IF
            DOTLINE
          THEN
          2 0
          DO
            J 2* BINS + @ FLOAT         % PUSH COUNTS FROM BIN
            RTOTAL @ FLOAT F/           % DIVIDE BY TOTAL COUNTS
            400. F*                     % SCALE FOR PERCENT PLOT
            INTEGER 0
            DO
              I -1 AXDOT
            LOOP
            AXLN
          LOOP
        LOOP
        AXLN>
        ;
```

```
75 `DTSUM ARRAY                       % ARRAY CONTAINING SUM OF ALL INST
                                     % DWELL TIMES
0 `BPTR1 VARIABLE                    % DTBUF POINTER
0 `BPTR2 VARIABLE                    % DTSUM POINTER
0 `LPSUM VARIABLE                    % TOTAL # OF FIXATIONS


% SUM DWELL TIME COUNTS FROM INDIVIDUAL INSTRUMENTS (FROM DTBUF)

`SUMUP :
        0 LPSUM !
        DTSUM 75 0FILL
        RDDATA                       % READ DATA INTO MEMORY
        H/DT-SORT                    % SORT DATA AND FILL DTBUF
        8 0
        DO
          DTSUM BPTR2 !              % SET POINTER TO TOP OF BUFFER
          I 75 * DTBUF + BPTR1 !     % SET POINTER LOCATION FOR NEXT INST
          75 0
          DO
            BPTR1 @ B@ BPTR2 @ @ +   % ADD DTBUF TO DTSUM
            BPTR2 @ !                % AND STORE IN DTSUM
            BPTR1 @ B@ LPSUM +!      % RUNNING TOTAL OF FIXATIONS
            BPTR1 1+!                % INCR TO NEXT BYTE OF DTBUF
            BPTR2 1+! BPTR2 1+!      % INCR TO NEXT WORD OF DTSUM
          LOOP
        LOOP
        ;


% DL INSERTS A VERTICAL DOTTED LINE IN OUTPUT

`DL :
        200 0
        DO
          I 10 /MOD
          UNDER EQZ
          IF
            I -1 AXDOT
          THEN
        LOOP
        ;

% PLOT DWELL TIME HISTOGRAM FOR ALL INSTRUMENTS COMBINED

`PLOTDT :
        18 TAB
        "PERCENT" MSG CR
```

```
        0 TAB
        `0 MSG
        20 TAB
        `10 MSG
        39 TAB
        `20 MSG CR
        <AXLN
        200 0
        DO
          I -1 AXDOT
        LOOP
        AXLN
        75 0
        DO
          I 1+ 15 /MOD UNDER
          EQZ
          IF
            DL                              % PRINT DOT LINE EVERY SEC
          THEN
          1 0
          DO
            J 2* DTSUM + @ FLOAT            % PUT COUNTS FOR BIN ON STACK
            #FIX @ FLOAT F/                 % DIVIDE BY TOTAL COUNTS
            200. F*                         % SCALE FOR PERCENT PLOT
            5. F*                           % TOP OF SCALE = 20 %
            INTEGER 0
            DO
              I -1 AXDOT                    % PLOT LINE FOR BIN
            LOOP
            AXLN
          LOOP
        LOOP
        AXLN>
        ;


% CALCULATE THE CUMULATIVE DISTRIBUTION FUNCTION OF THE DWELL TIME
% HISTOGRAM BEGINNING AT 1, 2, 3 AND 4 SEC

5 `CDF ARRAY                        % CUMULATIVE DISTRIBUTION FUNCTION ARR
0 `PTR VARIABLE                     % CDF ARRAY POINTER


`PCDF :
        CDF 5 0FILL
        0 PTR !
        61 15
        DO
```

```
          75 I
          DO
            I 2* DTSUM + @              % PUSH COUNTS ON STACK
            PTR @ CDF + +!              % STORE IN CDF
          LOOP
          PTR 1+! PTR 1+!               % INCR POINTER
          15
        +LOOP
        4 0
        DO
          "> " MSG
          I 1 + =
          " SEC  " MSG
          I 2* CDF + @ FLOAT            % PUSH COUNTS IN INTERVAL
          LPSUM @ FLOAT F/              % DIVIDE BY TOTAL COUNTS
          100. F*                       % CONVERT TO PERCENT
          PDEC                  % PRINT OUT RESULT
          " %" MSG CR
        LOOP
        CR CR
        ;


0 `CMAX VARIABLE                        % CONTAINS LARGEST BIN COUNT
0 `BMAX VARIABLE                        % CONTAINS BIN # FOR LARGEST BIN COUNT
0.0 `AVG FVARIABLE                      % CONTAINS MEAN OF D T HISTOGRAM

% CALCULATE THE MEAN OF THE DWELL TIME HISTOGRAM

`MEAN :
        0 CMAX !
        0 BMAX !
        0.0 AVG F!
        75 0
        DO
          I FLOAT                       % MULTIPLY BY BIN #
          0.06667 F*                    % MULTIPLY BY SEC/BIN
          0.03333 F+                    % ADD HALF INTERVAL WIDTH TO GET CENTER
          I 2* DTSUM + @ FLOAT F*       % PUSH COUNTS IN BIN
          AVG F+!                       % ADD TO RUNNING TOTAL
          I 2* DTSUM + @ CMAX @ GT      % TEST FOR LARGEST COUNT
          IF
            I BMAX !                    % STORE BIN # OF LARGEST COUNT
            I 2* DTSUM + @ CMAX !       % STORE LARGEST COUNT
          THEN
        LOOP
        "MODE " MSG
        BMAX @ FLOAT 0.06667 F*         % CALCULATE MODE
        0.03333 F+                      % ADD HALF INTERVAL WIDTH TO GET CENTER
        PDEC
        " SEC" MSG CR
```

```
        "MEAN " MSG
        AVG F@                          % PUSH TOTAL DWELL TIME
        LPSUM @ FLOAT F/                % DIVIDE BY TOTAL # OF DWELLS
        AVG F!                          % STORE MEAN
        AVG F@
        PDEC                            % PRINT OUT RESULT
        " SEC" MSG CR
        ;

0.0 `VAR FVARIABLE

% CALCULATE THE STANDARD DEVIATION

`STDEV :
        0.0 VAR F!
        75 0
        DO
          I FLOAT                       % FIND MIDPOINT OF INTERVAL
          0.06667 F*
          0.03333 F+
          DDUP F*                       % SQUARE INTERVAL MIDPOINT
          I 2* DTSUM + @ FLOAT F*       % MULT BY BIN FREQ
          VAR F+!                       % SUM UP
        LOOP
        "STD DEV " MSG
        VAR F@
        AVG F@ AVG F@ F*                % SQUARE THE MEAN
        LPSUM @ FLOAT F/                % DIVIDE MEAN SQUARED BY N
        F-
        LPSUM @ 1 - FLOAT F/            % DIVIDE BY N-1
        FSQRT                           % TAKE SQARE ROOT OF VARIANCE
        PDEC                            % PRINT RESULT
        " SEC" MSG CR
        ;


`G :
        4 0
        DO
          SUMUP
          "SUBJECT# " MSG
          SUBJECT# @ = CR
          "TASK DIFFICULTY " MSG
          I 1 + = CR CR
          PLOTDT
          5 ( CR )
        LOOP
        12 AX820
        ;

RADIX !
;F
```

```
% File : ACOR
% Author : A.T. Stephens
% Date : 8/81
% Date of Last Revision : 8/25/81

% ACOR plots the autocorrelation of the sequence of instrument #s
% vs time. AUTO requires a file with the number of samples in the first
% two bytes and a sequence of instrument #s sampled at equal time
% intervals in the following bytes.  The FFT of the autocorrelation
% may also be computed and plotted using FFTCOMP. This produces the
% power-spectral density of the instrument scan.  The results of the
% results of the autocorrelation are stored in the array I#FFT which
% is both the input and output array for the FFT program.  To run
% this program ISTIOC, FP, and FFT mult be loaded into the system
% and then, following the steps found in G at the end of this file,
% the autocorrelation and its FFT will be plotted as shown in
% Figures 29-32 and Figures 33-36, respectively.  There appears to
% be a problem with AUTO/FFT-STORE in writing either result onto
% the disk.  This problem has not been solved so use with caution.
% However, the plots may be produced without storing the results.


RADIX @ DECIMAL

0.0 `SUMDT FVARIABLE                    % CONTAINS PRODUCT OF TWO SIGNALS
0 `SHIFT VARIABLE                       % PHASE LAG OF TWO SIGNALS
0.0 `1STSUM FVARIABLE                   % NORM FACTOR (1ST VALUE)
0 `CNT VARIABLE                         % # OF SAMPLES

3000 `I#SEQ ARRAY                       % CONTAINS INST# VS TIME
0 `PTR1 VARIABLE                        % NON-SHIFTED POINTER
0 `PTR2 VARIABLE                        % SHIFTED POINTER
0 `CTR VARIABLE                         % CTR IS DO LOOP COUNTER FOR DOIT
0 `SUBJECT# VARIABLE
0 `RUN# VARIABLE
0 `NUM VARIABLE                         % SELECT INST# FOR AUTOCOR
0 `FLAG VARIABLE                        % TEST FOR OCCURRENCCE OF INST#
4096 `I#FFT ARRAY                       % INPUT AND OUTPUT ARRAY FOR FFT
0.0 `MEAN FVARIABLE                     % MEAN VALUE OF SEQ
0 `SUM VARIABLE                         % SUM OF x(t + shift)
2 `TD ARRAY                             % TASK DIFF FOR EACH LOADING LEVEL

0 TD 0 + B!
1 TD 1 + B!
2 TD 2 + B!
5 TD 3 + B!


% TMSTBL IS A MULTIPLICATION LOOKUP TABLE FOR #`S BETWEEN 0 & 7
0 `TMSTBL VARIABLE
0 B, 0 B, 0 B, 0 B, 0 B, 0 B,
0 B, 1 B, 2 B, 3 B, 4 B, 5 B, 6 B, 7 B,
```

```
0 B, 2 B, 4 B, 6 B, 8 B, 10 B, 12 B, 14 B,
0 B, 3 B, 6 B, 9 B, 12 B, 15 B, 18 B, 21 B,
0 B, 4 B, 8 B, 12 B, 16 B, 20 B, 24 B, 28 B,
0 B, 5 B, 10 B, 15 B, 20 B, 25 B, 30 B, 35 B,
0 B, 6 B, 12 B, 18 B, 24 B, 30 B, 36 B, 42 B,
0 B, 7 B, 14 B, 21 B, 28 B, 35 B, 42 B, 49 B,

% DOIT IS A CODE DEF FOR < DO IFILE GETBYTE OFILE GETBYTE TIMES M+ LOOP >
% IT SAVES APPROX 34 SECS PER 7 LINES PRINTED ON PLOTTER

`DOIT CODE<
        H POP,
        H INX,
        CTR SHLD,
        .                               % DO
          CTR LHLD,
          H DCX,
          H A MOV,
          L ORA,
          CTR SHLD,
          NEXT JZ,
          I#SEQ H LXI,
          XCHG,                         % I# ADDR IN D
          PTR1 LHLD,
          D DAD,                        % ADD POINTER TO I# ADDR
          XCHG,                         % I# ADDR + PTR1 IN D
          D LDAX,                       % INST# IN A
          0 B LXI,
          A C MOV,
          B PUSH,                       % PUSH INST# ON STACK
          XCHG,                         % PUT I# ADDR IN D,E
          PTR2 LHLD,
          D DAD,                        % H,L = I# ADDR + PTR2
          XCHG,
          D LDAX,                       % INST# IN A
          0 B LXI,
          A C MOV,
          B PUSH,                       % PUSH INST# ON STACK
          A ORA,                        % TEST FOR INST# = 0
          IFZ,
            SUM LHLD,
            H INX,                      % SUM x(t + shift)
            SUM SHLD,
          THEN,
          PTR1 LHLD,                    % INCR PTR1
          H INX,
          PTR1 SHLD,
          PTR2 LHLD,                    % INCR PTR2
          H INX,
          PTR2 SHLD,
          B POP,                        % TIMES
          D POP,
          E A MOV,
```

```
          RLC, RLC, RLC,
          C ADD,
          A E MOV,
          0 D MVI,
          TMSTBL H LXI,
          D DAD,
          M E MOV,
          D PUSH,
          B POP,                        % M+
          D POP,
          B A MOV,
          A ORA,
          IFP,
            D DCX,
          THEN,
          H POP,
          B DAD,
          XCHG,
          IFNC,
            H INX,
          THEN,
          D PUSH,
          H PUSH,
        JMP, >                          % LOOP



% TIMES MULTIPLIES TWO BYTES ON TOP OF STACK USING LOOKUP TABLE
% AND RETURNS RESULT

% `TIMES CODE<
%       B POP,
%       D POP,
%       E A MOV,
%       RLC, RLC, RLC,
%       C ADD,
%       A E MOV,
%       0 D MVI,
%       TMSTBL H LXI,
%       D DAD,
%       M E MOV,
%       D PUSH,
%       NEXT JMP, >


% FPR pushes the floating point # in H,L and B,C

. ASSEMBLER<
        H PUSH,                         % PUSH MSWORD ON STACK
        B PUSH,                         % PUSH SIGN/EXP ON STACK
        NEXT JMP, >
`FPR CONSTANT
```

```
% DFLOAT TAKES THE DOUBLE PRECISION NUMBER AT TOP,TOP-1 AND
% RETURNS A SINGLE PRECISION FLOATING POINT NUMBER

`DFLOAT CODE<
    H POP,              % POP HI BYTE OF DP INTEGER
    D POP,              % POP LO BYTE OF DP INTEGER
    32 B LXI,           % SET EXPO TO 32
    H A MOV,            % TEST FOR ZERO DP INTEGER
    L ORA,
    E ORA,
    D ORA,
    IFZ,
     H A MOV,
     .
      A ORA,
      FPR JM,           % CALL FPR WHEN MSB = 1
      E A MOV,          % ROTATE EACH BYTE LEFT 1 BIT
      RAL,
      A E MOV,
      D A MOV,
      RAL,
      A D MOV,
      L A MOV,
      RAL,
      A L MOV,
      H A MOV,
      RAL,
      A H MOV,
      C DCR,
     JMP,
    THEN,
    0 B LXI,
    FPR JMP, >


`SHIFT+2 CODE<
    SHIFT LHLD,
    H INX,
    H INX,
    SHIFT SHLD,
    NEXT JMP, >


% DL INSERTS A DOTTED LINE IN THE PLOT

`DL :
    300 0
    DO
     I 10 /MOD
     UNDER EQZ
     IF
      I -1 AXDOT
```

```
          THEN
        LOOP
        ;


% AUTO TAKES THE SEQUENCE OF INSTRUMENT# VS TIME AND PERFORMS AN
%       AUTOCORRELATION ON THE SEQUENCE. THE RESULTS ARE STORED IN
%       THE ARRAY I#FFT FOR EITHER STORAGE OR PERFORMING AN FFT TO
%       PRODUCE THE POWER-SPECTRAL DENSITY.

`AUTO :
        I#FFT 4096 0FILL
        IFILE OPEN                      % OPEN INST# VS TIME FILE
        0 0 SPOS
        GETBYTE                         % GET # OF SAMPLES AND STORE IN CNT
        GETBYTE
        256 *
        OR
        DROP                            % This and 2048 may be deleted to
        2048 CNT !                      %  allow autocorr on entire run
        2 0 SPOS
        CNT @ 0
        DO                              % TRANSFER INST#s FROM DISK TO MEMORY
          GETBYTE
          NUM @ EQ
          IF
            1 I#SEQ I + B!              % STORE A 1 IF SELECTED INST#
          ELSE
            0 I#SEQ I + B!              % STORE A 0 FOR ALL OTHER INST#S
          THEN
        LOOP
        CLOSE FLUSH
        0
        2048 0
        DO
          I#SEQ I + B@
          +
        LOOP
        FLOAT
        2048. F/
        MEAN F!                         % STORE MEAN OF INST#S
        0 SHIFT !
        AX-TER                          % ENABLE PRINTER
        06 AX820
        16.8-PRT/DEN
        "SUBJECT #" MSG
        SUBJECT# @ = CR
        "TASK DIFFICULTY ." MSG
        TD I + B@ = CR
        "FOR ATTITUDE INDICATOR " MSG CR CR
        30 TAB
        `R MSG CR
        0 TAB
```

```
`0.0 MSG
29 TAB
`0.5 MSG
59 TAB
`1.0 MSG CR
T-TER
<AXLN
4 0
DO
  0. 300. F*
  INTEGER -1 AXDOT
  .5 300. F*
  INTEGER -1 AXDOT
  1.0 300. F*
  INTEGER -1 AXDOT
  AXLN
LOOP
300 0                           % PRINT AXIS LINE
DO
  I -1 AXDOT
LOOP
AXLN
1024 0
DO
  0 0                           % INIT FOR DOUBLE PRECISION ADD
  SUM 0<-                       % ZERO SUM FOR MEAN SUBTRACT
  PTR1 0<-                      % INIT 1ST POINTER
  SHIFT @ PTR2 !                % INIT 2ND POINTER
  CNT @ SHIFT @ -               % PUSH UPPER LIMIT FOR LOOP IN DOIT
  DOIT                          % PUSH 2 INST#S, MULT, ADD TO SUM
                                %  AND REPEAT FOR ALL t
  SHIFT @ EQZ                   % STORE 1ST VALUE FOR NORMALIZING
  IF                            %  FACTOR
    DDUP
    DFLOAT
    SUM @ FLOAT
    MEAN F@
    F*
    F-                          % SUBTRACT MEAN
    CNT @ FLOAT
    F/
    1STSUM F!
  THEN
  DFLOAT                        % SP FLOAT FOR DP INTEGER ON STACK
  SUM @ FLOAT
  MEAN F@
  F*
  F-
  CNT @ FLOAT
  1STSUM F@
  F*
  F/                            % NORMALIZE TO LARGEST VALUE (1ST)
  DDUP
```

```
          I#FFT I 8 * + F!                % STORE VALUE OF AUTOCORR FOR FFT
          I 512 LE                        % PLOT FIRST 512 POINTS OF AUTOCORR
          IF
            I 1+ 50 /MOD UNDER
            EQZ
            IF
              DL                          % PRINT DOTLINE EVERY 10 SEC
            THEN
            300. F*                       % SCALE FOR PLOT
            INTEGER DUP                   % PLOT ONLY IF POSITIVE
            GEZ
            IF
              0
              DO
                I -1 AXDOT
              LOOP
              AXLN
            ELSE
              DROP
              0 -1 AXDOT
              AXLN
            THEN
          ELSE
            2DROP
          THEN
          SHIFT+2                         % SAMPLE EVERY OTHER POINT (5 Hz)
        LOOP
        AXLN>
        ;

% GET/SR# PRINTS THE SUBJECT# AND RUN#

`GET/SR# :
        IFILE OPEN
        0 0 SPOS
        GETBYTE GETBYTE
        256 *
        OR
        SUBJECT# !
        GETBYTE GETBYTE
        256 *
        OR
        RUN# !
        CLOSE
        FLUSH
        ;

% DLFFT PRINTS A DOTTED LINE AT PERIODS OF 10, 5, AND 2 SEC
%       ON THE PLOT OF THE POWER-SPECTRAL DENSITY

`DLFFT :
        375 0
        DO
```

```
          I 10 /MOD
          UNDER EQZ
          IF
            I -1 AXDOT
          THEN
        LOOP
        ;

% FFTCOMP COMPUTES THE FAST FOURIER TRANSFORM (FFT) OF THE
%         AUTOCORRELATION OF THE SCAN PATTERN IN I#FFT.
%         RESULTS ARE LEFT IN I#FFT.

`FFTCOMP :
        I#FFT 10 FFT
        AX-TER
        06 AX820
        16.8-PRT/DEN
        "FFT FOR SUBJECT #" MSG
        SUBJECT# @ = CR
        "TASK DIFFICULTY ." MSG
        TD I + B@ = CR
        "FOR ATTITUDE INDICATOR " MSG CR CR
        35 TAB
        `MAGNITUDE MSG CR
        0 TAB
        `0.0 MSG
        24 TAB
        `0.5 MSG
        49 TAB
        `1.0 MSG
        74 TAB
        `1.5 MSG CR
        T-TER
        <AXLN
        4 0
        DO
          0. 250. F*
          INTEGER -1 AXDOT
          .5 250. F*
          INTEGER -1 AXDOT
          1.0 250. F*
          INTEGER -1 AXDOT
          1.5 250. F*
          INTEGER -1 AXDOT
          AXLN
        LOOP
        375 0
        DO
          I -1 AXDOT
        LOOP
        AXLN
        512 0
        DO
```

```
            I 20 EQ
            I 41 EQ
            OR
            I 102 EQ
            OR
            IF
              DLFFT
            THEN
            I#FFT I 8 * + F@
            250. F*
            INTEGER 0
            DO
              I -1 AXDOT
            LOOP
            AXLN
          LOOP
          AXLN>
          ;

`PUTWORD :
          DUP
          255 AND
          PUTBYTE
          256 /
          PUTBYTE
          ;

`GETWORD :
          GETBYTE
          GETBYTE
          256 *
          OR
          ;


% AUTO/FFT-STORE STORES THE CONTENTS OF I#FFT ON THE DISK

`AUTO/FFT-STORE :
          IFILE WOPEN
          0 0 SPOS
          1024 0
          DO
            I#FFT I 8 * + F@
            SWAP
            PUTWORD PUTWORD
          LOOP
          SHRINK
          CLOSE
          FLUSH
          ;

% GETAUTO RETRIEVES A STORED AUTOCORRELATION FROM THE DISK FOR
%          FOR COMPUTATION OF THE FFT.
```

```
`GETAUTO :
        I#FFT 4096 0FILL
        IFILE OPEN
        1024 0
        DO
          GETWORD GETWORD
          I#FFT I 8 * + F!
        LOOP
        CLOSE FLUSH
        ;

% G PLOTS THE AUTOCORRELATION AND POWER-SPECTRAL DENSITY FOR THE
%   FOUR DATA FILES GIVEN. RESULTS OF THIS PROGRAM MAY BE FOUND IN
%   FIGURES 29-32 AND FIGURES 33-36.

`G :
        4 0
        DO
          12 TRACK
          DUP
          GET/SR#                             % GET SUBJECT AND RUN #
          0 TRACK
          AUTO                                % PLOT AUTOCOVARIANCE
          12 AX820                            % SEND FORM FEED
          FFTCOMP                             % COMPUTE & PLOT POWER-SPECTRAL DEN
          12 AX820                            % SEND FORM FEED
        LOOP
        12 AX820                              % SEND FORM FEED
        ;

`G1 : 3DROP DROP ;


RADIX !
;F
```

```
% FILE: H.HR

% Author : A.T. Stephens
% Date : 7/81
% Date of Last Revision : 8/5/81

% H.HR computes the entropy, entropy rate, and the redundancy factor for
% the pilot mental workload data obtained in the NASA/Langley experiments.
% H.HR differs from the other methods in that during the entropy computation,
% the maximum occurring sequence from each pass is deleted from the data.
% Therefore, NEVER RUN SETUP ON THE ORIGINAL DATA FILE !!  Temporary files
% are supplied for this purpose.  This deletion method removes the correlated
% sequences found in the old window method.  To run H.HR, follow the steps
% found in G at the end of this file.  G is set up to operate on a file
% containing all of the data file names in the order of desired presentation.
% The file GRIND contains these names for the 2/81 data files.  This file
% requires that the words in the experiment controlling files be defined.
% HHIST2 should be used.

RADIX @ DECIMAL

2 `N VARIABLE                           % SEQUENCE LENGTH
N @ `IA ARRAY                           % INSTRUMENT ARRAY
N @ `DTA ARRAY                          % DWELL TIME ARRAY
0 `DTSUM VARIABLE                       % CONTAINS SUM OF DT FOR CURRENT SEQ
0 `ADDR VARIABLE                        % CONTAINS ADDRESS OF CURRENT SEQ
0.0 `HCR FVARIABLE                      % ENTROPY(H) RATE (USING Hcorr)
0.0 `HOR FVARIABLE                      % ENTROPY(H) RATE (USING Ho)
0.0 `HO FVARIABLE                       % OBSERVED ENTROPY
0.0 `HCORR FVARIABLE                    % ENTROPY CORRECTED FOR SEQ LENGTH
8 `FACTOR VARIABLE                      % USED IN CONVERSION FROM INST# TO ADDR
0 `PAC/CNT VARIABLE                     % NUMBER OF N LENGTH SEQS IN RUN
`AFILE FILE                             % CREATE NEW FILE TABLE
0 `FLAG VARIABLE                        % FLAG FOR END OF LOOP TEST
1 `FLAG1 VARIABLE                       % PRINT LENGTH 1 SEQS = 1
                                        % OMIT LENGTH 1 SEQS  = 0
0 `#DIFPAC VARIABLE                     % # OF DIFFERENT SEQS
64 `8**N VARIABLE                       % 8 RAISED TO THE Nth POWER
0 `TD VARIABLE                          % TASK DIFFICULTY


5 `TOPN VARIABLE                        % # OF SEQS INCLUDED IN CALCULATIONS
TOPN @ `TTEN ARRAY                      % CONTAINS TOP TEN SEQS
TOPN @ 2 * `TTCNTS ARRAY                % CONTAINS FREQ OF TOP TEN SEQS
0 `OFFSET VARIABLE                      % OFFSET INTO TTCNTS FOR EACH RUN
4 `#SEQS ARRAY                          % CONTAINS TOTAL COUNTS FOR EACH RUN


HBUF2 VHBUF !
42. Q F!
`INIT-VAR CODE<
        0 H LXI,
```

```
    DTSUM SHLD,
    ADDR SHLD,
    8 H LXI,
    FACTOR SHLD,
    NEXT JMP, >


`-NSP :
    `TEMP4
    IFILE
    OPEN
    #FIX @ 0
    DO
     GETBYTE
     0 EQ
     IF
      GPOS
      SWAP
      1-
      SWAP
      SPOS
      -1 PUTBYTE
     THEN
    LOOP
    IFILE
    CLOSE
    FLUSH
    ;


`I#FILE :
    `TEMP1 IFILE OPEN
    `TEMP4 OFILE WOPEN
    IFILE
    0 0 SPOS
    GETBYTE
    OFILE
    0 0 SPOS
    PUTBYTE
    #FIX @ 1- 0
    DO
     IFILE
     GETBYTE
     GETBYTE
     2DROP
     GETBYTE                % GET INSTR#
     OFILE
     PUTBYTE                % STORE INSTR#
    LOOP
    IFILE
    CLOSE
    OFILE
    SHRINK
```

```
        CLOSE
        FLUSH
        ;


`-1? :
        DUP
        255 EQ                              % TEST FOR -1 IN BYTE
        IF
          DROP
          -1                                % PUSH -1 ONTO STACK
        THEN
        ;

`CONVERT :
        INIT-VAR                            % INITIALIZE VARIABLES
        N @ 0
        DO
          IA I 2* + @ DUP                   % PUSH & DUP FIRST INST#
          -1? GEZ                           % TEST FOR -1 (NO INSTR#)
          IF
            FACTOR @ *                      % CONVERT TO BASE OF # OF INSTR
            ADDR +!
            FACTOR @ 8 /
            FACTOR !
          ELSE
            DROP
            -1 ADDR !
            EXIT
          THEN
        LOOP
        ;

`ADDR/TST :
        ADDR @ MAXSEQ @ EQ                  % TEST FOR MAXSEQ
        IF
          GPOS
          SWAP                              % REVERSE HI/LO WORD ADDRESS
          N @ -                             % BACK UP THE LENGTH OF THE SEQ
          SWAP
          SPOS
          N @ 0
          DO
            -1 PUTBYTE                      % REPLACE USED SEQ WITH -1
          LOOP
        THEN
        ;

`START :
        N @ 0
        DO
          GETBYTE                           % GET INSTR#
          IA I 2* + !                       % STORE INSTR#
```

```
        LOOP
        CONVERT                             % CHANGE INSTR#S TO ADDR CODE
        ;

`PUTONE :
        #FIX @ N @ - 0
        DO
          N @ 1- 0
          DO
            IA I 2* 2 + + @                 % SHIFT INSTR#S FOR NEW SEQ
            IA I 2* + !
          LOOP
          GETBYTE                           % GET NEXT INSTR#
          IA N @ 1- 2* + !                  % STORE NEW INSTR#
          CONVERT                           % CHANGE INSTR#S TO ADDR CODE
          ADDR/TST                          % TEST FOR SELECTED ADDR
        LOOP
        ;

% FINDMAX is used in place of H-C-AUX for seq length 1
% It searches HBUF1 for the largest value and stores the counts in MAXCNT
% and the seq# in MAXSEQ.

 `FINDMAX :
        0 MAXSEQ !
        0 MAXCNT !
        8 0
        DO
          HBUF1 I 2* + @
          MAXCNT @
          GT
          IF
            HBUF1 I 2* + @
            MAXCNT !
            I MAXSEQ !
          THEN
        LOOP
        ;


`SETUP :
        IFILE                               % OPEN INST/DT FILE
        OPEN
        `TEMP2 6 CCONT
        `TEMP2 AFILE OPEN
        `TEMP3 OFILE WOPEN
        OFILE
        8**N @ 2* 0
        DO
          I 0 SPOS                          % ZERO DT FILE
          0 PUTBYTE
        LOOP
        AFILE
```

```
3072 0
DO
  I 0 SPOS
  0 PUTBYTE
LOOP
0 0 SPOS
IA N @ 0FILL
DTA N @ 0FILL
IFILE                           % SELECT INST/DT FILE
N @ 0
DO
  GETBYTE                       % PUSH INST#
  IA I 2* + !                   % STORE INST#
  GETWORD                       % PUSH DWELL TIME
  DTA I 2* + !                  % STORE DWELL TIME
LOOP
INIT-VAR                        % INITIALIZE VARIABLES
N @ 0
DO
  IA I 2* + @ FACTOR @ *        % CONVERT INST#S TO ADDR
  ADDR +!                       % STORE ADDRESS
  FACTOR @ 8 / FACTOR !         % COMPUTE NEW FACTOR
  DTA I 2* + @                  % PUSH DT
  DTSUM +!                      % STORE DT SUM
LOOP
OFILE                           % SELECT DT FILE
ADDR @ 2* 0 SPOS                % POSITION FILE POINTER
DTSUM @                         % PUSH CURRENT DT SUM
PUTWORD                         % STORE DT SUM IN DT FILE
#FIX @ N @ - 0
DO
  N @ 1- 0
  DO
    IA I 2* 2 + + @ IA I 2* + ! % SHIFT INST#S & DT FOR NEW SEQ
    DTA I 2* 2 + + @ DTA I 2* + !
  LOOP
  IFILE                         % SELECT INST/DT FILE
  GETBYTE                       % PUSH NEXT INST#
  IA N @ 1- 2* + !              % STORE NEW INST#
  GETWORD                       % PUSH DWELL TIME
  DTA N @ 1- 2* + !             % STORE DWELL TIME
  INIT-VAR                      % INITIALIZE VARIABLES
  N @ 0
  DO
    IA I 2* + @ FACTOR @ *      % CONVERT INST#S TO ADDR
    ADDR +!                     % STORE ADDRESS
    FACTOR @ 8 / FACTOR !       % COMPUTE NEW FACTOR
    DTA I 2* + @                % PUSH DT
    DTSUM +!                    % STORE DT SUM
  LOOP
  OFILE                         % SELECT DT FILE
  ADDR @ 2* 0 SPOS              % POSITION FILE POINTER
  GETWORD                       % PUSH TOTAL DT SUM
```

```
  DTSUM @ +                     % ADD CURRENT DT SUM FOR SEQ
  ADDR @ 2* 0 SPOS              % POSITION FILE POINTER
  PUTWORD                       % STORE NEW TOTAL DT SUM
LOOP
IFILE                           % SELECT INST/DT FILE
CLOSE FLUSH                     % CLOSE INST/DT FILE
`TEMP4 IFILE OPEN               % OPEN INSTR# FILE
0 #DIFPAC !
BEGIN
  #DIFPAC 1+!
  -1 FLAG !                     % SET END LOOP FLAG
  8**N @ H-C-AUX                % FIND & STORE MAX OCCURRING SEQ
                                % FINDMAX FOR N=1, # H-C-AUX OTHERWISE
  AFILE                         % SELECT SEQ/CNT FILE
  MAXSEQ @                      % PUSH MAX OCCURRING SEQ
  PUTWORD                       % PUT MAX OCCURRING SEQ IN FILE
  MAXCNT @                      % PUSH MAX COUNTS
  DUP
  #SEQS TD @ 2* + +!            % INCR TOTAL # OF SEQS
  PUTWORD                       % PUT MAX COUNTS IN FILE
  IFILE                         % SELECT INST/DT FILE
  0 0 SPOS                      % MOVE POINTER TO START OF FILE
  START                         % READ IN FIRST N INSTR#S
  ADDR @ MAXSEQ @ EQ            % TEST FOR MAX SEQ
  IF
    0 0 SPOS                    % BACK SPACE POINTER
    N @ 0
    DO
      -1 PUTBYTE                % REPLACE INSTR# WITH -1
    LOOP
  THEN
  PUTONE                        % REPLACE ALL MAXSEQ WITH -1
  Z-HBUF                        % ZERO HBUF*
  0 0 SPOS                      % MOVE POINTER TO START OF FILE
  START                         % READ IN 1ST N INSTR#S
  ADDR @ GEZ                    % TEST FOR UNUSED SEQ
  IF
    VHBUF @ ADDR @ + 1+!        % INCR SEQ COUNT
                                % INSERT 2* FOR SEQ LENGTH 1 ONLY
    0 FLAG !
  THEN
  #FIX @ N @ - 0
  DO
    N @ 1- 0
    DO
      IA I 2* 2 + + @           % SHIFT INSTR#S FOR NEW SEQ
      IA I 2* + !
    LOOP
    GETBYTE                     % GET NEXT INSTR#
    IA N @ 1- 2* + !            % STORE NEW INSTR#
    CONVERT                     % CHANGE INSTR# TO ADDR CODE
    ADDR @ GEZ                  % TEST FOR UNUSED SEQ
    IF
```

```
              VHBUF @ ADDR @ + 1+!     % INCR SEQ COUNT
                                       % INSERT 2* FOR SEQ LENGTH 1 ONLY
              0 FLAG !                 % STAY IN B-E LOOP
            THEN
          LOOP
          FLAG @ -1 EQ                 % TEST END LOOP FLAG
        END
        TD @ EQZ                       % IF 1ST RUN, STORE TOPN SEQS AND
        IF                             % CNTS FOR % OCCUR PLOT
          AFILE
          0 0 SPOS
          TOPN @ 0
          DO
            GETWORD
            TTEN I 2* + !
            GETWORD
            TTCNTS I + B!
          LOOP
        THEN
        IFILE
        CLOSE
        FLUSH
        ;



`E/CALC :
        0.0 HO F!
        0.0 HCORR F!
        0.0 HOR F!
        0.0 HCR F!
        AFILE                          % SELECT SEQ/CNT FILE
        0 0 SPOS                       % MOVE POINTER TO START OF FILE
        AX-TER                         % ENABLE PRINTER
        "SUBJ# " MSG
        SUBJECT# @ = CR                % PRINT SUBJ#
        "RUN# " MSG
        MAN/TRIAL# @ =                 % PRINT RUN#
        CR CR
        "SEQUENCE" MSG
        27 TAB
        "CNTS" MSG
        36 TAB
        "p(i)" MSG
        47 TAB
        "Ho" MSG
        58 TAB
        "Hcorr" MSG
        69 TAB
        "HRATE" MSG
        CR CR
        #FIX @ N @ /                   % COMPUTE MAX # OF SEQS IN RUN
        PAC/CNT !                      % STORE MAX # OF SEQS IN RUN
```

```
#DIFPAC @ 0
DO
  AFILE                                % SELECT SEQ/CNT FILE
  GETWORD                              % GET SEQ CODE
  MAXSEQ !                             % STORE SEQ CODE
  GETWORD                              % GET SEQ COUNTS
  MAXCNT !                             % STORE SEQ COUNTS
  FLAG1 @ 0 EQ                         % PRINT SEQS OF LENGTH 1 ?
  MAXCNT @ 1 EQ                        % COUNTS = 1 ?
  AND
  IF
    EXIT                               % IF SO, END LOOP
  THEN
  OFILE                                % SELECT DT FILE
  MAXSEQ @ 2* 0 SPOS                   % POSITION POINTER AT MAXSEQ
  GETWORD                              % GET DWELL TIME SUM
  DTSUM !                              % STORE DWELL TIME SUM
  N @ SEQLENGTH !
  SEQ-MAP                              % PRINT OUT SEQ
  27 TAB
  MAXCNT @ DUP =                       % PUSH, DUP, & PRINT SEQ COUNTS
  36 TAB
  FLOAT                                % FLOAT MAXCNT
  PAC/CNT @ FLOAT                      % PUSH TOTAL SEQ COUNT
  F/                                   % COMPUTE PERCENTAGE
  DDUP PDEC                            % DUP & PRINT p(i)
  47 TAB
  DDUP
  FLOG2
  F*
  FMINUS                               % = -p(i)log2p(i)
  DDUP PDEC                            % PRINT Ho
  58 TAB
  DDUP
  DDUP
  HO F+!
  DTSUM @ FLOAT
  30. F/                               % CONVERT TOTAL DWELL TIME TO SECONDS
  MAXCNT @ FLOAT F/                    % COMPUTE AVERAGE DWELL TIME
  F/                                   % COMPUTE ENTROPY RATE (HOR)
  HOR F+!                              % INCR Ho RATE
  Q F@ FLOG2                           % LOG2(Q)
  PAC/CNT @ FLOAT
  FLOG2                                % LOG2(PAC/CNT)
  F/                                   % LOG2(Q)/LOG2(PAC/CNT)
  F*                                   % = (-p(i)LOG2p(i))(LOG2(Q)/LOG2(P/C)
  DDUP PDEC                            % PRINT Hcorr
  69 TAB
  DDUP
  HCORR F+!                            % INCR Hcorr
  DTSUM @ FLOAT
  30. F/                               % CONVERT TOTAL DWELL TIME TO SECONDS
  MAXCNT @ FLOAT F/                    % COMPUTE AVERAGE DWELL TIME
```

```
                  F/                              % COMPUTE ENTROPY RATE (HCR)
                  DDUP PDEC
                  HCR F+!                         % INCR Hcorr RATE
                  CR
                LOOP
                CR CR
                "Ho " MSG
                HO F@ PDEC CR
                "Ho RATE " MSG
                HOR F@ PDEC CR
                "Hcorr " MSG
                HCORR F@ PDEC CR
                "Hcorr RATE " MSG
                HCR F@ PDEC CR
                "RF(Ho) " MSG
                HO F@
                PAC/CNT @ FLOAT FLOG2
                F/
                FMINUS
                1. F+
                PDEC CR
                "RF(Hcorr) " MSG
                HCORR F@
                PAC/CNT @ FLOAT FLOG2
                F/
                FMINUS
                1. F+
                PDEC CR
                "TOTAL # SEQS " MSG
                #SEQS TD @ 2* + @
                = CR
                "MAX POSS # OF SEQS " MSG
                PAC/CNT @ = CR
                "# DIFF SEQS " MSG
                #DIFPAC @ = CR
                "% OF POSS SEQS USED " MSG
                #SEQS TD @ 2* + @ FLOAT
                PAC/CNT @ FLOAT
                F/
                100. F*
                PDEC CR
%               "# OF BLINKS " MSG
%               #BLINK @ = CR
                CR CR CR
                T-TER
                AFILE
                CLOSE
                OFILE
                SHRINK
                CLOSE
                FLUSH
                ;
```

```
15 `DLL VARIABLE                   % DASHED LINE LENGTH FOR TOTAL % PLOT
5 `SPL VARIABLE                            % LENGTH OF SPACES FOR DASHED LINE
0 `DLC VARIABLE                            % CURRENT LENGTH OF LINE
0 `SPC VARIABLE                            % CURRENT NUMBER OF SPACES
4 `TOT% ARRAY                     % CONTAINS SUM OF % FOR TOP TEN SEQS
TOPN @ 4 * `HEIGHT ARRAY                   % CONTAINS Y VALUE OF POINTS
0 `POINT VARIABLE                          % CONTAINS CURRENT Y VALUE




% PLOTOP PLOTS THE PERCENTAGE OCCURRENCE VS LOADING LEVEL
% FOR TOP TEN SEQUENCES

`PLOTOP :
        0 DLC !
        0 SPC !
        0 POINT !
        HEIGHT TOPN @ 4 * 0FILL
        TOT% 4 0FILL
        AX-TER                    % ENABLE PRINTER
        06 AX820
        10-PRT/DEN
        21 TAB
        `PERCENT MSG CR CR
        `0 MSG
        22 TAB
        `50 MSG
        45 TAB
        `100 MSG CR
        <AXLN
        400 0
        DO
          I -1 AXDOT
        LOOP
        AXLN
        4 0
        DO
          TOPN @ 0
          DO
            TTCNTS I + TOPN @ J * + B@         % PUSH # OF COUNTS
            FLOAT
            #SEQS J 2* + @ FLOAT F/            % DIVIDE BY TOTAL COUNTS
            400. F*                            % SCALE FOR PERCENT PLOT
%           10. F*                             % TOP OF SCALE = 10 %
            INTEGER DUP
            HEIGHT I 2* + TOPN @ 2* J * + !        % STORE Y VALUE
%           5 /                                % TOP OF SCALE = 50% FOR TOT%
```

```
    TOT% J 2* + +!                         % INCR TOTAL %
  LOOP
LOOP
3 0
DO
  112 0
  DO
    DLC @ DLL @ LT                         % TEST FOR DESIRED LINE LENGTH
    IF
      TOT% J 2* + 2 + @                    % PUSH Y VAL @ X+1
      TOT% J 2* + @                        % PUSH Y VAL @ X
      DUP POINT !                          % STORE Y VAL @ X
      - FLOAT                              % FIND DIFF AND FLOAT RESULT
      112. F/                              % DETERMINE SLOPE OF LINE
      I 1 + FLOAT F* INTEGER               % DETERMINE NEW Y VAL
      POINT +!                             % STORE NEW Y VAL
      POINT @ -1 AXDOT                     % STORE NEW Y VAL FOR PLOT
      DLC 1+!                              % INCR COUNTER
    ELSE

      SPC @ SPL @ LT                       % TEST FOR DESIRED # OF SPACES
      IF
        SPC 1+!                            % INCR COUNTER
      ELSE
        0 SPC !                            % ZERO SPACE COUNTER
        0 DLC !                            % ZERO LINE COUNTER
      THEN
    THEN
    TOPN @ 0
    DO
      HEIGHT I 2* + TOPN @ 2* K 1 + * + @    % RETRIEVE Y VAL @ X+1
      HEIGHT I 2* + TOPN @ 2* K * + @        % RETRIEVE Y VAL @ X
      POINT !                              % STORE Y VAL @ X
      POINT @ - FLOAT                      % SUBT X FROM X+1 (Y VAL)
      112. F/                              % DETERMINE SLOPE OF LINE
      J 1 + FLOAT F* INTEGER               % DETERMINE NEW Y VAL
      POINT +!                             % STORE NEW Y VAL
      POINT @ -1 AXDOT                     % STORE NEW Y VAL FOR PLOT
    LOOP
    AXLN
  LOOP
  DOTLINE
LOOP
AXLN>
4 ( CR )
06 AXB20                                   % SELECT FIXED CHAR SPACING
TOPN @ 0
DO
  I 2* TTEN + @                            % PUSH SEQ#
  MAXSEQ !                                 % STORE IN MAXSEQ
  SEQ-MAP                                  % PRINT CORRESPONDING INST NAME
  CR
LOOP
```

```
        ;


`SEQSEARCH :
        `TEMP2 AFILE OPEN
        TOPN @ 0
        DO
          0 0 SPOS
          #DIFPAC @ 0
          DO
            GETWORD
            TTEN J 2* + @
            EQ
            IF
              GETWORD
              TTCNTS TD @ TOPN @ * + J + B!
            ELSE
              GETWORD
              DROP
            THEN
          LOOP
        LOOP
        AFILE CLOSE
        FLUSH
        ;



`G :
        0 TD !
        TTCNTS TOPN @ 2* 0FILL
        #SEQS 4 0FILL
        4 0
        DO
          12 TRACK
          RDDATA
          18 TRACK
          `TEMP1 H/DT-SORT
          I#FILE                        % SET UP INSTR# FILE
          -NSP                          % REMOVE NSP FROM COMPUTATION
          `TEMP1 SETUP
          E/CALC
          SEQSEARCH
          TD 1+!
          `TEMP1 DELETE                 % TEMP1 => INST/DT FILE
          `TEMP2 DELETE                 % TEMP2 => SEQ/CNT FILE
          `TEMP3 DELETE                 % TEMP3 => DT FILE
          `TEMP4 DELETE                 % TEMP4 => INSTR# FILE
          FLUSH
        LOOP
        PLOTOP                          % PLOT TOPN SEQS VS TD
        AX-TER
        FF
```

```
        CR CR
        T-TER
        ;

`Gl : 3DROP DROP ;

RADIX !
;F
```

APPENDIX B

PILOT QUESTIONNAIRE

1. Name ______________________________

2. Age ______

3. Left handed _______ Right handed ________ Either ________

4. Have you had any previous flying experience yes
   no

5. Total flight time: ______ Hrs.

6. Time in this simulator: _________ Hrs.

7. Time in aircraft similar to simulator _______ Hrs.

8. Total time in past 30 days _________ Hrs.

9. Part of time given in #6 and #7 obtained in past 30 days _______ Hrs.

APPENDIX C

Statistical Analysis

The statistical analyses performed on the data involved a two-way analysis of variance, testing differences between sample means using Student`s T-tests, and fitting curves to the data using multiple regression. The analyses were performed using the MINITAB Statistical Computing Package developed at Pennslyvania State University. The output from the following analyses is included in this Appendix:

1) Two-way analysis of variance on Hrate, with first factor being task difficulty, and the second factor being skill.

2) Student`s T-test for differences of means for each combination of different loading levels of Hrate. Labels indicate the two loading levels being considered. For example, 1-2 is the test for differences between the lowest level of imposed task difficulty and the second level.

3) Regression of Hrate on EXP(-TD)

4) Multiple regression of EXP((TD/Skill)**2) on four Predictors, the change from no-loading case for #TP, RMS/GS, RMS/LOC, %PWR/GS

Twoway Analysis of Variance on Hrate, with first factor being Task Difficulty, and second factor being Skill

```
ANALYSIS OF VARIANCE

DUE TO          DF         SS     MS=SS/DF
RTD              3     .30441      .10147
RSKI             8     .34714      .04339
ERROR           24     .15842      .00660
TOTAL           35     .80998

OBSERVATIONS
ROWS ARE LEVELS OF RTD    COLS ARE LEVELS OF RSKI

                 1          2          3          4          5          6
    1       1.0330      .7690     1.2160     1.0080      .8430      .8510
    2        .8140      .7210     1.0090      .9590      .7030      .8360
    3        .8580      .6750      .8670      .8700      .6510      .7720
    4        .7920      .6400      .5650      .9010      .4740      .7040
COL.
MEANS        .8742      .7012      .9142      .9345      .6677      .7907
                                               ROW
                 7          8          9     MEANS
    1        .7870      .8510      .8480      .9118
    2        .6520      .8120      .7630      .8077
    3        .7010      .6960      .5900      .7422
    4        .6410      .7030      .5250      .6606
COL.
MEANS        .6952      .7655      .6815      .7806

POOLED ST. DEV. =       .0812

INDIVIDUAL 95 PERCENT C. I. FOR LEVEL MEANS OF RTD
(BASED ON POOLED STANDARD DEVIATION)
       +---------+---------+---------+---------+---------+---------+
1                                           I******I******I
2                              I******I******I
3                      I******I******I
4            I******I******I
       +---------+---------+---------+---------+---------+---------+
     .560      .640      .720      .800      .880      .960     1.040

INDIVIDUAL 95 PERCENT C. I. FOR LEVEL MEANS OF RSKI
(BASED ON POOLED STANDARD DEVIATION)
       +---------+---------+---------+---------+---------+---------+
1                                    I*******I********I
2                  I*******I********I
3                                        I*******I********I
4                                          I*******I********I
5              I********I*******I
6                           I*******I*******I
7                 I********I*******I
8                        I********I*******I
9                I*******I********I
       +---------+---------+---------+---------+---------+---------+
      .50       .60       .70       .80       .90      1.00      1.10
```

Student's t-test for difference of means for each combination of different loading levels of Hrate.

```
TTEST MU=0.0, DATA IN C20

   1-2        N =    9      MEAN =       .10078     ST.DEV. =      .0695

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  4.349
   THE TEST IS SIGNIFICANT AT    .0024


TTEST MU=0.0 , DATA IN C21

   1-3        N =    9      MEAN =       .16622     ST.DEV. =      .0888

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  5.614
   THE TEST IS SIGNIFICANT AT    .0000


TTEST MU=0.0, DATA IN C22

   1-4        N =    9      MEAN =       .24689     ST.DEV. =      .177

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  4.179
   THE TEST IS SIGNIFICANT AT    .0031


TTEST MU=0.0, DATA IN C23

   2-3        N =    9      MEAN =      .065444     ST.DEV. =      .0760

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  2.584
   THE TEST IS SIGNIFICANT AT    .0324


TTEST MU=0.0, DATA IN C24

   2-4        N =    9      MEAN =       .14611     ST.DEV. =      .138

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  3.170
   THE TEST IS SIGNIFICANT AT    .0132


TTEST MU=0.0, DATA IN C25

   3-4        N =    9      MEAN =      .080667     ST.DEV. =      .102

   TEST OF MU =        .0000 VS. MU N.E.       .0000
   T =  2.367
   THE TEST IS SIGNIFICANT AT    .0455
```

Regression of Hrate on exp(-TD)

```
                                             ST. DEV.    T-RATIO =
      COLUMN      COEFFICIENT                OF COEF.    COEF/S.D.
NOCONSTANT
X1      E-TD                .9419              .0376         25.06

THE ST. DEV. OF Y ABOUT REGRESSION LINE IS
S =      .1404
WITH (  20- 1) =  19 DEGREES OF FREEDOM

ANALYSIS OF VARIANCE

 DUE TO      DF          SS       MS=SS/DF
REGRESSION    1    12.38042       12.38042
RESIDUAL     19     0.37470        0.01972
TOTAL        20    12.75512

             X1           Y     PRED. Y    ST.DEV.
ROW        E-TD          HR      VALUE    PRED. Y    RESIDUAL       ST.RES.
  1        0.82      0.8580     0.7712     0.0308     0.0868          0.63
  2        0.61      0.7920     0.5713     0.0228     0.2207          1.59
  3        0.90      1.0090     0.8523     0.0340     0.1567          1.15
  4        0.82      0.8670     0.7712     0.0308     0.0958          0.70
  5        0.61      0.5650     0.5713     0.0228    -0.0063         -0.05
  6        0.90      0.9590     0.8523     0.0340     0.1067          0.78
  7        0.82      0.8700     0.7712     0.0308     0.0988          0.72
  8        0.61      0.9010     0.5713     0.0228     0.3297          2.38R
  9        1.00      0.8430     0.9419     0.0376    -0.0989         -0.73
 10        0.90      0.7030     0.8523     0.0340    -0.1493         -1.10
 11        0.82      0.6510     0.7712     0.0308    -0.1202         -0.88
 12        1.00      0.8510     0.9419     0.0376    -0.0909         -0.67
 13        0.90      0.8360     0.8523     0.0340    -0.0163         -0.12
 14        0.82      0.7720     0.7712     0.0308     0.0008          0.01
 15        0.61      0.7040     0.5713     0.0228     0.1327          0.96
 16        1.00      0.7870     0.9419     0.0376    -0.1549         -1.14
 17        0.90      0.6520     0.8523     0.0340    -0.2003         -1.47
 18        0.82      0.7010     0.7712     0.0308    -0.0702         -0.51
 19        0.61      0.6410     0.5713     0.0228     0.0697          0.50
 20        1.00      0.8510     0.9419     0.0376    -0.0909         -0.67

R DENOTES AN OBS. WITH A LARGE ST. RES.


(X-PRIME X)INVERSE

                1
   1    .0716590
```

Multiple Regression of exp((TD/Skill)**2) on 4 Predictors, the change from no-loading case for #TP, RMS/GS, RMS/LOC, %PWR/GS

```
                                          ST. DEV.    T-RATIO =
       COLUMN      COEFFICIENT            OF COEF.    COEF/S.D.
        --              1.4483              .1456          9.95
X1     #TD               .0351              .0085          4.12
X2     RGD               .1765              .0321          5.50
X3     RLD              -.0366              .0078         -4.69
X4     %GSD              .0377              .0128          2.93

THE ST. DEV. OF Y ABOUT REGRESSION LINE IS
S =      .5209
WITH (  20- 5) =  15 DEGREES OF FREEDOM

R-SQUARED = 76.6 PERCENT
R-SQUARED = 70.4 PERCENT, ADJUSTED FOR D.F.

ANALYSIS OF VARIANCE

 DUE TO      DF          SS       MS=SS/DF
REGRESSION    4     13.3306         3.3326
RESIDUAL     15      4.0700         0.2713
TOTAL        19     17.4006

FURTHER ANALYSIS OF VARIANCE
SS EXPLAINED BY EACH VARIABLE WHEN ENTERED IN THE ORDER GIVEN

 DUE TO      DF          SS
REGRESSION    4     13.3306
#TD           1      1.2222
RGD           1      4.3520
RLD           1      5.4223
%GSD          1      2.3341
```

```
              X1          Y    PRED. Y    ST.DEV.
ROW          #TD       TERM      VALUE    PRED. Y    RESIDUAL      ST.RES.
  1          9.5      1.009      0.660      0.288       0.349        0.80
  2         40.1      1.031      1.333      0.335      -0.302       -0.76
  3          2.1      1.008      1.811      0.317      -0.803       -1.94
  4          0.0      1.012      1.011      0.206       0.001        0.00
  5         67.0      1.560      1.095      0.412       0.465        1.46 X
  6          7.1      1.001      0.813      0.223       0.188        0.40
  7          2.4      1.022      0.583      0.286       0.439        1.01
  8         39.3      1.007      2.021      0.355      -1.014       -2.66R
  9          0.0      1.085      1.448      0.146      -0.363       -0.73
 10          3.8      1.765      1.815      0.155      -0.050       -0.10
 11          1.1      2.474      2.010      0.199       0.464        0.96
 12          0.0      1.076      1.448      0.146      -0.372       -0.74
 13          1.9      1.107      1.358      0.152      -0.251       -0.50
 14          5.3      1.326      0.997      0.176       0.329        0.67
 15         36.6      1.829      1.858      0.319      -0.029       -0.07
 16          0.0      1.379      1.448      0.146      -0.070       -0.14
 17          7.5      3.841      2.882      0.240       0.959        2.07R
 18         10.5      2.382      2.669      0.275      -0.287       -0.65
 19         50.0      4.363      4.120      0.391       0.243        0.71 X
 20          0.0      1.554      1.448      0.146       0.106        0.21

R DENOTES AN OBS. WITH A LARGE ST. RES.
X DENOTES AN OBS. WHOSE X VALUE GIVES IT LARGE INFLUENCE.


(X-PRIME X)INVERSE

                 0           1           2           3           4
  0       .0781579
  1      -.0015565    .0002682
  2      -.0012401   -.0000159    .0037929
  3      -.0003402   -.0001720   -.0001682    .0002247
  4      -.0009825    .0000328    .0006742   -.0000288    .0006078
```

Plot of Standard Residuals vs Predicted Y-Value for Model Regression