NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE EFFECT OF THREE VARIABLES ON
SYNTHETIC SPEECH INTELLIGIBILITY IN NOISY
ENVIRONMENTS

by

Joyce C. Munlin

March 1990

Thesis Advisor: Gary K. Poock
Co-Advisor: Judith H. Lind

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification UNCLASSIFIED	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report Approved for public release; distribution is unlimited.
2b Declassification/Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School 6b Office Symbol *(If Applicable)* 365	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization 8b Office Symbol *(If Applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers Program Element Number / Project No / Task No / Work Unit Accession No

11 Title *(Include Security Classification)* THE EFFECT OF THREE VARIABLES ON SYNTHETIC SPEECH INTELLIGIBILITY IN NOISY ENVIRONMENTS

12 Personal Author(s) Munlin, Joyce C.

13a Type of Report	13b Time Covered	14 Date of Report *(year, month,day)*	15 Page Count
Master's Thesis	From To	1990, March	75

16 Supplementary Notation The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Synthetic Speech, Command and Control

19 Abstract *(continue on reverse if necessary and identify by block number*

Military Command and Control (C2) requires easy access to information needed for the commander's situation assessment and direction of troops. Providing this information via synthetic speech is a viable alternative, but additional information is required before speech systems can be implemented for C2 functions. An experiment was conducted to study several factors which may affect the intelligibility of synthetic speech. The factors examined were: 1) speech rate; 2) synthetic speech messages presented at lower, the same, and higher frequencies than background noise frequency; 3) voice richness; and 4) interactions between speech rate, voice fundamental frequency, and voice richness. Response latency and recognition accuracy were measured. Results clearly indicate that increasing speech rate leads to an increase in response latency and a decrease in recognition accuracy, at least for the novice user. No effect of voice fundamental frequency or richness was demonstrated.

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users	21 Abstract Security Classification Unclassified
22a Name of Responsible Individual J. Lind	22b Telephone *(Include Area code)* (408) 646-2543 22c Office Symbol ORLi

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited

The Effect of Three Variables on Synthetic Speech Intelligibility in Noisy Environments

by

Joyce C. Munlin
Lieutenant, United States Navy
B.S., University of California, Davis, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN
JOINT COMMAND, CONTROL, AND COMMUNICATIONS

from the

NAVAL POSTGRADUATE SCHOOL
March 1990

Author: Joyce C. Munlin
Joyce C. Munlin

Approved by: Gary K. Poock
Gary K. Poock, Thesis Advisor

Judith H. Lind
Judith H. Lind, Thesis Co-Advisor

Carl R. Jones
Carl R. Jones, Chairman,
Command, Control, and Communications Academic Group

ABSTRACT

Military Command and Control (C^2) requires easy access to information needed for the commander's situation assessment and direction of troops. Providing this information via synthetic speech is a viable alternative, but additional information is required before speech systems can be implemented for C^2 functions. An experiment was conducted to study several factors which may affect the intelligibility of synthetic speech. The factors examined were 1) speech rate, 2) synthetic speech messages presented at lower, the same, and higher frequencies than background noise frequency, 3) voice richness, and 4) interactions between speech rate, voice fundamental frequency, and voice richness. Response latency and recognition accuracy were measured. Results clearly indicate that increasing speech rate leads to an increase in response latency and a decrease in recognition accuracy, at least for the novice user. No effect of voice fundamental frequency or richness was demonstrated.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

AWACS	Airborne Warning and Control System
ANOVA	Analysis of Variance
C^2	Command and Control
C^3I	Command, Control, Communications, and Intelligence
CATCC	Carrier Air Traffic Control Center
DEC	Digital Equipment Corporation
NOSC	Naval Ocean Systems Center
PAM	Pulse-Amplitude Modulated
PCM	Pulse-Code Modulated

I. INTRODUCTION

A. BACKGROUND

1. Command and Control

Military commanders probably have dealt with how to control their forces since the beginning of time. However, only within recent history has command and control (C^2) been identified and studied as a separate discipline. C^2 is critical to any military commander. It has special significance to the United States and its allies, which depend on overcoming numerical inferiority with superior equipment and troop control. C^2 systems disseminate information and orders to various sites within the command structure in order to support the commander. These systems rely heavily on computers due to the vast amounts of data processing required. The interfaces between humans and computers therefore are very important for efficient C^2 operations.

An information chain is only as strong as its weakest link. The more interface layers between a decision maker and the information desired, the greater the likelihood of inaccuracies, delays, and frustration. Poorly designed and implemented interfaces result in user errors and confusion. Personnel may be hesitant to use computer information sources if they are awkward.

2. Computer Generated Speech as a Computer Interface

Computer generated speech has been proposed as an information output technique for computers that is acceptable to many users (Williges and Williges, 1982). This type of computer interface allows the computer to "speak" directly to the decision maker without other human involvement. Speech input and output as a human-computer interface method is garnering more attention as it becomes more economical and technologically feasible (Hakkinen and Williges, 1984).

Computer generated speech may be utilized in a variety of ways. Four general categories of use have been identified by DeHaemer (1989). These are:

1. Provide information by voice as a more natural and comfortable means for the user.
2. Increase the information bandpass, complementing visual information with aural information.
3. Decrease cognitive loading of the visual information channel by shifting information to the aural channel.
4. Facilitate "eyes on" a visual/spatial problem while providing verbal/aural instructions or information.

Computer generated speech presently is used in various industrial, military, and other federal applications. The U.S. Department of Energy has employed synthetic speech as an alarm system via public address and telephone for critical faults experienced during experimental investigation of the best way to dispose of radioactive waste (Digital Equipment Corporation, 1985). The National Aeronautics and Space Administration is utilizing synthetic speech to assist maintenance technicians in a task vital to the space shuttle program--maintaining the thermal protection system (Mollakarimi and Hamid, 1989). Synthetic speech is used to provide prompts, instructions, and feedback to the technicians. United Parcel Service utilizes a voice input/output system to free the hands and eyes of operators handling packages, which maximizes the efficiency of package handling and the speed of data entry (Verbex, 1988). The United Kingdom plans to install computer generated voice output in its Euro Fighter Aircraft to provide timely system and threat status reports to the pilot (Galletti and Abbott, 1989).

3. Computer Generated Speech Technology

The definition of "synthetic" speech depends on the user. One definition requires that, for a computer to generate true synthetic speech, the words that are spoken by the computer should not have been prespoken by a human (Cater, 1983). The method of storage--tape, or integrated circuit--is not relevant. If the words have been prespoken, then the speech is considered to be reconstructed speech. Thus direct waveform encoding and reconstruction of utterances is *reconstructed* speech. Under this definition, only one true "*synthetic*" speech method is included in this study: the analog formant frequency synthesis technique.

A second definition of "synthetic" speech is related to basic data sampling theory--Shannon's sampling theorem and the Nyquist rate (Cater, 1983; Stanley, 1982). According to these two theories, a signal must be uniformly sampled at a rate at least as high as twice the highest frequency in the signal's spectrum for adequate description of the analog waveform (Stanley, 1982). This means that, for satisfactory reconstruction of a voice signal with a maximum frequency of 3 kHz, the signal must be sampled at a rate of 6 kHz or higher. Adequate storage of each sampled signal in a computer requires at least four data bits per sample; this requires a bit sampling rate of 24,000 bits per second (6000 samples/sec x 4 bits/sample)(Cater, 1983). If the same quality of speech could be reconstructed with a reconstruction rate lower than the expected of 24,000 bits per second, then "synthetic" speech is produced instead of digitally reconstructed speech.

For this study, the term "synthetic" speech is used to mean that the words have not been prespoken by humans. The term "digitized" speech refers to speech generation methods that require that words be prespoken by humans. "Computer generated" speech is used for both synthetic and digitized speech.

a. Digitized Speech

There are many methods of producing digitized speech. As an example, one of the simplest methods of speech generation is the waveform encoding and reconstruction technique. For this process, a signal waveform is sampled by a unit sampling function at intervals T for a duration of t_0 (Inglis, 1988}. Figure 1 illustrates this process for a 3-data- bit analog-to-digital converter.

The original input signal, Figure 1(a), is the analog waveform which is to be sampled by a digital sampling system. For the signal to be reconstructible, the minimum sampling rate--the Nyquist rate--must be at least as high as twice the highest frequency in the spectrum to be sampled. The sampled speech is a pulse-amplitude modulated (PAM) signal, as shown in Figure 1(b). The PAM signal is a sampled-data signal consisting of a sequence of pulses in which the amplitude of each pulse is proportional to the analog signal at the corresponding sampling point. The signal is still analog. To translate it into digitized form, each sampled data pulse is replaced by one of a finite number of possible amplitude data values (Figure 1(c)). This process is called quantization; the pulses are now called pulse-code modulated

(PCM). The possible number of finite values is determined by the number of data bits used to represent the value of each pulse. For example, a two data bit system could represent 2^2 or 4 values, while a 3 bit system could represent 2^3

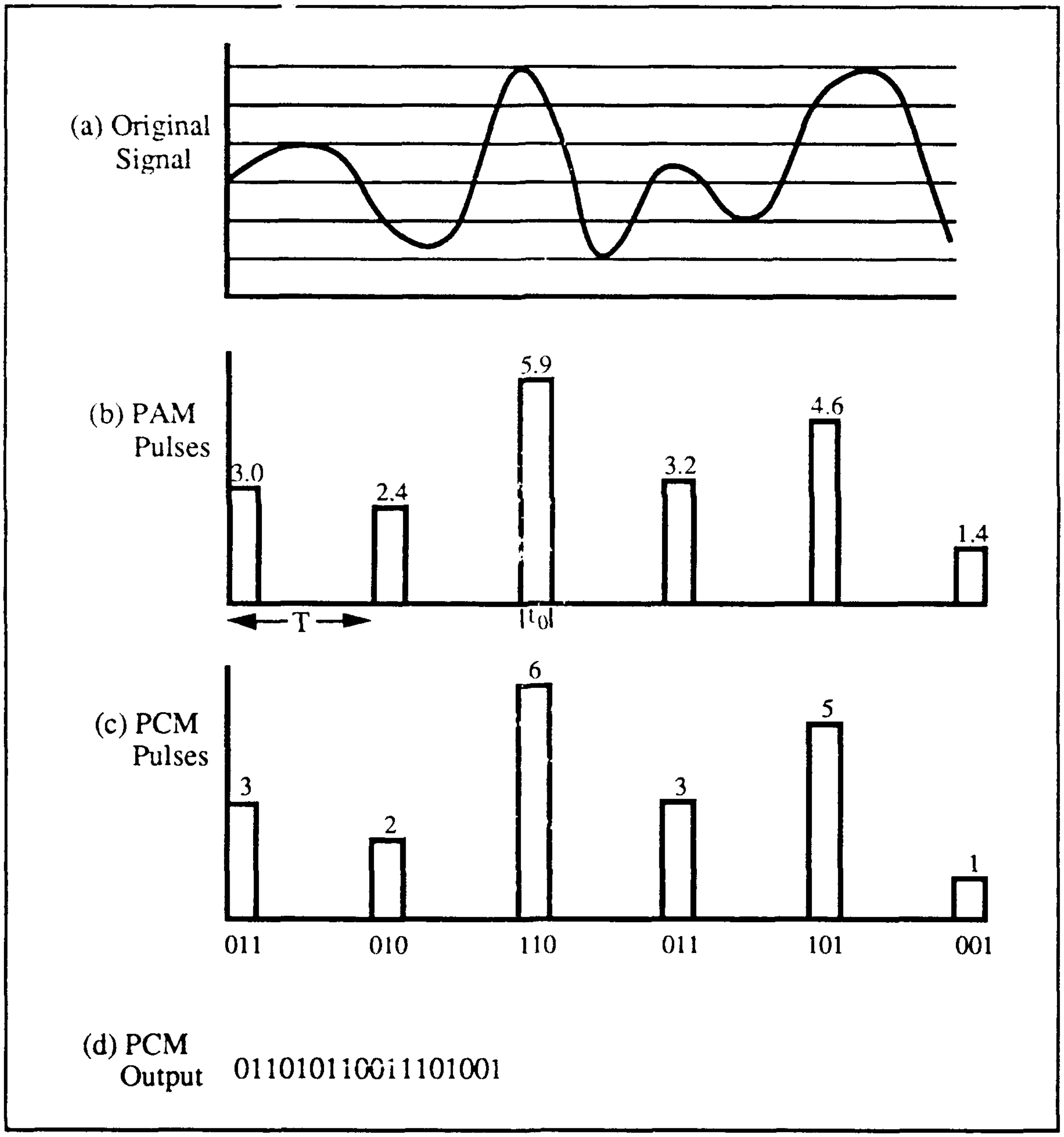

Figure 1. 3 Bit Analog-to-Digital Conversion

or 8 values. Finally, as shown in Figure 1(d), the pulses are stored in a computer as binary digits. The reconstruction process is quite similar to the encoding process.

A typical speech sampling and playback system is illustrated in Figure 2. Sound is transformed from acoustical to electrical energy by a microphone and then passed through a low pass filter to prevent aliasing by removing frequencies above one-half of the sampling rate. Aliasing occurs when the Nyquist minimum sampling rate requirement is not met and components of the original spectrum overlap and cannot be uniquely determined or separated (Teja and Gonnella, 1983). The amplifier intensifies the signal to a usable level. The sampler produces a signal of the type shown in Figure 1(b). The analog-to-digital converter is actually composed of two parts: the quantizer and the digital encoder. A 4-data-bit analog-to-digital converter is illustrated in Figure 2. However, converters may be designed for various numbers of bits--8,12,16, etc. A resident computer program sequentially stores the data in computer memory.

A playback program steps through the stored data and sequentially outputs it to the digital-to-analog converter. The PCM decoder deciphers the 4-bit code, converting it into the voltage level represented by the binary digits. A low pass filter removes undesirable high frequencies and the amplifier intensifies the signal to a usable level for the speaker system.

The quality of the output speech is highly dependent on the original sampling rate and on the number of bits used to represent the value of each pulse. Generally, the higher the sampling rate and the higher the number of bits, the better the quality of the resultant speech output.

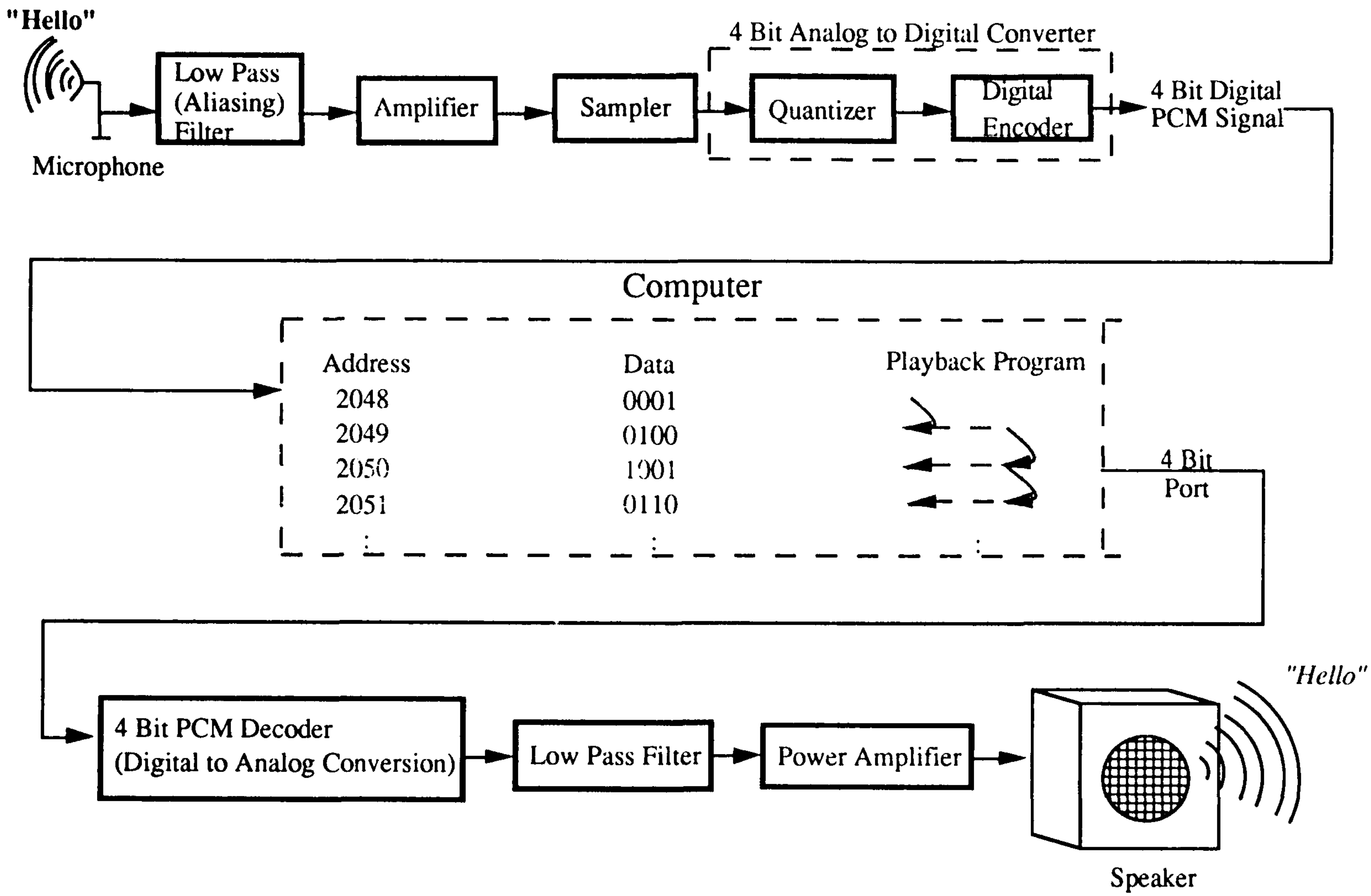

Figure 2. Typical Speech Sampling and Playback System (Adapted from Cater, 1983)

b. Synthetic Speech

Analog formant frequency synthesis is a typical synthetic speech methodology, used here as an illustration of the technique. The waveform encoding and reconstruction technique (discussed above) is similar to a "photograph" of speech. Analog formant frequency synthesis is more like an artist's rendition of speech. The principles behind the formant synthesizer are based on acoustic replication of the human vocal tract.

Basic understanding of human speech and linguistics is necessary in order to understand this synthesis technique. Typical pitch frequencies for male voices range from 130 Hz to 146 Hz, with an average frequency of around 141 Hz. The female voice pitch range is from 188 to 295 Hz, with a median frequency of approximately 233 Hz (Cater, 1983). These frequencies are the fundamental or glottal vibration frequencies created by the vocal chords.

Various resonance frequencies are created in the cavities within the vocal tract and are known as the formant frequencies. Three to four formant frequencies are required for adequate speech synthesis and range from approximately 200 to 2000 Hz from the first to the third formant. All of the formant frequencies exist simultaneously during speech. What is heard during speech is not a single frequency but rather a number of frequencies which have been created from the glottal vibration of the vocal chords.

In addition to the formant frequencies, fricatives, plosives, and nasal consonant sounds also are important to human understanding of speech. The fricatives and plosives are hissing and popping sounds primarily created by the teeth, lips, and tongue at the front of the mouth. Nasal consonant sounds (such as the ng in ring) are strongly dependent on the resonance of the nasal cavities.

A wide variety of sounds is necessary to produce normal human speech. Speech results from stringing together phonemes, which are basic sound units of speech. The English language uses approximately 40 of them (Cater, 1983). In addition, there are many variations of each phoneme, called allophones. The variations present in the allophones depend not only on the phoneme and word being spoken but also on the position of the phoneme within the word. Diphthongs are sounds which typically arise from the pronunciation of two vowel-type phonemes in series. Affricates are similar to diphthongs except that the unique sound arises from the pronunciation of two consonant-type phonemes in series.

Analog formant frequency synthesis begins with the entry of characters representing the words to be spoken into a computer (Figure 3). A keyboard is used to type and enter the words. The computer parses each word into its component phonemes, allophones, etc., and outputs the relevant control information for each unit of sound to the formant speech synthesizer. Bandpass filters are utilized to create resonance frequencies similar to human formant frequencies. The center frequency of each of the bandpass filters is adjustable to match the equivalent output of the human vocal system for a particular unit of sound. Fricative and nasal resonators are necessary in order to simulate the fricative and nasal consonants. To the human ear, the summation of the filter outputs resembles the output of the human voice.

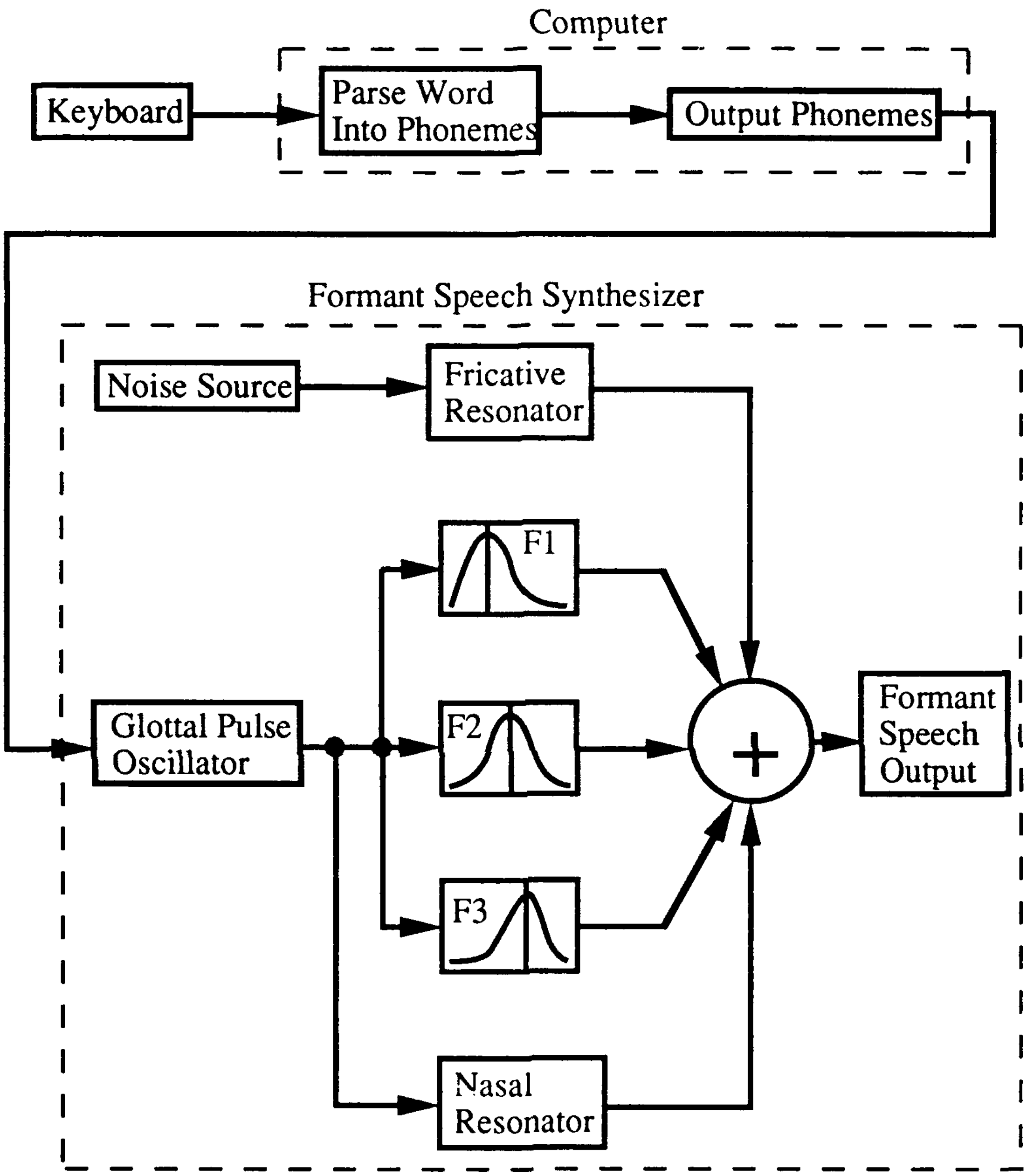

Figure 3. Formant Speech Synthesizer System (Adapted from Cater, 1983)

4. Computer Generated Speech Technology for Military Systems

Of particular interest to the U.S. military are voice input/output systems used to assist in managing aviation assets. One voice alert system currently installed in the F/A 18 Hornet aircraft provides verbal caution and warning messages to the pilot concerning his altitude, engine status, and fuel level. In the future this alerting system may be integrated with sophisticated computer programs in order to provide a vocal listing of outside threats in a coherent hierarchy, beginning with the most urgent (Kitfield, 1989).

Boeing Military Aircraft Company is conducting research related to improving the man-machine interface for the E-3 aircraft Airborne Warning and Control System (AWACS). AWACS is a command, control, communications, and intelligence (C^3I) system with onboard radar, surveillance, and data processing capabilities. It supports missions that identify and track airborne and surface targets for air traffic control, provides early warning of enemy threats, and directs interceptors to their targets. A prototyping approach is being used to evaluate voice input and output applicability for C^3I systems (Salisbury, 1989). The prototype has demonstrated the usefulness of voice input and output systems for several functions, including fuel updating, committing fighters, tactical and broadcast control, and sensor suite management. End users reported that they enjoyed the intuitive nature of the voice input/output interface (Chilcote, 1989).

The Speech Technology Group at the Naval Ocean Systems Center (NOSC) at San Diego began working with voice input and output systems in 1984. Systems of particular interest include voice controlled status boards in the Carrier Air Traffic Control Center (CATCC) and voice synthesis used for console message alerts in a U.S. Marine Corps mobile computer complex. For the CATCC NOSC found that voice input/output technology reduced manpower requirements, reduced errors, and increased the update and dissemination rate of the information (Johnson and Nunn, 1986).

The NOSC voice output system for the mobile computer complex provided alerts to the computer operator, who is often required to be away from his console. In addition, the system was programmed to provide translations of the otherwise cryptic two-character alert codes. Three benefits were noted by NOSC: (1) the computer operator received timely notice of important alerts which might otherwise have been delayed or entirely missed, (2) operator efficiency was increased through timely notification of system status and translation of cryptic status codes, and (3) less time was required to train the operator since status codes were already decrypted (Johnson and Nunn, 1986).

Military organizations of other countries also are interested in voice input and output systems. European countries in particular are conducting research and planning to field various systems. Areas of interest include fighter aircraft cockpits (France, United Kingdom, West Germany, and Italy), artillery target observation and reporting (United Kingdom), battlefield C^3I (West Germany), and helicopter operations (United Kingdom) (Partridge, 1989). Fighter aircraft for which voice input and output systems are planned include the French Rafale, the European Fighter Aircraft (West Germany, Italy, Spain and the United Kingdom), and the Tornado (United Kingdom).

B. FACTORS AFFECTING SYNTHETIC SPEECH INTELLIGIBILITY

When speech synthesis is used for military systems, it is critical that the listener understand the messages. It has been proposed that several factors affect speech intelligibility. These include (1) masking noise, (2) speech rate, (3) speech "richness", and (4) the type of voice synthesis system used.

1. Masking, Speech Rate, and Richness

Simpson and Marchionda-Frost (1984) tested the hypothesis that masking of the fundamental frequency of synthesized speech by high energy cockpit noise decreases the comprehensibility of the synthetic voice. They also evaluated whether response times to synthetic voice messages are diminished as the speech rate increases until an unknown maximum cognitive processing rate is reached. Under the experimental conditions tested, they found no significant differences in intelligibility due to masking noise at the same frequency as the synthesized speech. They also found no significant differences in intelligibility as a function of speech rate, within the range of 156 to 178 words per minute.

Other studies have been conducted on the effects of speech rate on comprehensibility. Slowiaczek and Nusbaum (1985) found "...significant decrements in intelligibility with increased speaking rate". Marics and Williges (1988) also found decreased intelligibility and increased response latency with an increase in speech rates.

Possible explanations for the apparent discrepancies in findings among researchers may be due to the unique experimental procedures utilized by Simpson and Marchionda-Frost. Subjects in their experiments were trained until they scored at a 100% recognition level for all vocabulary words, then were maintained at 100% word recognition status during the course of the experiment. All messages were structured using the same general format: threat type/position/status, in that order. The vocabulary word set and number of different messages were small.

Synthetic voice devices are generally considered of maximum usefulness for employment in systems which require a large or even unlimited vocabulary (Allen, 1981). Under these conditions neither vocabulary nor sentence structure may be 100% pre-trained. Compared to the Simpson and Marchionda-Frost study, Slowiaczek and Nusbaum (1985) and Marics and Williges (1988) utilized larger vocabularies and more varied structures to test the effects of speech rates on intelligibility. Thus, it is not surprising that results were different. Tests of the effects of masking noise also might yield different results, with large vocabularies and test subjects that are not trained to 100% word recognition capability.

A "rich" voice is one that is full and mellow in tone and quality. Digital Equipment corporation offers the following description of the richness parameter of its DECtalk system:

> The opposite of a soft breathy voice is a rich, brilliant voice. *This voice type carries well in a noisy environment* [emphasis added]. It is forceful and intelligible, although not always the most friendly sounding voice...For example, you might turn up the richness factor when you need a voice that conveys emergency procedures or warnings.

No known research has been completed to determine the degree to which increasing voice richness will increase intelligibility. Contact with DEC did not yield any further information on research related to voice richness (Telephone Conversation, 1989).

2. Voice Synthesis Systems

There are many different synthetic voice output devices on the market today. Pratt (1987) compared the performances of eight of these synthetic voice systems: DECtalk (four different voices), Calltext, Infovox, Prose 2000, TI-Speech, JSRU, Namal Type & Talk, and Computer Concepts. Pratt tested intelligibility under noisy and clear conditions, using semantic differential scaling and diagnostic and and modified rhyme tests. Under all conditions, Perfect Paul, Beautiful Betty, and Frail Frank--three of the four DECtalk voices tested--rated in the top three. The combined results of all the tests ranked Perfect Paul overall as the most intelligible voice. In another study, Greene, Manous, and Pisoni evaluated the DECtalk version 1.8 speech synthesis system and concluded that "...we have found the synthetic speech to be substantially better than any of the other test-to-speech systems we have studied in our laboratory over the last five years" (Indiana University, 1984).

C. STUDY GOAL AND OBJECTIVES

The goal of this study is to provide the U.S. military with a better understanding of factors that affect comprehensibility of synthetic speech as a human-computer interface. Emphasis is on the understanding of messages spoken by several kinds of voices while in a noisy environment. A laboratory experiment was conducted in order to meet this goal.

The objectives of the experiments are as follows:

1. To determine the effect of speech rate on accuracy and response latency in the presence of background noise.
2. To determine the effect on accuracy and response latency of synthetic speech messages presented at lower, the same, and higher frequencies than the background noise.
3. To determine the effect of increasing the "richness" parameter of a synthetic voice on accuracy and response latency in noisy environments.
4. To determine the interactions between voice richness, voice frequency and speech rate, as these affect accuracy and response latency.

D. SCOPE

This study is limited to examining three factors at three levels each which may affect human perception of the computerized synthetic voice output of the DECtalk Computer System, Version 1.8, when used in a noisy environment. The three factors are speech rate, average pitch of the voice as this relates to the pitch of background noise, and voice richness. The levels chosen for each factor are: 1) Speech rate--160, 175, and 190 words per minute, 2) Average pitch of the voice--95 Hz, 115 Hz, and 135 Hz, 3) Richness--10, 50, and 90 as defined by DEC. The following sections present a detailed description of the experiment that was conducted, along with results, data analysis, and conclusions regarding the effects of the three factors on speech understanding in a noisy environment.

II. METHODOLOGY

A. EQUIPMENT

Experiments for this study were performed in a Controlled Acoustic environment chamber developed by Industrial Acoustics Company. The inside dimensions of the chamber are 78 inches high by 76 inches wide by 72 inches deep. Figure 4 illustrates the experimental equipment configuration.

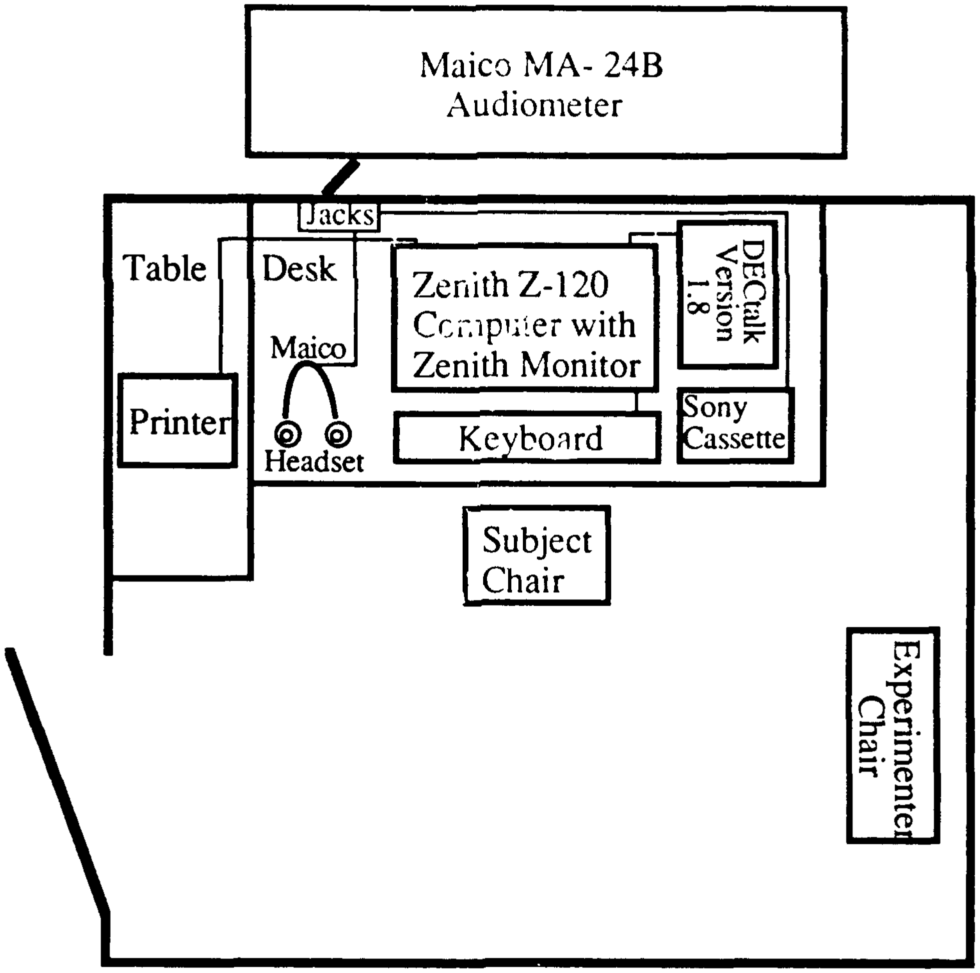

Figure 4. Experiment Equipment Configuration

DECtalk version 1.8 voice synthesis computer system, developed by the Digital Equipment Corporation (DEC), produced the synthetic voice output used for this study. Ten different voices are provided by DEC which are customizeable for various parameters. For this study, the Perfect Paul voice was utilized throughout with modified speech rate, richness, and average pitch. The designed fundamental frequency of this voice is 120 Hz. The DECtalk system was driven by a Zenith Z-120 personal computer. This computer also was used to store and control the verbal sentence material and variable voice parameters.

Background noise used for masking was an actual shipboard recording of the USS Kitty Hawk's pump room. A Sony cassette deck model TC-124 played the tape of the pump room noise for the experiment. A spectrogram of the sound frequencies of this noise from 95 to 145 Hz, as obtained using the cassette deck and an HP 3562A signal analyzer system, is provided in Figure 5. As may be observed, the spectrogram shows a series of spikes with the maximum at 115 ± 4 Hz, at a root-mean-square (RMS) sound power level of between -18 dB and -26dB. A spectrogram of the sound frequencies produced by the signal analyzer/cassette system electronics (Figure 6) shows an energy spike at 120 Hz, but at a sound power level well below the level of the pump room noise recording. The pump room noise, as played on the Sony cassette deck, was considered adequate to provide masking for the Perfect Paul synthetic voice with the fundamental frequencies used for these tests.

Sound from both the DECtalk system and noise cassette tape were fed into a Maico model MA-24B research and clinical audiometer, consisting of twin

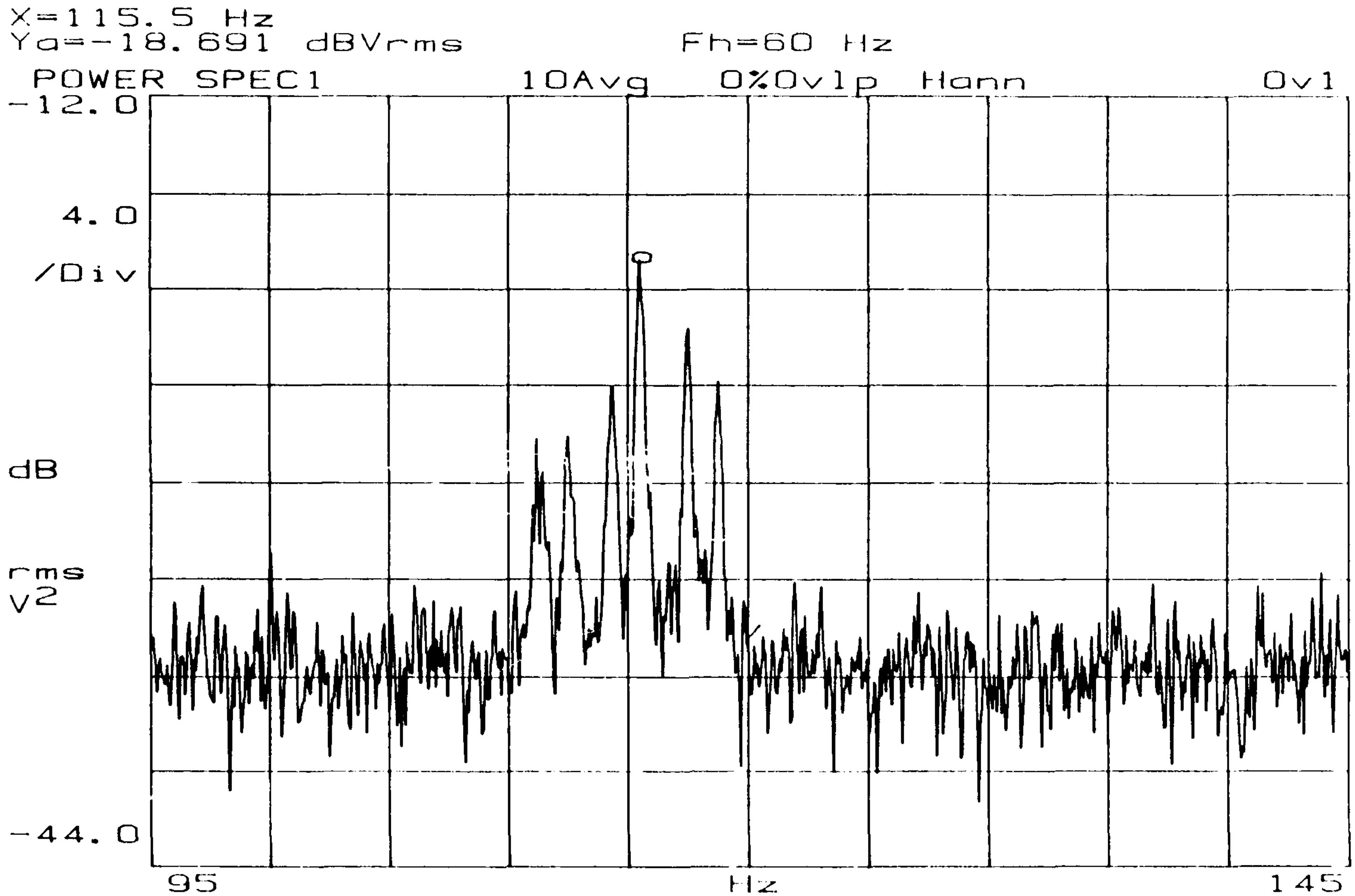

Figure 5. Spectrogram of Recorded Noise from USS Kitty Hawk Pump Room, as Played on Sony Cassette Tape Deck

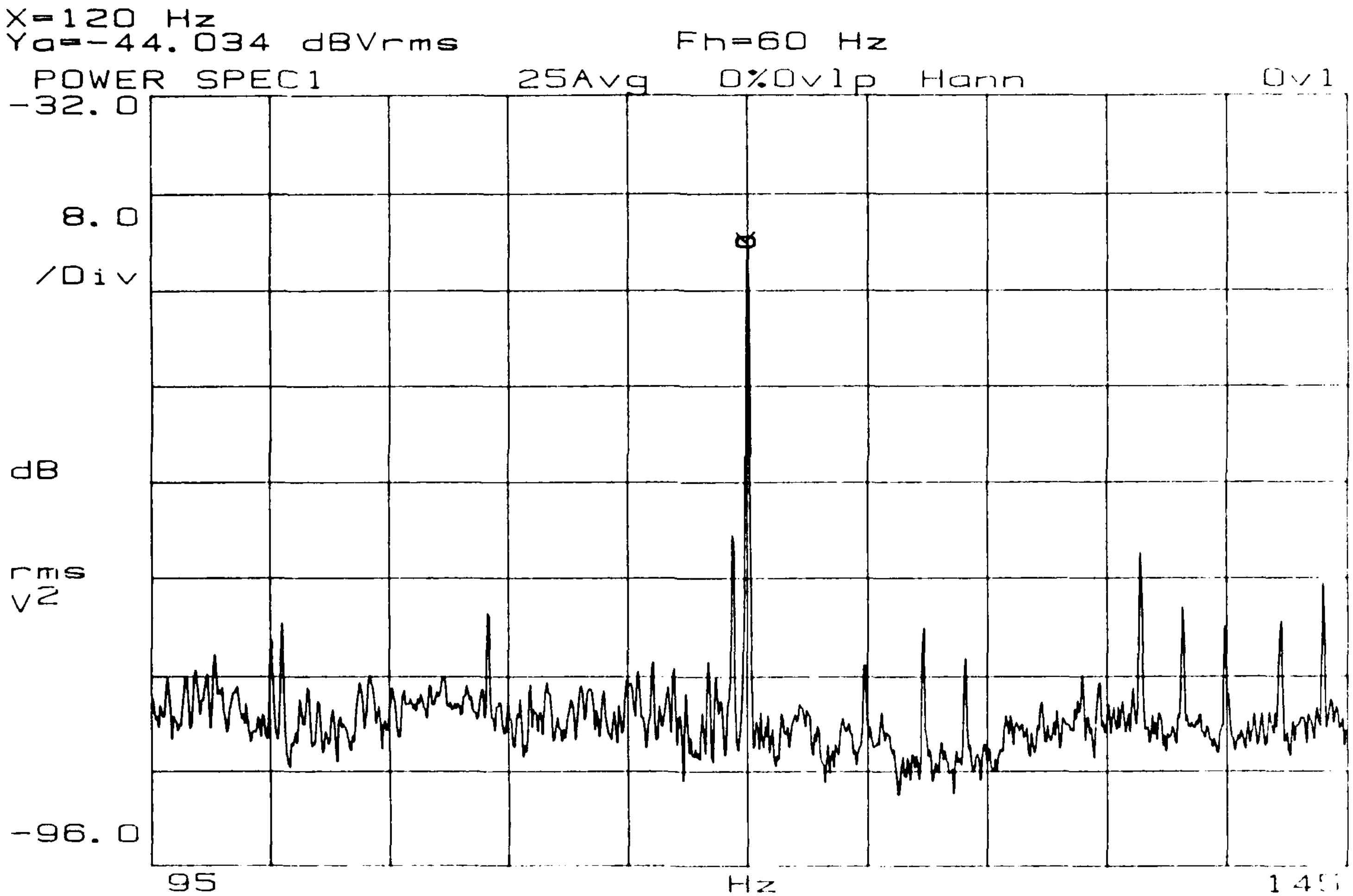

Figure 6. Spectrogram of Noise Generated by Sony Cassette Tape Deck and HP3562A Signal Analyzer System. Note that the RMS Power Level at 120 Hz is less than –40 dB

audiometer channels and an accessory control section. The two signals were intensified by separate left and right calibrated amplifiers and mixed by the Maico audiometer. That signal was then fed into the Maico test headsets worn by the subjects. A HP 427A voltmeter was used at the input jacks to the acoustic chamber to determine the difference in dB levels between the noise and the synthetic voice as they were delivered to the Maico test headsets. The noise signal was maintained at a level of 10 dB stronger than the voice signal.

B. STUDY VARIABLES

Three independent variables were tested during this study. First was the speech rate of the synthetic voice. Three levels were tested: 160, 175, and 190 words per minute. Second, the fundamental frequency of the synthetic voice was tested. Three levels were selected, to be lower, the same, and higher than the high energy frequency of the background noise. The levels were 95, 115, and 145 Hz. The richness of the voice also was tested at three settings: 10, 50 and 90. All other factors, including background noise frequency and volume, were held constant.

Two measures were taken to serve as dependent variables for this study. First was the accuracy with which subjects transcribed synthetic voice messages as they heard them. Second was response latency--elapsed time from the end of the vocal presentation of each sentence until the subject typed the first character of his response.

C. EXPERIMENTAL DESIGN

A three-way factorial design was used for this study. Each of the three independent variables was tested at three levels. The resulting 3^3 data matrix is shown in Figure 7. The 27 cells of the data matrix represent the 27 tested conditions. All subjects were evaluated under all 27 conditions.

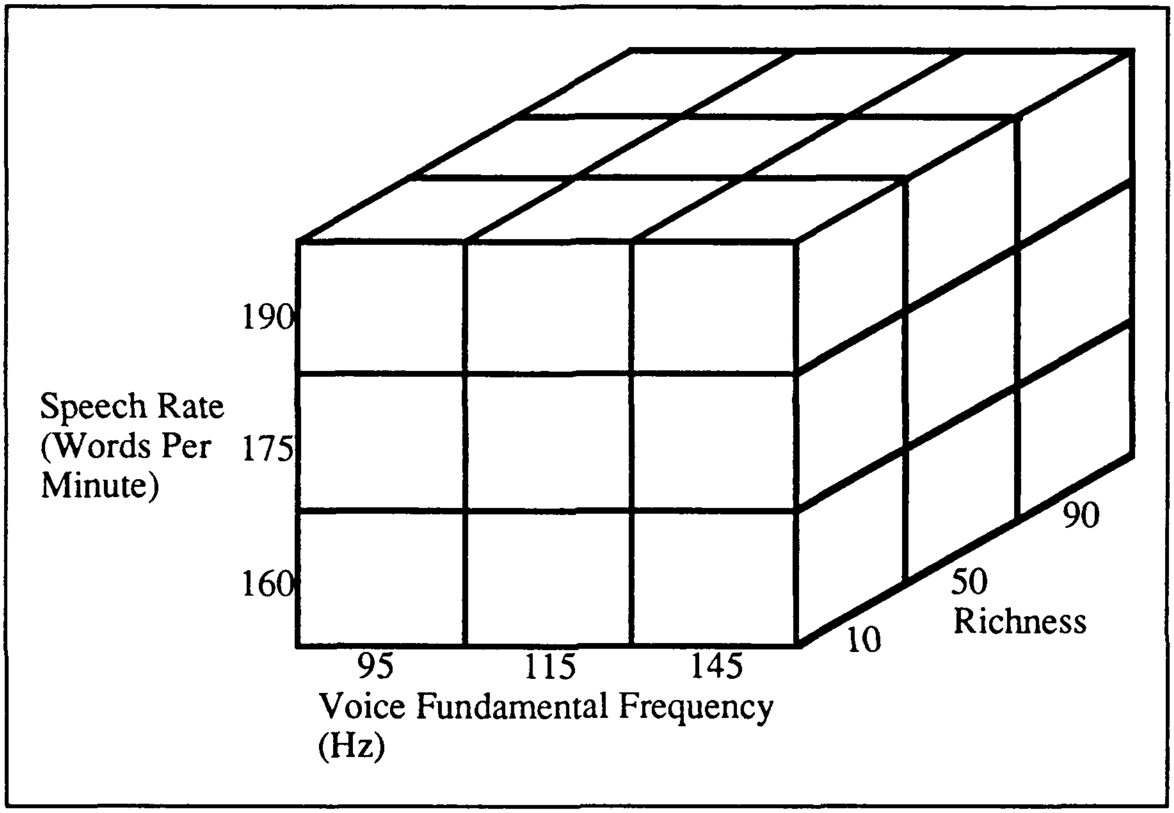

Figure 7. Experimental Design Matrix

The design of this experiment is called a mixed model. When all of the levels of an experiment are chosen by the experimenter then the design of the experiment is a fixed model. If all levels are randomly chosen, the design is a random model. However, if some levels are chosen by the experimenter and some levels are randomly selected, the design is a mixed model. In this experiment, the levels of speech rate, fundamental frequency, and richness were chosen by the experimenter. However, the subjects were chosen randomly, resulting in a mixed model.

D. STUDY PARTICIPANTS

A total of 19 Naval Postgraduate School students from various curricula participated in this study on a voluntary basis. Of these subjects, 18 were male and one was female. The participants ranged from 26 to 38 years of age, all were U.S. military officers from various branches of service, and all were native English speakers. All subjects indicated that they consider themselves to have normal hearing.

Participants were asked about previous experience with synthetic voice output. Four male participants indicated that they had some previous experience. Two indicated that they had experienced synthetic voice output on a home personal computer, one had experience as a user with a phone trouble desk, and one had seen a demonstration at a science center. Test results from these individuals were not analyzed separately.

E. PROCEDURE

Each participant first filled out a questionnaire which asked for the date, name of the participant, date of birth, sex, whether the subject had normal hearing, and if the subject had any previous experience with synthetic voice output and if so where. The participant was then seated in the acoustic booth. The following set of instructions was then read by the experimenter to the subject:

These are the instructions for the synthetic voice experiment. If you have any questions regarding these instructions please ask and I will repeat any part or all of the instructions. This is an experiment with the DECtalk computerized synthetic voice output device in a noisy environment. The noise you will be exposed to is from the USS Kitty Hawk's pump room. Different sentences will be spoken by the DECtalk unit. You will respond as quickly and accurately as possible after the DECtalk has completed the sentence by entering the words you thought you heard on the key board. The sentence you type will be seen on the screen in the bottom left corner. There is no editing capability. If you make a typo you can tell me after you have completed typing the sentence. If you do not understand all of the words, enter what you do understand. Make your best effort to enter any and all words you heard. Spelling nor typing skill is of concern. After you enter your name begin the experiment by pressing the return key. Soon thereafter, DECtalk will present a sentence to you. Respond as quickly and accurately as possible after the completion of the sentence. End the entry of your sentence with a return. A prompt will then appear on the screen asking if you are ready for another sentence. When you are ready press the return key again. This will continue for 27 times and we will do that twice. I will be here during the experiment if you have any difficulties or questions.

After the instructions were read, the participant placed the Maico headset on his or her head and adjusted it, then began the experiment when ready.

Stimulus materials consisted of 100 syntactically correct and meaningful sentences, spoken to the subjects by the synthesized voice. The sentences were derived from Egan (1948), and are commonly known as the Harvard sentences. Each contains five content words (main nouns, adjectives, and verbs), plus articles and pronouns as necessary to make a smooth-flowing sentence. Examples of the sentences are :

1. A plump hen is well fitted for stew.

2. The ape grinned and gnashed his yellow teeth.
3. The birch canoe slid on the smooth planks.

For each test run, 27 sentences were randomly chosen from the 100 sentence file.

A program written in the Pascal programming language by Professor David Wadsworth was used to control the experiment (Appendix A). The program randomly presented each of the 27 test conditions in each 27-sentence test run. The program also timed response latency--the difference in time between when the DECtalk finished speaking and when the subject first pushed a key on the keyboard in response. The program then recorded the sentence that was spoken, the response latency of the subject, and the sentence typed on the keyboard by the subject.

Participants verbally noted when they had made typographical errors while typing the sentence that they thought they heard. The experimenter noted these errors for reference when scoring the responses. The program was run twice for each subject, for a total of 54 sentence presentations per subject.

III. RESULTS AND DISCUSSION

A. DATA ANALYSIS

1. Measurement of Response Latency

Response latency was measured in milliseconds from the time the DECtalk synthesized voice finished a sentence until the subject first pushed a key on the keyboard in response. The overall average value was 2497 milliseconds.

Inspection of the data points led to the conclusion that a total of 25 response latency values should be considered outliers. Responses of less than half of a second and of more than 9 seconds were removed. The remaining values then were averaged for all of the experimental design matrix cells. The resulting cell average values were used to replace the outliers so that clearly erroneous values would not unduly influence the analysis and a complete data matrix would be available for further analysis.

2. Measurement of Accuracy

As noted earlier, each of the 100 Harvard sentences includes five content words. Accuracy was measured as a percentage of the number of content words correctly transcribed. For a given test run (using 27 sentences, each including five content words) 100% accuracy required correct transcription of a total of 135 words. Only the content words in each sentence were considered in determining whether a subject transcribed each sentence correctly.

If a content word was missing it was graded as incorrect. The omission or addition of prefixes or suffixes was scored as incorrect. Substitution of a word with the same sound but different meaning was scored as correct, e.g., bear substituted for the word bare.

3. Analysis of Variance

An analysis of variance (ANOVA) was used to determine the level of the effects of speech rate, voice fundamental frequency, and richness on response latency and accuracy. Results were used to identify statistically significant differences in the variance of the mean accuracy and response latency between the three levels of speech rate, three levels of voice fundamental frequency, and three levels of richness. ANOVA was also used to test for interactions between the combinations of each of the three factors and of all three levels. Due to the mixed model experimental design, during data analysis all the main effects and interactions of fixed factors were tested by the corresponding interaction of the fixed part and the random one. The random main effect and the interactions of the random factor by the fixed parts were tested against the error term.(Anderson and McLean, 1974). That is, the main effect of speech rate was tested against the interaction of speech rate with subject, and the interaction of speech rate with fundamental frequency was tested against the interaction of speech rate with fundamental frequency with subject, etc.

The ANOVA was used to determine whether there was a significant difference in mean performance levels as a function of the three levels tested for each variable. For each main effect found to be significant at the 0.05 level or higher, a Newman-Keuls test was conducted to determine which means were significantly different than the others. Initially each test was conducted at the 0.05 level of significance. If the means were significantly different at the 0.05 level then they were tested at the 0.01 level of significance.

The Statistical Analysis System (SAS) software run on an IBM 3033/4381 computer was used to perform the ANOVA. The Newman-Keuls test was conducted by the experimenter according to the procedure provided in Hicks (1973).

B. RESULTS

1. Analysis of Response Latency Data

The effects on response latency of speech rate, voice fundamental frequency, and richness were analyzed first for both runs combined. In addition, all interactions were analyzed. The results are shown in Table 1.

As may be observed both the speech rate main effect and the four-way interaction, speech rate by fundamental frequency by richness by data collection run, show a level of significance at 0.05 or above. This indicates that the mean response latency values for at least two of the three speech rates are significantly different from each other, and that only five times out of 100 would these results be expected to occur randomly. The interaction effect indicates that the combination of the four factors has an effect on the response latency. The effect of data collection run itself is significant at the 0.1 level. No other effects or interactions are significant at the 0.1 level or above.

A Newman-Keuls test was performed to determine which of the three speech rate levels were significantly different from the others. Speech rates 160 words per minute and 190 words per minute were significantly

TABLE 1. ANALYSIS OF VARIANCE FOR RESPONSE LATENCY FOR BOTH DATA COLLECTION RUNS COMBINED

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	766	4.75+
Fundamental Frequency (FF)	2	324	2.33
SR * FF	4	174.8	1.08
Richness (Ri)	2	123.5	0.51
SR * Ri	4	26.25	0.27
FF * Ri	4	324	1.69
SR * FF * Ri	8	108.6	0.47
Subject (S)	18	2280	
SR * S	36	161.2	
FF * S	36	139.1	
SR * FF * S	72	161.2	
Ri * S	36	244.5	
SR * Ri * S	72	97.99	
FF * Ri * S	72	192.2	
SR * FF * Ri * S	144	230.5	
Run (R)	1	995	3.52 •
SR * R	2	191.5	1.11
FF * R	2	109.5	0.51
SR * FF * R	4	243.8	1.56
Ri * R	2	143.5	0.67
SR * Ri * R	4	41.5	0.29
FF * Ri * R	4	131.5	0.63
SR * FF * Ri * R	8	651.1	4.59 †
S * R	18	283	
SR * S * R	36	172.5	
FF * S * R	36	214.3	
SR * FF * S * R	72	156.1	
Ri * S * R	36	213.8	
SR * Ri * S * R	72	143.5	
FF * Ri * S * R	72	207.8	
SR * FF * Ri * S * R	144	141.9	

* Indicates interaction between sources
• Shows significance at the 0.1 level
+ Shows significance at the 0.05 level
† Shows significance at the 0.01 level

different from each other, at the 0.05 level. The difference between mean response latencies at 175 and at 160 words per minute was not significant, nor was the difference between 190 and 175 words per minute.

The mean response latency values for run one and run two as a function of the three levels of speech rate--160, 175, and 190 words per minute--are displayed in Figure 8. The graph demonstrates a clear trend of increasing response latency for increasing speech rate, with response latency consistently lower in run two than in run one.

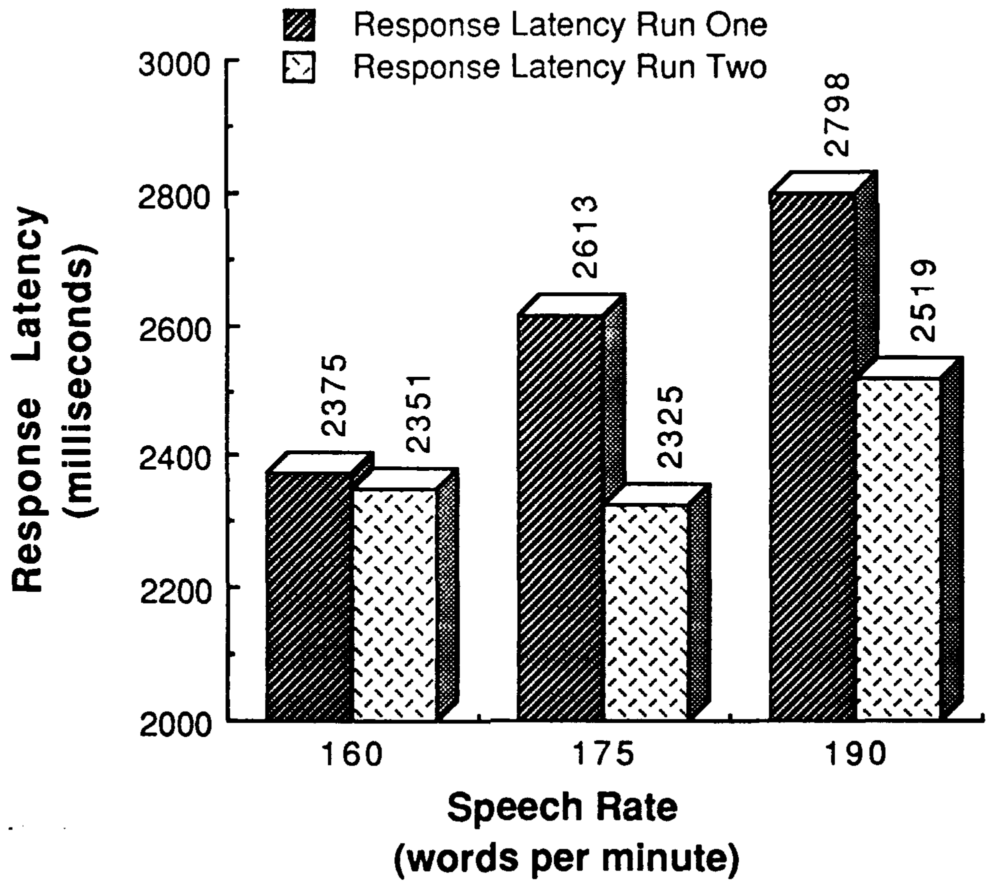

Figure 8. Comparison of Mean Response Latencies for Speech Rate on Run One and Two.

Because of the interaction between speech rate, fundamental frequency, richness, and run, two more ANOVAs were conducted, one for each of the two individual data collection runs. The results for the ANOVA on data collection run one are given in Table 2 and results for the ANOVA on run two are shown in Table 3.

Table 2 indicates that for run one, the effect of speech rate on response latency is significant at the 0.01 level. In addition, the three-way interaction of speech rate with fundamental frequency with richness is significant at the 0.05 level. No other effects or interactions are significant at the 0.05 level or above. A Newman-Keuls test was performed to determine at which speech rate levels the response latency values were significantly different from each other. Speech rates of 160 words per minute and 190 words per minute were significantly different from each other at the 0.01 level. The difference between the mean response latencies at 175 and 160 words per minute was not significant, nor was the difference between 190 and 175 words per minute.

For run one, the three-way interaction (speech rate by fundamental frequency by richness), as these factors affect response latency, is displayed as three graphs in Figure 9. The mean response latency for each of the three speech rates is depicted for every richness value--10, 50, and 90--and for every fundamental frequency setting--95, 115, and 135 Hz-- at each of the three richness values.

As may be observed in Table 2, the ANOVA of response latency for run two indicates that no effect or interaction is significant at the 0.05 level or above. However, the three-way interaction speech rate by fundamental frequency by richness is significant at the 0.1 level. This run two three-way interaction is displayed graphically in Figure 10. The mean response latency is shown as a function of each speech rate, for each richness value and each fundamental frequency setting.

TABLE 2. ANALYSIS OF VARIANCE FOR RESPONSE LATENCY FOR DATA COLLECTION RUN ONE

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	768.5	6.34 †
Fundamental Freqeuemcy (FF)	2	90.5	0.44
SR * FF	4	340.3	2.18
Richness (Ri)	2	69.0	0.27
SR * Ri	4	11.0	0.09
FF * Ri	4	149.0	0.76
SR * FF * Ri	8	399.6	2.18 +
Subject (S)	18	1100	
SR * S	36	121.3	
FF * S	36	207.9	
SR * FF * S	72	156.3	
Ri * S	36	256.6	
SR * Ri * S	72	123.0	
FF * Ri * S	72	197.1	
SR * FF * Ri * S	144	183.0	

* Indicates interaction between sources
\+ Shows significance at the 0.05 level
† Shows significance at the 0.01 level

2. Analysis of Accuracy Data

Each sentences was scored for the percentage correct of the five content words; possible values were 0, 0.2, 0.4, 0.6, 0.8 and 1.0. As a result, variances and means were not independent. To stabilize the variances, the values were transformed to $2 * \arcsin \sqrt{x}$, following the recommendation of Winer, 1971. For both runs combined, the effects on accuracy of speech rate, voice fundamental frequency, and richness were analyzed, along with all interactions. The results are presented in Table 4.

TABLE 3. ANALYSIS OF VARIANCE FOR RESPONSE LATENCY FOR DATA COLLECTION RUN TWO

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	189.0	0.89
Fundamental Frequency (FF)	2	343.0	2.36
SR * FF	4	78.25	0.49
Richness (Ri)	2	198.0	0.98
SR * Ri	4	57.0	0.48
FF * Ri	4	306.5	1.51
SR * FF * Ri	8	360.1	1.90 •
Subject (S)	18	1464	
SR * S	36	212.3	
FF * S	36	145.5	
SR * FF * S	72	160.9	
Ri * S	36	201.7	
SR * Ri * S	72	118.5	
FF * Ri * S	72	202.8	
SR * FF * Ri * S	144	189.4	

* Indicates interaction between sources
• Shows significance at the 0.1 level

For the combined results, the ANOVA indicates that speech rate has a significant effect on accuracy at the 0.01 level. The main affect of the run number, and two interactions (speech rate by fundamental frequency by run, and speech rate by fundamental frequency by richness by run) were significant at the 0.1 level. This indicates that the mean accuracy values for at least two of the three speech rates are significantly different from each other, and that only one time out of 100 would these results be expected to occur randomly. No other effects or interactions were significant at the 0.1 level or above.

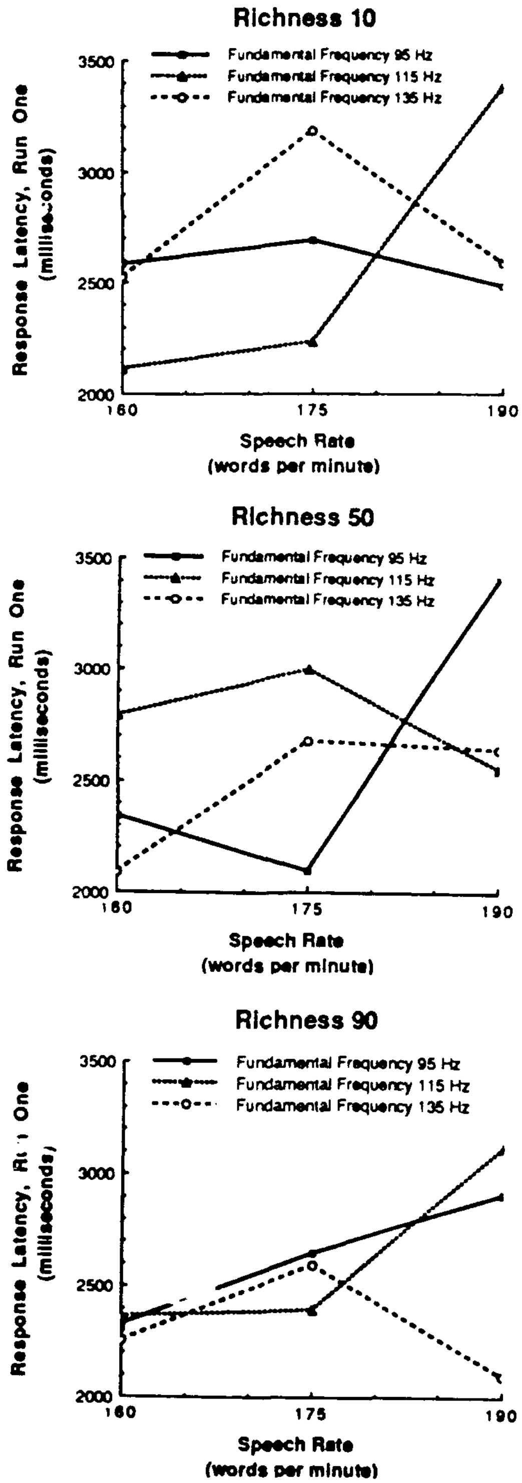

Figure 9. Response Latency Run One Three-Way Interaction of Speech Rate, Fundamental Frequency, and Richness.

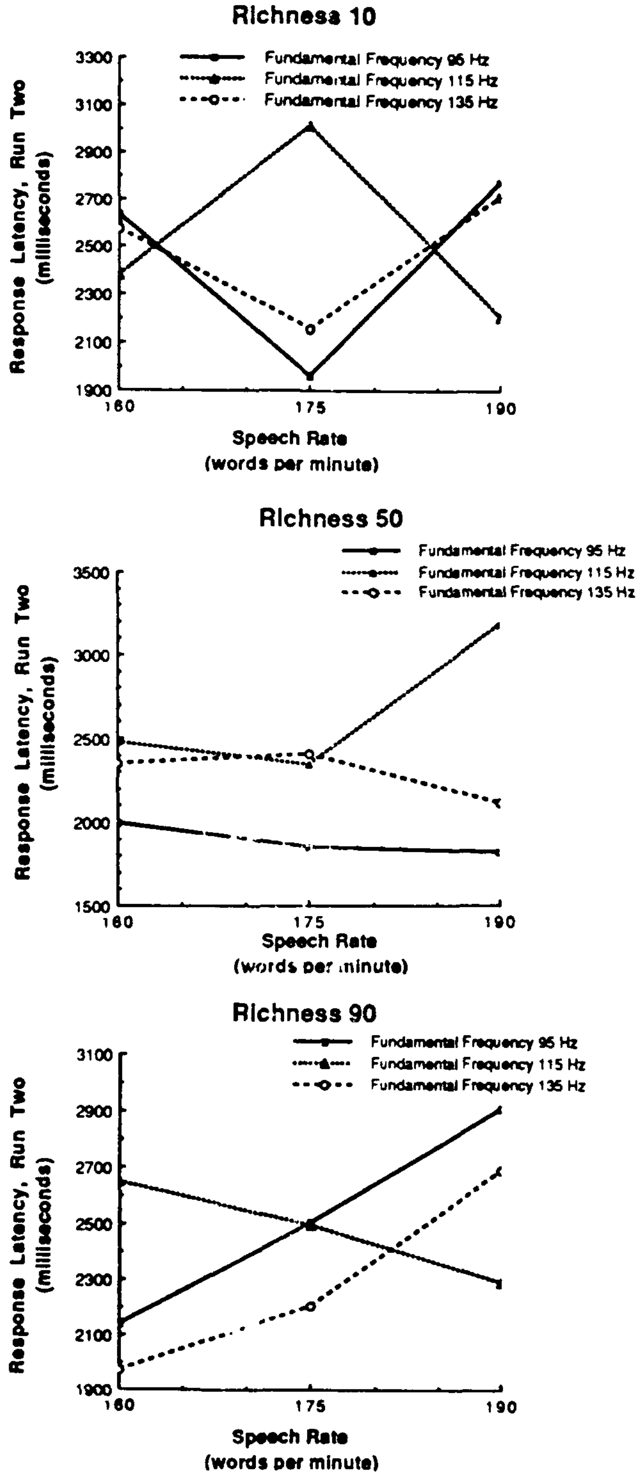

Figure 10. Response Latency Run Two Three-Way Interaction of Speech Rate, Fundamental Frequency, and Richness.

TABLE 4. ANALYSIS OF VARIANCE FOR ACCURACY FOR BOTH DATA COLLECTION RUNS COMBINED

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	6.571	10.64 †
Fundamental Frequency (FF)	2	0.5880	0.89
SR * FF	4	0.6565	0.88
Richness (Ri)	2	1.781	1.86
SR * Ri	4	0.5378	0.62
FF * Ri	4	1.062	1.53
SR * FF * Ri	8	0.4938	0.72
Subject (S)	18	3.101	
SR * S	36	0.6174	
FF * S	36	0.6628	
SR * FF * S	72	0.7496	
Ri * S	36	0.9597	
SR * Ri * S	72	0.8610	
FF * Ri * S	72	0.6948	
SR * FF * Ri * S	144	0.6902	
Run (R)	1	3.429	4.09 •
SR * R	2	0.9985	1.53
FF * R	2	1.067	1.64
SR * FF * R	4	1.501	2.27 •
Ri * R	2	0.4220	0.67
SR * Ri * R	4	0.6613	0.77
FF * Ri * R	4	0.0130	0.02
SR * FF * Ri * R	8	1.036	1.83 •
S * R	18	0.8394	
SR * S * R	36	0.6512	
FF * S * R	36	0.6498	
SR * FF * S * R	72	0.6620	
Ri * S * R	36	0.6258	
SR * Ri * S * R	72	0.8566	
FF * Ri * S * R	72	0.7752	
SR * FF * S * R	144	0.5667	

* Indicates interaction between sources
• Shows significance at the 0.1 level
† Shows significance at the 0.01 level

A Newman-Keuls test was performed to determine which of the speech rate levels were significantly different from one another. Accuracy values for speech rates of 160 words per minute and 190 words per minute were significantly different from each other at the 0.01 level of significance. The difference between mean accuracy values for speech rates of 190 and 175 words per minute was significant at the 0.05 level, as was the difference between accuracy values for 175 and 160 words per minute.

The mean accuracy values for run one and run two, as a function of the three levels of speech rate are displayed in Figure 11. Two trends are evident. First accuracy decreases with increases in speech rate. Second, accuracy generally is better for the second run than for the first.

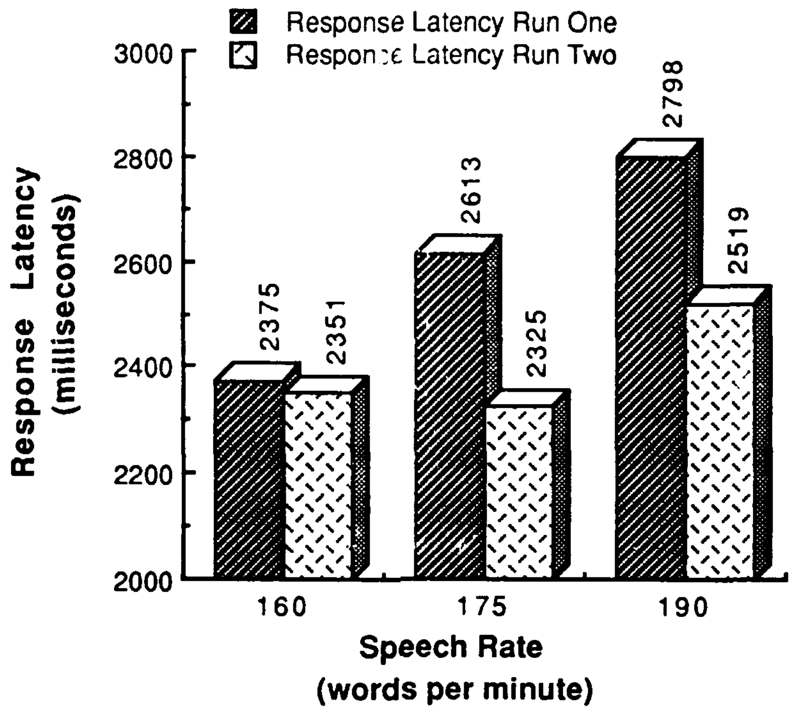

Figure 11. Comparison of Transformed Mean Accuracies for Speech Rate on Run One and Two.

Although the effects of run number and the interactions cited above were significant only at the 0.1 level, (meaning that 10 out of 100 times the same results could be obtained by chance), additional ANOVA tests were indicated since two of the three effects were also observed with the other dependent variable, response latency. Two more ANOVAs were conducted,one for each individual data collection run. The ANOVA results for run one are given in Table 5 and the results for run two are shown in Table 6. As may be observed in the latter, no main effects or interactions were found to significant at the 0.1 level or above, for run two. Table 5 indicates that, for the first run, the main effect of speech rate is significant at the 0.01 level. No other effects or interactions were significant at the 0.05 level or above.

TABLE 5. ANALYSIS OF VARIANCE FOR ACCURACY FOR DATA COLLECTION RUN ONE

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	6.31	8.55 †
Fundamental Frequency (FF)	2	0.8535	1.05
SR * FF	4	0.2500	0.31
Richness (Ri)	2	0.2578	0.28
SR * Ri	4	0.4095	0.47
FF * Ri	4	0.5808	0.67
SR * FF * Ri	8	0.5940	0.92
Subject (S)	18	1.989	
SR * S	36	0.7383	
FF * S	36	0.8136	
Sr * FF * S	72	0.8057	
Ri * S	36	0.9342	
SR * Ri * S	72	0.8631	
FF * Ri * S	72	0.8607	
SR * FF * Ri * S	144	0.6468	

* Indicates interaction between sources

† Shows significance at the 0.01 level

TABLE 6. ANALYSIS OF VARIANCE FOR ACCURACY FOR DATA COLLECTION RUN TWO

SOURCE	DEGREES OF FREEDOOM	MEAN SQUARE	F RATIO
Speech Rate (SR)	2	1.259	2.37
Fundamental Frequency (FF)	2	0.8015	1.59
SR * FF	4	0.7638	1.26
Richness (Ri)	2	1.945	2.99
SR * Ri	4	0.7895	0.92
FF * Ri	4	0.4940	0.81
SR * FF * Ri	8	0.9356	1.53
Subject (S)	18	1.952	
SR * S	36	0.5303	
FF * S	36	0.5044	
SR * FF * S	72	0.6060	
Ri * S	36	0.6514	
SR * Ri * S	72	0.8546	
FF * Ri * S	72	0.6092	
SR * FF * Ri * S	144	0.6101	

*Indicates interaction between sources

A Newman-Keuls test was performed to determine at which speech rate levels accuracy values were significantly different from each other. This test indicated that the mean accuracy values for speech rates of 160 and 190 words per minute were significantly different from each other at the 0.01 level. The mean accuracy values for speech rates of 160 and 175 were significantly different at the .05 level. The difference between mean accuracy values for 175 and 190 words per minute was not found to be significantly different at the 0.05 level or above.

For run one, the interaction of speech rate, fundamental frequency, richness, and run, as these affect accuracy, are graphically depicted in Figure 12. For ease of comparison, the graphs shown in Figures 9 and 12 are combined for Figure 13. The similarity of the trends is striking for both of the dependent variables, as a function of speech rate.

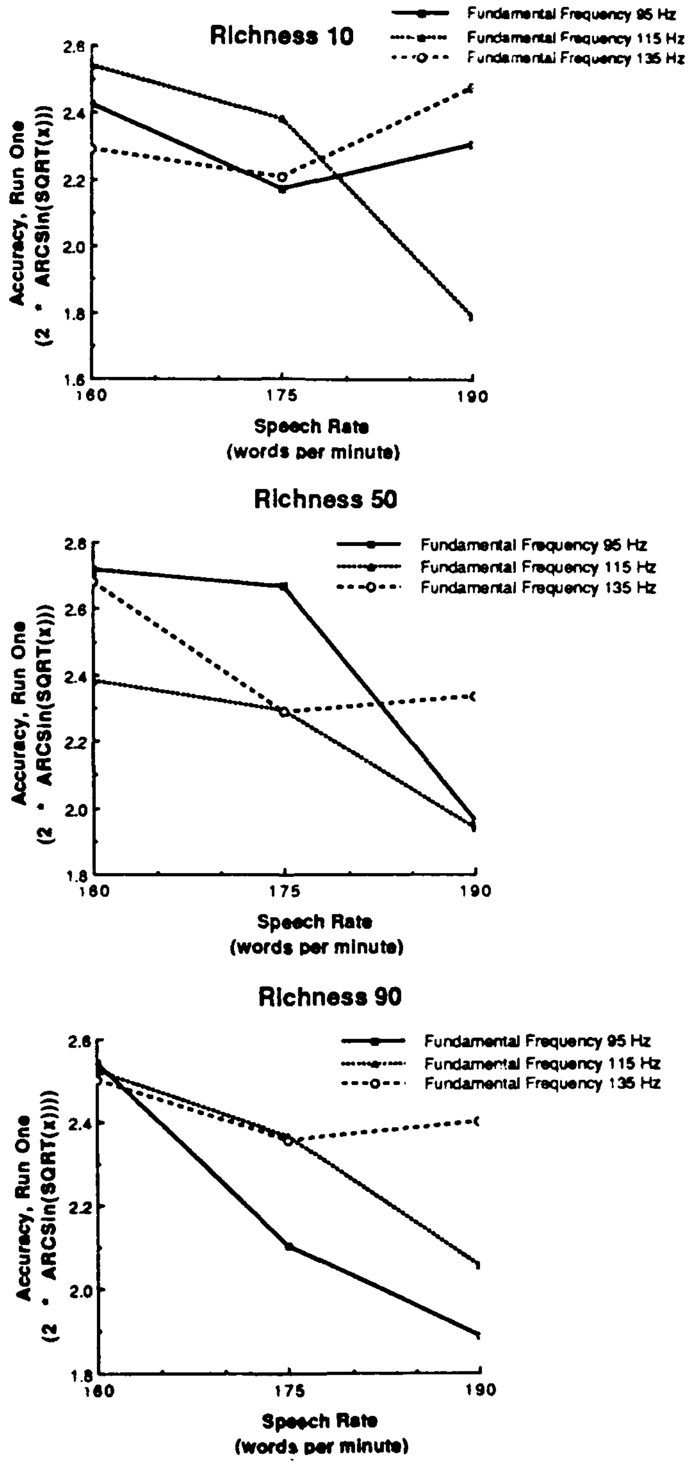

Figure 12. Accuracy Run One Three-Way Interaction of Speech Rate, Fundamental Frequency, and Richness

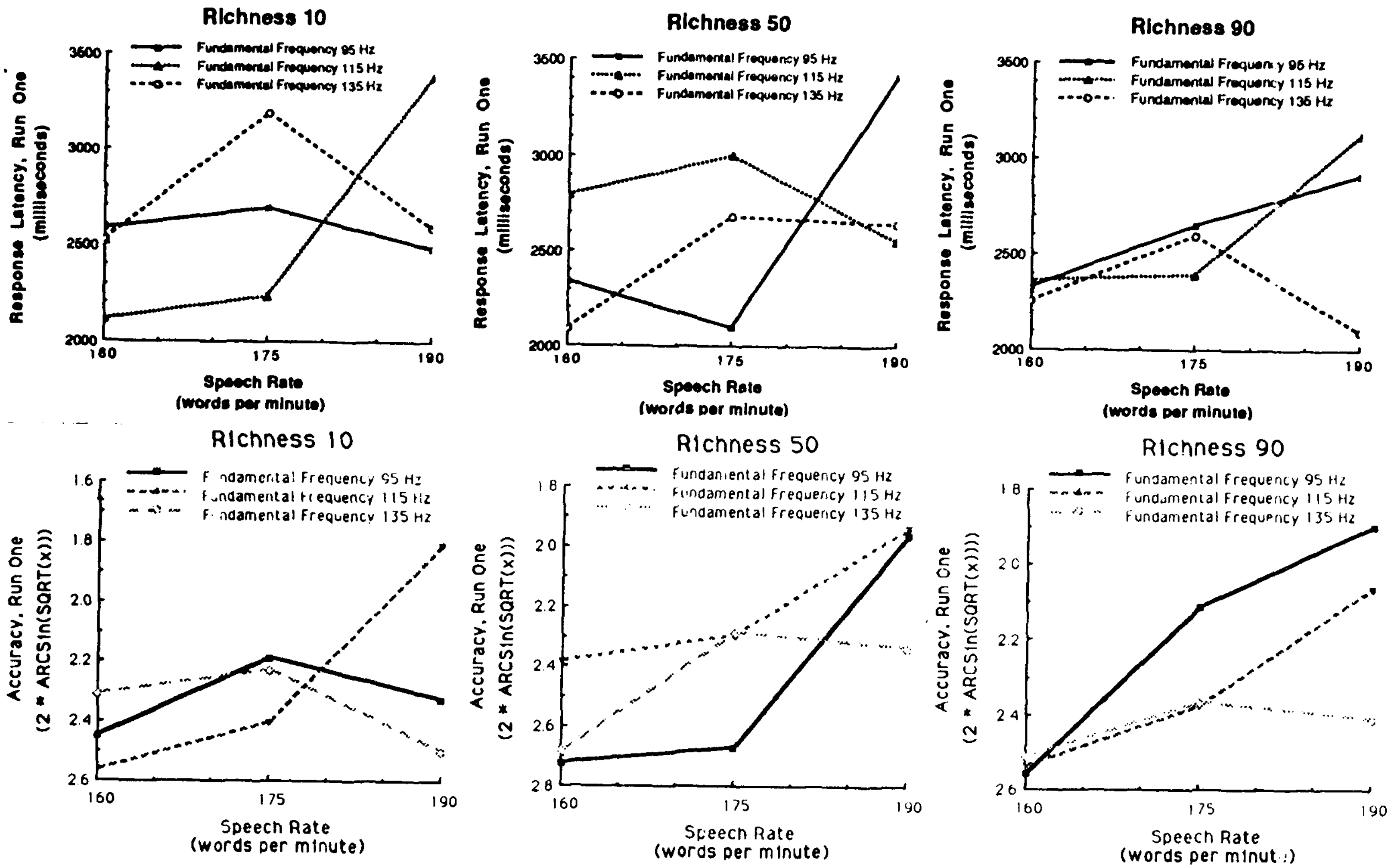

Figure 13. Comparison of Three-Way Interaction of Speech Rate, Fundamental Frequency, and Richness on Response Latency Run One and Accuracy Run One. Note that the Accuracy Graph Y-Axis has been Inverted, for Ease of Comparison.

IV. CONCLUSIONS AND RECOMMENDATIONS

The goal of this experiment was to enhance the U.S. military's understanding of factors which may affect the intelligibility of synthetic speech. The specific objectives were to gain knowledge about speech rate, voice fundamental frequency, and richness, particularly in a noisy environment. Two dependent variables, response latency and accuracy, were chosen as surrogate measurements for intelligibility.

A. EFFECT OF SPEECH RATE ON RESPONSE LATENCY AND ACCURACY

This study has demonstrated clearly that increasing speech rate leads to an increase in response latency and a decrease in accuracy, at least for the novice user in a noisy environment. Analysis of the collected data indicates a 0.01 or higher level of significance for a difference in the values of response latency and accuracy means as a function of speech rate. When the first and second data collection runs are analyzed separately, however, results indicate that differences among speech rates are significant only for run one. This indicates that considerable learning is taking place in a relatively short period of instruction--27 sentences--and that the effect of speech rate on intelligibility of synthetic speech decreases rapidly with experience.

An important question is what the exact levels of speech rate are that affect intelligibility. The results from this study are somewhat mixed. For both response latency and accuracy during run one the differences between mean values obtained at the upper and lower rates tested, 160 and 190 words per minute, are significant at the 0.01 level. Clearly, there is a difference in intelligibility as speaking speed increases from 160 to 190 words per minute.

The evidence is not so conclusive with respect to the middle rate of speech intelligibility, at 175 words per minute. When compared with the intelligibility of speech at 160 or 190 words per minute, response latency differences for run one are not significant at the 0.05 level. However, for accuracy measurements, on both runs the effect of speech rate resulted in significant differences between 160 and 175 words per minute and between 190 and 175 words per minute, at the 0.05 level. Considering run one alone, the difference in mean values for accuracy levels for 160 and 175 words per minute is significantly different at the 0.05 level, whereas the difference in means for 175 and 190 words per minute is not significantly different at the 0.05 level.

B. EFFECT OF SYNTHETIC SPEECH MESSAGES PRESENTED AT LOWER, THE SAME, AND HIGHER FREQUENCIES THAN THE BACKGROUND NOISE

The results of this study were unambiguous with respect to the effect of the pitch of the synthetic voice on both response latency and accuracy. Under the conditions of background noise and voice frequency tested, the means of response latency and accuracy were not significantly different at the 0.05 level or higher,regardless of whether the voice fundamental frequency was higher, lower, or approximately the same as the frequency of the background noise. This was true for both runs combined and for the runs separately. No effect of voice fundamental frequency on the intelligibility of synthetic speech was demonstrated.

C. EFFECT OF RICHNESS ON RESPONSE LATENCY AND ACCURACY

The experiment described here demonstrated no significant differences in response latency or accuracy at the 0.05 level or higher as a function of the richness of the synthetic voice. This was the case for data from both runs combined and from the runs analyzed separately. Under the conditions tested, richness does not appear to have an effect on the intelligibility of synthetic speech.

D. INTERACTIONS BETWEEN VOICE RICHNESS, VOICE FUNDAMENTAL FREQUENCY, AND SPEECH RATE

One significant interaction--speech rate by fundamental frequency by richness--was discovered during this experiment. In run one, the three-way interaction was found to be significant for the response latency dependent variable at the 0.05 level. With respect to accuracy on run one,no such interaction, was observed. Figure 13 compares the results of the effect of the three-way interaction on both the response latency and accuracy for run one. It may be observed that the trends are quite similar for this run, though no similar effect was found for run two. It would appear that the interaction of the three factors is significant only for the novice user. Even a minor amount of training--27 sentences--appears to negate the effects of the interaction.

E. RECOMMENDATIONS

Under the conditions of this study, speech rate has been shown to be a major factor in determining response latency and accuracy, for the novice user. However, within the range tested--160 to 190 words per minute--speech rate was not found to be a factor for the experienced user and experience seems to be gained quickly. More research is needed to determine how quickly learning takes place. It would also be very useful to determine the upper limit of speech rate that still results in intelligible speech, for an experienced user.

The relationship of voice fundamental frequency to the frequency of background noise does not appear to affect the intelligibility of synthetic speech directly in either the novice or the more experienced user. Yet, for the novice user an interaction between speech rate, fundamental frequency, and richness appears to have a rather large effect on intelligibility. The nature of this interaction is not readily apparent, partly because richness has not been defined clearly and the nature of this factor and its various effects are not known. More research is needed to shed light on the interaction, and on voice richness in general.

APPENDIX A

```
Program DecTalk;

Uses CRT,DOS,
     TPTimer,
     DecComm;

CONST MaxMsg = 100;
      MaxParm = 3;
      ComPort = 1;
      ComSpeed = 9600;

VAR OutF : Text;
    Iter,NMsg : Word;
    T1,T2,T3,T4 : LongInt;
    PC1,PC2,PC3,P1,P2,P3,Msg,Resp : String;
    MsgBase : Array[1..MaxMsg] OF ^String;
    Parm1,Parm2,Parm3 : Array[1..MaxParm] OF ^String;
    ParmUsed: Array[1..MaxParm,1..MaxParm,1..MaxParm] OF Boolean;
    firstchar : Char;
    Closed    : Boolean;
    exit_save: Pointer;

PROCEDURE Initialize;
VAR i,j,k : Word;

BEGIN
  Iter := 0;
  T1 := 0;
  T2 := 0;
  T3 := 0;
  T4 := 0;
  P1 := '';
  P2 := '';
  P3 := '';
  Msg := '';
  Resp := '';
  FOR i := 1 TO MaxMsg DO New(MsgBase[i]);
  FOR i := 1 TO MaxParm DO
    BEGIN
      New(Parm1[i]);
      New(Parm2[i]);
      New(Parm3[i]);
      FOR j := 1 TO MaxParm DO
        FOR k := 1 TO MaxParm DO ParmUsed[i,j,k] := False;
    END;
  ComInit(ComPort,ComSpeed);
  Randomize;
END;

PROCEDURE SetData;
CONST msgdb = 'SPEECH.DAT';
      parmdb = 'PARMS.DAT';

VAR   dbfile : Text;
      str : String;
      i : Word;

BEGIN
  Assign(dbfile,msgdb);
  Reset(dbfile);
```

```
  i := 1;
  REPEAT
    ReadLn(dbfile,str);
    IF NOT(Eof(dbfile)) THEN MsgBase[i]^ := str;
    Inc(i);
  UNTIL (i>MaxMsg) OR (EOF(dbfile));
  NMsg := Pred(i);
  Close(dbfile);
  Assign(dbfile,parmdb);
  Reset(dbfile);
  ReadLn(dbfile,PC1);
  FOR i := 1 TO MaxParm DO ReadLn(dbfile,Parm1[i]^);
  ReadLn(dbfile,PC2);
  FOR i := 1 TO MaxParm DO ReadLn(dbfile,Parm2[i]^);
  ReadLn(dbfile,PC3);
  FOR i := 1 TO MaxParm DO ReadLn(dbfile,Parm3[i]^);
  Close(dbfile);
END;

PROCEDURE Beep;

BEGIN
  Sound(1000);
  Delay(500);
  NoSound;
END;

PROCEDURE GetDataFileName;
VAR str : String;

BEGIN
  REPEAT
    ClrScr;
    Write('Enter subject name: ');
    ReadLn(str);
    str := Copy(str+'        ',1,8);
    IF (str='        ') THEN
      BEGIN
        Beep;
        WriteLn('Please enter a name');
      END;
  UNTIL (str<>'        ');
  Assign(OutF,str+'.EXP');
  ReWrite(OutF);
END;

PROCEDURE SelectParms;
VAR i,j,k,i1,i2,i3,j1,j2,j3,k1,k2,k3 : Word;

BEGIN
  REPEAT
    i := Succ(Random(MaxParm));
    j := Succ(Random(MaxParm));
    k := Succ(Random(MaxParm));
  UNTIL (ParmUsed[i,j,k]=False);
  P1 := Parm1[i]^;
  P2 := Parm2[j]^;
  P3 := Parm3[k]^;
  ParmUsed[i,j,k] := True;
  WriteLn(OutF,'Parm. 1: ',P1,' Parm. 2: ',P2,' Parm. 3: ',P3);
```

```
END;

PROCEDURE SelectMessage;
VAR i : Word;

BEGIN
  REPEAT
    i := Succ(Random(NMsg));
  UNTIL (MsgBase[i]^<>'');
  Msg := MsgBase[i]^;
  MsgBase[i]^ := '';
  WriteLn(OutF,'Msg: ',Msg);
END;

PROCEDURE AwaitStart;

BEGIN
  GoToXY(25,12);
  Write('Press ENTER for ');
  IF (Iter=1) THEN
    Write('first')
  ELSE IF (Iter=MaxParm*MaxParm*MaxParm) THEN
    Write('last')
  ELSE
    Write('next');
  Write(' message.');
  ReadLn;
  ClrScr;
END;

FUNCTION AwaitResponse : Char;

BEGIN
  AwaitResponse := ReadKey;
END;

PROCEDURE GetResponse(fc : Char);
CONST cr = #13;

VAR c : Char;
    s : ARRAY[1..80] OF Char;
    st : String;
    i,len : Word;

BEGIN
  s[1] := fc;
  st := fc;
  i := 2;
  Resp := '';
  GoToXY(1,23);
  Write(st);
  REPEAT
    c := ReadKey;
    s[i] := c;
    Inc(i);
    st := st + c;
    GoToXY(1,23);
    Write(st);
  UNTIL (c=cr) OR (i>80);
  len := Pred(i);
```

```
  FOR i := 1 TO Len DO
    BEGIN
      CASE s[i] OF
        #8 : s[i] := #225;
        #9 : s[i] := #231;
        #0..#31 : s[i] := #155;
        #128..#256 : s[i] := #175;
      END;
      Resp := Resp + s[i];
    END;
END;

PROCEDURE FlushKeys;
VAR c : Char;

BEGIN
  WHILE KeyPressed DO c := ReadKey;
END;

{$F+}
PROCEDURE Exit_Proc;

BEGIN
 IF NOT(Closed) THEN Close(OutF);
 ExitProc := exit_save;
END;
{$F-}


Begin
  Closed := True;
  exit_save := ExitProc;
  ExitProc := @Exit_Proc;

  ClrScr;
  Initialize;
  SetData;
  GetDataFileName;
  Iter := 1;
  Closed := False;
  REPEAT
    WriteLn(OutF,'Iter.: ',Iter);
    SelectParms;
    SelectMessage;
    AwaitStart;
    T1 := ReadTimer;
    SendMsg(Msg,PC1,P1,PC2,P2,PC3,P3);
    FlushKeys;
    T2 := ReadTimer;
    firstchar := AwaitResponse;
    T3 := ReadTimer;
    GetResponse(firstchar);
    T4 := ReadTimer;
    WriteLn(OutF,'Resp.: ',resp);
    WriteLn(OutF,'T1: ',ElapsedTimeString(T1,T2),' T2: ',ElapsedTimeString
                       (T2,T3),' T3: ',ElapsedTimeString(T3,T4));
    WriteLn(OutF);
    Inc(Iter);
  UNTIL (Iter>MaxParm*MaxParm*MaxParm);
  Close(OutF);
```

```
  Closed := True;
End.
```

```
Unit DecComm;

Interface

USES IBMCOM,TPTimer;

PROCEDURE ComInit(port,speed : Word);

PROCEDURE SendMsg(msg,PC1,P1,PC2,P2,PC3,P3 : String);

{======================================================================}
Implementation
CONST esc = #27;
      xon = #19;
      xoff = #17;
      dt_term = esc+'\';
      dt_speak = esc+'P0;12;1;z'+ dt_term;
      dt_photext = esc+'P0;0z;';
      dt_query_reply = esc + 'P0;21;40z' + dt_term;
      dt_reply = esc + 'P;31;40z' + dt_term;

Procedure ComInit(port,speed : Word);
VAR error : Word;

BEGIN
  com_install(port,error);
  IF (error<>0) THEN
    BEGIN
      WriteLn('Cannot install communications package.');
      Halt;
    END;
  com_set_speed(speed);
  com_set_parity(com_none,1);
  com_raise_dtr;
END;

PROCEDURE Send(s : String);
VAR cr,cs : Char;
    i : Word;

BEGIN
  com_flush_rx;
  FOR i := 1 TO Length(s) DO
    BEGIN
      cs := s[i];
      com_tx(cs);
      IF (com_rx=xoff) THEN
        REPEAT
        UNTIL (com_rx=xon);
    END;
END;

PROCEDURE Wait;
VAR c : Char;
    s : String;
```

```
BEGIN
  s := '';
  REPEAT
    REPEAT
      c := com_rx;
    UNTIL (c<>#0);
    s := s + c;
  UNTIL (c='\');
END;

PROCEDURE SendMsg(msg,PC1,P1,PC2,P2,PC3,P3 : String);
VAR s : String;
    ts,te : LongInt;

BEGIN
  s := dt_photext + PC1 + P1 + ':DV '+ PC2 + P2 + PC3 + P3 + dt_term;
  s := s + msg + dt_query_reply + #13;
  Send(s);
  Wait;
END;

Begin
End.
```

```
UNIT ibmcom;

{Version 3.0}

{This unit is the communications port interrupt driver for the IBM-PC.
It handles handles all low-level i/o through the serial port.  It is
installed by calling com_install.  It deinstalls itself automatically
when the program exits, or you can deinstall it by calling com_deinstall.

Donated to the public domain by Wayne E. Conrad, January, 1989.
If you have any problems or suggestions, please contact me at my BBS:

    Pascalaholics Anonymous
    (602) 484-9356
    2400 bps
    The home of WBBS
    Lots of source code
}


INTERFACE

USES
  Dos;


TYPE
  com_parity = (com_none, com_even, com_odd, com_zero, com_one);


PROCEDURE com_flush_rx;
PROCEDURE com_flush_tx;
FUNCTION  com_carrier: Boolean;
FUNCTION  com_rx: Char;
FUNCTION  com_tx_ready: Boolean;
FUNCTION  com_tx_empty: Boolean;
FUNCTION  com_rx_empty: Boolean;
PROCEDURE com_tx (ch: Char);
PROCEDURE com_tx_string (st: String);
PROCEDURE com_lower_dtr;
PROCEDURE com_raise_dtr;
PROCEDURE com_set_speed (speed: Word);
PROCEDURE com_set_parity (parity: com_parity; stop_bits: Byte);
PROCEDURE com_install
  (
  portnum  : Word;
  VAR error: Word
  );
PROCEDURE com_deinstall;


IMPLEMENTATION


{Summary of IBM-PC Asynchronous Adapter Registers.  From:
  Compute!'s Mapping the IBM PC and PCjr, by Russ Davis
  (Greensboro, North Carolina, 1985: COMPUTE! Publications, Inc.),
  pp. 290-292.

Addresses given are for COM1 and COM2, respectively.  The names given
```

in parentheses are the names used in this module.

3F8/2F8 (uart_data) Read: transmit buffer. Write: receive buffer, or baud rate divisor LSB if port 3FB, bit 7 = 1.

3F9/2F9 (uart_ier) Write: Interrupt enable register or baud rate divisor MSB if port 3FB, bit 7 = 1.
PCjr baud rate divisor is different from other models;
clock input is 1.7895 megahertz rather than 1.8432 megahertz.
Interrupt enable register:

```
    bits 7-4  forced to 0
    bit 3     1=enable change-in-modem-status interrupt
    bit 2     1=enable line-status interrupt
    bit 1     1=enable transmit-register-empty interrupt
    bit 0     1=data-available interrupt
```

3FA/2FA (uart_iir) Interrupt identification register (prioritized)

```
    bits 7-3  forced to 0
    bits 2-1  00=change-in-modem-status (lowest)
    bits 2-1  01=transmit-register-empty (low)
    bits 2-1  10=data-available (high)
    bits 2-1  11=line status (highest)
    bit 0     1=no interrupt pending
    bit 0     0=interrupt pending
```

3FB/2FB (uart_lcr) Line control register

```
    bit 7   0=normal, 1=address baud rate divisor registers
    bit 6   0=break disabled, 1=enabled
    bit 5   0=don't force parity
            1=if bit 4-3=01 parity always 1
              if bit 4-3=11 parity always 0
              if bit 3=0 no parity
    bit 4   0=odd parity,1=even
    bit 3   0=no parity,1=parity
    bit 2   0=1 stop bit
            1=1.5 stop bits if 5 bits/character or
              2 stop bits if 6-8 bits/character
    bits 1-0  00=5 bits/character
              01=6 bits/character
              10=7 bits/character
              11=8 bits/character

    bits 5..3: 000 No parity
               001 Odd parity
               010 No parity
               011 Even parity
               100 No parity
               101 Parity always 1
               110 No parity
               111 Parity always 0
```

3FC/2FC (uart_mcr) Modem control register

```
    bits 7-5  forced to zero
    bit 4     0=normal, 1=loop back test
    bits 3-2  all PCs except PCjr
    bit 3     1=interrupts to system bus, user-designated output: OUT2
    bit 2     user-designated output, OUT1
    bit 1     1=activate rts
```

```
     bit 0     1=activate dtr

3FD/2FD (uart_lsr) Line status register
     bit 7  forced to 0
     bit 6  1=transmit shift register is empty
     bit 5  1=transmit hold register is empty
     bit 4  1=break received
     bit 3  1=framing error received
     bit 2  1=parity error received
     bit 1  1=overrun error received
     bit 0  1=data received

3FE/2FE (uart_msr) Modem status register
     bit 7  1=receive line signal detect
     bit 6  1=ring indicator (all PCs except PCjr)
     bit 5  1=dsr
     bit 4  1=cts
     bit 3  1=receive line signal detect has changed state
     bit 2  1=ring indicator has changed state (all PCs except PCjr)
     bit 1  1=dsr has changed state
     bit 0  1=cts has changed state

3FF/2FF (uart_spr) Scratch pad register.}


{Maximum port number (minimum is 1) }

CONST
  max_port = 4;


{Base i/o address for each COM port}

CONST
  uart_base: ARRAY [1..max_port] OF Integer = ($3F8, $2F8, $3E8, $2E8);


{Interrupt numbers for each COM port}

CONST
  intnums: ARRAY [1..max_port] OF Byte = ($0C, $0B, $0C, $0B);


{i8259 interrupt levels for each port}

CONST
  i8259levels: ARRAY [1..max_port] OF Byte = (4, 3, 4, 3);


{This variable is TRUE if the interrupt driver has been installed, or FALSE
if it hasn't.  It's used to prevent installing twice or deinstalling when not
installed.}

CONST
  com_installed: Boolean = False;


{UART i/o addresses.  Values depend upon which COMM port is selected.}

VAR
```

```
  uart_data: Word;              {Data register}
  uart_ier : Word;              {Interrupt enable register}
  uart_iir : Word;              {Interrupt identification register}
  uart_lcr : Word;              {Line control register}
  uart_mcr : Word;              {Modem control register}
  uart_lsr : Word;              {Line status register}
  uart_msr : Word;              {Modem status register}
  uart_spr : Word;              {Scratch pad register}

{Original contents of IER and MCR registers.  Used to restore UART
to whatever state it was in before this driver was loaded.}

VAR
  old_ier: Byte;
  old_mcr: Byte;


{Original contents of interrupt vector.  Used to restore the vector when
the interrupt driver is deinstalled.}

VAR
  old_vector: Pointer;


{Original contents of interrupt controller mask.  Used to restore the
bit pertaining to the comm controller we're using.}

VAR
  old_i8259_mask: Byte;


{Bit mask for i8259 interrupt controller}

VAR
  i8259bit: Byte;


{Interrupt vector number}

VAR
  intnum: Byte;


{Receive queue.  Received characters are held here until retrieved by
com_rx.}

CONST
  rx_queue_size = 128;    {Change to suit}
VAR
  rx_queue: ARRAY [1..rx_queue_size] OF Byte;
  rx_in   : Word;         {Index of where to store next character}
  rx_out  : Word;         {Index of where to retrieve next character}
  rx_chars: Word;         {Number of chars in queue}


{Transmit queue.  Characters to be transmitted are held here until the
UART is ready to transmit them.}

CONST
```

```
  tx_queue_size = 16;      {Change to suit}
VAR
  tx_queue: ARRAY [1..tx_queue_size] OF Byte;
  tx_in    : Integer;      {Index of where to store next character}
  tx_out   : Integer;      {Index of where to retrieve next character}
  tx_chars: integer;       {Number of chars in queue}


{This variable is used to save the next link in the "exit procedure"
chain.}

VAR
  exit_save: Pointer;


{$I ints.inc}    {Macros for enabling and disabling interrupts}


{Interrupt driver.  The UART is programmed to cause an interrupt whenever
a character has been received or when the UART is ready to transmit another
character.}

{$R-,S-}
PROCEDURE com_interrupt_driver; INTERRUPT;

VAR
  ch    : Char;
  iir   : Byte;
  dummy: Byte;

BEGIN

  {While bit 0 of the interrupt identification register is 0, there is an
  interrupt to process}

  iir := Port [uart_iir];

  WHILE NOT Odd (iir) DO
    BEGIN

    CASE iir SHR 1 OF

      {iir = 100b: Received data available.  Get the character, and if
      the buffer isn't full, then save it.  If the buffer is full,
      then ignore it.}

      2:
        BEGIN
        ch := Char (Port [uart_data] );
        IF (rx_chars <= rx_queue_size) THEN
          BEGIN
          rx_queue [rx_in] := Ord (ch);
          Inc (rx_in);
          IF rx_in > rx_queue_size THEN
            rx_in := 1;
          rx_chars := Succ (rx_chars);
          END;
        END;

      {iir = 010b: Transmit register empty.  If the transmit buffer
```

```
    is empty, then disable the transmitter to prevent any more
    transmit interrupts.  Otherwise, send the character.

    The test of the line-status-register is to see if the transmit
    holding register is truly empty.  Some UARTS seem to cause transmit
    interrupts when the holding register isn't empty, causing transmitted
    characters to be lost.}

    1:
      IF (tx_chars <= 0) THEN
        Port [uart_ier] := Port [uart_ier] AND NOT 2
      ELSE
        IF Odd (Port [uart_lsr] SHR 5) THEN
          BEGIN
          Port [uart_data] := tx_queue [tx_out];
          Inc (tx_out);
          IF tx_out > tx_queue_size THEN
            tx_out := 1;
          Dec (tx_chars);
          END;

    {iir = 001b: Change in modem status.  We don't expect this interrupt,
    but if one ever occurs we need to read the line status to reset it
    and prevent an endless loop.}

    0:
      dummy := Port [uart_msr];

    {iir = 110b: Change in line status.  We don't expect this interrupt,
    but if one ever occurs we need to read the line status to reset it
    and prevent an endless loop.}

    3:
      dummy := Port [uart_lsr];

    END;

  iir := Port [uart_iir];
  END;

 {Tell the interrupt controller that we're done with this interrupt}

 Port [$20] := $20;

END;
{$R+,S+}


{Flush (empty) the receive buffer.}

PROCEDURE com_flush_rx;
BEGIN
  disable_interrupts;
  rx_chars := 0;
  rx_in    := 1;
  rx_out   := 1;
  enable_interrupts;
END;
```

```
{Flush (empty) transmit buffer.}

PROCEDURE com_flush_tx;
BEGIN
  disable_interrupts;
  tx_chars := 0;
  tx_in    := 1;
  tx_out   := 1;
  enable_interrupts;
END;


{This function returns TRUE if a carrier is present.}

FUNCTION com_carrier: Boolean;
BEGIN
  com_carrier := com_installed AND Odd (Port [uart_msr] SHR 7);
END;


{Get a character from the receive buffer.  If the buffer is empty, return
a NULL (#0).}

FUNCTION com_rx: Char;
BEGIN
  IF NOT com_installed OR (rx_chars = 0) THEN
    com_rx := #0
  ELSE
    BEGIN
    disable_interrupts;
    com_rx := Chr (rx_queue [rx_out] );
    Inc (rx_out);
    IF rx_out > rx_queue_size THEN
      rx_out := 1;
    Dec (rx_chars);
    enable_interrupts;
    END;
END;


{This function returns True if com_tx can accept a character.}

FUNCTION com_tx_ready: Boolean;
BEGIN
  com_tx_ready := (tx_chars < tx_queue_size) OR NOT com_installed;
END;


{This function returns True if the transmit buffer is empty.}

FUNCTION com_tx_empty: Boolean;
BEGIN
  com_tx_empty := (tx_chars = 0) OR NOT com_installed;
END;


{This function returns True if the receive buffer is empty.}

FUNCTION com_rx_empty: Boolean;
BEGIN
```

```
  com_rx_empty := (rx_chars = 0) OR NOT com_installed;
END;


{Send a character.  Waits until the transmit buffer isn't full, then puts
the character into it.  The interrupt driver will send the character
once the character is at the head of the transmit queue and a transmit
interrupt occurs.}

PROCEDURE com_tx (ch: Char);
BEGIN
  IF com_installed THEN
    BEGIN
    REPEAT UNTIL com_tx_ready;
    disable_interrupts;
    tx_queue [tx_in] := Ord (ch);
    IF tx_in < tx_queue_size THEN
      Inc (tx_in)
    ELSE
      tx_in := 1;
    Inc (tx_chars);
    Port [uart_ier] := Port [uart_ier] OR 2;
    enable_interrupts;
    END;
END;


{Send a whole string}

PROCEDURE com_tx_string (st: String);
VAR
  i: Byte;
BEGIN
  FOR i := 1 TO Length (st) DO
    com_tx (st [i] );
END;


{Lower (deactivate) the DTR line.  Causes most modems to hang up.}

PROCEDURE com_lower_dtr;
BEGIN
  IF com_installed THEN
    BEGIN
    disable_interrupts;
    Port [uart_mcr] := Port [uart_mcr] AND NOT 1;
    enable_interrupts;
    END;
END;


{Raise (activate) the DTR line.}

PROCEDURE com_raise_dtr;
BEGIN
  IF com_installed THEN
    BEGIN
    disable_interrupts;
    Port [uart_mcr] := Port [uart_mcr] OR 1;
    enable_interrupts;
```

```
    END;
END;


{Set the baud rate.  Accepts any speed between 2 and 65535.  However,
I am not sure that extremely high speeds (those above 19200) will
always work, since the baud rate divisor will be six or less, where a
difference of one can represent a difference in baud rate of
3840 bits per second or more.}

PROCEDURE com_set_speed (speed: Word);
VAR
  divisor: Word;
BEGIN
  IF com_installed THEN
    BEGIN
    IF speed < 2 THEN speed := 2;
    divisor := 115200 DIV speed;
    disable_interrupts;
    Port  [uart_lcr]  := Port [uart_lcr] OR $80;
    Portw [uart_data] := divisor;
    Port  [uart_lcr]  := Port [uart_lcr] AND NOT $80;
    enable_interrupts;
    END;
END;


{Set the parity and stop bits as follows:

  com_none     8 data bits, no parity
  com_even     7 data bits, even parity
  com_odd      7 data bits, odd parity
  com_zero     7 data bits, parity always zero
  com_one      7 data bits, parity always one}

PROCEDURE com_set_parity (parity: com_parity; stop_bits: Byte);
VAR
  lcr: Byte;
BEGIN
  CASE parity OF
    com_none: lcr := $00 OR $03;
    com_even: lcr := $18 OR $02;
    com_odd : lcr := $08 OR $02;
    com_zero: lcr := $38 OR $02;
    com_one : lcr := $28 OR $02;
    END;
  IF stop_bits = 2 THEN
    lcr := lcr OR $04;
  disable_interrupts;
  Port [uart_lcr] := Port [uart_lcr] AND $40 OR lcr;
  enable_interrupts;
END;

{Install the communications driver.  Portnum should be 1..max_port.
Error codes returned are:

  0 - No error
  1 - Invalid port number
  2 - UART for that port is not present
  3 - Already installed, new installation ignored}
```

```
PROCEDURE com_install
  (
  portnum  : Word;
  VAR error: Word
  );
VAR
  ier: Byte;
BEGIN
  IF com_installed THEN
    error := 3
  ELSE
    IF (portnum < 1) OR (portnum > max_port) THEN
      error := 1
    ELSE
      BEGIN

      {Set i/o addresses and other hardware specifics for selected port}

      uart_data := uart_base [portnum];
      uart_ier  := uart_data + 1;
      uart_iir  := uart_data + 2;
      uart_lcr  := uart_data + 3;
      uart_mcr  := uart_data + 4;
      uart_lsr  := uart_data + 5;
      uart_msr  := uart_data + 6;
      uart_spr  := uart_data + 7;
      intnum    := intnums [portnum];
      i8259bit  := 1 SHL i8259levels [portnum];

      {Return error if hardware not installed}

      old_ier := Port [uart_ier];
      Port [uart_ier] := 0;
      IF Port [uart_ier] <> 0 THEN
        error := 2
      ELSE
        BEGIN
        error := 0;

        {Save original interrupt controller mask, then disable the
        interrupt controller for this interrupt.}

        disable_interrupts;
        old_i8259_mask := Port [$21];
        Port [$21] := old_i8259_mask OR i8259bit;
        enable_interrupts;

        {Clear the transmit and receive queues}

        com_flush_tx;
        com_flush_rx;

        {Save current interrupt vector, then set the interrupt vector to
        the address of our interrupt driver.}

        GetIntVec (intnum, old_vector);
        SetIntVec (intnum, @com_interrupt_driver);
        com_installed := True;
```

```
          {Set parity to none, turn off BREAK signal, and make sure
          we're not addressing the baud rate registers.}

          Port [uart_lcr] := 3;

          {Save original contents of modem control register, then enable
          interrupts to system bus and activate RTS.  Leave DTR the way
          it was.}

          disable_interrupts;
          old_mcr := Port [uart_mcr];
          Port [uart_mcr] := $A OR (old_mcr AND 1);
          enable_interrupts;

          {Enable interrupt on data-available.  The interrupt for
          transmit-ready is enabled when a character is put into the
          transmit queue, and disabled when the transmit queue is empty.}

          Port [uart_ier] := 1;

          {Enable the interrupt controller for this interrupt.}

          disable_interrupts;
          Port [$21] := Port [$21] AND NOT i8259bit;
          enable_interrupts;

          END;
        END;
END;


{Deinstall the interrupt driver completely.  It doesn't change the baud
rate or mess with DTR; it tries to leave the interrupt vectors and
enables and everything else as it was when the driver was installed.

This procedure MUST be called by the exit procedure of this module before
the program exits to DOS, or the interrupt driver will still
be attached to its vector -- the next communications interrupt that came
along would jump to the interrupt driver which is no longer protected and
may have been written over.}


PROCEDURE com_deinstall;
BEGIN
  IF com_installed THEN
    BEGIN

    com_installed := False;

    {Restore Modem-Control-Register and Interrupt-Enable-Register.}

    Port [uart_mcr] := old_mcr;
    Port [uart_ier] := old_ier;

    {Restore appropriate bit of interrupt controller's mask}

    disable_interrupts;
    Port [$21] := Port [$21] AND NOT i8259bit OR
     old_i8259_mask AND i8259bit;
    enable_interrupts;
```

```
    {Reset the interrupt vector}

    SetIntVec (intnum, old_vector);

    END;
END;


{This procedure is called when the program exits for any reason.  It
deinstalls the interrupt driver.}

{$F+} PROCEDURE exit_procedure; {$F-}
BEGIN
  com_deinstall;
  ExitProc := exit_save;
END;


{This installs the exit procedure.}

BEGIN
  exit_save := ExitProc;
  ExitProc := @exit_procedure;
END.
```

LIST OF REFERENCES

Allen, J., "Linguistic-based Algorithms Offer Practical Text-to-Speech Systems," *Speech Technology*, v. 3, pp.12-16, 1981.

Anderson, V. L. and McLean, R. A., *Design of Experiments: A Realistic Approach*, Dekker, Inc., p 112, 1974.

Cater, J. P., Electronically Speaking: Computer Speech Generation, Howard W. Sams & Co., Inc., 1983.

Chilcote, J., "Investigating Voice I/O for the Airborne Warning and Control System (AWACS)," *Proceedings of the Military and Government Speech Technology*, pp. 47-52, 1989.

DeHaemer, M. J., "The Use of Protocol Analysis to Determine the Effects of Adding Synthesized Voice to Computer Graphics," *Proceedings of the American Voice Input/Output Society Voice I/O Systems Applications Conference*, pp. 242-246, 1989.

Digital Equipment Corporation, *DECtalk User Manual*, 1985.

Digital Equipment Corporation, *Straight Talk*, v. 2, p. 6, 1985.

Egan, J. P., "Articulation Testing Methods," *Laryngoscope*, v. 58, pp. 955-991, 1948.

Galletti, I., and Abbott, M., "Advanced Airborne Speech Recogniser," Proceedings of the American Voice Input/Output Society Voice I/O Systems Application Conference," pp. 127-135, 1989.

Hakkinen, M. T., and Williges, B. H., "Synthesized Warning Messages: Effects of an Alerting Cue in Single- and Multiple-Function Voice Synthesis Systems," *Human Factors*, v. 26, pp. 185-195, 1984.

Hicks, C. R., *Fundamental Concepts in the Design of Experiments*, Holt, Rinehart and Winston, Inc., pp. 2-10, 31-36, 173, 1973.

Indiana University Department of Psychology Speech Research Laboratory Report No. 10, *Perceptual Evaluation of DECtalk: A Final Report on Version 1.8*, by B. G. Greene, L. M. Manous, and D. B. Pisoni, p.16, 1984.

Inglis, A. F., *Electronic Communications Handbook*, McGraw-Hill, Inc., p. 11-12, 1988.

Johnson, R. R., and Nunn, S. W., "Voice I/O Applications at NOSC," Proceedings of Speech Technology 1986, Voice Input/Output Applications Show and Conference, pp. 255-257, 1986.

Kitfield, J., "Danger Inside the Cockpit," *Military Forum*, pp. 34-64, May 1989.

Marics, M. A., and Williges, B. H., "The Intelligibility of Synthesized Speech in Data Inquiry Systems," *Human Factors*, v. 30, pp. 719-732, 1988.

Mollakarimi, C. L., and Hamid, T. S., "Remote Voice Training: A Case Study on Space Shuttle Applications," *Proceedings of the American Voice Input/Output Society Voice I/O Systems Applications Conference*, pp. 257-263, 1989.

Partridge, A., "A Review of Some Major European Military Speech Technology Programmes," *Proceedings of the Military and Government Speech Technology*, pp. 218-222, 1989.

Pratt, R. L., "Quantifying the Performance of Text-to-Speech Synthesizers," *Speech Technology*, v. 3, pp. 54-64, 1987.

Salisbury, M. W., "A Method for Designing Command, Control, Communications, and Intelligence (C^3I) Systems with Voice Input/Output," *Proceedings of the American Voice Input/Output Society Voice I/O Systems Applications Conference*, pp.155-160, 1989.

Simpson, C. A., and Marchionda-Frost, K., "Synthesized Speech Rate and Pitch Effects on Intelligibility of Warning Messages for Pilots," *Human Factors*, v. 26, pp. 509-517, 1984.

Slowiaczek, L. M., and Nusbaum, H. C., "Effects of Speech Rate and Pitch Contour on the Perception of Synthetic Speech," *Human Factors*, v. 27, pp. 701-712, 1985.

Stanley, W. D., *Electronic Communications Systems*, Prentice-Hall, Inc., pp. 262-310, 1982.

Teja, E. R., and Gonnella, G. W., *Voice Technology*, Reston Publishing Company, Inc., pp 14-31, 1983.

Telephone conversation between T. Vitale, Digital Equipment Corporation, and J. C. Munlin, 15 November 1989.

Verbex Voice Systems, Inc., *About Verbex Voice Systems, Inc.*, 1988.

Williges, B. H., and Williges, R. C., "Structuring Human/Computer Dialogue Using Speech Technology," *Workshop on Standardization for Speech I/O (Input/Output)Technology*, pp.145-150, 1982.

Winer, B. J., *Statistical Principles in Experimental Design*, McGraw-Hill Book Company, pp. 11, 309-321, 1971.

INITIAL DISTRIBUTION LIST

No. of Copies

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

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

3. Prof. Carl R. Jones, Code CC 1
   C3 Academic Group
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Professor Judith H. Lind, Code OR/Li 4
   Department of Operation Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Professor Gary K. Poock, Code OR/Pk 2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. LT. Joyce Munlin 4
   Naval Communications Unit
   East Machias, ME 04630

7. Tony Vitale 1
   Digital Equipment Corporation
   30 Forbes Rd.
   NR05/I4
   North Boro, MA 01532