ESD-TR-72-168

Semiannual Technical Summary

Speech

31 May 1972

Prepared for the Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-70-C-0230 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

22

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified	2b. GROUP None
3. REPORT TITLE Speech		
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Semiannual Technical Summary, 1 December 1971 through 31 May 1972		
5. AUTHOR(S) *(Last name, first name, initial)* Forgie, James W.		
6. REPORT DATE 31 May 1972	7a. TOTAL NO. OF PAGES 24	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. F19628-70-C-0230	9a. ORIGINATOR'S REPORT NUMBER(S) Semiannual Technical Summary, 31 May 1972	
b. PROJECT NO. ARPA Order 2006	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-72-168	
c.		
d.		
10. AVAILABILITY/LIMITATION NOTICES Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency, Department of Defense	

13. ABSTRACT

Programs for processing speech waveforms to yield spectral analysis and phonetic segmentation have been under development on the FDP-Univac 1219 facility. Available outputs now include: (1) homomorphic (cepstrally smoothed) spectra; (2) a set of parameters extracted from the spectra; (3) some preliminary segmentation markers; (4) linear predictive coding coefficients, spectra, and formant estimates; and (5) a fundamental frequency estimate. Current efforts are directed toward refining the segmentation algorithm.

Implementation of the supporting software for the TX-2 Speech Data Base is proceeding well. Many of the required modules are operational, and others are being checked out. A multiple-user network server has been designed to allow network access to the data base without requiring normal log-in to TX-2. The Hughes LCSC-1 scan converter is being satisfactorily used to produce speech spectrograms.

The TSP hardware is now operating well enough to pass acceptance test, and emphasis is shifting from hardware checkout to system programming.

The BCPL compiler on TX-2 has been extended to allow conditional compilation and to supply a symbol table for a new symbolic debugger which has become operational. Software for the Xerox LDX printer is operational, and the printer is being used extensively with a variety of character fonts.

14. KEY WORDS

man-machine communications	computer graphics	ARPA Network
speech understanding systems	Terminal Support Processor (TSP)	TX-2 computer
LDX printer		

1a

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

SPEECH

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
ADVANCED RESEARCH PROJECTS AGENCY

1 DECEMBER 1971 – 31 MAY 1972

ISSUED 11 JULY 1972

LEXINGTON MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Advanced Research Projects Agency of the Department of Defense under Air Force Contract F19628-70-C-0230 (ARPA Order 2006).

SUMMARY

Programs for processing speech waveforms to yield spectral analysis and phonetic segmentation have been under development on the FDP-Univac 1219 facility. Available outputs now include: (1) homomorphic (cepstrally smoothed) spectra; (2) a set of parameters extracted from the spectra; (3) some preliminary segmentation markers; (4) linear predictive coding coefficients, spectra, and formant estimates; and (5) a fundamental frequency estimate. The spectra and parameters can be displayed conveniently for evaluation or threshold setting. Current efforts are directed toward refining the segmentation algorithm.

Implementation of the supporting software for the TX-2 Speech Data Base is proceeding well. Many of the required modules are operational, and others are being checked out. A multiple-user network server has been designed to allow network access to the data base without requiring normal log-in to TX-2. The Hughes LCSC-1 scan converter is being satisfactorily used to produce speech spectrograms, and will be further integrated into the TX-2 system.

The TSP hardware is now operating well enough to pass acceptance test, and emphasis is shifting from hardware checkout to system programming. The system software modules being implemented first are the overall system monitor and the keyboard-echo process.

TX-2 activities have included further changes and extensions to the BCPL compiler to allow conditional compilation and to supply a symbol table for a new symbolic debugger which has become operational. Hardware changes include an increase in the number of cycle-stealing IO channels, and changes in the address transformation hardware to accommodate additional main core memory. Software for the Xerox LDX printer is operational, and the printer is being used extensively with a variety of character fonts.

Accepted for the Air Force
Joseph R. Waterman, Lt. Col., USAF
Chief, Lincoln Laboratory Project Office

CONTENTS

Preceding page blank

GLOSSARY

APEX	The TX-2 time-sharing system
BCPL	Basic Combined Programming Language – an intermediate-level language for computer programming
FDP	Fast Digital Processor – a Lincoln Laboratory computer designed for waveform processing applications
SPC	Speech Processing Controller – a sub-operating system supporting speech programming on TX-2
TAP	A TX-2 assembler producing relocatable binary code compatible with BCPL
TELNET	The software which allows a console on one network computer to function as the console for another
TSP	Terminal Support Processor

SPEECH

I. SPEECH

Current work on speech understanding systems is primarily concerned with the development of algorithms for achieving phonetic recognition, and with the development of software to support a large Speech Data Base which is intended to serve the needs of the phonetic recognition effort as well as the requirements of other ARPA contractors. The recognition work is currently centered on the Laboratory's Fast Digital Processor (FDP), while the data base is being built on TX-2, which has a connection to the ARPA network. TX-2 will also handle the more complex recognition logic and the linguistic processing required to achieve a complete speech understanding system. We expect that a direct connection between these facilities will be established later in the program, but with the recent addition of a 7-track tape unit to TX-2, communication via digital tapes appears adequate for near-term needs.

In the linguistic area, we have chosen a task domain for our experimental speech understanding system and have begun the design of some experiments to assess the value of linguistic constraints in overcoming the errors and ambiguities to be expected in the output of any phonetic recognition subsystem. The task domain will be the vocal command of the Lincoln speech data analysis and retrieval system. This task combines the properties of a command and control system with some aspects of an information retrieval system. Techniques developed for this domain should be applicable to other problem areas by changing the data base and those vocabulary elements directly related to the data base contents. By using the Lincoln Speech Data Base as the task domain data base, we expect to achieve a potentially useful short-term system which can be evaluated in a real-world context by speech workers from our project or other ARPA projects. The use of speech data also avoids the expenditure of extra resources to build up some other data base and to invent likely commands and retrieval requests for a less-understood application.

The following two sections discuss the current status of the work on phonetic recognition and data base development. Further discussion of the task domain and linguistic experiments will be deferred until the next report in this series.

A. Waveform Analysis and Segmentation

Processing of the speech waveform is being carried out on the FDP-1219 computer facility. Available outputs of this processing now include: (1) homomorphic (cepstrally smoothed) spectra; (2) a set of parameters extracted from the homomorphic spectra; (3) some rather preliminary segmentation markers based on these parameters; (4) linear predictive coding coefficients, and spectra and parameters (including formant estimates) derived from these coefficients; and (5) a fundamental frequency estimate. The spectra and parameters can be displayed conveniently for evaluation or for experimental setting of thresholds. Current efforts are directed toward refining the segmentation program; new waveform measurements are added as they are required for segmentation. This section will begin with a brief description of the FDP-1219 computer facility, and proceed to a discussion of the specific speech processing algorithms.

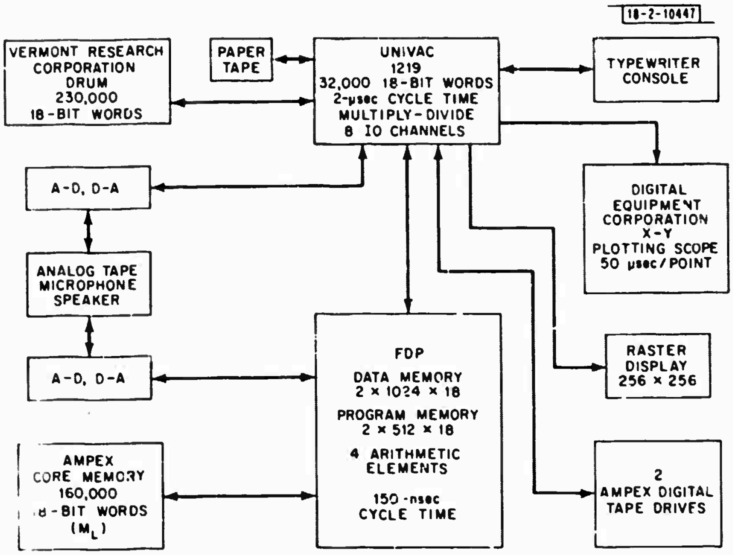

Fig. 1. FDP-1219 speech processing facility.

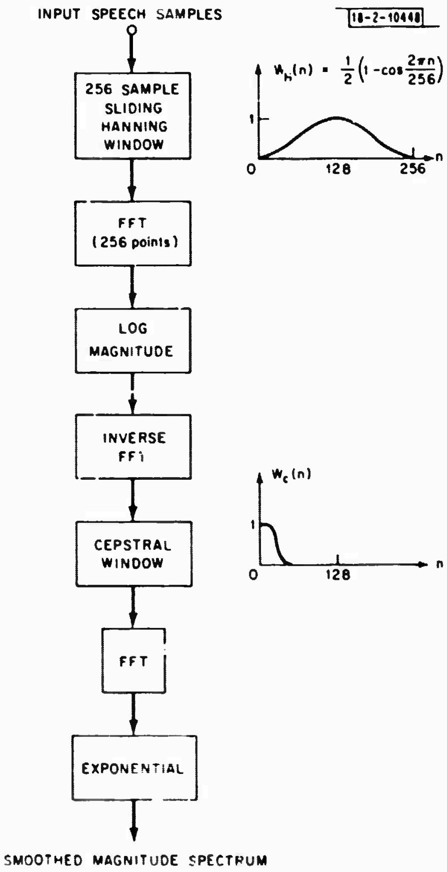

Fig. 2. Block diagram of homomorphic spectral analysis system.

1. The FDP-1219 Facility

The important components of the speech processing facility, and their interconnections, are indicated in Fig. 1. The heart of the facility is the FDP,[1] a general-purpose computer designed and constructed at Lincoln Laboratory to have special capability for fast execution of signal processing algorithms. The speed of the FDP is such that either the homomorphic or linear predictive spectrum analysis can be performed in real time. The Univac 1219 controls the loading of the various programs from the drum, services the displays, and is utilized for editing, assembling, and debugging.

The uses of the various components of the facility are best illustrated by example. In a typical running session, the procedure is as follows. The necessary programs are transferred from a digital tape, through the 1219, and to the drum for ready access. The initial 1219 and FDP programs are loaded and started. Speech is played from the analog tape, through the A-D, and into the FDP where (for example) a homomorphic spectrogram is computed in real time and stored in the Ampex core memory (M_L) which is a peripheral to the FDP. A second FDP program is then loaded automatically under 1219 control, and the spectrogram is processed to extract pertinent parameters including segmentation markers. These data are both stored in M_L and sent to 1219 for display. For displaying spectrograms and time-aligned measurements above the spectrograms, a 256- × 256-point raster display, which refreshes 10 times per second, is available. Display of spectral cross sections is more convenient on the DEC X-Y point plotting scope. An alternative to the above mode of operation is for the speech input to come from digital tape and flow through the 1219 into the FDP. Digitized speech data from the data base will be handled in this way.

2. Homomorphic Spectrum Analysis

In homomorphic spectrum analysis, smoothed spectral cross sections are obtained by windowing the cepstrum to eliminate the effects of the excitation function.[2] A block diagram of the analysis system implemented in the FDP is shown in Fig. 2. The input speech is passed through a 6-dB/octave pre-emphasis filter and a 5-kHz cutoff low-pass filter, sampled at 10 kHz, and sent to the FDP through a 12-bit A-D converter. The FDP computes the cepstrally smoothed spectral cross sections in real time, storing both the spectra and the input speech samples in M_L. In the FDP, the speech is first windowed with a 25.6-msec (256-sample) Hanning window, which is shifted by 6.4 msec between spectral cross-section computations. The required 256-point discrete Fourier transform (DFT) of the windowed speech is actually accomplished by means of a 128-point fast Fourier transform (FFT), since the speech data are real. The result of the log magnitude and inverse FFT is the cepstrum, a time function consisting of an additive combination of the effects of the vocal tract response and the excitation function. The vocal tract information is concentrated near the origin of the time axis in the cepstrum, while the effects of the excitation function (at least for voiced speech) primarily consist of peaks at the pitch period and its multiples. Thus, the cepstrum is windowed by a function which is unity near the origin and tapers to zero before the first pitch peak. The window utilized is of the form

$$w_c(n) = \begin{cases} 1 & 0 \leqslant n < \tau \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{n-\tau}{\Delta}\right)\right] & \tau \leqslant n < \tau + \Delta \\ 0 & n \geqslant \tau + \Delta \end{cases}$$

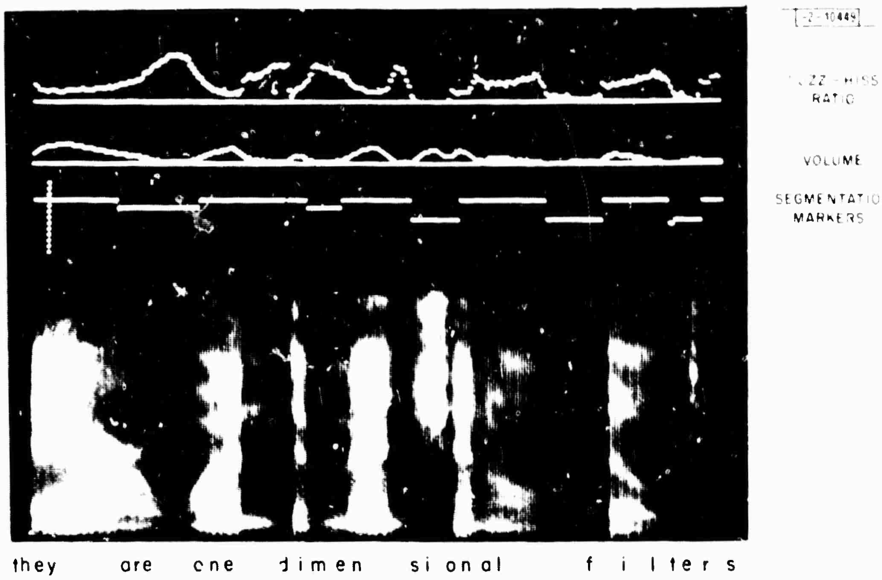

Fig. 3. Spectrogram and parameters for sentence "They are one dimensional filters."

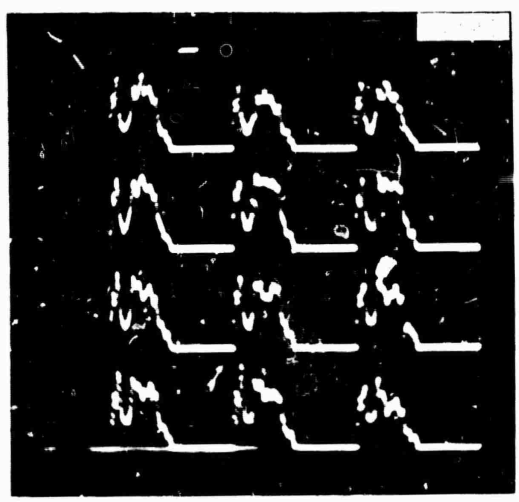

Fig. 4. Twelve consecutive spectral cross sections starting at point in Fig. 3. First cross section is at top left and ordering is left to right, then top to bottom. Frequency range of each cross section is 0 to 5000 Hz.

where τ and Δ can be set at run time. For male voices, $\tau = \Delta = 20$ was a typical choice of these parameters. At the 10-kHz sampling rate, this corresponds to a 2.0-msec "passband" and 2.0-msec transition region. The smoothed log spectrum is then computed as the DFT of the windowed cepstrum, and the smoothed magnitude spectrum is obtained by exponentiation. All the preceding computation and storage of speech and spectrum data in M_L are accomplished in real time. The storage capacity of M_L is such that about 4.5 sec of speech, spectra, and associated parameters (see below) can be held at one time. The 256×256 raster scan display permits convenient display of 256 spectral frames, or about 1.6 sec of speech.

3. Parameter Extraction and Display

The next step in the analysis is to pass through the spectral data, compute certain measurements on each frame, and store these measurements in a parameter table in M_L. Currently, space is allocated for 20 parameters per frame, but this number can be easily increased as new measurements are added. The initial set of measurements which have been programmed are aimed toward providing data for a preliminary segmentation of the speech.

Parameters now computed and stored in M_L include: (a) a measure of total spectral energy; (b) a measure of spectral energy in the 300- to 5000-Hz range; (c) a ratio of spectral energy in the 0- to 880-Hz range to total spectral energy (rudimentary buzz-hiss detector); (d) a spectral derivative defined as the sum of the magnitudes of the differences of the corresponding spectral samples 12.8 msec apart; (e) a ratio of energy in 0- to 400-Hz range to energy in 400- to 3000-Hz range (rudimentary nasal indicator). An additional parameter which is computed directly from the speech waveform (not from the spectrum) is a pitch measurement, obtained from a peak-processing algorithm due to Gold and Rabiner.[3] This algorithm includes four independent pitch detectors which find peaks in bandpass filtered speech and measure the periodicity of these peaks. If a consistent period is found among the different pitch detectors, it is designated the pitch period. If no consistent period is found, an indication of an unvoiced region is given.

For display of the spectra and parameters, two display scopes, both controlled by the 1219, are used: a fast raster scan scope capable of displaying a 256×256 sample intensity modulated picture with a refresh time of 1/10 sec, and a DEC scope with a speed of 50 µsec/displayed point. A typical display from the raster scope is shown in Fig. 3 for the sentence "They are one dimensional filters." Above the spectrogram of 1.6 sec of speech are displayed three time-aligned waveforms. The top waveform is the rudimentary buzz-hiss indicator mentioned above. The second waveform represents total energy in each spectral cross section, which may be referred to casually as a volume function. The third waveform consists of preliminary segmentation markers based primarily on the top two measurements, and will be explained in the next section. There is a vertical time marker on the spectrogram, and Fig. 4 displays 12 consecutive spectral cross sections beginning at the marker. The first cross section is at top left, second at top middle, etc.

4. Segmentation

A segmentation program is under development whose early goal is to separate the speech waveform into broad phonetic classes. The inputs to the segmentation are the types of parameters discussed above, and thresholding algorithms produce segmentation markers. The current algorithm produces only four types of segments: high-volume vowel-like segment, volume dip within a vowel-like segment, fricative-like segment, and silence or stop. The decisions made are now rather rudimentary. For example, the vowel-fricative decision is based on a single measurement, the ratio of energy is the 0- Hz region to the total energy in a spectral cross section. However, the program is struc to facilitate experiments with additional measurements and thresholding. Even at the current stage of segmentation, some editing is included. For example, fricative-like segments less than 3 frames (19.2 msec) in duration are eliminated.

An example of the segmentation program output is shown just above the spectrogram in Fig. 3 for the sentence "They are one dimensional filters." In order to indicate visually the results of the segmentation, the four segment classes are coded as different amplitude levels on a piecewise-constant waveform. The top level represents high volume vocalization, as in the initial [ei] sound. During the [r] - [w] glide, a dip in the volume function is found and marked by a drop to the second highest level. Fricative-like segments are marked by the next lower level, as in the [sh] of "dimensional," the [f] of "filters," and the aspirated release of the [t] in filter. The lowest level marks stops and silences, as in the [d] of dimensional and the [t] of "filters." Notice that the [n] of "dimensional" is missed by the volume dip detector. This is because the program marked only those dips where a minimum in the volume function was surrounded by two maxima, both of which occurred during the voiced segment. In this example, the volume minimum during the nasal is followed by a maximum occurring during an unvoiced segment. The program is currently being modified to mark dips of this latter type, as well as to incorporate additional measurements and segmentation indications.

5. Linear Predictive Coding

...ar predictive coding,[4] the vocal tract transfer function is represented by an all-pole ...r, whose coefficients are found by solving a set of linear equations based on an autocorrelation matrix derived from a section of the speech waveform. Within this framework, there are many versions of the algorithm for obtaining the coefficients, differing in detail. In the algorithm implemented on the FDP, a non-pitch synchronous technique is used,[5] and predictor coefficients are computed every 6.4 msec on the basis of 25.6 msec (256 samples) of speech. This framing rate is compatible with the framing rate for the homomorphic analysis. Spectral cross sections are derived from the predictive coefficients by a calculation of the magnitude of the transfer function, and a good estimate of the formant frequencies may be obtained by peak detection on the spectral cross section. Figure 5 is a comparison of homomorphic and predictive coding (with 12 coefficients) spectrograms of the same sentence. The smoother structure of the predictive coding spectrogram is to be expected from the nature of the model that is imposed. The current plan of operation is to use the homomorphic spectra for initial segmentation, and the predictive coding data to yield the formant information necessary for phoneme identification.

B. Speech Data Base

The Speech Data Base is intended to provide fast, automatic access to the entire range of data associated with each of many utterances. The functional requirements and organization of the data base were discussed in some detail in the previous report in this series (30 November 1971, DDC AD-735326). Work during the present reporting period has been primarily concerned with implementation and documentation of the supporting software. The following sections discuss the status of the principal software packages involved in data base support.

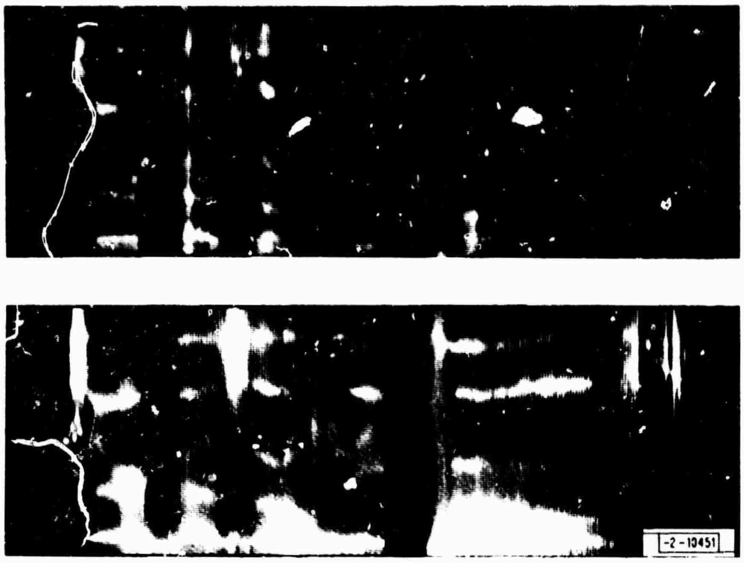

Fig. 5. Homomorphic (top) and linear predictive (bottom) spectrograms of sentence "The filter has only poles."

1. Speech Processing Controller (SPC)

The SPC is a sub-operating system being used to provide a programming environment for most other speech software. The SPC has been operational for the last several months, and some extensions and modifications have been introduced as a result of experience.

2. APEX Data Base Extension

The APEX executive system on TX-2 is being extended to handle the secondary memory management requirements of the data base. Coding for the necessary changes is complete and is being checked out.

3. SR Interface

The SR Interface is a package of BCPL programs designed to facilitate communication between both SPC processes and user-generated software and the APEX storage and retrieval facilities. The basic routines in this package have been written and checked out. Some evolution has already taken place in this software as a result of changing external specifications.

4. SR Commands

SR Commands facilitate user level requests for conditional searches of the data base. A translator will interpret the commands and call routines in the SR Interface package to effect the search. The design for the command language and parsing rules for the translator are nearly complete.

5. Display and Labeling

Programs to display envelope functions, spectral cross sections, spectrograms, etc., and to interact via tablet and keyboard have been demonstrated. Further work is indicated to smooth the interaction in certain cases and to provide the specific routines needed to build the time-event arrays corresponding to a manual labeling of phonetic events.

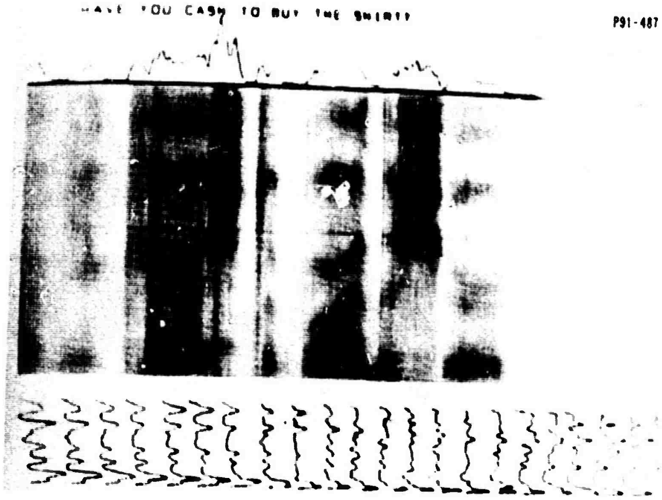

Fig. 6. Various features of sentence "Have you cash to buy the shirt?" displayed together. Top: maximum amplitude in spectral section vs time; high-frequency boost emphasizes fricatives. Center: 0- to 5-kHz homomorphic spectrogram synchronized with above function. Bottom: Spectral sections: amplitude (left to right) vs frequency (bottom to top); sections are 6.4 msec apart.

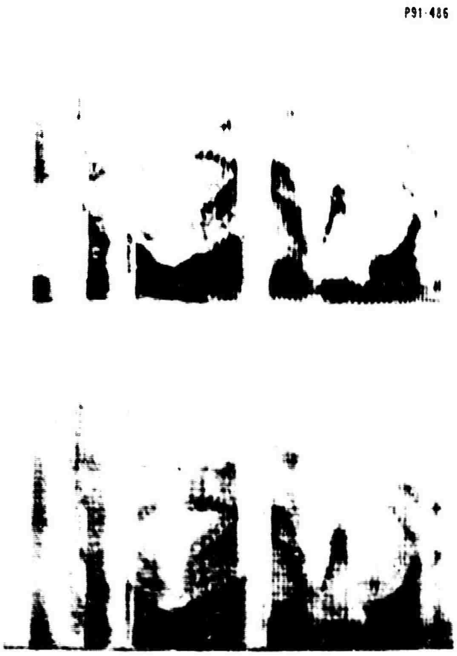

Fig. 7. Two spectrograms of same speech sample shown in Fig. 6, but using different intensity-to-gray-level transformations. Upper spectrogram uses linear transformation; lower one uses logarithmic transformation.

6. SURNET

A specialized network server facility called SURNET has been designed to provide multiple-user network access to the data base without the necessity of logging in to TX-2 as a normal (TELNET) user. The design is ready for presentation at the June 1972 meeting of the ARPA Speech Data Base Working Group. Discussion of the design will be deferred until the user community has had an opportunity to review the design in relation to their requirements.

7. Documentation

A draft of a document describing the data base facility has been prepared for the June meeting of ARPA Speech contractors. The document is intended to evolve into a user's manual for the facility.

C. Display of Speech Spectrograms

In an earlier Semiannual Technical Summary (30 November 1971, DDC AD-735326), the preliminary evaluation of a Hughes scan converter unit (LCSC-1) was reported. We knew then that a better storage tube would be forthcoming to upgrade the performance of the unit, an have since installed the new tube in the unit and have indeed observed improvements. The resolution at 50-percent modulation was increased from 1000 TV lines to about 1400 TV lines. The discernible gray levels were increased from five to nine. The performance of the unit in representing speech spectrograms can be seen in Figs. 6 and 7 which are photographs of the 1000-line TV monitor used to display the picture stored in the scan converter unit.

The spectrograms of Figs. 6 and 7 are generated by the combined operation of the TX-2 digital-to-analog output hardware and the display generator. The display generator produces a raster of vertical lines almost close enough to each other to merge. Each vertical line corresponds to a single spectral section. The amplitude information, quantized to eight levels, is fed to the scan converter from the digital-to-analog output in synchronism with the drawing of the vertical lines. In this mode of operation, a spectrogram is drawn quite rapidly. With 128 frequency bands (as shown in Figs. 6 and 7) a spectrogram with 400 time samples is generated in a little more than one-quarter of a second.

The restricted range of grays available requires some kind of amplitude compression and/or normalization to produce acceptable spectrograms. In the figures, a logarithmic relation between gray level and spectral amplitude was used, and amplitudes were scaled to present all values above half the maximum at the darkest gray level. No claims are made that this scheme is optimal, and further experimentation is planned.

We are sufficiently encouraged by results to date to proceed with the integration of the scan converter into the TX-2 display system. Unlike the direct view storage scopes which have been in use for some time, the scan converter has no direct way of dealing with cursors and tracking "bugs." We expect that some combination of the selective erase capability of the scan converter and injection of an appropriate signal into the video output can be utilized to achieve an interactive capability.

II. TERMINAL SUPPORT PROCESSOR (TSP) SYSTEM

The TSP system is a small-scale computer system intended to support interactive graphics users of a computer network. The design aims at providing basic interactive graphics services for a number of consoles, each consisting of a keyboard, a tablet, and a pair of storage scopes.

The system provides a language called LIL, which a user can utilize to control interactions between his console input and output devices and between the TSP and other computers in the network. The TSP itself consists of three microprocessors sharing 65,536 words of 900-nsec core memory arranged in 8 banks of 8192 words each. Previous reports in this series have described the user specifications for LIL (31 May 1970, DDC AD-709187) and the system architecture of the TSP (30 November 1970, DDC AD-716817).

During this reporting period, the Meta 4 computer system passed its acceptance tests. Therefore, emphasis is beginning to shift from hardware checkout to system programming. The two areas being worked on at the moment are the overall system monitor and the keyboard-echo process.

The overall system design is fairly simple. Each processor is dedicated to certain tasks; thus, it is not necessary to write a single monitor which worries about scheduling both processors In fact, since the sections of the system we want to write first are all handled by the same processor, it is not necessary to be concerned about scheduling the second processor at all.

The monitor has to solve two problems: the synchronization of system processes with input/output devices, and the scheduling of system processes to insure that each one runs often enough to provide satisfactory response. Both problems are solved by using pending bits. An IO device signals an event by causing a hardware interrupt. A short assembly language program is immediately awakened. This interrupt handler sets a pending bit and goes back to sleep.

Whenever the system is otherwise idle, a dispatcher continuously checks for nonzero pending bits. When one is found, the appropriate pending process, a BCPL subroutine, is called. The pending processes update the system data base to reflect IO events and initiate other IO operations. Interlocking is provided by the fact that each pending process runs to completion before returning to the dispatcher, which may then activate other pending processes.

A pending process may also set pending bits. This makes it possible to schedule the processes for satisfactory response. When a single job is too long to be handled by one pending process, it is broken up into several subprocesses. Before returning to the dispatcher, each subprocess sets a pending bit to activate its subsequent subprocess. Thus, the dispatcher has an opportunity to run a higher priority pending process between the subprocesses.

The first set of interrupt handlers and pending processes being written collects input typed from a console's keyboard and echoes it on the console's storage scope. This section also handles simple editing by allowing the user to delete the previous character or the current line. Since the TSP is intended to be an intelligent terminal for ARPA network hosts, we are giving the user the ability to control the character that is echoed when he hits a particular key. Thus, users of TX-2 can define a keyboard which resembles a TX-2 keyboard, while users of the IBM 360 can define a keyboard which looks like an IBM 2741 keyboard. Until the user changes his keyboard definition, all TSP keyboards will resemble the Network Virtual Terminal keyboard, an ASCII device which is becoming the network standard. All keyboards have one key which cannot be redefined and which restores the standard TSP keyboard definition. Thus, users are prevented from losing control of the terminal.

Echoing on a scope is somewhat different from echoing on a typewriter. An additional complication is that the system must supply the user with some way to erase his scope and continue at the top of the page. An advantage is the speed at which lines can be displayed. For example, if the user erases his scope in the middle of a line, there is no particular problem in repainting the line at the top of the new page. This speed also allows clearer echoing of edited lines.

While the user is deleting individual characters from a line, the system displays a cursor which points to the last character deleted. When the user types another input character, the system crosses out the entire line and repaints a clean copy. This style of echoing has been used for several years in the TX-2 storage scope editor. Users have found it very pleasant to work with.

Allowing the user to redefine his keyboard creates another human engineering problem. What happens when the user asks the system to redefine his keyboard but then goes ahead and types before the system has a chance to make the change? Some systems simply lock the keyboard whenever the user completes a line. This leads to a good deal of user frustration, since most lines do not cause a change of keyboard definition. Some systems accept the first part of the line in the old keyboard definition and the rest of the line in the new. This can cause considerable confusion. The TSP accepts the entire line in the new keyboard definition, and crosses out and repaints any characters the user slipped in under the old definition.

III. TX-2 ACTIVITIES

A. BCPL

1. BCPL Changes and Extensions

The structure facility discussed previously in this series of reports (31 May 1971, DDC AD-726534) is complete and has seen extensive use, including use in the latest version of the compiler itself. It seems to be a useful idea and is winning acceptance among the programmers. On the basis of suggestions, a few additional improvements have been made to the facility.

A conditional compilation facility has been added to the BCPL compiler. Any expression that can be evaluated at compile time is evaluated, and optimizing of conditional expressions is done whenever possible. For example, consider the two-armed conditional

test B ifso S1 ifnot S2

where B is some expression and S1 and S2 are statements. If B can be evaluated at compile time, then only one of S1 or S2 is compiled, the other being totally ignored. The first application of this facility – indeed, the reason for its creation at this time – has been for use of the compiler. We are currently supporting on TX-2 BCPL compilers for three different computers: TX-2, BCOM and the SEL. Maintenance of the source code for the three compilers, with incorporation of improvements in all three in a coordinated manner, has become an increasingly burdensome task. With the conditional compilation facility, all code for all three compilers has been combined into one package. Three manifest constants have been defined in the compiler: COMP_TX2, COMP_BCOM and COMP_SEL; and, in any compilation, one of these is true and the other two false. Suitable conditionals have been incorporated into the text so that only that portion of the code needed for a given compiler is compiled. We have already achieved from this facility the addition of structures into the BCOM compiler with considerably reduced effort.

The compiler now emits a symbol table for use in debugging, and a program has been written to produce a human-readable listing of the information in the symbol table. Soon to be completed is the necessary interface to permit the debugger (reported in the next section) to read the symbol table so that, even more than at present, debugging can be done in source program terms. The symbol table contains enough data so that, if an error is detected at run time, it is possible for the debugger to display the line of source text that was being obeyed when the error occurred. Further, the programmer may interrogate the values of variables by name.

A new calling sequence has been specified for calling one BCPL-coded program from another. The new form permits the called program to determine how many arguments were passed to it by the caller, an ability that is needed in the Speech Processing Controller. We have been experimenting with the new calling sequence enough to establish that it works, and we will shortly make the final switchover to it. Doing so is a drastic step, since it renders obsolete all existing compiled programs, necessitating recompilation. Programs written in TAP – assembly code – need to be modified. All the BCPL libraries have been changed and tested thoroughly.

A library facility for relocatable programs has been specified and coded as an SB thesis by an M.I.T. senior. The code is in an advanced state of debugging. The facility includes a new program, the library maintainer LIB, and changes to the loader 5NLOAD. LIB is a tool that lets the user combine compiled programs into libraries. A library entry may be either the compiled module itself or a pointer to the module. In the latter case, the module may live in either the same directory as the library or in some other directory. The changes to 5NLOAD permit it to search libraries, loading only those modules needed.

2. Symbolic Debugger

The first phase of a new symbolic debugging system for programs written in the BCPL language on TX-2 has been completed. The system is interactive, and makes use of the hardware and software facilities on TX-2 for causing program interrupts (breakpoints) to occur on specified instruction or data references, without the need to modify the user's program in any way. It also makes extensive use of information from the relocatable loader, and from symbol tables generated by the latest BCPL compiler.

In addition to the standard features of a good symbolic debugging system, the debugger has the following features:

(a) The ability for a user to extend the command repertory of the debugger controller in a straightforward manner.

(b) The ability for a user to define a BCPL function which he can then associate with a selected breakpoint in his program or data and cause to be executed when the breakpoint occurs. Applications for this facility include conditional trapping, program performance monitoring, and program animation.

The capability to define BCPL subroutines and associate them with breakpoints has been used to:

(a) Add a feature which will maintain and display a dynamic trace of an arbitrary BCPL program's execution history.

(b) Add a feature which will maintain and display a dynamic trace of references to specified variables as a program is executed.

(c) Animate the event scheduler data structure of a large (air traffic control) simulation program.

B. ARPA Network

The TELENET SERVER and LOGGER programs which allow ARPA Network log-in to TX-2 are now fully operational. Recent implementation of a complete translation between the TX-2 character set and network standard ASCII, and the provisions for a remote user to activate a "help request" at TX-2 allow full use of TX-2 keyboard-typewriter software from the network.

Except for occasional special situations, TX-2 is generally up for network use whenever it is up for normal time-sharing use. Occasional use from remote sites has demonstrated the ability of the system to recover from most of the abnormal events which can occur in network usage.

In anticipation of future demands, plans are proceeding on extending the software to allow more than one remote user to access TX-2 at the same time. This LOGGER extension is being designed to mesh with the SURNET speech data-base service which will enable a user to store and retrieve speech data without having to log into TX-2 as a normal user.

C. TX-2 System Changes

A number of changes have been made to TX-2 in order to improve performance and facilitate software development.

Cycle-stealing IO devices which previously shared IO channels were made autonomous so that as many as eight devices can now be active simultaneously. This not only increases the total IO bandwidth for these devices, but also reduces the programming overhead in operating them.

The main core memory is being increased by two new 32,536-word modules, bringing the total memory capacity of TX-2 to approximately 232,000 words. This is divided into eight banks of roughly equal capacity, thereby maximizing opportunities for overlapping randomly chosen banks. The control for the four processor ports on the memory bank switch is being further activated (TX-2 has used only two of the four ports) and speeded up, so that shortly the controller for the IO cycle-stealing channels and the controller for a new raster display will be able to use separate memory ports and operate at close to memory speeds ($\approx 1 \mu sec$).

The address transformation logic has been modified in order to facilitate the management of virtual memories embedded in the enlarged real main memory. The basic TX-2 address transformation is a two-stage process offering both segmentation and paging. Each stage makes use of a small fast memory to hold the transformation values. The modification simply allows the APEX executive program to specify whether a segment is to be paged or not, thereby permitting a saving in the use of page address memory space whenever a whole segment can be assigned to contiguous main memory addresses. The resulting reduction in the utilization of page address memory space will allow the increased main memory to be accommodated without undue increase in system overhead related to the management of page address memory space.

D. LDX Printer

Several years ago, it was decided that the TX-2 would need to supplement its charactron-driven Xerox printer to serve projected needs for hard copy using ASCII character sets. It was decided that it would be best to produce a very general hard-copy facility, so plans were made to use a Xerox LDX printer driven by a minicomputer. The LDX (Long Distance Xerography) printer is a raster-scan device designed for cross-country document transfer. Consequently,

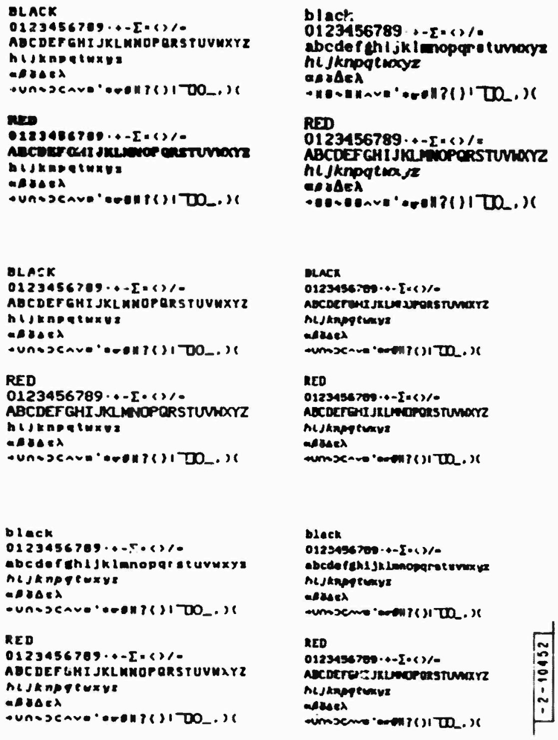

Fig. 8. Six standard character sets available for LDX printer on TX-2.

Fig. 9. Speech spectrogram produced on LDX printer. Dot patterns represent 16 gray levels.

by appropriate hardware and software, any imaginable character set can be generated, as well as arbitrary diagrams such as [illegible]rated circuit masks. The LDX model obtained for TX-2 produces 135 scan lines per inch. A scan line is about 8 inches in length and requires 969 bits to be transmitted from the minicomputer (a PDP-8/L with 4k memory). The paper is $8\frac{1}{2}$ inches wide, and page length is under program control. Paper speed is slightly more than $1\frac{1}{2}$ inches per second.

The software for the Xerox LDX printer has been sufficiently completed to allow regular use of the LDX. Six standard character sets have been designed, and a number of variations on these have been implemented by individual users. Gray-scale output for spectrograms is routinely available. Figures 8 and 9 show examples of LDX output.

In the current implementation, the TX-2 executive simply transmits what it presumes to be text, without analysis, to the PDP-8 and accepts no recoverable error messages. The PDP-8 has rules to give somewhat reasonable output for any input and crashes on any unforeseen conditions. The PDP-8 will chop any lines too long to fit across or too complex to fit the real-time constraints.

The PDP-8 program can be considered to have three priority levels. The lowest level accepts input from TX-2 and generates character lines, chopping as necessary. The middle level generates individual raster-scan lines for these character lines, including compound characters (overstrikes). The highest level responds to LDX interrupts and sets up the next scan line to be shipped to the LDX.

In order to minimize the possibility of the PDP-8 running amuck and becoming unable to respond to TX-2, the memory protection feature is used to prevent 128 PDP-8 registers from being written. This area of memory contains the code to process crashes and reload from TX-2. The program for LDX output is always loaded just before output.

Currently, automatic page headers are available only with a reduced character set. A more general scheme involving interaction with TX-2 at each page break and requiring no reduction in character set is under consideration.

REFERENCES

1. B. Gold, I. L. Lebow, P.G. McHugh and C.M. Rader, "The FDP, A Fast Programmable Signal Processor," IEEE Trans. Computers C-20, 33 (1971), DDC AD-728092.
2. A.V. Oppenheim, "Speech Analysis-Synthesis System Based on Homomorphic Filtering," J. Acoust. Soc. Am. 45, 458 (1969), DDC AD-689574.
3. B. Gold and L. Rabiner, "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain," J. Acoust. Soc. Am. 46, 442 (1969).
4. B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Am. 50, 637 (1971).
5. V.W. Zue, "Speech Analysis by Linear Prediction," Quarterly Progress Report No. 105, Research Laboratory of Electronics, M.I.T. (April 1972), pp. 133-142.