CSRE: A SPEECH RESEARCH ENVIRONMENT

Donald G. Jamieson[1], Terrance M. Nearey[2], and Ketan Ramji[1]

[1]Speech Communication Laboratory

Hearing Health Care Research Unit

Department of Communicative Disorders

University of Western Ontario

London, Ontario

and

[2]Department of Linguistics

University of Alberta

Edmonton, Alberta.

ABSTRACT

CSRE (The Canadian Speech Research Environment) is designed as a comprehensive, integrated, inexpensive, micro-computer-based workstation to support speech research. The focus of the project has been to develop the basic functionality required by speech researchers, using mass-produced, widely-available hardware. In its present form, CSRE includes: 1) waveform editors to capture, cut, paste, and measure signals; 2) facilities to produce colour spectrograms and waterfall displays from sampled data using a variety of spectral analysis techniques; 3) a parametric digital speech synthesizer, based on that developed by Klatt (1980);888 4) and an experiment generator, developed to speed the programming of experimental sequences. The system requires an IBM/AT compatible computer (typically, Zenith Model 386 or Model 248, with math coprocessor and EGA or VGA graphics), a data acquisition system (e.g., Data Translation DT2801-A or ARIEL DSP-16, plus microphone, preamplifier and filters), and a Microsoft-compatible mouse.

RESUMÉ

Le CSRE, "Canadian Speech Research Environment" est un système complet et peu coûteux d'étude de la parole à partir d'un micro-ordinateur. Nous avons d'abord cherché à intérgrer les équipements de base disponibles sur le marché et accessibles aux chercheurs. Dans son état actuel, le CSRE comprend trois programmes d'analyse de formes d'ondes acoustiques, deux programmes d'analyse spectrale et un synthétiseur de parole fondé sur le programme de Klatt (1980). Le système recquiert un ordinateur IBM/AT (ou un ordinateur équivalent, tel que le Zénith modèle 386 ou 248), un dispositif d'acquisition de données (Data Translation DT2801-A ou ARIEL-DSP16), un microphone, un pré-amplificateur et des filtres ainsi qu'une souris Microsoft.

CSRE is designed as a comprehensive, integrated, inexpensive, micro-computer-based workstation to support speech research. Such research requires five basic facilities: 1) a facility to record, store, and play back high quality natural speech signals; 2) a facility to edit stored natural speech signals -- to eliminate unwanted portions of the signal and to concatenate (paste together) two different signals; 3) a facility to analyze and measure speech signals -- to accurately measure the duration and amplitude of different parts of the speech signal and to accurately measure the frequency and amplitude of critical portions of the speech signal; 4) a facility to parametrically synthesize speech signals; and 5) a facility to control speech output -- as for experiments or assessment procedures.

Our initial focus has been to implement these basic functions which speech researchers require, using mass-produced, widely-available hardware. Subsequently, we have sought to develop these functions within an interactive environment, in which researchers can move readily between alternative applications, and to improve the ratio of the user's productive "investigative" time to "clerical" time.

The system we have developed requires an IBM/AT compatible computer (typically, Zenith Model 386 or Model 248, with math coprocessor (Intel 80x87) and EGA (16 colors) or VGA graphics), a data acquisition system (e.g., Data Translation DT2801-A or ARIEL DSP-16, plus microphone, preamplifier and filters), and a Microsoft-compatible mouse. The components of the software package have now been tested extensively and the system is presently being distributed to all interested university-based researchers. The most important functions available within the package are described in the following sections.

COMPONENT PROGRAMS OF THE CSRE PACKAGE

I. WAVEFORM EDITORS

I.A. CAPTURE (the Waveform Capture Editor) is used to capture (digitize and edit) a portion of a speech signal. This signal may be recorded directly from microphone or tape input, using CAPTURE, or taken from a disk file. Mouse-driven menu commands are used to select functions -- for example to display a signal, to isolate the desired portion of the displayed signal, to play the selected signal, to cut the selected signal and save it to disk, and so forth (See Figure 1). The capture functions permit the user to display, in the window at the top of the screen, the entire signal which has been previously sampled or digitized by CAPTURE. The signal display includes time in milliseconds on the horizontal axis, and amplitude as coded by the ADC on the vertical axis, the name of the file being displayed, and the location of the starting and ending data points. Alternatively, the users may sample (record, digitize) a sound from an external source. After selecting this function, sampling will begin (and end) on the user's command (by pressing the mouse button to begin and end). At this point, the full signal may be "spoken"; alternatively, a window may be positioned on any portion of the signal in the top viewport and that part played back.

```
NAME      : DEMO2.ADF      TIME 1 :  1101.3  ms      TIME 2 :  1311.2  ms      SF        :  10.0 kHz
                           AMP. 1 :    -0.1  V       AMP. 2 :    -0.5  V
                           dT     :   209.9  ms      RMS    :     2.1  V       LENGTH    :  1711 ms
```
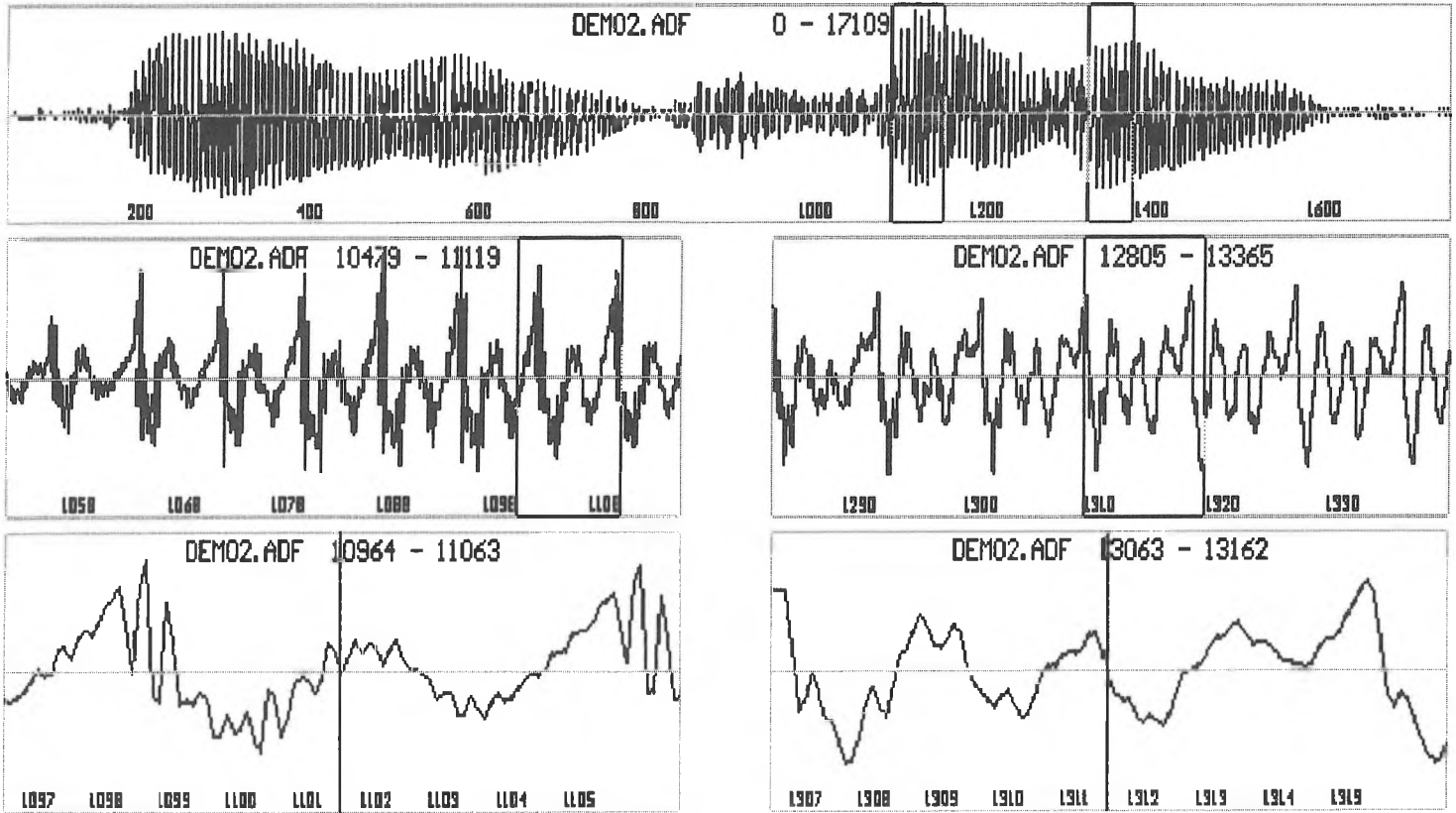

Figure 1. An example of the format of the CAPTURE screen when a signal is being displayed. Signals are initially displayed in the uppermost window, with the initial and final portions of the selected signal subsequently displayed at increasingly higher resolution within the lower four windows.

CAPTURE is designed specifically to capture (cut, or isolate) part of a signal. To do so, the user must indicate, precisely, both the beginning and the end of the signal to be captured (cut), by placing markers (cursors) within the primary signal -- i.e., the signal in the top window. A series of windows are used to indicate precisely where the **beginning** and **end** of the captured potion of the signal should be (middle windows) and to "fine-tune" both the beginning and the endpoints of the selected signal in preparation for cutting the signal (lower windows). As the user "tunes" the positions of the markers, current time/amplitude values are displayed in real time. When both markers have been specified, the display will also indicate the time difference and the RMS value in voltage of all points between these markers. To listen to the signal between the markers (what the user has "captured"), there is also an option to play back this portion. Finally, a **save** option permits the user to save the edited signal as a named disk file.

THE COMPLETE ACOU

Our state-of-the-art instruments do far more than measure the intensity of sound in any given situation. With this system, Brüel & Kjaer
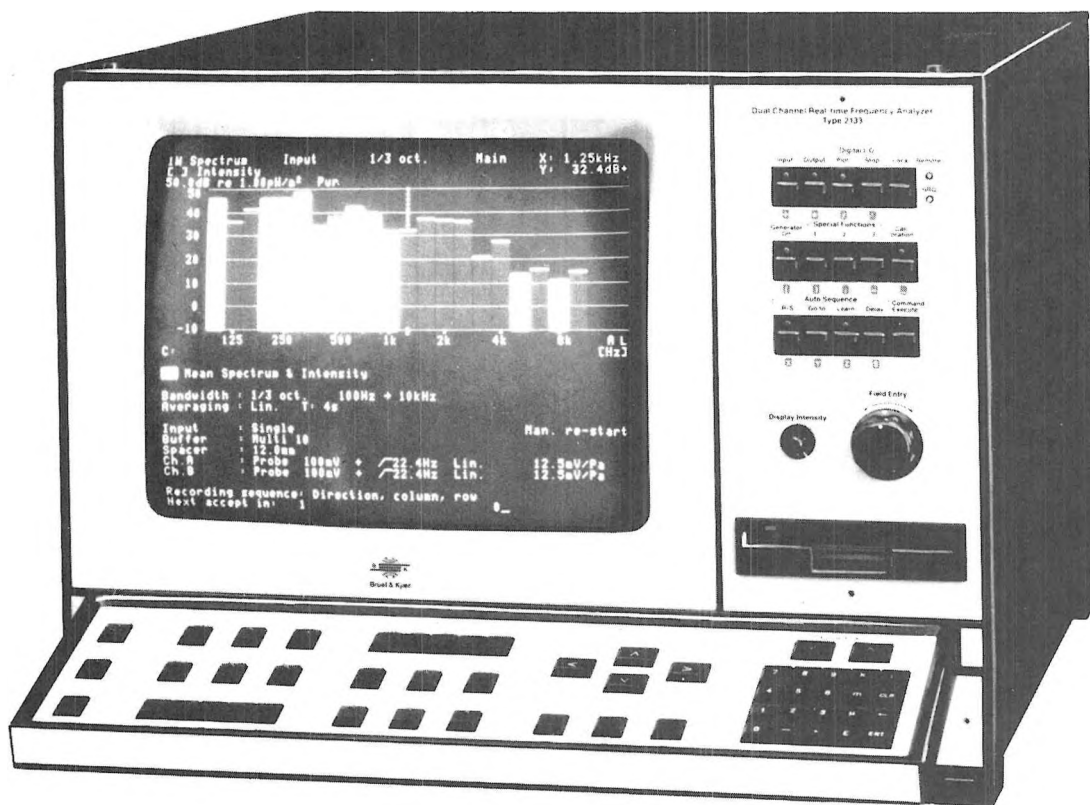
The dual-channel real-time analyzer Type 2133: It has extensive data processing features along with built-in storage capabilities. One of two such analyzers created by Brüel & Kjaer, its measurements range from 1/3-octave dual-channel real-time analysis to 11 kHz (1/1-octave to 22 kHz).
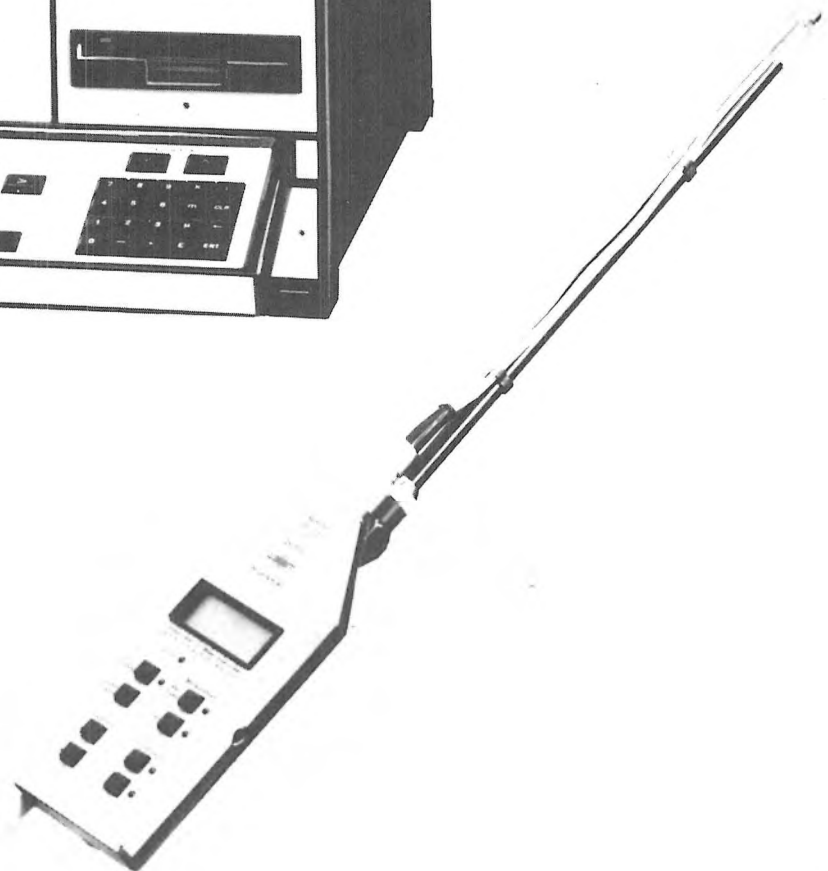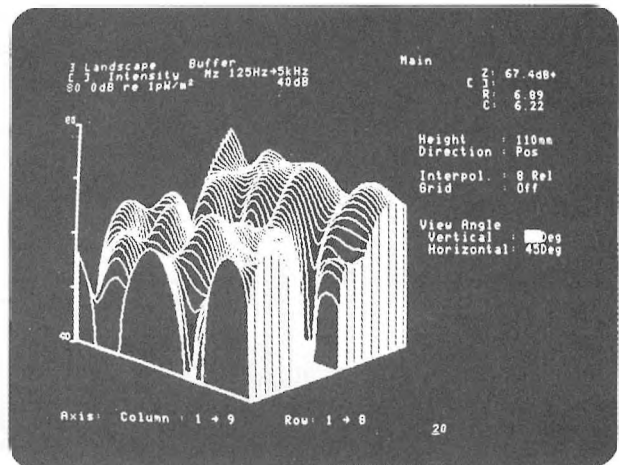
STICS LABORATORY

instruments can map, graph, record and analyze sound events. It's like having an entire laboratory calibrated to the highest possible accuracy.
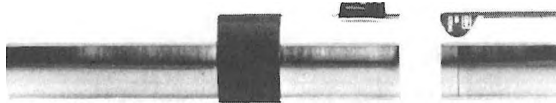

New intensity probe 3545 and remote control unit ZH 0354, for direct connection to, and control of, Analyzer 2133.
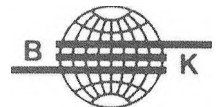

An MS-DOS compatible data storage is used in the built-in disc drive. An optional intensity mapping software package, the BZ 7021, is also available.

Sound intensity calibrator 3541, an innovation in intensity measurements.

Brüel & Kjær

I.B. PASTE allows user to concatenate portions of two digitized signals. Both original signals are displayed in the top and bottom windows respectively and both signals can also be played back. The user can zoom into a portion of the signal in the top viewport to indicate the approximate desired point at which the initial part of the signal should end. This magnified signal will appear in the middle window (left). The user can then fine tune the initial segment using a mouse-controlled marker. Similarly, the magnified final portion of the signal from the bottom window will also appear in the middle window (right) with a mouse-controlled marker used to fine tune the final segment. (See Figure 2). The concatenation will be such that all points from the beginning of the signal in the top window to the marker in the middle-left window will be pasted **in front** of all points beginning at the marker in the middle-right window to the end of the signal in the bottom window. Both individual signals and the concatenated result can be heard individually using the **playback** options. Finally, the **save** option permits the user to save the resulting concatenated signal to a named disk file.
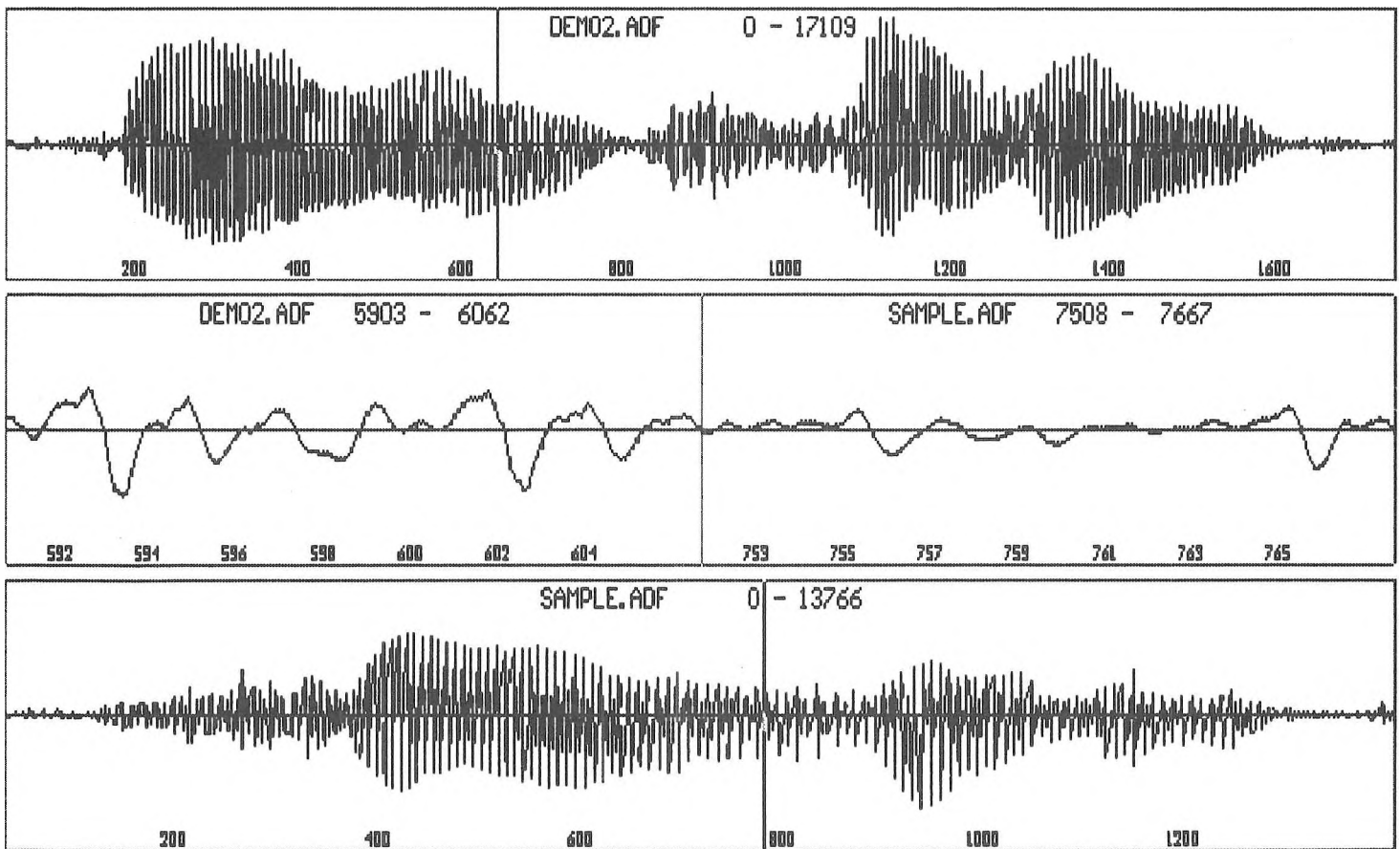

Figure 2. An example of the format of the PASTE screen used to concatenate two signals to form a third signal. The viewports in the middle of the display indicate the exact point at which the initial signal (from the top viewport) will be joined with the final signal (from the bottom viewport).

II. SPECTRAL ANALYSIS PROGRAMS

II.A. COLOR SPECTROGRAM. The Spectrogram program produces a perceptually-compelling time/frequency/amplitude display of a digitized signal, with time on horizontal axis, frequency on the vertical axis and amplitude as one of 16 possible colors to code the amplitude of the signal. The dB range represented by each color is shown as part of the display, and the range of dB values displayed can be altered by the user. The user is also offered several alternative mappings from intensity to color: a rainbow scale, a temperature (heat) scale, and a brightness scale. For users having a suitable monochrome monitor, a true gray scale is also available, together with a reverse gray scale (brightness). Inexpensive, high-quality, colour hardcopy output is straightforward, using an HP PaintJet printer, with a print utility such as Pizzaz Plus.

```
NAME        : DEMO2.ADF    TIME       :   263.5 ms   SF        : 10.0 kHz   DB SCALE  : RELATIVE
FILTERS     : 256          FREQUENCY  :  1816.4 Hz   PROC, LAG : ARMA, 15   WINDOW    : RECTANGLE
RES, SHFT   : 5, 34        AMPLITUDE  :   -10.03 dB  PRE-EMPH. : ON         LENGTH    : 1711 ms
```
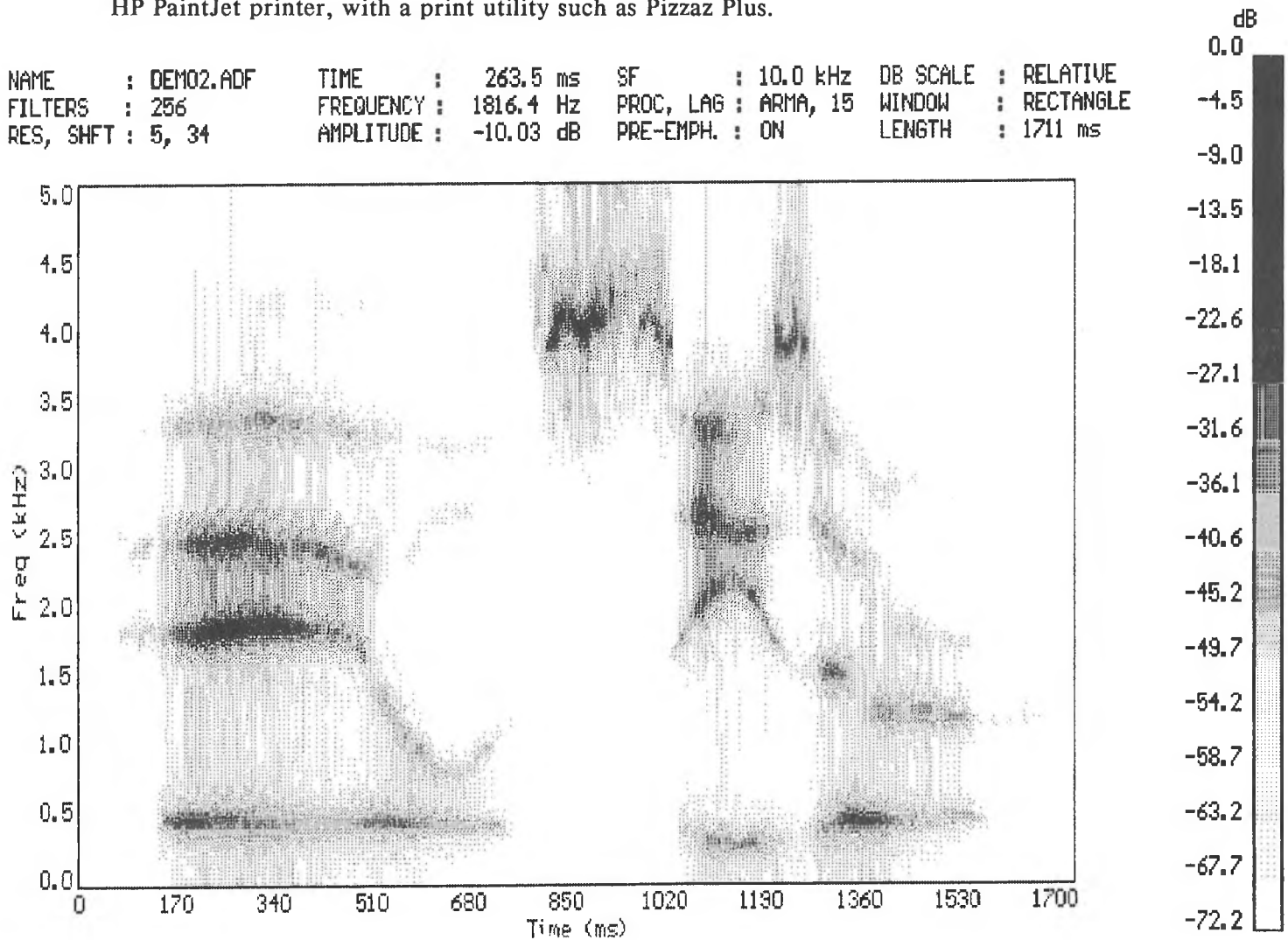

Figure 3. An example of the format of the SPECT screen showing the results of a computed spectrogram in true gray-scale mode, with direct measurement from the display. The program provides a time/frequency/amplitude display of a digitized signal, with time on the horizontal axis, frequency on the vertical axis and amplitude coded either as one of 16 possible colors or, in monochrome mode, as the relative brightness of the display. The dB range represented by each color (gray-level) is shown to the right of the display, and can be altered by the user. The user is also offered several alternative mappings of intensity, and a large number of analysis options (see text for details). After a spectrogram has been computed and displayed, the mouse can be used to define and play a portion of the signal, and/or to obtain a direct readout of the time, frequency, and amplitude coordinates of any desired point on the display.

For any desired sampled signal, the power spectrum density is calculated and a spectrogram can be generated under menu control (See Figure 3). Each parameter has a default value, but users can override this default to specify: 1) the desired frequency resolution (number of filters); 2) the desired temporal resolution; 3) the lower and upper frequency boundaries of the spectrogram (in Hz); 4) the sampling frequency of the signal; 5) the type of analysis procedure to be used (fast-Fourier Transform (FFT), Auto-regressive (AR), or modified AutoCovariance methods); 6) the color scale to be used; 7) the range in dB (subset of colors) to be displayed; 8) whether the intensity of the spectrogram is to be scaled **relative** to the within-signal maximum value, or **independent** of signal intensity -- with scaling relative to the maximum signal which could be recorded using a 12 (or 16) bit analog-to-digital converter; and 9) whether or not the signal is to be preemphasized.
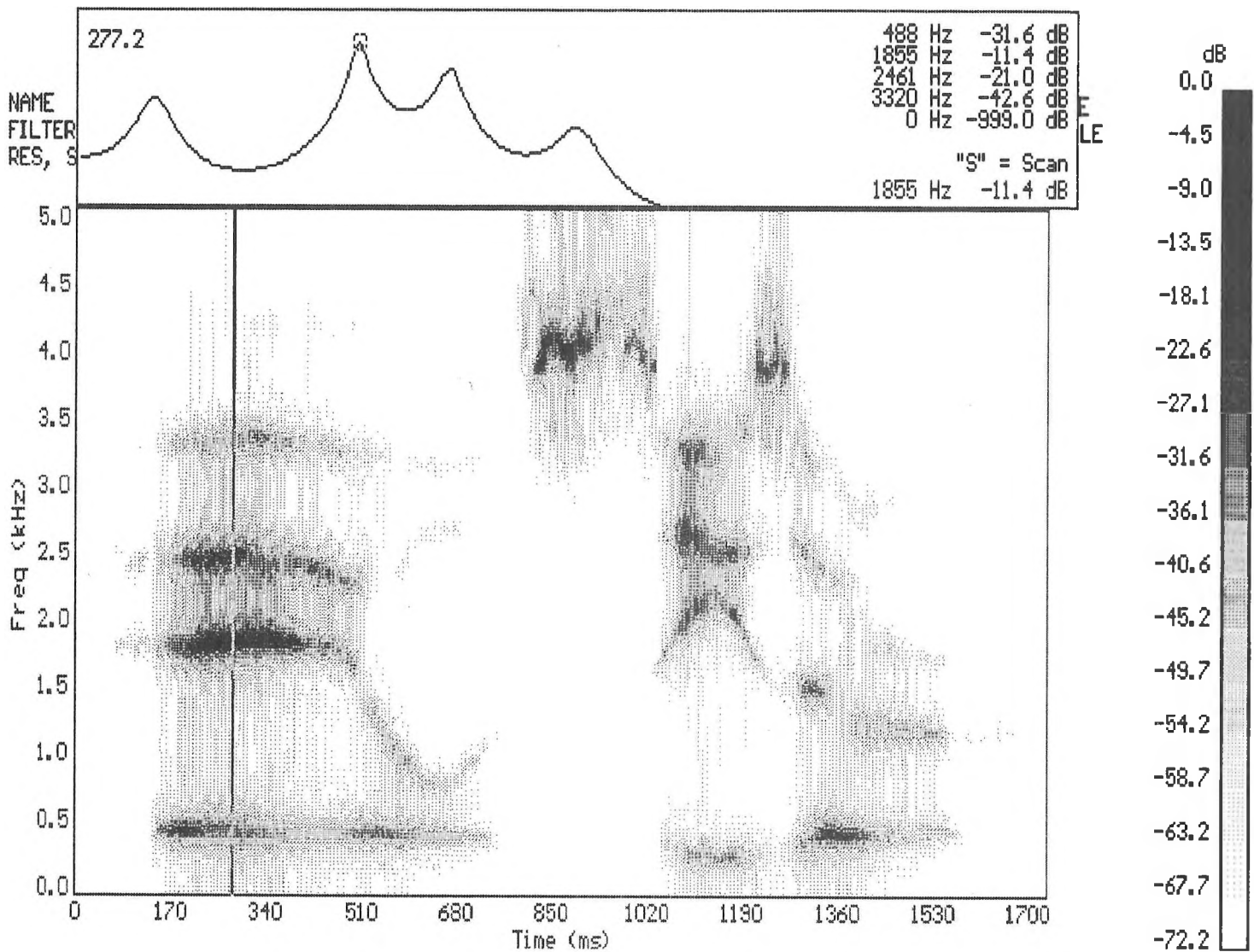

Figure 4. An example of the screen format of SPECT as the user selects a single "spectral time slice" to be displayed just above the actual spectrogram display. The frequencies and amplitudes of the major peaks for this slice are automatically extracted and listed. SPECT also provides the option to "scan" the slice for frequency/amplitude values along the time slice.

Users may also compute a second color spectrogram, using a different signal or different settings with the same signal, and store this on a separate graphics page, toggling back and forth between the two displays, as desired. Once a spectrogram has been displayed, users may move a cursor under mouse control to define and play a region of the signal to which they wish to listen, or to obtain a direct readout of the time, frequency, and amplitude coordinates of any desired point (See Figure 3). Users may also select a "spectral slice" (amplitude-by-frequency display) at a particular time point in the signal, for display in conjunction with the spectrogram (See Figure 4). When such a spectral slice is displayed, the major peaks from this spectral slice are automatically extracted and listed; in addition, a mouse-or key-controlled cursor can be used to obtain amplitude values for any desired frequency.

II.B. WATERFALL DISPLAY. A waterfall display (time by frequency by amplitude in 3 dimensions) is another way to display a power spectrum density resulting from the spectral analysis of a digitized signal. The result is displayed as a series of "stacked" time slices (See Figure 5).
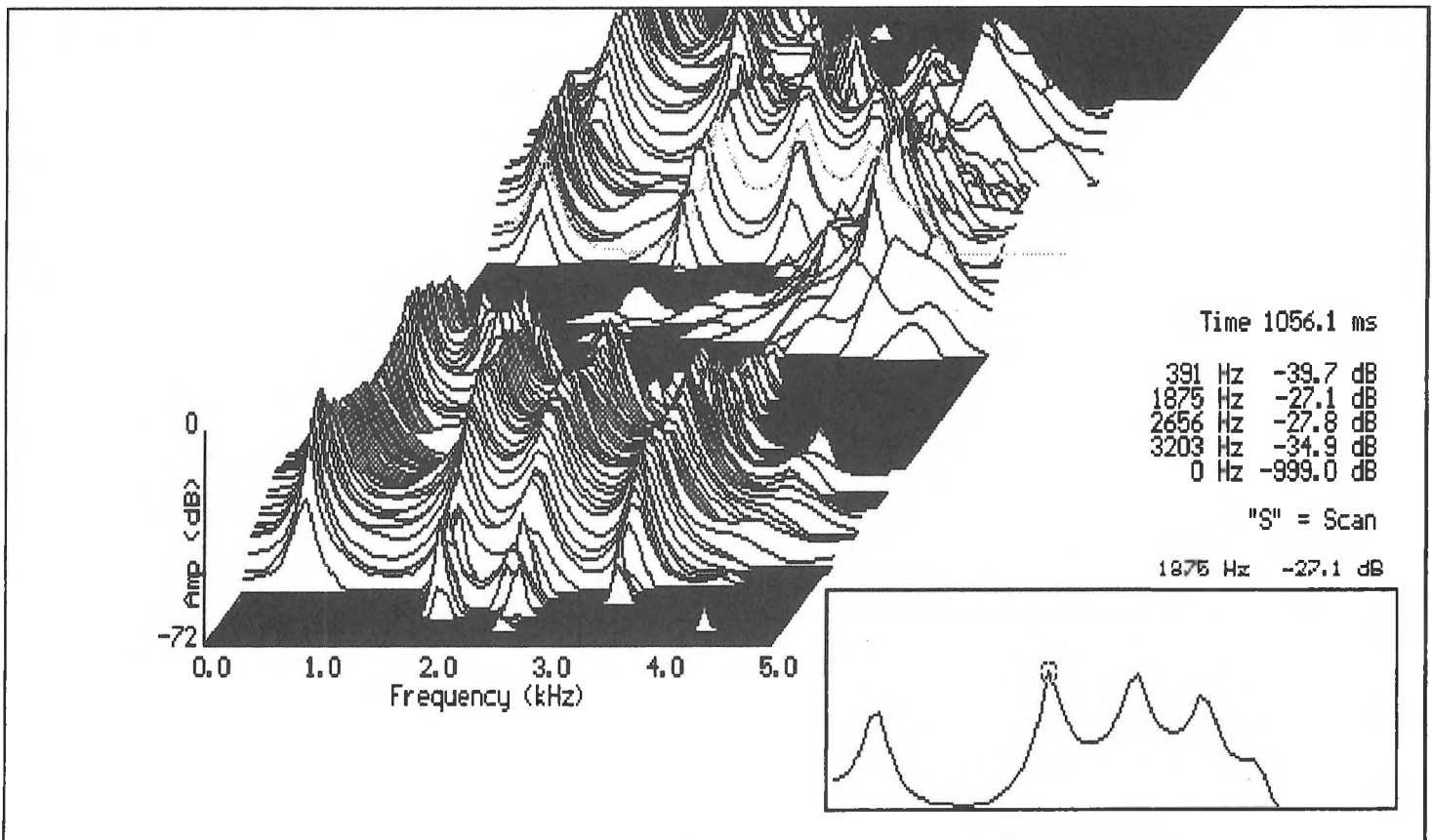

Figure 5. An example of a waterfall display (time-by-amplitude-by-frequency) in three dimensions. The display shows the results of a spectral calculation as a series of "stacked" time slices. Any time slice in the display can be selected using the mouse or cursor keys; once a time slice has been selected, a marker can be moved along this time slice to obtain instantaneous frequency and amplitude values for any desired portion of the signal.

The program also allows one to select any time slice from the display with mouse control. Upon this selection, one can also run a special marker along this time slice to obtain instantaneous frequency and amplitude values for any desired portion of the signal.

III. PARAMETRIC SPEECH SYNTHESIZER

SYNTH is a parametric speech synthesizer which generates **audio data** files from the specifications of 40 control parameters -- the 39 parameters of the KLATT80 synthesis, plus parameter 40, CORRSW, added by Kewley Port. In this implementation, the synthesizer is controlled by an INTERPolated file, which specifies the values of each parameter, updated every 5 ms (See Figure 6). The INTERPolated files are generated from KNOT files, which define "skeletonized" parameter tracks, specifying "piecewise linear" functions of the desired parameter tracks. (These KNOT files are fed to an interpolator to generate the complete files, prior to synthesis.)

```
CSRE Knot Editor
=============================================================================
Klatt Parameter Information          #  Time Value      #  Time Value

 ┌──────────────────────────────────┐  ┌──────────────┐  ┌──────────────┐
 │ Fundamental freq of voicing      │  │ 1     0   100│  │16            │
 │                                  │  │ 2    50   125│  │17            │
 │ f0_   V        0        500    0 │  │ 3   100   120│  │18            │
 │                                  │  │ 4   200   110│  │19            │
 └──────────────────────────────────┘  │ 5            │  │20            │
 Name   Type    Min      Max    Def     │ 6            │  │21            │
                                        │ 7            │  │22            │
                                        │ 8            │  │23            │
Options  [G]                            │ 9            │  │24            │
                                        │10            │  │25            │
G)et Parameter                          │11            │  │26            │
P)rocess                                │12            │  │27            │
S)ave                                   │13            │  │28            │
E)xit                                   │14            │  │29            │
                                        │15            │  │30            │
Editing  [Q]                            └──────────────┘  └──────────────┘

A)dd          Time Value
C)hange      ┌─────┬─────┐            File Name
D)elete      │     │     │            ┌────────────────────────────────────┐
Q)uit <esc>  └─────┴─────┘            │ KNOTBASE.KFE                       │
                                      └────────────────────────────────────┘
```

Figure 6. An example of a SYNTH screen, showing initial input to the parametric speech synthesizer. Using SYNTH, the 39 control parameters for the synthesizer are defined at each of the critical "change" points within the signal. Subsequently, these "skeletonized" parameter tracks, specifying "piecewise linear" functions of the desired parameter tracks, are INTERPolated to produce a file which specifies the values of each parameter, updated every 5 ms. The INTERPolated files are then used to control the digital synthesizer.

IV. CONTINUUM GENERATOR

Many studies using synthesized speech involve, at one stage or another, an exploration of the perceptual effects of systematically varying the synthesis parameters. Typically, stimuli which bear such a parametric relationship to each other are conceptualized in terms of a "continuum" in which adjacent stimuli differ by a step in one (or more) parameter(s). Using conventional procedures, the creation of such a parametrically-varied continuum is slow and requires a high degree of attention, if errors are to be avoided; moreover the task is distinctly uninspiring. To facilitate such explorations, we have therefore developed a "Continuum Generator", in the form of a high-level "programming language".

In its simplest application, the Continuum Generator allows a researcher to specify a range of values which a variable parameter is to assume -- starting value, number of steps, and size of steps -- and then to synthesize all of the speech signals which would result from the combination of each of the possible values of the variable parameter, in turn, together with a specified set of fixed values for all of the other parameters which are required to control the synthesizer. In this way, the Continuum Generator operates much like a "do" or "for" loop in a conventional high-level programming language such as Basic or Fortran. By extrapolation, nested loops can also be generated, resulting in the full set of synthesized speech signals produced by the factorial combination of two (or more) sets of variable parameters.

Using these procedures, we have been able to quickly generate large numbers of speech signals for use in speech perception experiments, with a minimum of effort, and with the details of the synthesis parameters controlled automatically (and without error). Indeed, some experiments using these procedures involve several hundred parametrically-synthesized signals -- offering new opportunities to explore combinations of parameters which would previously have been impossible because of the time constraints of traditional synthesis techniques. One application in which the Continuum Generator has proved to be of particular value involves the study of interactions between parameters; another valuable application involves the identification of combinations of parameters which result in very-high quality synthesized speech.

IV. EXPERIMENT GENERATOR

The Experiment Generator is a menu driven program which permits an experimenter to implement an audio experiment by filling a form specifying values for the predictable parameters of audio stimulus file names, number of times each signal is to be presented per block, rate of presentation, time permitted for response, inter-trial interval, and so forth. The program then produces an experiment specification file which is read and implemented by the Experiment Controller to carry out the experiment as specified. The response timing within the experiment is controlled by a clock which is preprogrammed to produce a resolution of 1 ms. The parameters such as stimulus and block lists which make up the experiment can be generated/edited through the same menu driven system.

V. FUTURE PLANS

To date, CSRE has been used in four ways: 1) to edit natural speech tokens to produce standardized tests of speech intelligibility for hearing impaired listeners (e.g., Jamieson, DellOrletta, and Ramji, 1988); 2) to prepare synthetic speech signals for experiments on the perception of native-language contrasts (Cheesman, 1989) and non-native speech sounds, during second language training (Jamieson and Morosan, 1989); 3) to analyze the temporal and spectral properties of English and Mandarin Chinese fricative speech sounds; and 4) to analyze lung and airway sounds associated with different respiratory tract disorders (Slawinska, Jamieson, McMillan, and Wake, 1987).

In our current phase of the project, we are expanding the range of application of CSRE in these areas, adding new features such as F0 extraction, formant tracking, and parameterization for burst and fricative sounds, and investigating a range of applications in the area of the disorders of human communication. In addition to expanding the range of applications of the CSRE software, our future work will focus on three topics: 1) the development of higher-level tools to support speech research; 2) the implementation of specific signal- and graphics-processing procedures, using more powerful processors; 3) the comparative study of alternative processing and display procedures, including, where required, the development of new algorithms and procedures.

ACKNOWLEDGEMENT AND AVAILABILITY

CSRE is being developed and distributed on a nonprofit, cooperative/shareware basis. Researchers who are interested in using CSRE can contribute to the project in one of two ways. First, they can contribute programs and/or utilities which they have written, which will increase the usefulness of the package, and can be adapted to form an integrated package, working together with the other parts of CSRE, as part of subsequent releases of the package. Second, researchers can make a financial contribution (presently $250) in return for the CSRE software. The funds obtained in this manner will be directed to the further development of the package. Interested researchers should contact Dr. D.G. Jamieson at the indicated address, or by e-mail (JAMIESON@UWOVAX.UWO.CA)

REFERENCES

Cheesman, M.F. Acoustical factors underlying speech perception. Doctoral dissertation, University of Minnesota, 1989.

Kania, A., Jamieson, D.G., Nearey, T.M., and Ramji, K.V. (1988). A comparison of alternative procedures for digital spectral analysis of speech. **Journal of the Acoustical Society of America, 84, S1.**

Jamieson, D.G. and Morosan, D. (1989). Evaluation of a technique for training new speech contrasts: Generalization across voices, but not word-position or task. **Journal of Speech and Hearing Research, 32, 501-511.**

Jamieson, D.G. and Nearey, T.M. (1988). CSRE : A Speech Research Environment. **Abstracts of the Association for Research In Otolaryngology, 11,** 228.

Jamieson, D.G., Seewald, R.C., and Raftery, E. (1988). A general-purpose hearing aid prescription, simulation, and testing system. **Canadian Acoustical Association,** annual meeting, Toronto, October, 1988.

Slawinski, E.B., Jamieson, D.G., McMillan, D.D., Wake, C.H. and Hambly, M.D. (1987). Acoustical analysis of respiration sounds in neonates. **Proceedings of the Canadian Acoustical Association,** 85-91.

Yaphe, M., Raftery, E., and Jamieson, D.G.. (1989). A general-purpose facility for adaptive testing in psychoacoustics. **Behavioral Research Methods, Instrumentation, and Computers, 21,** 275-280.