ASPA: Automatic speech-pause analyzer*

D. GERVERT† and G. DINELEY
University of Durham, Durham, England

This paper describes a suite of computer programs for monitoring patterns of speaking and pausing from twin-channel tape-recorder output.

The programs to be described below were developed in connection with research on patterns of speaking and pausing in simultaneous conference interpretation. The aim was to monitor, in real time, twin-channel tape recordings of speech in order to produce as output a digital record of the presence or absence of speech on each channel, together with the length of time the channels remained in any particular state:

Channel 1 on - Channel 2 on
Channel 1 on - Channel 2 off
Channel 1 off - Channel 2 on
Channel 1 off - Channel 2 off

The programs were developed with the following points in mind:

(1) In order to provide a more consistent waveform for sampling, the envelope of the speech waveform, rather than the speech waveform itself, was to be sampled.

(2) In order to reduce the amount of data handled by the programs, data should be initially processed in the sampling program.

(3) Since the signal being sampled (i.e., the envelope of the speech waveform) would be of a low frequency, and in order to reduce the amount of data handled, a relatively slow sampling rate (100 times/sec) was decided upon.

(4) For speed and convenience, the data were to be stored in disk files.

(5) The criterion for a pause was to be ¼ sec. Any break in speech on either channel of less than ¼ sec was to be discounted, and the interval regarded as continuous with the speech signal on either side of it on the relevant channel.

(6) For simplicity, and in order to facilitate debugging, the programs should be developed in modular form.

The flow chart in Fig. 1 illustrates the sequence of eight programs. These are described more fully below.

EQUIPMENT

(1) Computer: IBM 1130, with

*The research for which these programs were developed was supported by the Social Science Research Council.

†Requests for reprints should be addressed to Dr. D. Gerver, Department of Psychology, Queens University, Kingston, Ontario, Canada.

8,000 16-bit words of core storage and disk drive holding 512,000 words; also a 1134 paper-tape reader and a 1131 console typewriter output.

(2) Interface: With analog input and output, variable-rate interrupt generator, and clock.

(3) Oscilloscope: Four-track slow scan.

(4) Stereo tape recorder.

(5) Two envelope follower circuits, each consisting of a simple rectifier with capacitor smoothing and a decay resistor.

ASPA: The Programs

Since the actual details of interface sampling, disk storage routines, etc., will depend upon the user's own particular computer configuration, the programs are presented in flow-chart form with textual descriptions.

(1) CLEAR

CLEAR works entirely with disk output. The sole purpose of CLEAR is to clear all disk data files used to zero. This is necessary at the beginning of each run in order to avoid pickup of data from previous runs. See Fig. 2.

(2) OMOTS

OMOTS consists essentially of three programs that sample the digitized envelope of the speech waveform at the interface and store the results of the sampling on disk. The configuration of equipment necessary for operating this program is shown in Fig. 3 and is explained in the text following.

Fig. 1. Sequence of programs and operations in ASPA.

Fig. 2. CLEAR.

The composite program first reads a piece of paper tape which defines the maximum number of disk file sectors which can be used, the trigger levels, and bias for sampling the envelope of the speech signal. The program then prepares itself to accept interrupts and waits. This allows the operator to prepare the equipment, i.e., advance the experimental tapes to the point required and start the replay. When the Program Start key on the computer keyboard is pressed, the generation by TIMON of Level 3 interrupts at the interface is initiated. These are timed to arrive at 10-msec intervals. On receipt of an interrupt, the program samples two of the interface analog inputs and decides whether they exceed a certain critical

level, which is independently variable on both channels by means of the trigger levels and bias defined on the paper tape. The program then decides whether one or both channels have changed from one state to another since the previous sample. It constructs a data set of the form TTTS and stores it in a buffer in the program; where TTT = time in 1/100 sec since the last change, and S = present state of the channels:

0 = both off
 1 = Channel 1 on, 2 off
 2 = Channel 1 off, 2 on
 3 = both on

This program has two buffers so that when the first is full it can interchange them, storing data in the second while writing the contents of the first onto disk. This process of double buffering enables the program to continue sampling unimpaired, even though it is simultaneously sampling and transferring data to disk and despite the fact that data are being produced from an irregular sampling of the channels. On receiving a signal from the keyboard (i.e., on termination of the relevant portion of tape being analyzed) the program commands the interface to stop generating interrupts, writes the contents of the last buffer onto the disk and ends, leaving the computer ready for the next program. The program also has a facility for clamping both channels so that neither can change state more frequently than a predetermined rate (in this case 5/100 sec); this was incorporated to reduce the amount of data produced. OMOTS and the sequence of subprograms are shown in Figs. 4 and 5.

Fig. 3. Input configuration.

Fig. 4. OMOTS.

The following inputs and outputs are employed with OMOTS:

(1) Paper-tape input describing various parameters: length of disk file to be used, trigger level, bias for differential trigger levels, if required.

(2) Analog inputs on the interface for monitoring tape recorder output. It should be stressed that the tapes to be analyzed should be as free as possible of any background noise. Where necessary, a bandpass filter should be used to eliminate unwanted high- or low-frequency interference.

(3) Disk output, for recording the data.

(4) Analog outputs from the interface. These (as shown in Fig. 3) are displayed on two channels of the four-channel oscilloscope and provide the operator with a picture of what the program "thinks it is hearing." These can be compared with the auditory signals and their displays on the other two oscilloscope channels, and the level of the tape-recorder output can be adjusted until the operator is satisfied that the computer is satisfactorily tracking the recorded signals.

Before analyzing each tape, OMOTS is run in order to set the appropriate levels. This is done using a test tape of known signal-silence times, and the actual experimental tape in question.

When setting up the equipment for use, the test tape should be prepared and run through ASPA, comparing the final output with the known test tape events and event times.

(3) SHUFF

This program (see Fig. 6) reads from disk and writes back onto disk.

It inverts the data produced by OMOTS, converting an assembler array into a FORTRAN-compatible array. It also converts the data words into a new form, TTTS, by building up a new word, using the time from the present data word and the state from the previous data word. One data word is lost in this process. This program is necessary, since the data produced by OMOTS represent the time the system was in a state *before* it changed to its current state, whereas the final analysis requires the time the system is actually in a particular state.

Fig. 5. OMOTS (SET, TIMON, RET).

Fig. 7. DROPM.

Fig. 8. CMPRS.

Fig. 9. CLEAN.

(4) DROM

This program (Fig. 7) reads from disk and writes back onto disk, preparing the data from SHUFF so that pauses of less than $\frac{1}{4}$ sec can be eliminated by effectively switching back on any channel which has switched off for less than $\frac{1}{4}$ sec. This interval can, of course, be varied to suit the user.

(5) CMPRS

This program (Fig. 8) also works from disk to disk, taking the data from DROM and compressing any sequences of data words which have the same state into one data word which has as its time the sum of sequence times.

(6) CLEAN

CLEAN is used to eliminate noise at the start of the record. This could be due to switching on the tape recorder, noise on the tape, etc. The program (Fig. 9) prints out the start (e.g., the first 10 events and times) of the data produced by the previous programs, and waits for a number to be typed in. The number is selected by the operator on inspecting the data printout. For instance, in the experiment for which these programs were developed, all data sets should have started with a short interval (e.g., .8 sec) with Channel 1 on, Channel 2 off. If the data set showed 0102 0053 0041 0033 0801, the operator would type in the last figure, and all preceding data words would be eliminated for the final analyses.

(7) CRASH

CRASH reads from disk, analyzes the data, and prints the results. Since the type of analysis required (e.g., time x state frequency distributions, means and variances of times spent in each state, etc.) depends on the user, no description of this program is provided here.

(8) SHOWD

Reads from disk files and prints the stored data. Since this depends on the user's individual requirements, no further description is provided.