Seventh Quarterly Progress Report

March 26 through June 26, 1985

NIH Contract N01-NS-2356

Speech Processors for Auditory Prostheses

Prepared by

Blake S. Wilson, Charles C. Finley and Dewey T. Lawson

Neuroscience Program Office
Research Triangle Institute
Research Triangle Park, NC 27709

CONTENTS

I. Introduction

The purpose of this project is to design and evaluate speech processors for auditory prostheses. Ideally, the processors will extract (or preserve) from speech those parameters that are essential for intelligibility and then appropriately encode these parameters for electrical stimulation of the auditory nerve. Work in the present quarter included the following:

1. Psychophysical and speech-testing studies with the most-recent patient in the experimental series at the University of California at San Francisco (UCSF);

2. Further development of software and hardware for support of such studies;

3. Initial development of an ensemble model of the spatial and temporal patterns of neural discharge produced by intracochlear electrical stimulation; and

4. Preparation for two major presentations at the Gordon Conference on "Implantable Auditory Prostheses."

In this report we will describe the speech-testing studies indicated in point 1 above. Briefly, the patient (LP) presented a tremendous challenge to the UCSF and RTI teams in that his psychophysical performance along almost every measured dimension was worse than any previous patient in the UCSF series. Moreover, only one of his scores on speech tests (voice/unvoice) was above chance for the "compressed analog outputs" strategy used in the present UCSF processor. With these discouraging results in mind, our approach was first to reproduce one version of the analog-type UCSF processor in the software of our block-diagram compiler; then determine if the simulated processor produced results essentially identical to those obtained with the UCSF "tabletop" analog processor; and finally to evaluate other processing strategies in an attempt to improve LP's understanding of speech. The basic plan of these other processors was to reduce in steps the temporal and spatial overlap between channels and introduce (in the last two processors tested) a representation of the

linear-prediction residual signal. In all, the block-diagram compiler was used to simulate 5 distinctly different processing strategies. As we will describe in detail in the next section of this report, some of these processors produced percepts that were clearly in the "speech mode," that were spontaneously recognized as the speech test tokens delivered to the processor, and that produced test scores well above chance on confusion-matrix material.

Additional sections in this report include (1) an update on the current status of software development for the block-diagram compiler (Appendix 1) and (2) our plans for the next quarter of project work. Descriptions of the ensemble model and of our psychophysical tests with patients LP and EHT (the previous patient in the UCSF series) are deferred for now, but will appear in future reports.

II. Speech-Testing Studies with Patient LP

The most-recent patient in the UCSF experimental series (patient LP) was intensively studied by the UCSF and RTI teams during the months of June and July, 1985. As mentioned in the Introduction of this report, the psychophysical performance of LP along almost every measured dimension was worse than any previous patient in the UCSF experimental series. Among the findings of the basic psychophysical studies were the following:

a. thresholds to stimuli delivered between bipolar electrode pairs were much higher than thresholds to the same stimuli delivered between a single monopolar electrode and a remote reference electrode, on all channels;

b. dynamic ranges to sinusoidal and pulsatile stimuli were extremely narrow compared to all other patients studied by the UCSF team (see, e.g., Table 1);

c. temporal and spatial interactions between channels were very severe for the middle channels in the electrode array and somewhat less severe for the apical-most and basal-most pairs in the array;

d. LP was able to distinguish percepts elicited by stimulation with different pairs in the electrode array, if the stimuli were delivered one at a time;

e. excitation of the middle channels strongly inhibited percepts elicited by excitation of the apical and basal channels;

f. thresholds and uncomfortable loudness levels (UCLs) were labile, and these measures exhibited shifts both within and between sessions (UCL was somewhat more stable than threshold), which further limited the useful dynamic range; and

g. the onset of UCL from most-comfortable loudness (MCL) was very sudden for most channels and most conditions of stimulation.

Not surprisingly, LP's case has been informally described by members of the UCSF team as "one tough nut to crack" and "off the map." With the exception of the voice/unvoice test of the MAC battery, none of his scores on speech tests with the present UCSF processor were above chance; indeed, heroic efforts were required just to map the processor outputs into LP's useable dynamic range. In all, the results mentioned above, along with others, were consistent with a picture of very poor survival of peripheral neural elements along the middle portion of the electrode array and at least some survival in the apical and basal segments.

RTI's active involvement in the tests with LP began in early July after several significant problems with the RTI patient stimulator and associated software were fully resolved. We were ably assisted in many of the tests by members of the UCSF team, particularly by Mark White. Our approach was first to reproduce one version of the analog-type UCSF processor in the software of our block-diagram compiler; then determine if the simulated processor produced results essentially identical to those obtained with the UCSF "tabletop" analog processor; and finally to evaluate other processing strategies in an attempt to improve LP's understanding of speech. The basic plan of these other processors was to reduce in steps the temporal and spatial overlap between channels and to introduce (in the last two processors tested) a representation of the linear-prediction residual signal. In all, the block-diagram compiler was used to simulate 5 distinctly different processing strategies. As mentioned before, some of these processors produced percepts that were clearly in the "speech mode," that were spontaneously recognized as the speech test tokens delivered to the processor, and that produced test scores well above chance on confusion-matrix material.

Signal-processing elements used in various ways by the 5 processors are shown in the block diagram of Fig. 1. Speech tokens are read from an input disk file and then high-pass filtered to flatten the speech spectrum and diminish the otherwise overwhelming influence of F1. The output of the high-pass filter is then fed to a bank of bandpass filters whose center frequencies span the combined range of F1 and F2. The RMS energy in each band is sensed by a full-wave rectifier and low-pass filter connected in series to each bandpass filter output. Finally, the "analog" outputs of the bandpass filters are mapped onto the dynamic range of the patient with

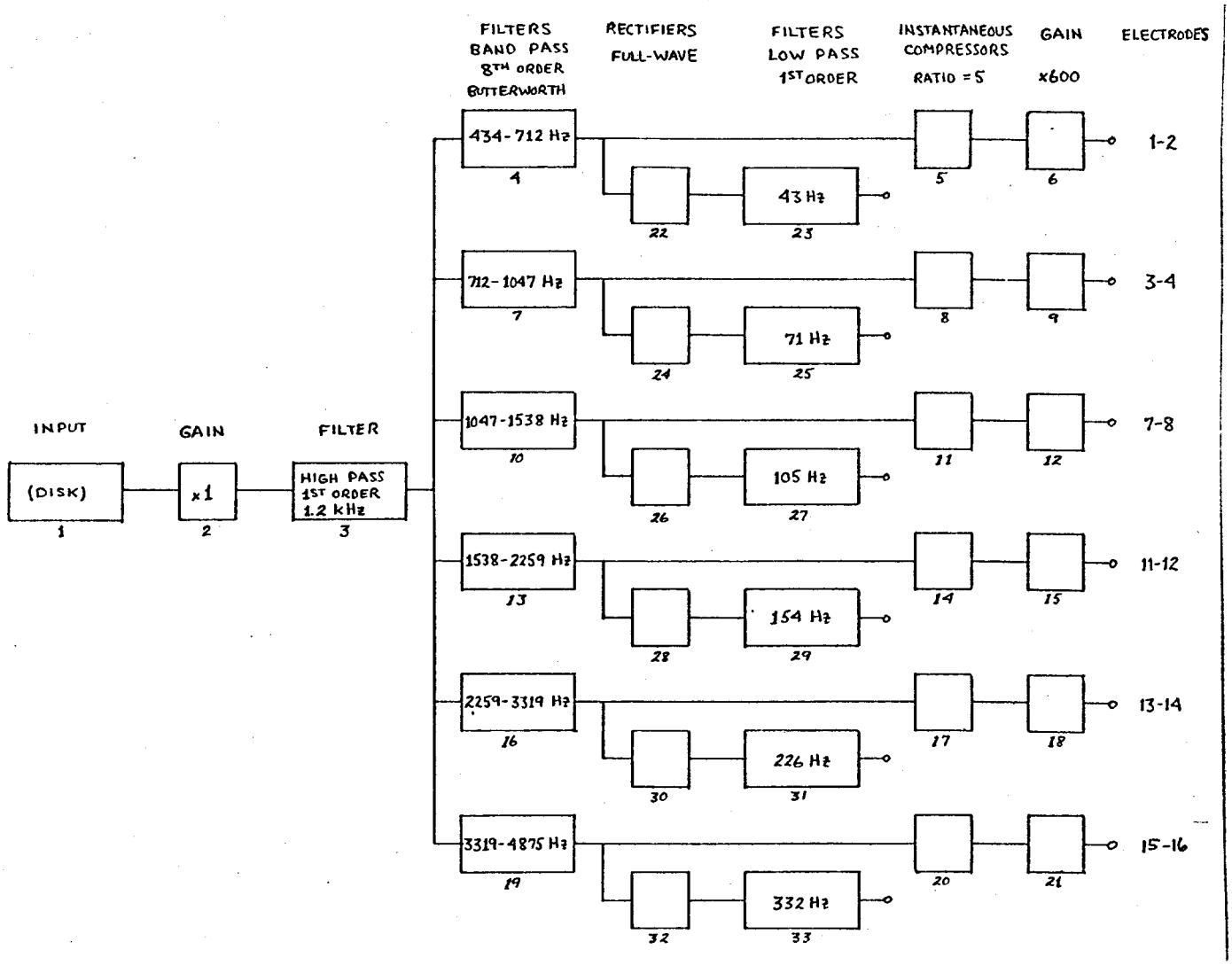


Fig. 1. Basic design used for simulation of 5 processing strategies for tests with patient LP, June and July, 1985.

instantaneous compressors (of the "compressor-compressor" type, see Appendix 1, pp. 43-47) and gain blocks for each channel.

Simulation of one variant of the basic UCSF processing strategy was accomplished by using all elements in the block diagram of Fig. 1 except those used to sense the RMS levels in each band of frequencies. Post-compression of the filter bank outputs with instantaneous compressors was selected instead of pre-compression before the filter bank with a noninstantaneous compressor (as in the usual configuration of the UCSF processor) because (1) we were concerned that "overshoots" produced at the onset of compression with noninstantaneous compressors would greatly exceed LP's UCLs and (2) we estimated, for LP's very narrow dynamic range, that the spectral distortion produced by pre-compression would be more damaging to the frequency representation at the channel outputs than would be the "de-emphasis" of this representation produced by post-compression.

Typical outputs of this simulation of the UCSF processing strategy are presented in Fig. 3 and typical outputs of the RMS level detectors are presented for reference in Fig. 2. The speech input token is the word "BOUGHT." As illustrated in Fig. 2, this token has strong energy components in the lower two frequency bands during the initial part of the vowel, a relatively strong peak in the third band near termination of the vowel, a "blip" in bands 1-5 at the end of the vowel, and burst of energies in bands 3-6 at the "t." Fig. 3 shows, in the upper-left panel, that these features are represented to some degree in the "compressed analog" outputs of the simulated UCSF processor. Details of the outputs during the initial part of the vowel and just after the onset of the "t" (right panels) further illustrate that the representation across frequencies is highly compressed and that features reflecting the excitation of the vocal tract are indistinct. Finally, the lower-left panel shows histograms of the output levels of stimuli presented during the token. In all, Fig. 3 portrays a dense and highly compressed representation of the speech token with ongoing activity in all channels during and beyond the utterance.

As was expected from the results of previous tests with the analog "tabletop" unit, results obtained with the processor of Fig. 3 were miserable. First, attenuators between the isolated outputs and implanted electrodes had to be carefully adjusted before each session (and often before each token!) to attain suprathreshold levels without exceeding the perilously-close UCLs on each channel. This activity usually consumed at

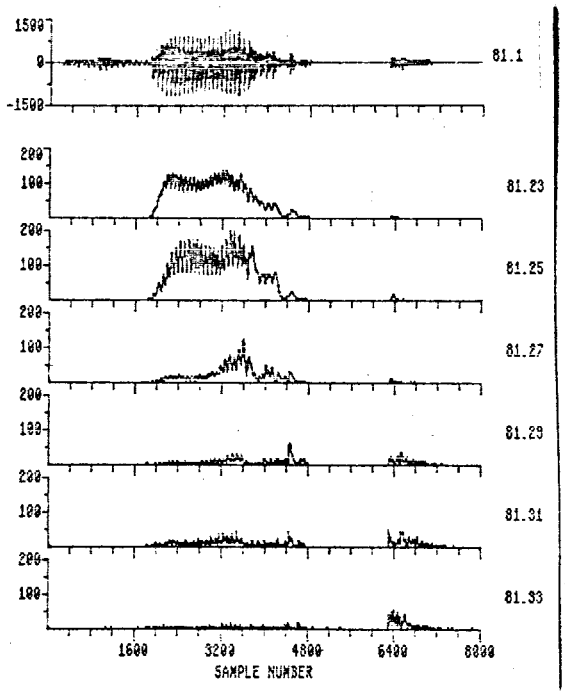


Fig. 2. Speech input (top) and band energies for the token "BOUGHT." Numbers to the right of each trace refer to outputs of the blocks with the same numbers in Fig. 1, for design # 81. The sampling frequency is 10 kHz in this and subsequent figures.

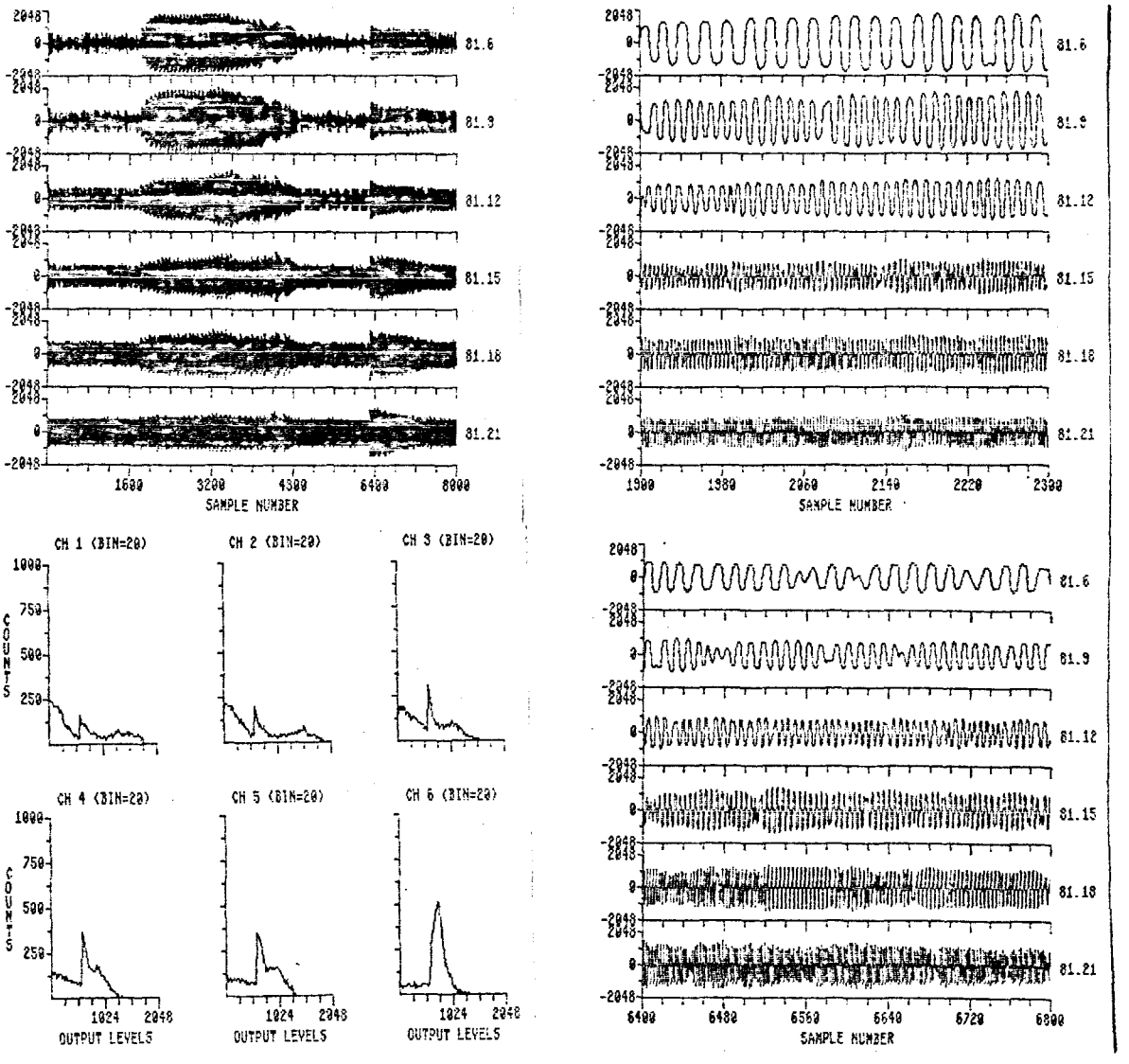


Fig. 3. Displays of outputs for speech-processing strategy 1 used with patient LP. This strategy is a simulation of the basic UCSF speech processor, in which band-pass filters divide the speech input into frequency ranges that span the first and second formants. The outputs of the filters are then compressed to map the signals onto the limited dynamic range of implant patients. Details of the particular design used for patient LP are presented in the block diagram of Fig. 1. As noted in the text, this patient had an extremely limited dynamic range (on the order of 2-3 dB on every channel for short biphasic pulses and somewhat greater for low-frequency sinusoids) which required relatively-high compression ratios for mapping. The upper-left panel shows the channel outputs for the entire token ("BOUGHT"), and the right panels show details of the output during the initial part of the vowel segment (upper right, starting at sample number 1900) and during the "t" (lower right, starting at sample number 6400). Y-axis calibrations are in digital-to-analog converter (dac) units; 2048 dac units correspond to 800 μA of current at the electrodes. Finally, the lower-left panel shows histograms of the channel outputs for the token. The bin width is 20 and the output levels are in dac units.

least half of a two-hour session. Next, once the attenuators were adjusted, the percepts were rarely speech-like in character. Instead, the processed tokens sounded "mushy," "drawn out," "indistinct" or "on all the time." The general picture that emerged from these anecdotal remarks was one of a poor representation of temporal events, possibly produced by severe channel interactions. In no case were speech tokens spontaneously identified as the words delivered to the speech processor; these words included several tokens of the vowel confusion test ("BEET," "BIT," "BOAT," "BOOT" and "BOUGHT") and of the consonant confusion test ("ADA," "AKA," "ALA," "ANA," "ASA," "ATA," "ATHA" and "AZA").

The suggestion that channel interactions might have been largely responsible for the poor results just mentioned led us to evaluate another processing strategy in which the number of simultaneous channel outputs was reduced from 6 to 2. In this design the RMS energy levels in each bandpass of frequencies were compared for every sample. The "compressed analog" output of a channel would be delivered to its assigned electrodes only if (1) it was one of the two channels with the greatest RMS energy for the present sample and (2) the RMS energy was above a preset "noise threshold." The number of channels allowed to send their outputs to the electrodes could therefore decline to one or none during relatively-quiet intervals.

Typical outputs of this second speech-processing strategy are presented in Fig. 4. As is obvious from the figure, this strategy has a greatly reduced bandwidth of transmission to the electrode array compared to the previous strategy illustrated in Fig. 3. In particular, there are quiet intervals during silent or low-energy segments of the input speech token; no more than two channels are "on" at a time; and, as the histograms in the lower-left panel show, the number of suprathreshold samples delivered to the electrode array is greatly reduced (note the scale change on the histograms). Unfortunately, however, this processing strategy did not produce a marked improvement in performance. The tokens did sound more "speech-like" in character but not one was spontaneously identified as a word, much less the correct word. The percepts from this processor had better "temporal dynamics" than the first processor, and the tokens were no longer described as "on all the time." As before, much of the time available in each session was consumed in repeated adjustments of the attenuators.

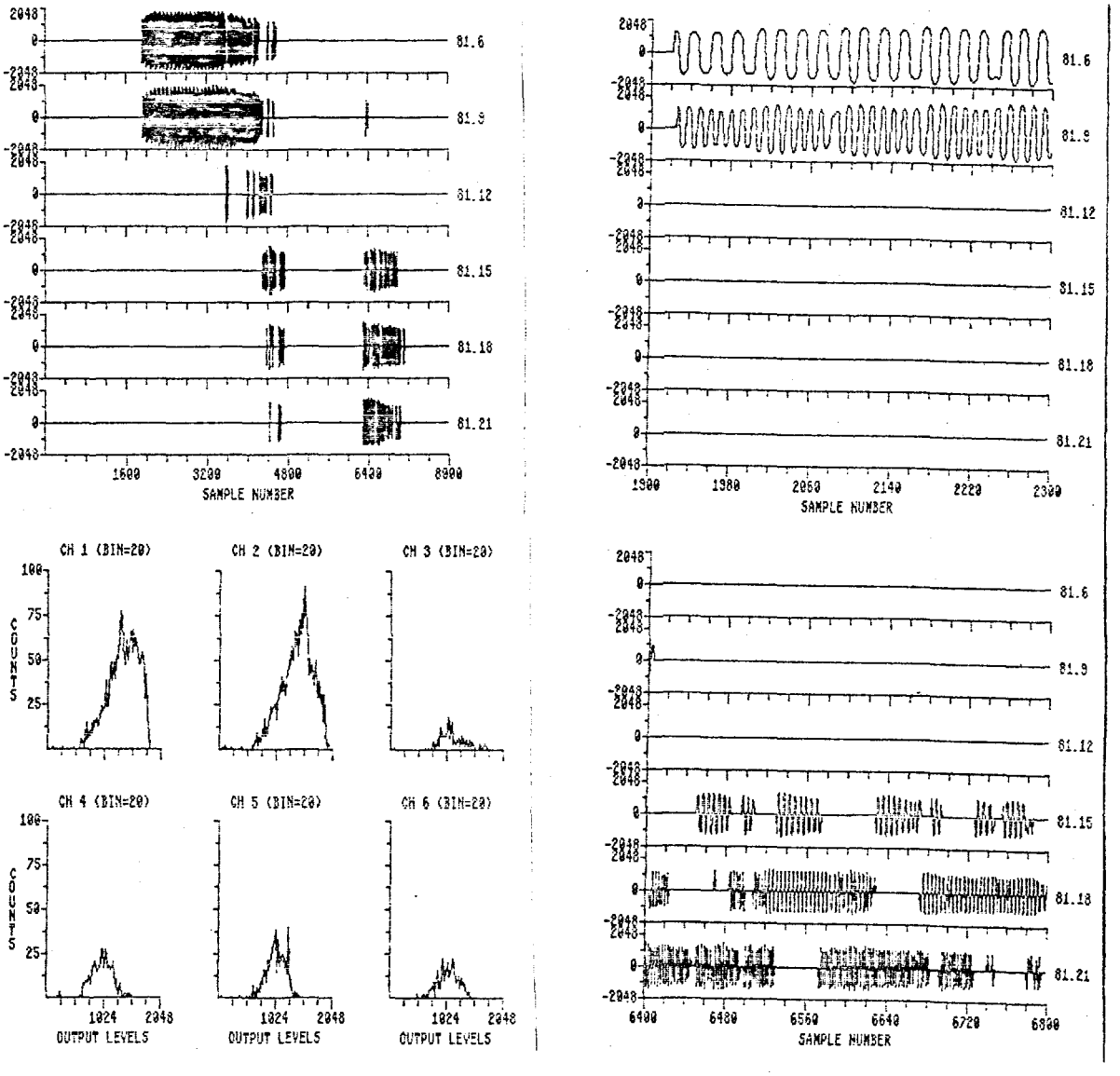


Fig. 4. Displays of outputs for speech-processing strategy 2 used with patient LP. This strategy is the same as that illustrated in Fig. 3, except that only two channels are allowed to send their outputs to the electrodes at any one time. The channels selected for output to the electrodes are those (1) with the greatest RMS energy in their frequency bands at each sampling interval and (2) that have RMS energies above a preset "noise threshold." The number of channels allowed to send outputs to the electrodes can decline to one or none during relatively-quiet intervals. The overall strategy illustrated in this figure has a greatly reduced spatial bandwidth of transmission to the electrode array compared to the previous strategy illustrated in Fig. 3.

The next step in our sequence of processing strategies was to reduce even further the bandwidth of transmission to the electrode array and to abandon "compressed analog" stimulation in favor of pulsatile stimulation. The latter choice was made because LP's thresholds and UCLs were somewhat more stable for short-duration biphasic pulses than for sinusoids (of frequencies that approximated the center frequencies of the bandpass filters) and because we could exert better control over temporal interactions with pulses. As with the strategy of Fig. 4, the RMS energy levels in each band were compared at each sample to select the two or fewer electrode pairs in the array to receive stimuli. The stimuli delivered to the selected electrodes were balanced biphasic pulses with a duration of 300 μsec/phase. This was the minimum duration that would allow us to span the range from threshold to UCL on all channels. Information reflecting the relative levels of the selected bandpass channels was transmitted to the electrode array through a compression law of the form:

$$\text{pulse intensity} = A \times \log(\text{RMS level}) + k,$$

where the parameters "A" and "k" were determined for each channel according to the threshold and UCL for pulses on that channel. The parameters used, along with target thresholds and UCLs for each channel, are presented in Table 1.

A final aspect of the design of processor 3 was that the pulses delivered to the two selected channels (when two channels were selected) were interleaved so that the onset of a pulse on one channel would never follow the offset of a pulse on another channel within an interval of less than 1 msec. Because short-term temporal integration fell off rapidly at and beyond 1 msec for LP, we hoped this interleaving of stimuli would reduce temporal interactions between channels.

The outputs of processor 3 for the input token "BOUGHT" are shown in Fig. 5. As would be expected from the description above, pulsatile outputs to the selected channels are strictly interleaved and the bandwidth of transmission to the electrode array is greatly reduced compared to strategy 2 (Fig. 4). In addition, some voicing and voice/unvoice (v/uv) information is presented to the electrode array in that the pulse amplitudes coarsely follow the "ripples" in the outputs of the RMS level detectors that correspond to pitch periods. Although this information is difficult to

Table 1. Parameters used for mapping pulsatile stimuli onto LP's psychophysical space.

Channel	Electrode Pair	Min RMS	Max RMS	Target Threshold*	Target UCL*	A	k
1	1-2	10	139	380	1080	612	-232
2	3-4	10	207	1400	2000	456	944
3	7-8	10	190	1250	2000	587	664
4	11-12	10	157	650	745	74	576
5	13-14	10	150	800	1090	247	553
6	15-16	10	81	480	760	308	172

*Digital-to-analog converter (dac) units; 2048 dac units correspond to 800 μA at the electrodes.

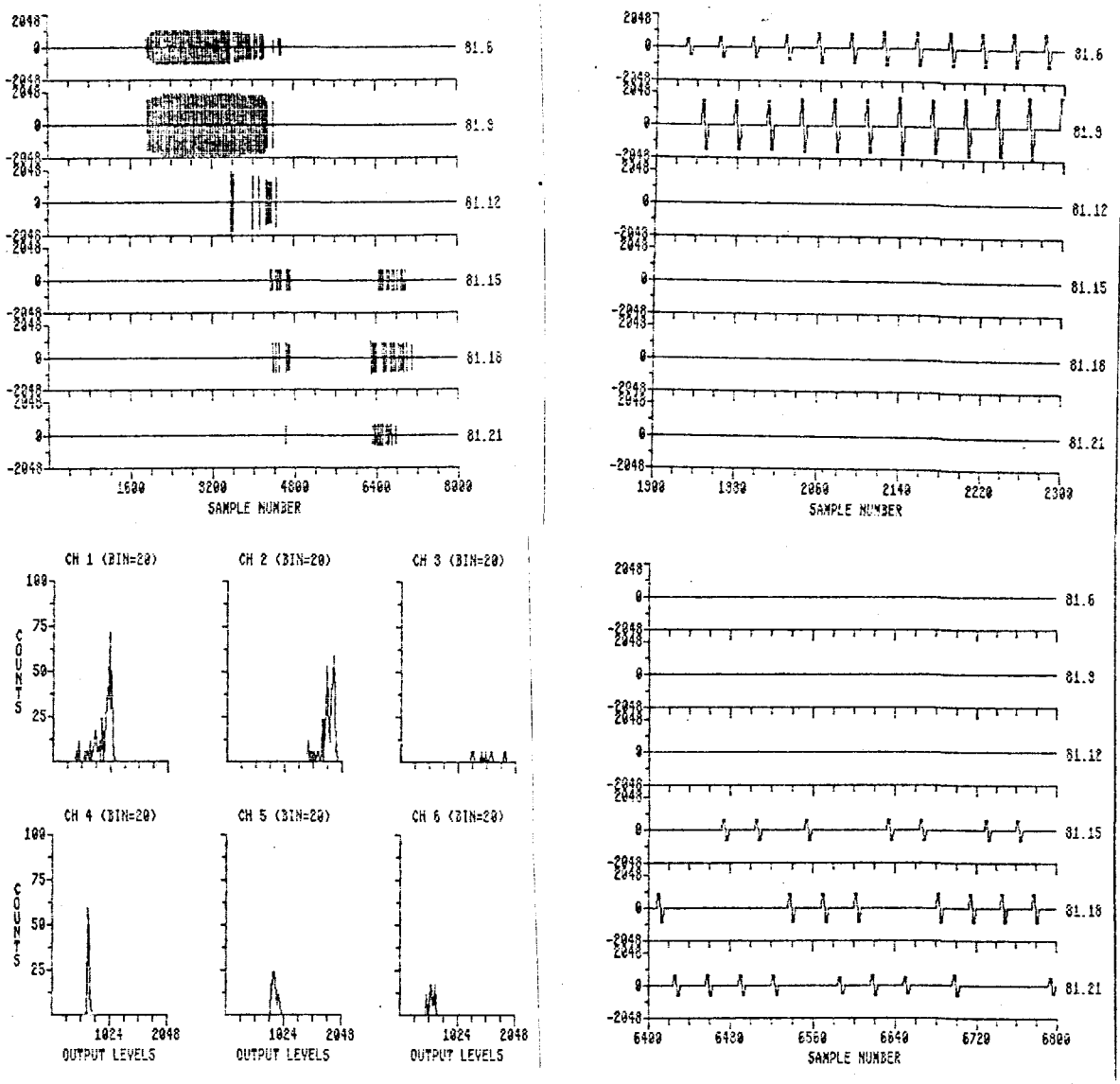


Fig. 5. Displays of outputs for speech-processing strategy 3 used with patient LP. This strategy selects channels for output to the electrode array, as in strategy 2 illustrated in Fig. 4, but presents balanced biphasic pulses instead of the compressed analog signals of the previous strategies. The pulse amplitudes are determined by a compression law of the form:

$$\text{pulse intensity} = A \times \log(\text{RMS level}) + k,$$

where parameters "A" and "k" are preset for each channel according to the threshold and uncomfortable loudness level (UCL) for pulses on that channel. The duration of all pulses is 300 μsec/phase, which allowed us to span the dynamic range on all channels for patient LP (200 μsec/phase was too short). Finally, the pulses are interleaved so that the onset of a pulse on one channel never follows the offset of a pulse on another channel within an interval of less than 1 msec. Because short-term temporal integration declined rapidly at and beyond 1 msec for LP, we hoped this interleaving of stimuli would reduce temporal interactions between channels.

appreciate in the highly-compressed mapped stimuli shown in Fig. 5, every second or third pulse has an amplitude greater than its neighbors in the upper-right panel (which shows a voiced interval). Obviously, this representation of voicing is extremely crude and would be much improved for higher pulse frequencies; however, strict interleaving of the pulses precluded this possibility.

We are pleased to report that the percepts elicited with processing strategy 3 were all in the "speech mode," that most of the tokens in the vowel confusion matrix were spontaneously recognized as the correct words, and that half of the tested tokens in the consonant confusion matrix were spontaneously recognized as the correct words. The improvement over processing strategy 2 was immediate and compelling. Moreover, the mapping rules of strategy 3 produced (for the first time) a tolerable range of loudnesses across tokens. Although formal tests were not conducted, these loudnesses also appeared to have far greater stability than the loudnesses of tokens produced with strategy 2. In all, it was clear to us and clear to the patient that speech information was making its way onto the nerve. A record of LP's initial reports in listening to the outputs of processor 3 is presented in Table 2. Of the 11 tokens presented after our first adjustment of processor outputs, 7 were immediately and spontaneously recognized as the correct words. Unfortunately, time ran out in the session before we were able to conduct formal tests on vowel and consonant confusions.

The apparent confusions between "ASA," "AZA" and other consonant pairs (see Table 2), and the possible need to fit LP with an extracochlear electrode (we were informed that LP might have to have his intracochlear device removed due to potential infection), led us to the design and simulation of strategy 4, in which we sought a better representation of the excitation of the vocal tract. A relatively-new technique of speech analysis, "multipulse excitation," was applied. The derivation of multipulse excitation waveforms is illustrated in Fig. 6. First, the linear prediction (LP) residual signal is extracted from a high-order (e.g., 10) LP analysis of the speech wave. Then procedures similar to those described by Atal and Remde (1982) are used to derive a sequence of balanced biphasic pulses whose amplitudes and positions in time correspond to perceptually-significant events in the LP residual signal. As before, the duration of the pulses is 300 μsec/phase and no pulse follows another within 1 msec. The LP residual and multipulse excitation sequence for the token "BOUGHT"

Table 2. Initial reports made by LP in listening to the outputs of processor 3.

Token*	Report
BOOT	near threshold ("not loud enough to make it out")
BOUGHT	spontaneous recognition ("a perfect 'BOUGHT'")
BOAT	spontaneous recognition ("you're saying 'BOAT'; the sound is nice and has a good loudness")
BIT	spontaneous recognition
BEET	spontaneous recognition ("the 'EE' is high in pitch; the word 'BEET' is very clear")
ADA	spontaneous recognition ("close to 'ATA', but is clearly 'ADA'; that's a good 'ADA'")
AKA	not recognized ("could be 'ADA' or 'ATA'")
ANA	spontaneous recognition ("sounds just like 'ANA'; a beautiful 'ANA'!")
ASA	not recognized ("can't tell")
ATA	spontaneous recognition
AZA	not recognized ("could be 'ASA' or 'AZA'; there's no way I could tell the difference between those two")

*Note: Tokens 'ALA' and 'ATHA' were not presented in the initial tests with processor 3.

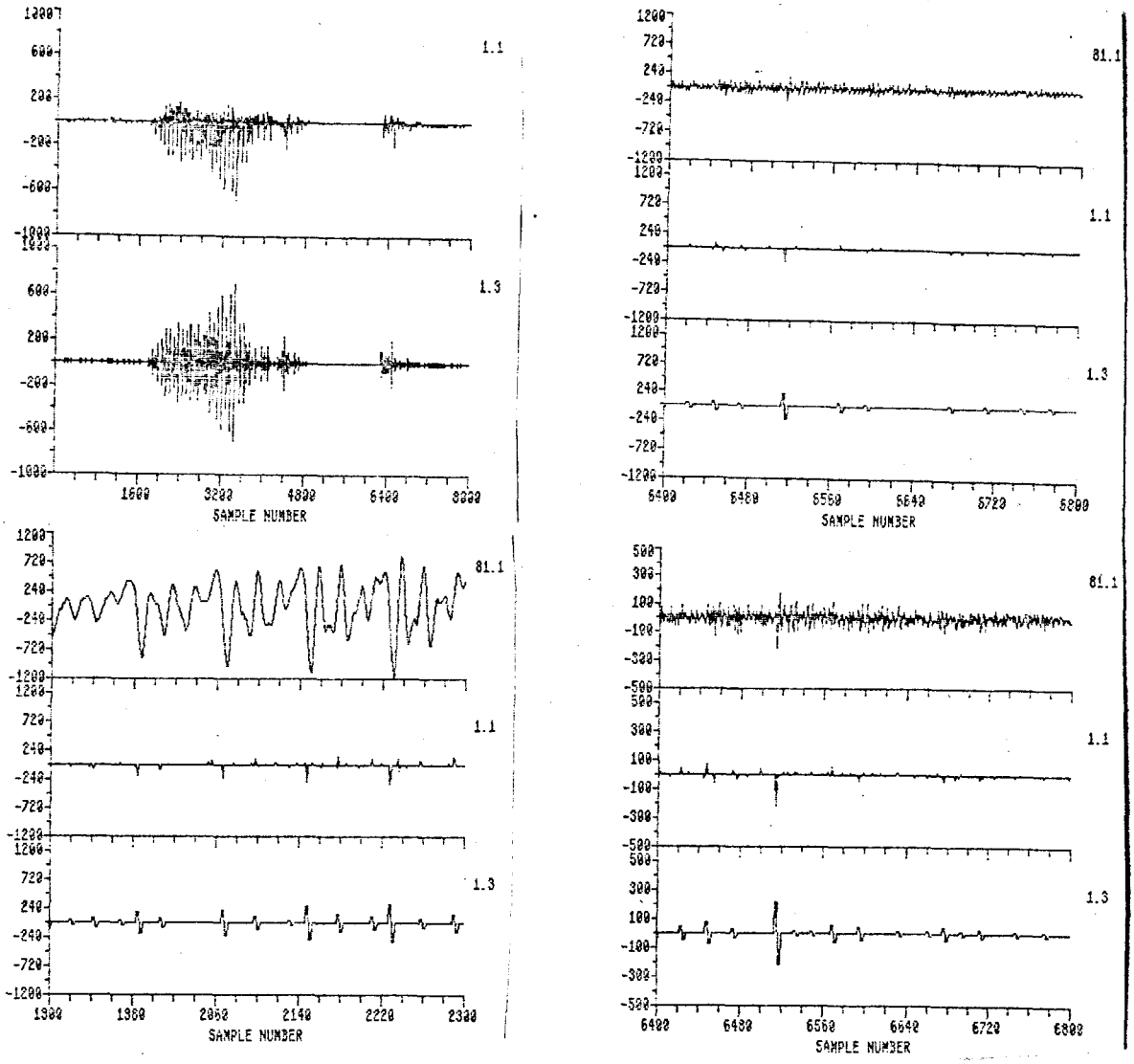


Fig. 6. Derivation of "multipulse excitation" waveforms. The upper-left panel shows the linear-prediction residual signal for a 10th-order model of the input speech token "BOUGHT" (top trace) and the derived multipulse excitation sequence (bottom trace; see Atal and Remde, 1982). The pulses in the multipulse excitation sequence are 300 μsec/phase, balanced biphasic pulses, as in strategy 3 (Fig. 5). As before, no pulse follows another within 1 msec, and no more than 6 pulses are selected for each 10 msec frame. The lower-left panel shows a detail of waveforms typical of voiced-speech segments. The upper trace is the raw speech waveform, the middle trace is the LP residual of this waveform, and the bottom trace is the multipulse excitation sequence for this residual signal. Note that the multipulse excitation sequence is highly periodic and reflects well the major periodic events in the LP residual and input speech waveforms. Finally, the upper-right panel shows a detail of waveforms typical of unvoiced speech. The waveforms are centered on the "T" of "BOUGHT." Here, the raw speech, LP residual and multipulse excitation signals are aperiodic and their amplitudes are generally less than the corresponding signals of the voiced-speech panel. The lower-right panel is a magnified view of the upper-right panel, to illustrate better the characteristics of the multipulse excitation sequence found for unvoiced consonants.

are shown in the upper-left panel of Fig. 6. Details of these waveforms and the input speech waveforms are shown in the remaining panels of the figure. The lower-left panel presents waveforms typical of voiced speech. These waveforms demonstrate that the multipulse excitation sequence is highly periodic for voiced speech and that it reflects well the major periodic events in the LP residual and input speech waveforms. The right panels show details of waveforms typical of unvoiced speech; here, the raw speech, LP residual and multipulse excitation signals are aperiodic and their amplitudes are generally less than the amplitudes of the corresponding signals for voiced speech sounds. In all, the multipulse excitation sequence represents well the fundamental frequency of voiced speech sounds and whether the present speech sound is voiced or unvoiced (or a mixture of the two). In addition, some information on the amplitude of excitation of the vocal tract is present in the multipulse sequence.

Application of the derived multipulse excitation sequence in a speech processor for an auditory prosthesis is illustrated in Fig. 7. This strategy (strategy 4) selects channels for output to the electrode array, as in strategies 2 and 3, but presents the multipulse excitation sequence instead of compressed analog signals (strategy 2) or rigidly-periodic biphasic pulses (strategy 3). As with strategy 3, the pulses of the multipulse excitation sequence are interleaved for the two selected channels. However, the minimum time interval between the offset of a pulse on one channel and the onset of the next pulse on another channel is reduced to 400 μsec, in order to preserve the character of the multipulse excitation sequence. Finally, the pulses are mapped onto the psychophysical space of the implant subject with a compression law of the form:

$$\text{pulse intensity} = A \times \log(\text{multipulse excitation level}) + k,$$

where parameters "A" and "k" are specified for each channel according to the threshold and UCL for pulses on that channel (see Table 1). This compression law differs from the one used for strategy 3 in that the multipulse excitation level is represented instead of the RMS energy level in the selected channels. Thus, one might expect that strategy 4 would provide a better representation of vocal-tract excitation than strategy 3, but only at the expense of increased temporal overlap between channels and no representation of the relative RMS levels between the two selected

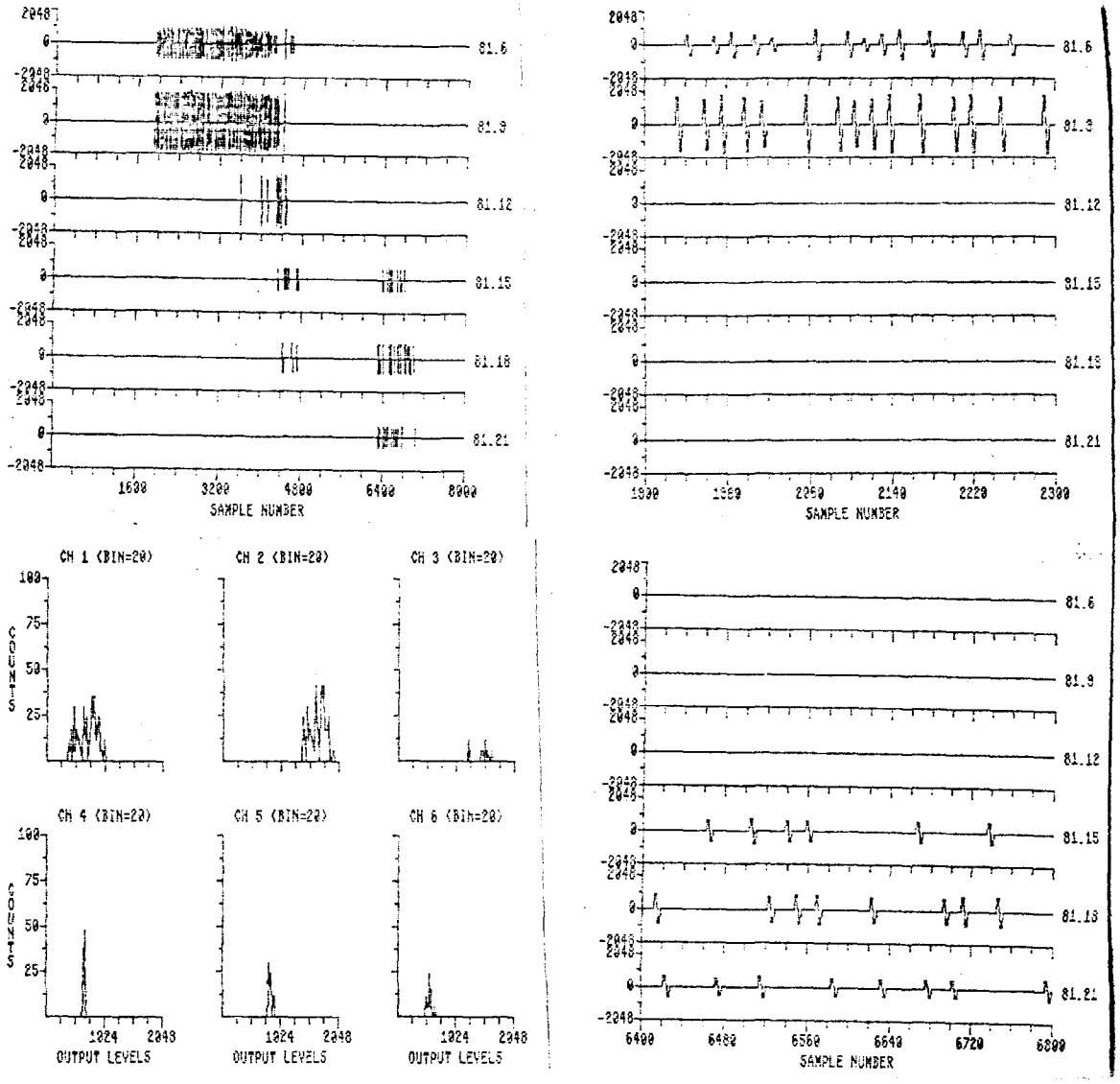


Fig. 7. Displays of outputs for speech-processing strategy 4 used with patient LP. This strategy selects channels for output to the electrode array, as in strategies 2 and 3, but presents the multipulse excitation sequence instead of compressed analog signals (strategy 2) or rigidly-periodic biphasic pulses (strategy 3). The multipulse excitation sequence is derived using the procedures illustrated in Fig. 6, and each pulse is mapped onto the psychophysical space of the implant patient with a compression law of the form:

$$\text{pulse intensity} = A \times \log(\text{multipulse excitation level}) + k,$$

where parameters "A" and "k" are preset for each channel according to the threshold and UCL for pulses on that channel. As with strategy 3, the pulses of the multipulse excitation sequence are interleaved for the two selected channels. However, the minimum time interval between the offset of a pulse on one channel and the onset of the next pulse on another channel is reduced to 400 μsec, in order to preserve the character of the multipulse excitation sequence on both channels.

outputs.

The results obtained with strategy 4 were generally encouraging. First, and somewhat surprisingly, single-channel stimulation with the gated output of channel 1 produced spontaneous recognition of several vowel tokens. For example, when we "brought up" channel 1 (by decreasing the attenuation between the isolated driver and electrode pair 1-2) at the beginning of a session, LP would instruct us to "stop there! You already have a good 'BOUGHT', but the 'b' and 't' are very weak." Although we did not have time to conduct formal tests to measure performance with this single-channel paradigm, it was our strong impression that voicing information was well-represented by the multipulse excitation sequence. Indeed, LP remarked that "this processor sounds more natural than the one you tried yesterday" (strategy 3) and later remarked that the six channel processor using multipulse excitation sounded "even more like speech" than strategy 3. LP never described the percepts elicited with strategy 3 as speech-like until at least 3 of the 6 channels had their outputs mapped onto his dynamic range.

Several variants of the "multichannel, multipulse excitation" processor were then evaluated along with a reduced 4-channel version of the "interleaved pulses" processor (strategy 3). The variants of the multichannel, multipulse excitation processor included (1) a reduced 4-channel version; (2) a reduced 5-channel version, in which the output to the most-inhibitory of the original 6 channels was disconnected; and (3) the same 5-channel strategy as in 2 above, except that only one channel was selected for output at each time frame. Evaluation of the 4-channel processors was initiated in response to a request from the UCSF team to identify, as best we could with the few remaining sessions we had with LP, a more-or-less "optimal" configuration for LP's portable processor. The request was made because the UCSF team needed additional information to (1) make an informed risk/benefit decision on whether to remove the device (the medical situation was not improving) and (2) decide on which 4 electrode pairs in the array should be used for the 4-channel transcutaneous link, if a decision was made to leave the electrode array in place. The reduced 5-channel processors were evaluated in an effort to improve the performance of the 6-channel processors by eliminating the strongly-inhibitory inputs of channel 4 (pair 11-12). Finally, we evaluated the last 5-channel strategy, in which only one channel was selected for output at a time, to gauge the

effects of a further reduction in the spatial and temporal bandwidths of transmission to the electrode array (displays of outputs for this last strategy are presented in Fig. 8).

The results of tests with randomized presentations of tokens of the vowel confusion matrix are summarized in Table 3. All scores are well (and significantly) above chance (20%) but below levels one might expect from the very encouraging anecdotal reports we obtained with the 6-channel, interleaved pulses processor (Table 2). In addition, the scores obtained with the multipulse excitation processors are not significantly better than the score obtained with the interleaved pulses processor. That is, despite the fact that speech tokens sounded much more "natural" with the multipulse excitation sequence, vowel discrimination in most cases actually declined. Finally, we note that the results of tests with consonant confusion matrices are only partially analyzed at this time. The "preliminary returns" suggest that consonant-confusion scores will be somewhat lower than the vowel-confusion scores presented in Table 3.

To conclude this section on our experience with patient LP, we will present the following preliminary observations and interpretations of results:

a. Although some patients have good or excellent performance with multichannel processors that present "compressed analog" signals at their outputs (e.g., EHT, the previous experimental patient at UCSF), other patients have miserable performance with these processors;

b. The patients who have little or no recognition with the "compressed analog outputs" processor are likely to exhibit various manifestations of poor nerve survival and severe channel interactions;

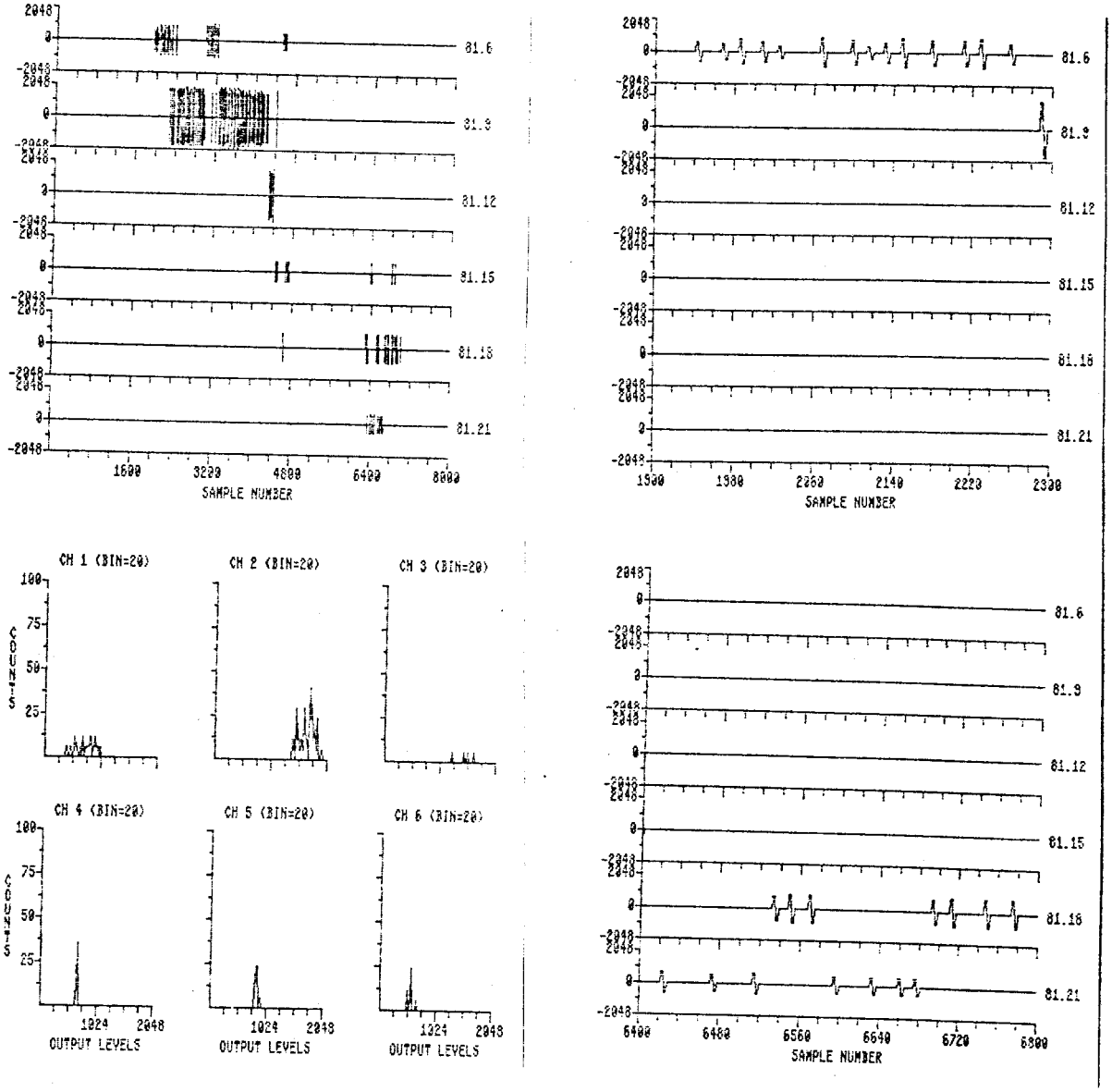


Fig. 8. Displays of outputs for speech-processing strategy 5 used with patient LP. This strategy is the same as the one illustrated in Fig. 7 (strategy 4), except that only one channel is selected for output at each time frame. Strategy 5 has greatly reduced spatial and temporal bandwidths of transmission to the electrode array compared to strategies 4 and 3.

Table 3. Summary of vowel-confusion results for several multichannel processing strategies.

Strategy	%-Correct Identifications
4-channel, interleaved pulses	56 and 70*
4-channel, multipulse excitation	58
5-channel, multipulse excitation (one band left out)	32 and 40**
5-channel, multipulse excitation, 1 channel output at a time (one band left out)	48

*Token "BOOT" was not heard for this condition; if this inaudible token is removed from the confusion matrix (as "not heard"), the percentage of correct identifications increases to 70.

**Token "BEET" was not heard for this condition; if this inaudible token is removed from the confusion matrix (as "not heard"), the percentage of correct identifications increases to 40.

c. In tests with patient LP, reduction of the spatial bandwidth of transmission of compressed analog outputs to the electrode array (by reducing the number of simultaneous output channels from 6 to 2) did not improve performance over the "6-channel, all outputs on" strategy (zero performance for both);

d. An immediate and compelling increase in speech recognition can be obtained (at least in LP and probably in patients with similar patterns of nerve survival) with a 6-channel strategy in which interleaved pulses are delivered to the two channels that have the highest RMS energy in each time frame (strategy 3);

e. Use of the multipulse excitation sequence (strategies 4 and 5) can improve the "naturalness" of speech percepts and apparently also can convey much information over a single channel of stimulation;

f. However, performance on speech tests is in general lower with the "multichannel, multipulse excitation" strategies than with the "multichannel, interleaved pulses" strategy, possibly because the RMS energy levels in the selected channels are not represented in the former;

g. A hybrid approach, in which the magnitudes of pulses in the multipulse excitation sequence are convolved with the magnitudes of RMS energies in the channels, might produce natural-sounding speech with good discrimination of v/uv boundaries and formant frequencies;

h. Many more tests are obviously required to confirm and extend these preliminary (but encouraging) findings;

i. Tools with the power and flexibility of RTI's block-diagram compiler (and associated software and hardware) are essential to quick and productive reactions to the situations presented by individual implant patients; and

j. It is highly likely, in our opinion, that different processing strategies will be required for patients with different classes of neural pathology.

III. Plans for the Next Quarter

The most important activity for the next quarter will be the conduct of tests with the first implant patient at Duke Medical Center. We expect that this patient will be implanted during the last week of September and that our tests with her will begin in mid October. Before testing commences we plan to (1) revise and expand the software for psychophysical and speech tests according to the lessons we have learned from our experience with the last two patients at UCSF; (2) develop further the software for the block-diagram compiler, as outlined in Appendix 1; (3) modify somewhat the design of the hardware interface between the Eclipse computer and implanted electrodes, to reduce further the levels of leakage current and to add "manual" controls for attenuation of channel outputs; (4) complete construction and checkout of the analog-to-digital converter side of the hardware interface, to allow us to measure intracochlear potentials with this patient; and (5) install and verify the correct operation of all remaining components in the cochlear-implant laboratory at Duke.

Finally, in addition to these activities, we plan to present various aspects of our work at major conferences on cochlear implants and related topics. Scheduled presentations for the next quarter are the following:

Wilson, BS and CC Finley: Speech processors for auditory prostheses. Invited paper to be presented in the special session on signal processing for the hearing impaired, IEEE Bioengineering Conf., Sept. 27-30, 1985;

Finley, CC and BS Wilson: Models of neural stimulation for electrically evoked hearing. Invited paper to be presented in the special session on neurostimulation, ACEMB Meeting, Sept. 30-Oct. 2, 1985;

Finley, CC and BS Wilson: A simple finite-difference model of field patterns produced by bipolar electrodes of the UCSF array. Invited paper to be presented at the special session on cochlear implants, IEEE Bioengineering Conf., Sept. 27-30, 1985.

IV. References

Atal, B. S. and Remde, J. R., "A new model of LPC excitation for producing natural-sounding speech at low bit rates," Conf. Rec. 1982 IEEE Int. Conf. Acoust., Speech and Signal Processing, pp. 614-617, 1982.

Appendix 1

Present Status and Functional Description of the Block-Diagram Compiler

PREFACE

Because we have received many requests for current information on the development (and further development) of the block-diagram compiler, we present in this appendix descriptions of the present status and specifications of each major module in the system.

1. Introduction

The block-diagram compiler and associated software are in an advanced state of development. At present, all core programs listed below in Table A.1.1 have been written, debugged, and used to simulate various speech-processing strategies in tests with an implant patient. In addition, all modules listed in the main menus of the DESIGN program (see below) have been written and debugged as stand-alone programs. Our general procedure for incorporating a new module into the block-diagram compiler is to (1) write and debug the module in stand-alone form so that it can be evaluated and refined independently of the complex logic of programs DESIGN and EXECUTE; (2) incorporate dialog, displays and logic within DESIGN to allow the investigator to specify the function with appropriate feedback on performance and design parameters; and (3) incorporate the algorithm or "engine" to execute the function within EXECUTE with code that is consistent with the logic of EXECUTE and that is optimized for speed of computation. This procedure has been fully implemented for about half of the functions listed in the main menus of the DESIGN program. The functions now fully implemented include all those necessary for complete simulation of the basic UCSF speech processor, variants of this processor, and several multichannel strategies that use pulsatile stimulation.

2. DSP Functions

The major categories of modules in the block-diagram compiler are indicated in the main menus of the DESIGN program. These menus are presented in Fig. A.1.1. The first major category is for "DSP" (digital signal processing) modules and includes the functions of filtering, Fourier analysis, cepstral analysis and data windowing. As described in some detail on pages 23-28 in our 4th quarterly report, specification and subsequent execution of classic IIR ("infinite duration impulse response") filters are fully incorporated into the block-diagram compiler. The present set of options for filter design include the specification of (1) lowpass, highpass or bandpass response; (2) the class of filter response, where the choices include Butterworth, Chebychev and elliptic functions; (3) the break frequency or frequencies; and (4) a direct or indirect input of filter order. Because filter calculations are all done with floating-point

Table A.1.1. Programs Used in the Block-Diagram Simulator

CPEXEC -- executive program for managing communications between and execution of other programs in the set;

DESIGN -- program for the design of a signal-processing system, in which the user specifies the function and position of each block within a network of blocks;

MODIFY -- program to modify signal-processing systems previously defined by program DESIGN;

PREPARE -- program that transforms the files generated by program DESIGN into files that are used by program EXECUTE;

EXECUTE -- program that executes the simulation of a signal-processing system;

SHOWNTELL -- program for display of outputs generated by EXECUTE, either as graphs on the computer console or as acoustic signals produced over the D/A converter;

SAMPLE -- program to sample speech and other data with the A/D converter, and to store these data on disk in contiguous files with identifying headers;

ASNELEC -- program to assign electrode channels to receive data from the outputs of EXECUTE, and to transform these data into the format required for control of and communication with the hardware interface interface between the computer and implanted electrodes;

TEST -- program to send data out to the electrodes from the files generated by program ASNELEC, and to monitor and log patient responses to processed speech stimuli.

Screen 1

ENTER ONE OF THE FOLLOWING OPTIONS FOR THE FUNCTION OF BLOCK N:

MODULE CATEGORY	OPTION		FUNCTION
DSP:	1	=	FILTER
	2	=	FFT ANALYZER
	3	=	CEPSTRUM ANALYZER
	4	=	WINDOW
SPEECH ANALYSIS:	5	=	LPC ANALYZER
	6	=	FORMANT TRACKER
	7	=	PITCH EXTRACTOR
SIGNAL SOURCE:	8	=	NOISE GENERATOR
	9	=	SIN/COS GENERATOR
	10	=	PULSE-TRAIN GENERATOR
	11	=	DISK FILE
MATH OPERATIONS:	12	=	SUMMER
	13	=	MULTIPLIER/INVERTER
	14	=	DIVIDER
	15	=	LOGARITHMIC CALCULATOR
	16	=	INTEGRATOR
OTHER:	17	=	SHOW REMAINING OPTIONS

ENTER OPTION >

Screen 2

ENTER ONE OF THE FOLLOWING OPTIONS FOR THE FUNCTION OF BLOCK N:

MODULE CATEGORY	OPTION		FUNCTION
CIRCUIT FCNS:	18	=	COMPRESSOR
	19	=	ZERO-CROSSING COUNTER
	20	=	PEAK DETECTOR
	21	=	WINDOW COMPARATOR
	22	=	LEVEL COMPARATOR
	23	=	ONE SHOT (MONOSTABLE MULTIVIBRATOR)
	24	=	FLIP-FLOP
	25	=	SWITCH
	26	=	RECTIFIER
	27	=	UNIT DELAY OPERATOR
OTHER:	28	=	READ SUBSYSTEM FOR PRESENT BLOCK FROM ANOTHER DESIGN
	29	=	SELECT A USER-DEFINED RULE
	30	=	IDENTIFY A USER-DEFINED RULE
	31	=	SHOW TOPOLOGY OF PRESENT SYSTEM
	32	=	RETURN TO PREVIOUS SCREEN
	33	=	REVISE A BLOCK
	34	=	EXIT FROM DESIGN PROGRAM

ENTER OPTION >

Fig. A.1.1. Two main menus of the DESIGN program.

numbers, user-specified frequencies for passband edges and center frequencies can be precisely approximated so long as the Nyquist criterion is satisfied (i.e., these frequencies must be less than one-half of the sampling frequency). Limits on filter order depend on the class of filter response and the proximity of bandpass edges to the Nyquist frequency. In general, however, these limits are arbitrarily high for the present application and always exceed an order of 8 for filters that have passband edges at least 10% below the Nyquist frequency.

Plans for further development of the filter functions for the block-diagram compiler include incorporation of IIR filters with user-specified locations of poles and zeros in the s and z planes (transformation from s to z planes is accomplished using the bilinear transform) and incorporation of various FIR (finite impulse response) filters, primarily to achieve maximally-flat group delays within the filter passbands. The code for the remaining IIR filter will be drawn from an existing stand-alone program and the code for the FIR filters will be drawn from programs published by McClellan et al. (1979), Rabiner et al. (1979) and Kaiser (1979). These latter programs are part of the IEEE library of programs for digital signal processing and have been used in our laboratory for many years as stand-alone programs.

The next function in the DSP category is that of Fourier analysis. The "engine" for Fourier analysis, called by many programs in our computing facility, is a fast-Fourier transform (FFT) algorithm that is optimized for speed of execution in the assembly language of our Eclipse computers. This subroutine has been used for many and varied tasks over the last 10 years including spectral analyses of bat sonar pulses, speech signals and cochlear-microphonic waveforms. Examples of such analyses are presented on the following pages in Figs. A.1.2-A.1.4. Options for FFT analysis within the construct of the block-diagram compiler are listed in Table A.1.2. These options more than encompass the range normally used for speech analysis. For example, speech spectrograms usually have frequency resolutions that approximate 300 Hz and 45 Hz; these resolutions correspond to options 1 and 4 in Table A.1.2 for the assumed sampling frequency of 10 kHz.

Next in the series of functions listed in the DSP category is the "cepstrum analyzer." This function implements a "homomorphic deconvolution" of the speech waveform in which parameters describing the excitation of the

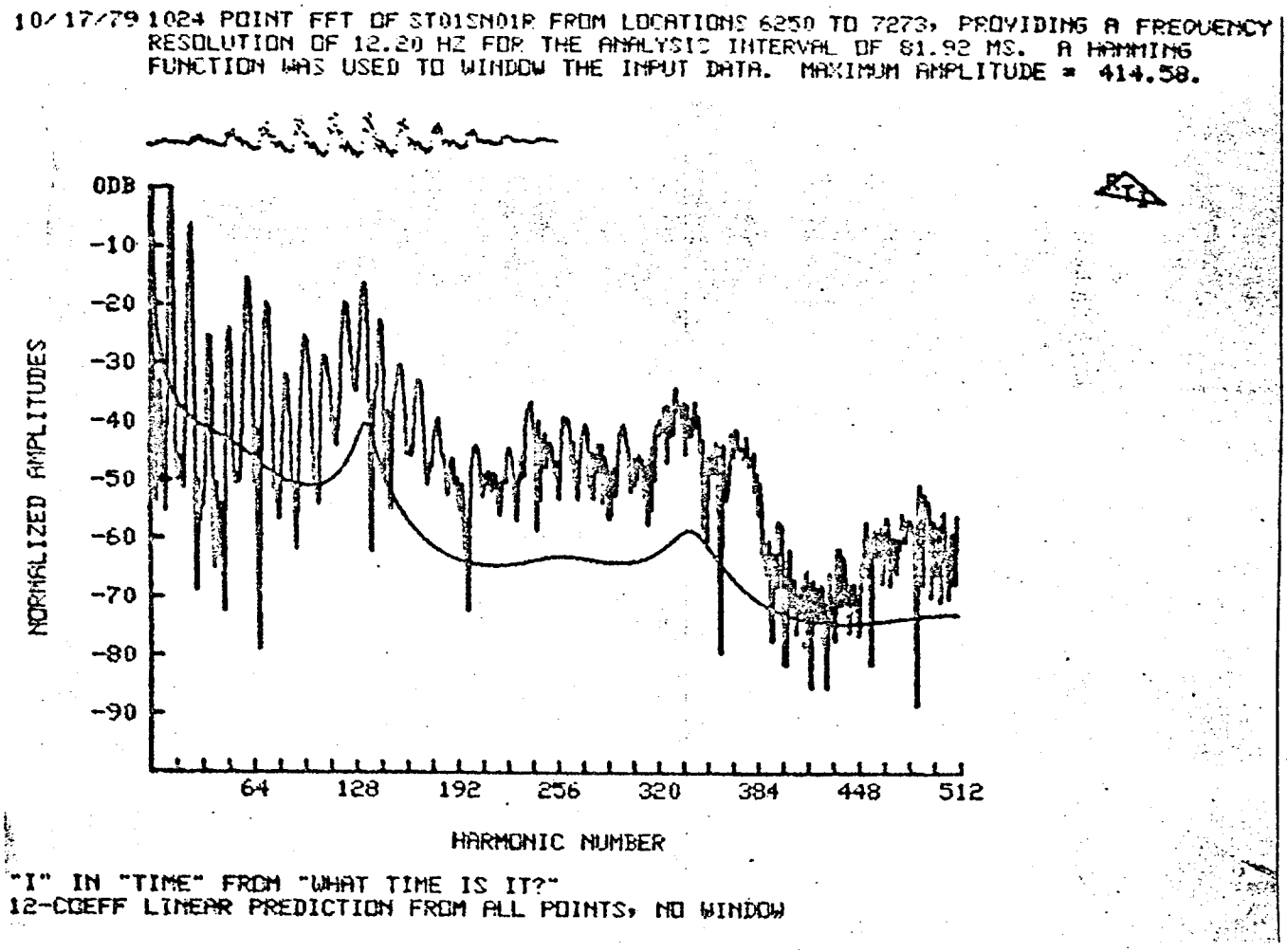


Fig. A.1.2. 1024-point FFT of the first "i" in the sentence "What time is it?". The periodicity in the spectrum reflects the harmonic structure of glottal excitation for voiced speech sounds, and the broad peaks reflect the formants of this vowel. The smooth curve showing the formant structure without the "clutter" of the FFT was obtained with a 12-coefficient linear-prediction model of the data.

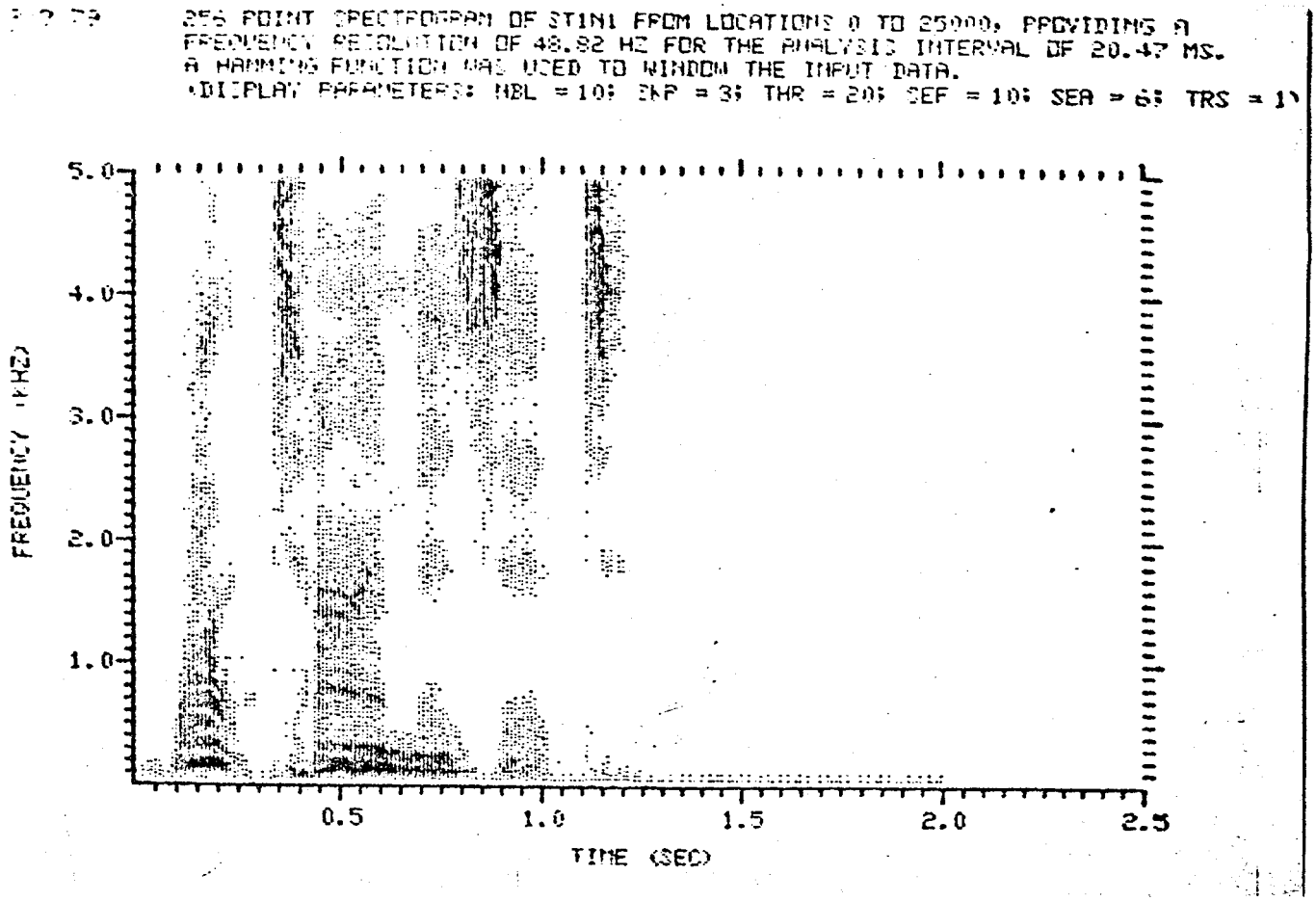
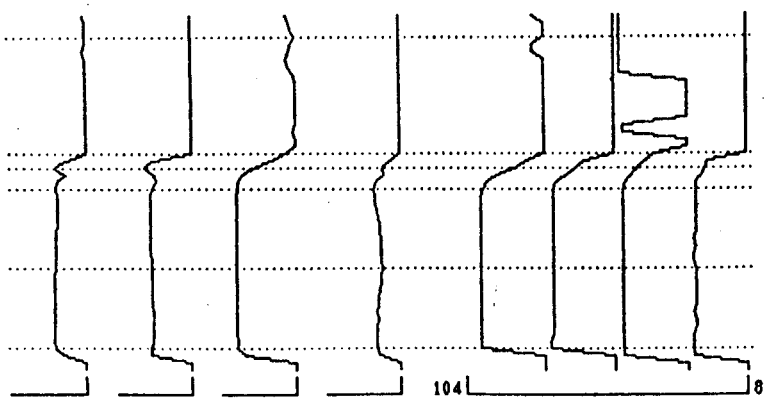
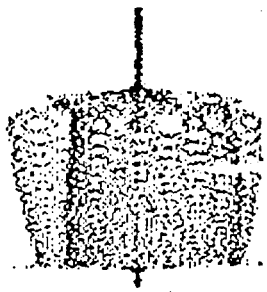
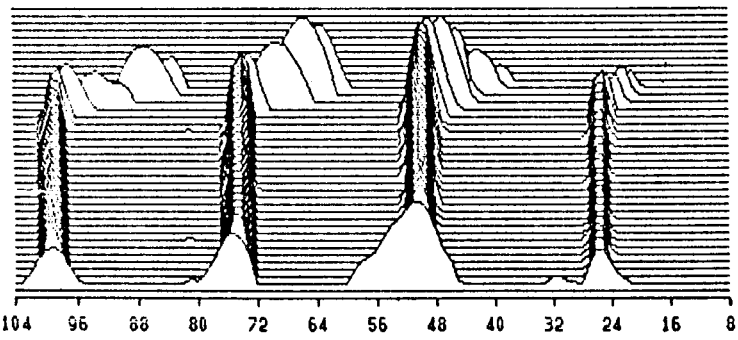


Fig. A.1.3. Spectrogram of the sentence "What time is it?". The high resolution in frequency afforded by selection of a 256 point FFT shows clearly the harmonic structure of voiced speech sounds. However, rapid changes in the spectrum due to individual glottal puffs are not resolved. Such a temporal structure is reflected in 64 and 128 point FFTs for this sentence. Display parameters for the generation of an artificial gray scale are those first described by Strong and Palmer (1975).

0.00	0.00	6.72	0.00	95.00	65.00	47.50	8.75
20.65	32.16	28.28	11.53	107.50	81.25	55.00	26.25
19.98	27.08	39.26	17.97	121.25	91.25	61.25	30.00
20.45	27.40	40.29	12.41	122.50	91.25	61.25	30.00
20.40	26.62	39.47	16.15	122.50	92.50	61.25	31.25







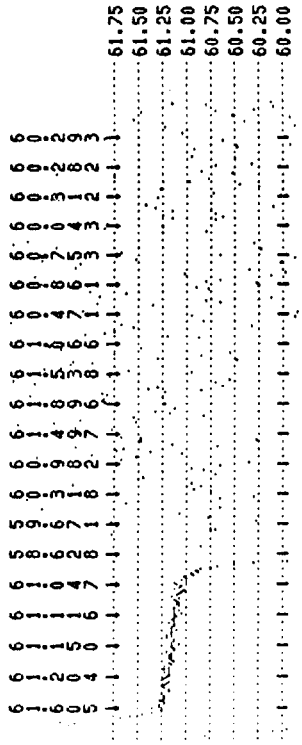


Fig. A.1.4. Frequency analysis of the sonar pulse of the mustache bat, Pteronotus p. parnellii. The upper trace shows the time waveform, and the data presented immediately below it are sequential FFTs plotted in a perspective display. Hidden lines are removed. The information presented at the right of the perspective display shows the amplitudes (top 4 traces) and frequencies (bottom 4 traces) of the major components in the waveform. Significant events in these traces are marked by dotted lines, and values at these events are shown on the right-hand side of the figure. Finally, the bottom plot shows the output of a software phase-locked loop, in which the Doppler shift of the bat's cry and echo can be seen (from Kobler et al., in press).

Table A.1.2. Options for FFT analysis. Entries for frequency resolution and analysis interval are based on a 10 kHz sampling frequency.

Option	Points	Resolution (Hz)	Analysis Interval (msec)
1	32	312.5	3.2
2	64	156.3	6.4
3	128	78.1	12.8
4	256	39.1	25.6
5	512	19.5	51.2
6	1024	9.8	102.4

vocal tract are identified and separated from parameters describing the short-time configuration of the vocal tract (see Rabiner and Schafer, 1978, pp. 355-391, for a complete discussion). Cepstrum analysis is a particularly powerful (but slow) technique for accurate determination of the fundamental frequency of voiced-speech sounds (see, e.g., Noll, 1967) and is therefore also included among the options in the block-diagram compiler for voice pitch extraction (option 7). Processing steps in the computation of short-time cepstra for each time frame of input speech include (1) windowing each time frame of speech to attenuate end effects; (2) calculating the FFT of the windowed segment; (3) computing the log magnitude of the FFT components; and (4) calculating the inverse FFT of the log-magnitude function. The "workhorse" of analysis is again the FFT algorithm. At present we have a debugged subroutine to manage these tasks for calculation of both the complex and real (from minimum-phase reconstruction) cepstrum on input frames of sampled data. In addition, we have two stand-alone main programs to deliver the windowed segments of data to the cepstrum subroutine and to display the results of the analysis. Code from these programs will be incorporated into the DESIGN and EXECUTE programs of the block-diagram compiler, as previously described.

The final module in the DSP category is that of data windowing. We have two subroutines for this module; the first to calculate the window function and the second to multiply (repetitively) this function and sequential frames of sampled data. The window function itself is computed only once to save processing time and multiplications are carried out only for segments of the window that do not have a value of 1.0 (e.g., multiplications are not required for the long middle segment of an extended-cosine-bell window). Options for data windowing include "boxcar" (rectangular window), Hanning, Hamming, and extended-cosine-bell functions. The subroutines for data windowing are already integral parts of our existing programs for FFT analysis, cepstral analysis, and linear-prediction analysis. These subroutines will be explicitly incorporated in the block-diagram compiler so that other analysis programs within the compiler can have access to windowed data.

3. Speech Analysis Functions

The functions in the category of speech analysis include linear-prediction analysis, formant extraction and "pitch" (fundamental frequency and voice/unvoice) extraction. For linear-prediction (LP) analysis, both the autocorrelation and covariance methods (Markel and Gray, 1976) are implemented in stand-alone programs that have been used for many years in our laboratory. An example of LP analysis is presented in Fig. A.2.2, which shows a 1024-point FFT and 12-coefficient LP trace for the first "i" in "What time is it?". The smooth curve of the LP trace clearly identifies the formant structure of the vowel. One of the formant trackers (of function 6, Fig. A.1.1) exploits this property by operating on the output of the LP analyzer. This tracker uses the subroutines FINDPK and FORMNT presented in chapter 7 of Markel and Gray (1976).

Another option for formant tracking is a simulation of the procedure we use in the Autocuer (a speech-analyzing lipreading aid for profoundly-deaf users), which involves interpolation of the peak outputs of a 16-channel filter bank every 10 msec. Post-processing logic is used to smooth the resulting formant tracks and to reject spurious "formant" peaks (in conjunction with an algorithm for end-point detection). The accuracy of formant-frequency detection approximates plus or minus 25 Hz over the band that encompasses F1 and F2. The inclusion of this option for formant tracking in the block-diagram compiler is particularly appropriate for the present application because (1) formant trackers that operate directly on a filter-bank representation of the short-time spectra of speech (the representation obtained either with discrete filters or an FFT) generally have better performance in noisy environments than trackers that operate on representations derived from LP analysis of speech and (2) the Autocuer formant extraction scheme has already been implemented in a portable, real-time speech processor with documented extraction performance.

The options for pitch extraction (i.e., determination of the fundamental frequency of voiced-speech sounds and determination of whether the present speech sound is voiced or unvoiced) include the Gold-Rabiner algorithm (1969), the Average-Magnitude-Difference function (Ross et al., 1974; Sung and Un, 1980; Un and Yang, 1977), the Tucker and Bates algorithm (Tucker and Bates, 1977), the "simplified inverse filter tracking" or "SIFT" algorithm (Markel, 1972; Markel and Gray, 1976, pp. 206-211), and, as

mentioned before, the cepstral method (Noll, 1967). The code used for the Average-Magnitude-Difference function is a simulation in FORTRAN of the machine-language code used in the portable, real-time processor we described on pages 8-11 of our sixth quarterly progress report. With this FORTRAN simulation, we can fully evaluate the potential performance of the portable processor in tests with implant patients using the block-diagram compiler.

In addition to the explicit methods of pitch extraction just described, we have available the residual signal from our programs for LP analysis of speech. Use of the LP residual signal, and the "multipulse excitation" transform of it (Atal and Remde, 1982), was described in section II of this report. In general, the residual and multipulse signals are excellent representations of the excitation of the vocal tract for all speech sounds (including those with mixed voiced and unvoiced components). The multipulse signal in particular may have direct utility in speech processors for auditory prostheses.

With the exception of the FORTRAN simulation of the Average-Magnitude-Difference function, all modules described in this section on speech analysis exist as debugged, stand-alone programs. Most of these programs have been used for many years in our laboratory and are highly reliable. In addition, the code for extraction and manipulation of the LP residual signal has been fully incorporated within the block-diagram compiler. As previously described, this module was used in the simulation of two speech processing strategies for tests with the most recent implant patient at UCSF. Finally, we want to mention that software links have been incorporated in the block-diagram compiler to access the speech-analysis programs of the "Interactive Laboratory System" (ILS) package marketed by Signal Technology, Inc. These programs generally duplicate a subset of our speech analysis modules, including the modules for LP analysis, cepstral analysis, pitch extraction using the SIFT algorithm, and pitch extraction using the cepstral method. The duplication in functions is useful, of course, as a cross-check on program performance and accuracy; however, we rarely use the links to the ILS software because (1) the ILS code is generally slower (sometimes an order of magnitude slower) than our code; (2) we are much more familiar with our code and therefore consider it to be more trustworthy and adaptable in our hands; and (3) we are only licensed to use the ILS code on the UCSF computer.

4. Signal-Source Functions

The signal-source functions include a noise generator, a sine/cosine generator, a pulse-train generator, and input from a disk file. The last two are fully incorporated into the block-diagram compiler and we describe applications of these modules in section II of this report. Several noise generators exist as stand-alone programs and will be incorporated into the block-diagram compiler in the near future. Subroutines for generating uniform, pseudo-random sequences (from 0.0 to 1.0) include the RANDOM and DRANDOM functions of the Data General FORTRAN V subroutine library and the program UNI, written by Alan Gross for the IEEE library of programs for digital signal processing. The outputs of these programs are statistically acceptable for all anticipated needs.

5. Math Operations

All math operations, with the exception of the integrator (and associated RMS-to-dc converter), are functioning components of the block-diagram compiler. The implemented functions include a (1) summer; (2) multiplier/inverter; (3) divider; and (4) logarithmic calculator. The summer and multiplier/inverter can accept many inputs and the divider can accept two inputs. These inputs can be any combination of floating-point or integer signal vectors or of floating-point or integer constants. All math operations are optimized for speed of execution utilizing the floating-point processors of our Eclipse computers. Accuracy of the computations depends on the type of representation used for the numbers; the word length for integers in the Eclipse is 16 bits and that for floating-point numbers is 32 bits (1 sign bit, 7 exponent bits, 24 mantissa bits, approximately 7.2 decimal digits of significance).

6. Circuit Functions

In addition to the modules for digital signal processing and speech analysis, various circuit functions likely to find application in speech processors for auditory prostheses have been, or soon will be, incorporated into the block-diagram compiler. Because the dynamic range of stimulus intensities from threshold to uncomfortable loudness is much lower in

implant patients than in normal listeners, the compressor circuit function has obvious importance for the present application. Some sort of compression is used in all extant speech processors for auditory prostheses. The compressor module in the block-diagram compiler is fully implemented and provides for unprecedented flexibility in the selection of design parameters. The main options for compressor design are presented in the first menu of the compressor module, which is reproduced here in Fig. A.1.5. The major types of compressors included among these options are single- and multiple-segment "compressor-compressors" (where each segment of the steady-state compression function forms a straight line on a log-log plot), an automatic volume control (noninstantaneous, infinite compression), a logarithmic compressor, and a hard limiter. The "compressor-compressors" can be instantaneous or noninstantaneous.

```
SELECT CLASS OF COMPRESSOR FROM THE FOLLOWING OPTIONS:

       1 = CLASSIC 'COMPRESSOR-COMPRESSOR', NONINSTANTANEOUS
       2 = CLASSIC 'COMPRESSOR-COMPRESSOR', INSTANTANEOUS
       ---------------------------------------------------
       3 = PIECEWISE 'COMPRESSOR-COMPRESSOR', NONINSTANTANEOUS
       4 = PIECEWISE 'COMPRESSOR-COMPRESSOR', INSTANTANEOUS
       ---------------------------------------------------
       5 = CLASSIC AVC (NONINSTANTANEOUS, INFINITE COMPRESSION)
       ---------------------------------------------------
       6 = LOG COMPRESSOR, INSTANTANEOUS
       ---------------------------------------------------
       7 = HARD LIMITER

ENTER )
```

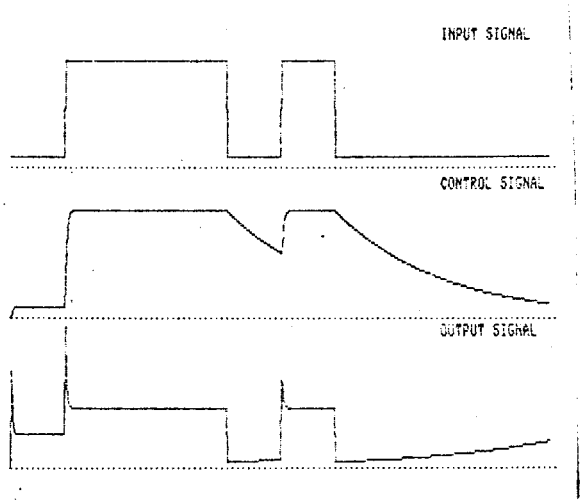
Fig. A.1.5. First menu of options for the compressor module.

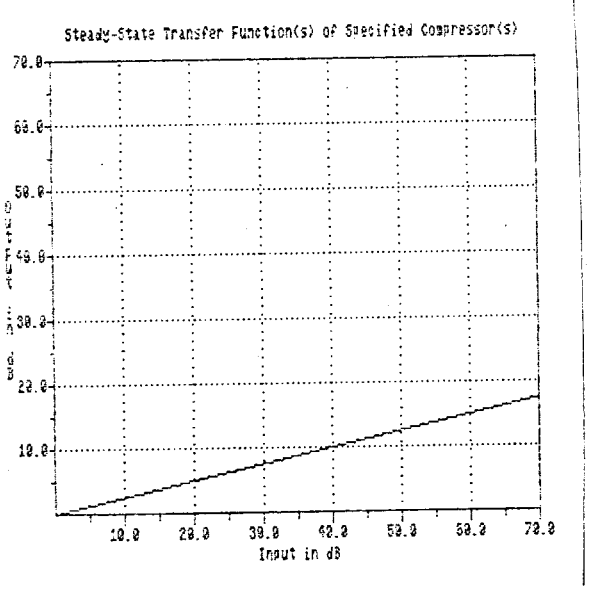
To assist the investigator in the design of compressors, several options are presented at the end of each specification pass for display of compressor outputs and characteristics. The menu presenting these options is reproduced here as Fig. A.1.6, and example displays are presented in Figs. A.1.7 and A.1.8. In general, the ranges of compression ratios, attack times and release times exceed those practically realizable in analog circuits. Also, as noted before, the range of compressor types greatly exceeds the range that normally would be entertained for implementation in analog circuits. The options for piecewise compression, in particular, provide a powerful means for precise mapping of inputs onto the psychophysical space of implant patients.

```
SELECT ONE OF THE FOLLOWING OPTIONS FOR DISPLAY OF COMPRESSOR CHARACTERISTICS:

      1 = SHOW DYNAMIC RESPONSE
      2 = SHOW STEADY-STATE TRANSFER FUNCTION
      3 = DESIGN NEW COMPRESSOR
      4 = RETURN TO BLOCK-DIAGRAM DESIGN PROGRAM

ENTER >
```

Fig. A.1.6. Menu of options for the display of compressor characteristics

Other circuit-function modules now fully incorporated into the block-diagram compiler are the rectifier and unit-delay operator. The rectifier can be full- or half-wave and can operate on integer or floating-point numbers. The unit-delay operator is used in the specification of closed feedback loops with the block-diagram compiler for networks that do not have built-in delays (i.e., for networks that do not have embedded filters or integrators, etc.). The remaining circuit functions are relatively simple to implement and will be incorporated into the block-diagram compiler in the near future. Among these remaining functions, the zero-crossing detector has the highest priority for inclusion because it will be required for a full simulation of the present speech-processing strategy used in the Australian device.





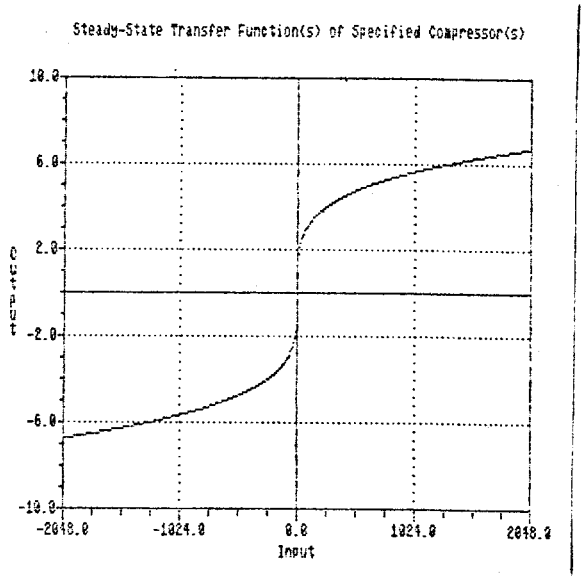


Fig. A.1.7. Dynamic and steady-state characteristics of a single-segment "compressor-compressor." The parameters of this compressor are: Attack time = .5 msec; Release time = 5 msec; Compression ratio = 4; Threshold for onset of compression = 0.0; Amplification of compression stage = 1.0. The upper panel shows the dynamic responses of the specified compressor to the rectangular forcing function in the top trace. The middle trace is the derived control signal for adjusting the instantaneous gain of the compression stage and the bottom trace is the output signal (x 100). Intervals of attack, release and compression are clearly evident in the output signal. Finally, the bottom two panels show the steady-state response of the compressor on log-log and lin-lin plots. The "compressor-compressor" function forms a straight line on a log-log plot whose slope reflects the selected compression ratio.

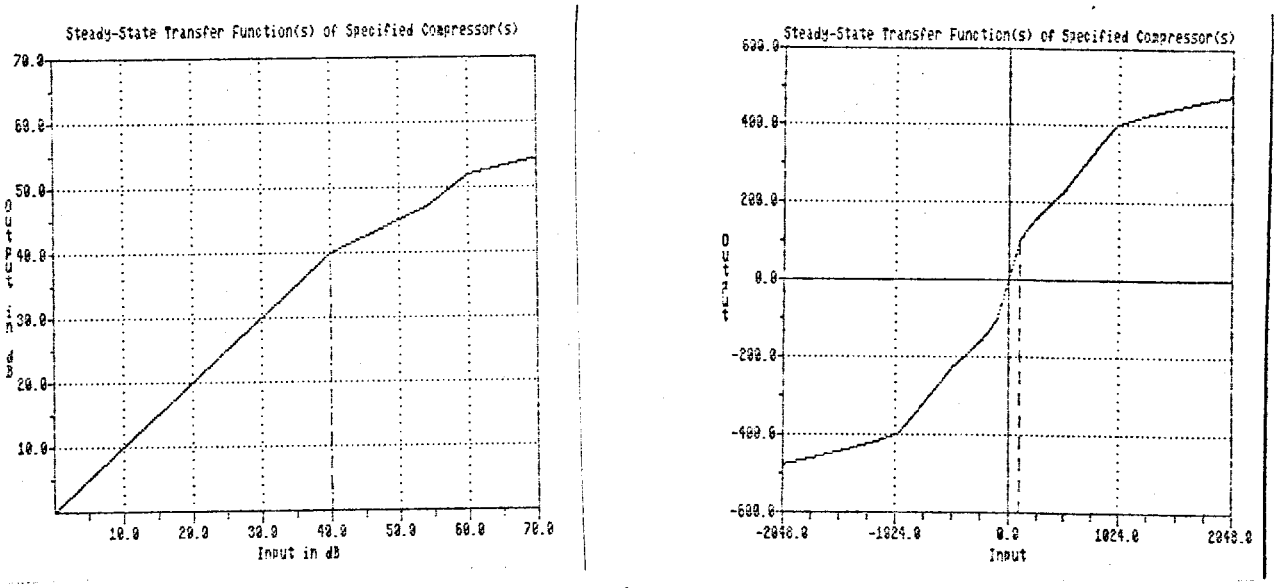


Fig. A.1.8. Steady-state transfer function of a piecewise "compressor-compressor" shown on log-log and lin-lin plots. This compressor has a threshold for the onset of compression at 100 (marked by the vertical dashed lines) and three segments of compression thereafter. The compression ratio of segment 1, for input values from 100 to 500, is 2.0; the ratio of segment 2, for inputs from 500 to 1000, is 1.2; and the ratio of segment 3, for inputs above 1000, is 4.0. The options for piecewise compression provide a powerful means for precise mapping of inputs onto the psychophysical space of implant patients.

7. Other Functions

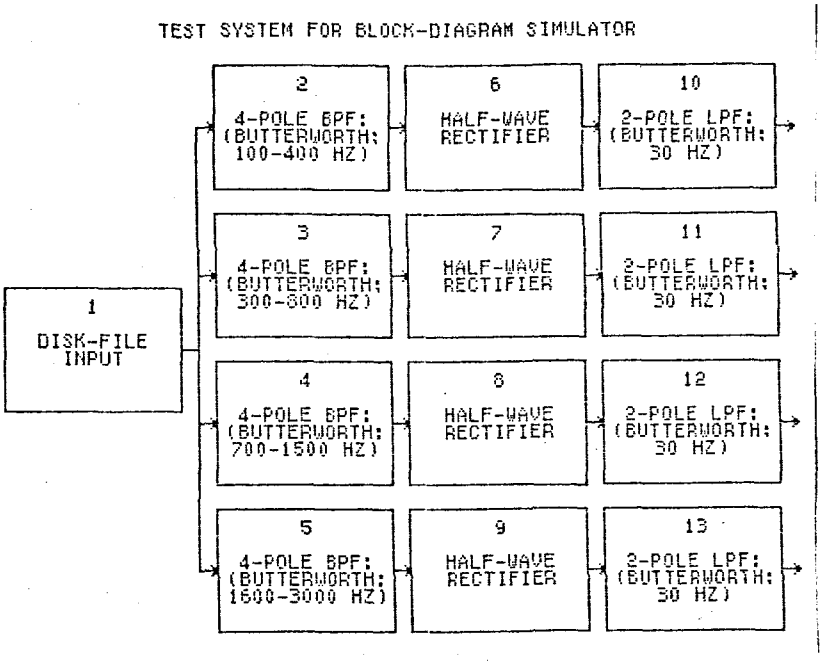
Various control functions are listed under the "OTHER" category of modules in the second main menu of the DESIGN program (see Fig. A.1.1), and include facilities to (1) read a subsystem for the present block from another design stored on the disk; (2) select a user-defined rule, the code for which is also stored on the disk; (3) identify a new user-defined rule to be added to the library; (4) show the topology of the present system as defined by the investigator up to this point; (5) return to the previous screen, the first main menu of the DESIGN program; (6) revise a previously specified block; and (7) exit from the DESIGN program. All of these control functions with the exception of 2 and 3 are now incorporated in the block-diagram compiler.

As indicated in section II of this report, several new functions for the control and selection of processor outputs have been added to our basic list of processor modules; these are called by the first option listed above. Finally, an example of the output produced in response to a request to show the topology of a network is presented in Fig. A.1.9.

8. Concluding Remarks

At present, the block-diagram compiler is fully capable of simulating most speech-processing strategies proposed or in use to date. Incorporation of the remaining stand-alone modules, as described in the Introduction to this Appendix, will allow us to simulate every extant speech processor for auditory prostheses (as described in the open literature) and many new processors that may provide improved performance. In our experience with the modules now fully incorporated, we have learned that composition and debugging of the stand-alone modules take much more time than the two following steps to incorporate the modules into programs DESIGN and EXECUTE. These last two steps generally require between 1 and 3 days of programming effort to accomplish. Therefore, most of the work for the block-diagram compiler is behind us and, as demonstrated with patient LP (section II, this report), the present software already constitutes an extremely-powerful tool for designing and evaluating speech processors for auditory prostheses.

As a final note in this appendix on the block-diagram compiler, we would like to mention that this software is only one component in a much-



TOPOLOGY OF DESIGN 1:

BLOCK	INPUT(S) BLK/OUT/IN	DISK OUTPUT(S)	SFREQ OF OUTPUT(S)	DATA TYPE OF OUTPUT(S)	FUNCTION
1	TUNA		20000.0	INT	DISK INPUT
2	1/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
3	1/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
4	1/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
5	1/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
6	2/ 1/ 1		20000.0	INT	RECTIFIER
7	3/ 1/ 1		20000.0	FLT	RECTIFIER
8	4/ 1/ 1		20000.0	FLT	RECTIFIER
9	5/ 1/ 1		20000.0	FLT	RECTIFIER
10	6/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
11	7/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
12	8/ 1/ 1	YES/ 1	20000.0	FLT	FILTER
13	9/ 1/ 1	YES/ 1	20000.0	FLT	FILTER

THE FUNCTION OF BLOCK 5 IS DESCRIBED BELOW:

BANDPASS BUTTERWORTH FILTER WITH THE FOLLOWING PARAMETERS:

FILTER ORDER = 4;
SAMPLING FREQUENCY = 20000.0;
BREAK FREQUENCIES = 1600.0 AND 3000.0.

Fig. A.1.9. Response to a request to show the topology of a network specified with the DESIGN program. The top panel is a block diagram of the user-specified system and the middle panel is the topology printout describing this network. The topology information fully specifies for each block its inputs and outputs; whether and how many of its outputs are to be written to the disk; the sampling frequency of its output(s); the data type of its outputs; and its major category of function. Specific information on the design parameters for any single block also can be requested, as illustrated in the bottom panel for block 5.

larger set of programs we have developed for design and evaluation of speech processors for auditory prostheses. These other programs include ones to (1) conduct the "miniMAC" and "MAC" tests with processed speech tokens in a fully-automated and statistically-rigorous fashion; (2) conduct many basic psychophysical tests (e.g., pulse thresholds, noise thresholds, tone thresholds, loudness measures, loudness comparisons, gap detection, loudness and frequency DL's, and more), also in a fully-automated and statistically-rigorous fashion; (3) model the electric field patterns produced in ears implanted with electrodes of various geometries; (4) model the neural responses evoked by such electric fields for various patterns of neural survival and stimulus waveforms; and (5) generate complex stimuli for presentation and evaluation of "stimulus primitives." Our plans for near future include further refinement and development of the block-diagram compiler and the other software just listed. While much work remains, we are pleased to note that most of the work has been completed and that all the "core" programs are in place and fully debugged. The software for the block-diagram compiler, for example, now consists of 161 separate programs and more than 15,000 lines of FORTRAN and machine-language code. We expect that completion of this software (to incorporate the remaining modules into DESIGN and EXECUTE) will require several thousand more lines of code and between two and three person-months of programming effort.

9. References

Atal, B. S. and Remde, J. R., "A new model of LPC excitation for producing natural-sounding speech at low bit rates," Conf. Rec. 1982 IEEE Int. Conf. Acoust., Speech and Signal Processing, pp. 614-617, 1982.

Gold, B. and Rabiner, L. R., "Parallel processing techniques for estimating pitch period of speech in the time domain," J. Acoust. Soc. Am., 46 (1969) 442-449.

Kaiser, J. F., "Design subroutine (MXFLAT) for symmetric FIR low pass digital filters with maximally-flat pass and stop bands." In C. J. Weinstein et al. (Eds.), Programs for Digital Signal Processing, IEEE Press, New York, 1979.

Kobler, J. B., Wilson, B. S. and Henson, O. W., Jr., "Echo intensity compensation by echolocating bats," Hear. Res. (accepted for publication).

Markel, J. D., "The SIFT algorithm for fundamental frequency estimation," IEEE Trans. Audio and Electroacoust., AU-20 (1972) 367-377.

Markel, J. D. and Gray, A. H., Jr., Linear Prediction of Speech, Springer-Verlag, Berlin, Heidelberg and New York, 1976.

McClellan, J. H., Parks, T. W. and Rabiner, L. R., "FIR linear phase filter design program." In C. J. Weinstein et al. (Eds.), Programs for Digital Signal Processing, IEEE Press, New York, 1979.

Noll, A. M., "Cepstrum pitch determination," J. Acoust. Soc. Am., 41 (1967) 293-309.

Rabiner, L. R., McGonegal, C. A. and Paul, D., "FIR windowed filter design program--WINDOW." In C. J. Weinstein et al. (Eds.), Programs for Digital Signal Processing, IEEE Press, New York, 1979.

Rabiner, L. R. and Schafer, R. W., Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, N. J., 1978.

Ross, M. J., Shaffer, H. L., Cohen, A., Frendberg, R. and Manley, H. J., "Average magnitude difference function pitch extractor," IEEE Trans. Acoust., Speech and Signal Processing, ASSP-22 (1974) 353-362.

Strong, W. J. and Palmer, E. P., "Computer-based sound spectrograph system," J. Acoust. Soc. Am., 58 (1975) 899-904.

Sung, W. Y. and Un, C. K., "A high-speech pitch extractor based on peak detection and AMDF," J. Korea Inst. Electr. Eng., 17 (1980) 38-44.

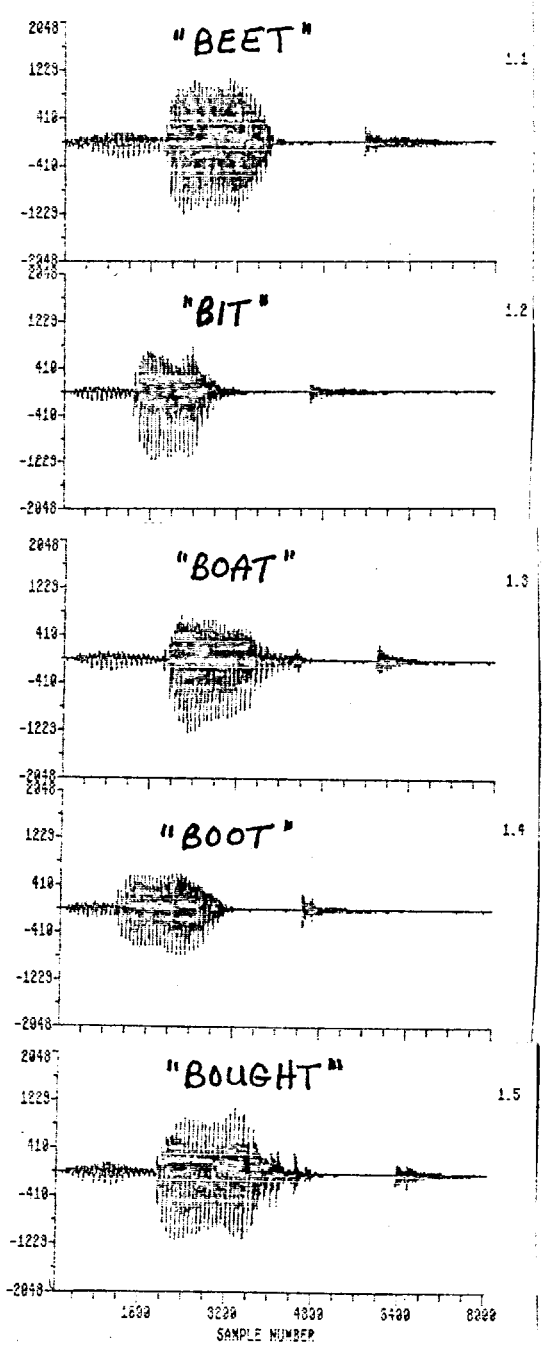
Tucker, W. H. and Bates, R. H. T., "Efficient pitch estimation for speech and music," Electronics Lett., 13 (1977) 357-358.

Un, C. K. and Yang, S.-C., "A pitch extraction algorithm based on LPC inverse filtering and AMDF," IEEE Trans. Acoust., Speech and Signal Processing, ASSP-25 (1977) 565-572.

Appendix 2

Vowel Confusion Tokens and Processor Outputs,
Strategy 4, 7/85

Vowel Tokens



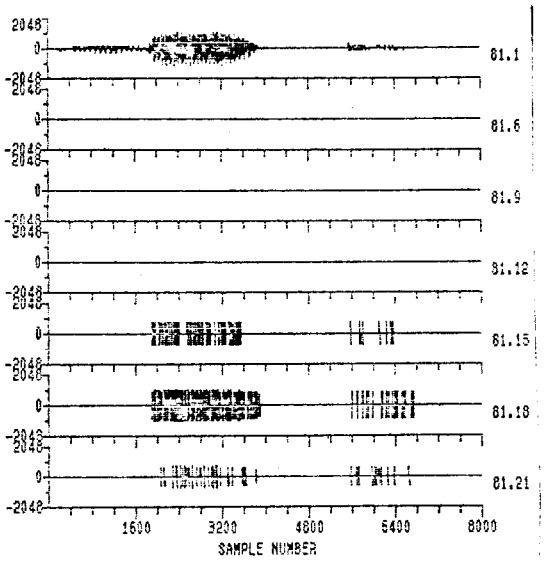
Key to Appended Plots

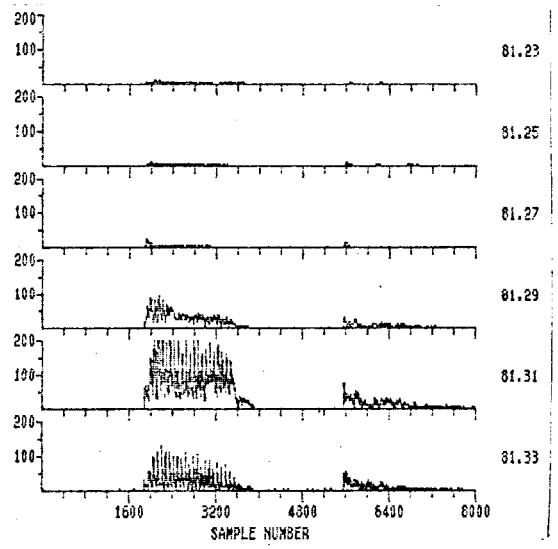
[Label numbers are in the form 81.x where x is the processor block number shown in Fig. A.3.1 on page 25 above. Each trace represents the output of the corresponding block.]

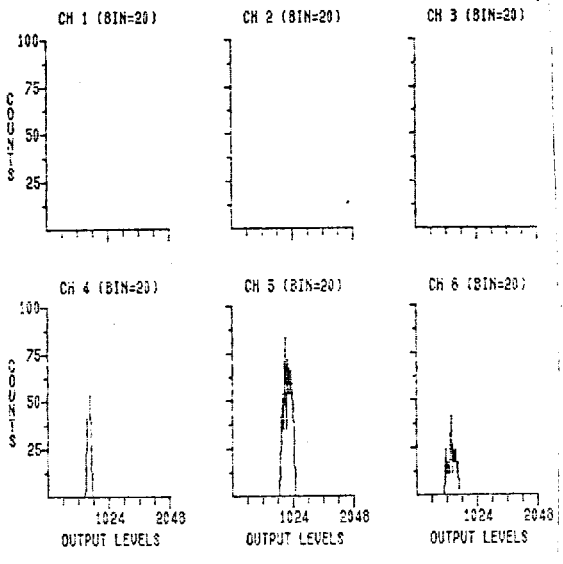
81.1	input speech signal
81.6	434- 712 Hz band output signal
81.9	712-1047 Hz band output signal
81.12	1047-1538 Hz band output signal
81.15	1538-2259 Hz band output signal
81.18	2259-3319 Hz band output signal
81.21	3319-4875 Hz band output signal
81.23	434- 712 Hz band RMS energy
81.25	712-1047 Hz band RMS energy
81.27	1047-1538 Hz band RMS energy
81.29	1538-2259 Hz band RMS energy
81.31	2259-3319 Hz band RMS energy
81.33	3319-4875 Hz band RMS energy

Samples-per-output-level histograms for each channel.

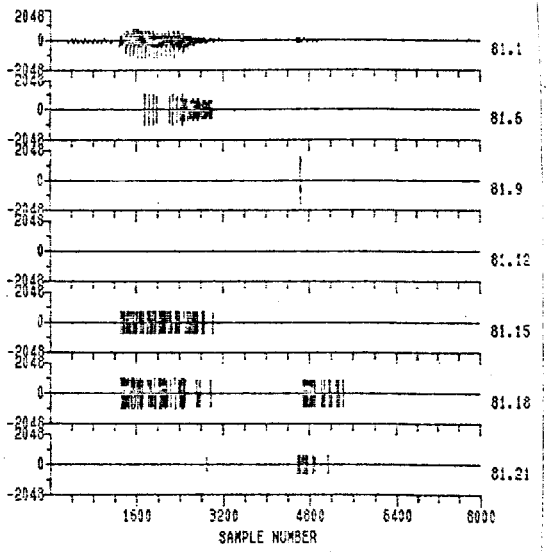
BEET

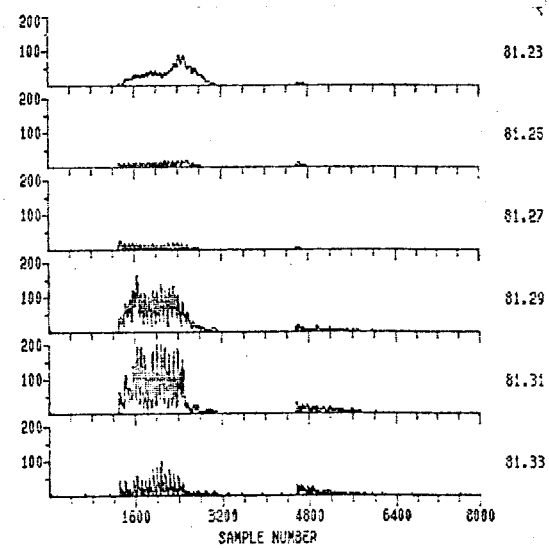

81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

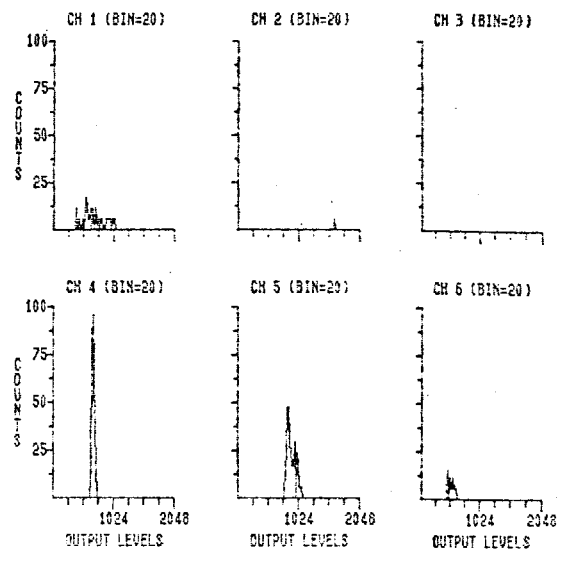

81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
100
75
50
25
1024
2048
OUTPUT LEVELS

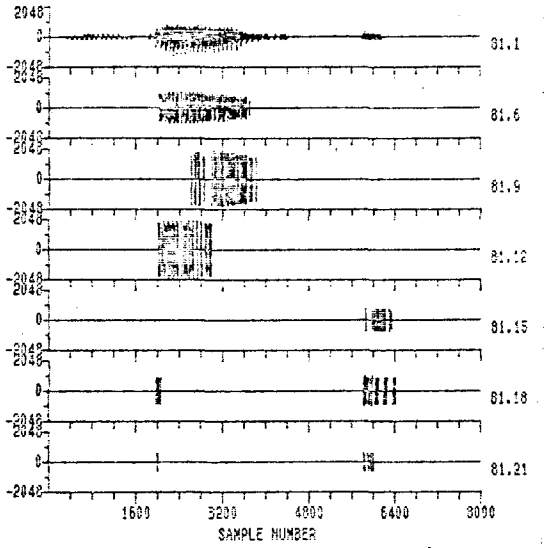
BIT

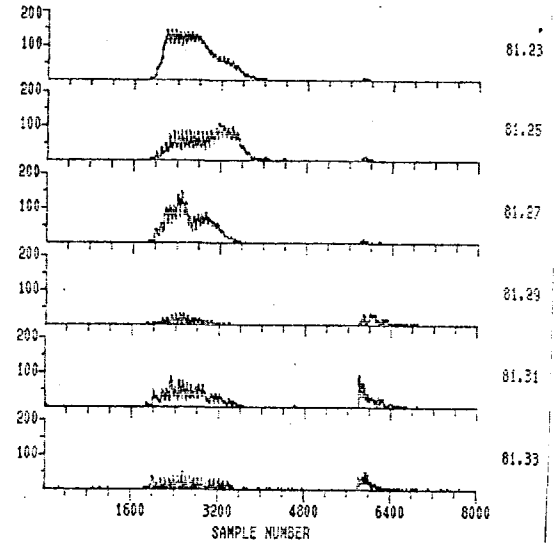

81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

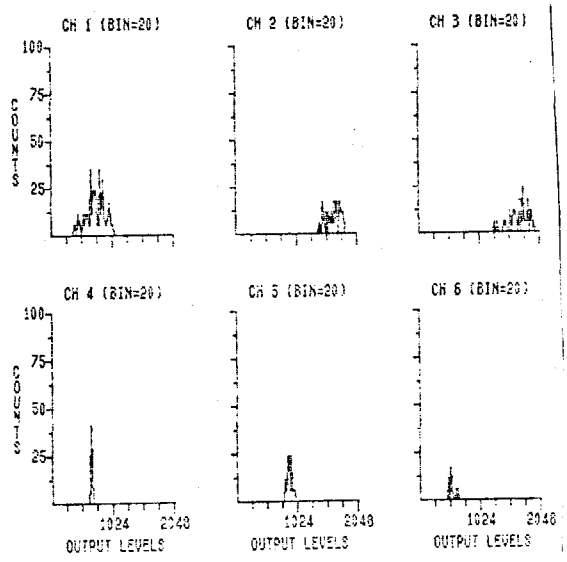

81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
100
75
50
25
1024
2048
OUTPUT LEVELS

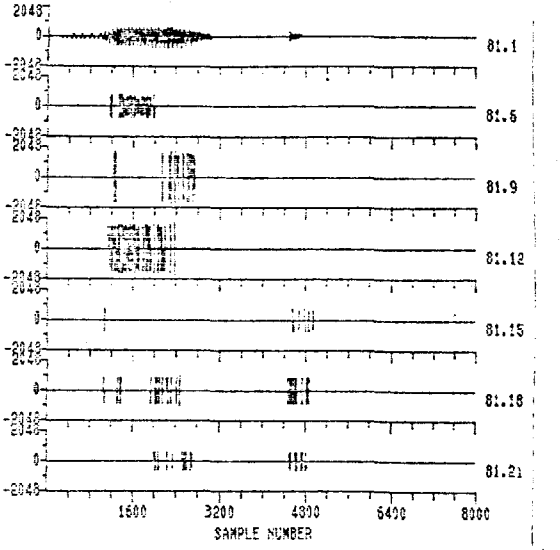
BOAT

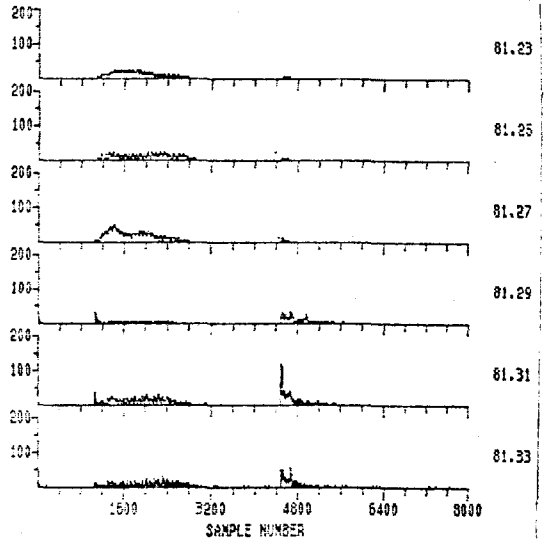

2048
0
-2048
81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

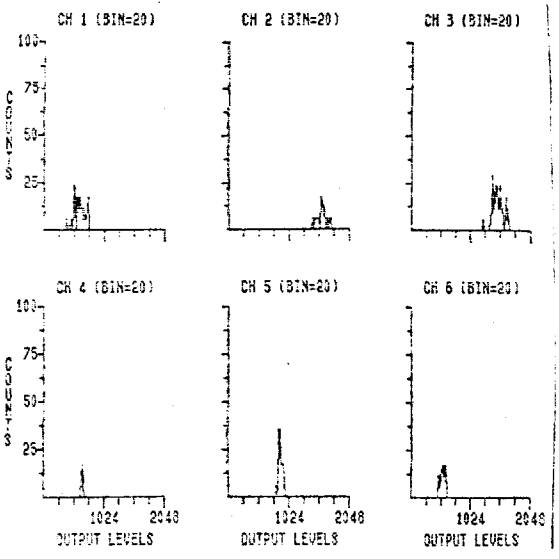

200
100
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
100
75
50
25
SECOND
1024
2048
OUTPUT LEVELS

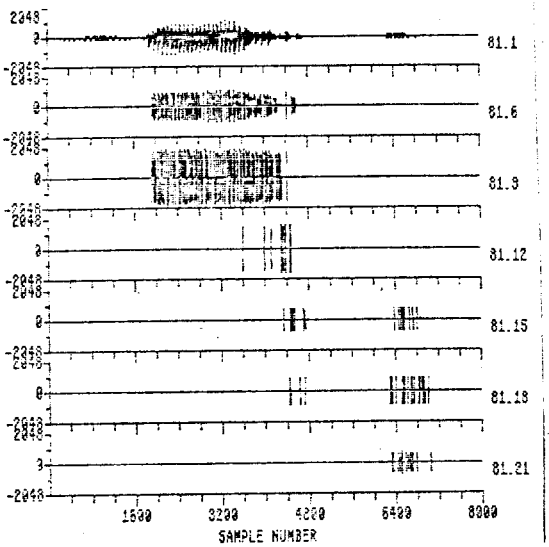
BOOT

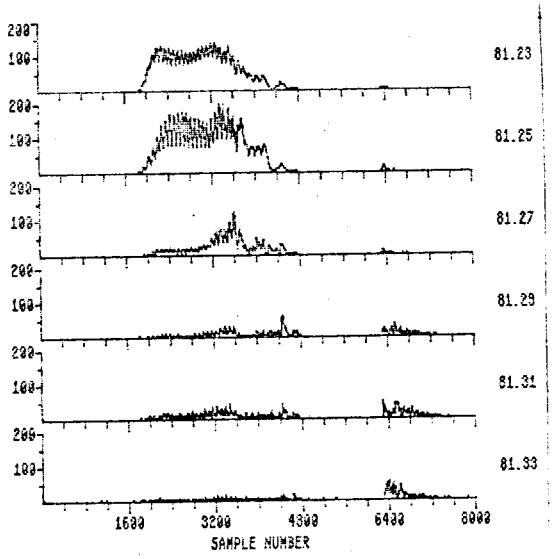

2048
0
-2048
81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

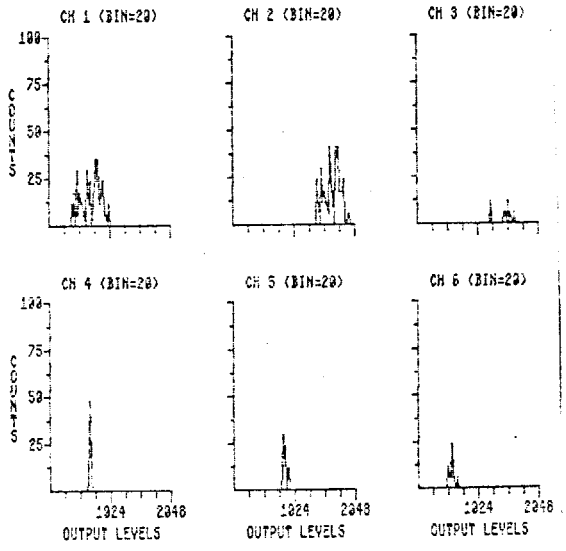

200
100
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
100
75
50
25
SECOND
1024
2048
OUTPUT LEVELS

BOUGHT


81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

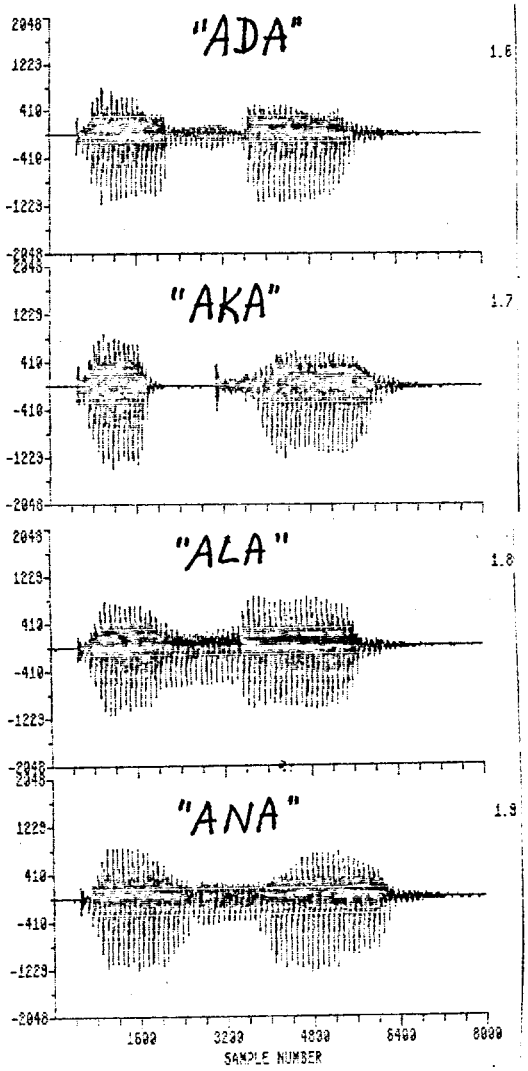

81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER

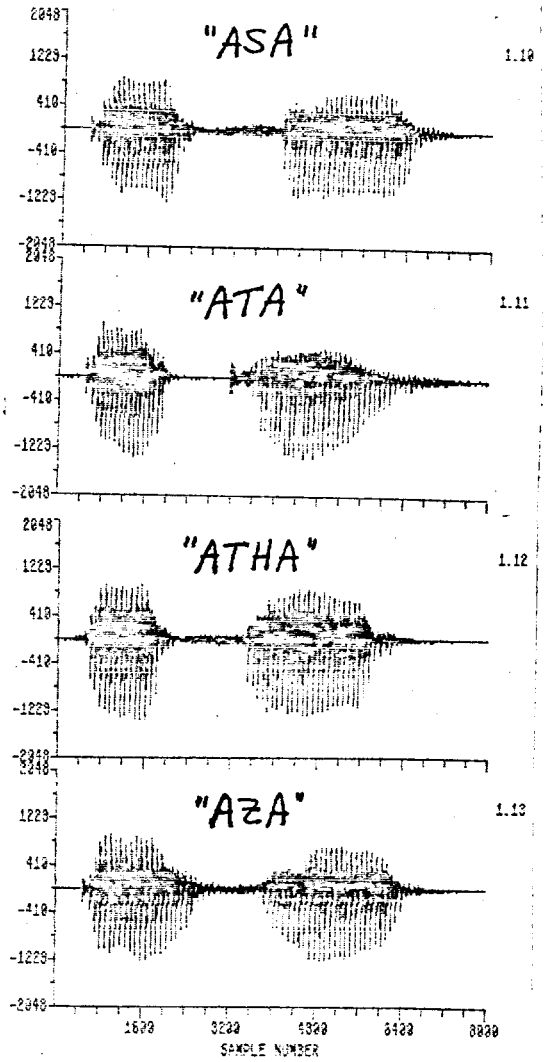

CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
100
75
50
25
1024
2048
OUTPUT LEVELS

Appendix 3

Consonant Confusion Tokens and Processor Outputs,
Strategy 4, 7/85

Consonant Tokens





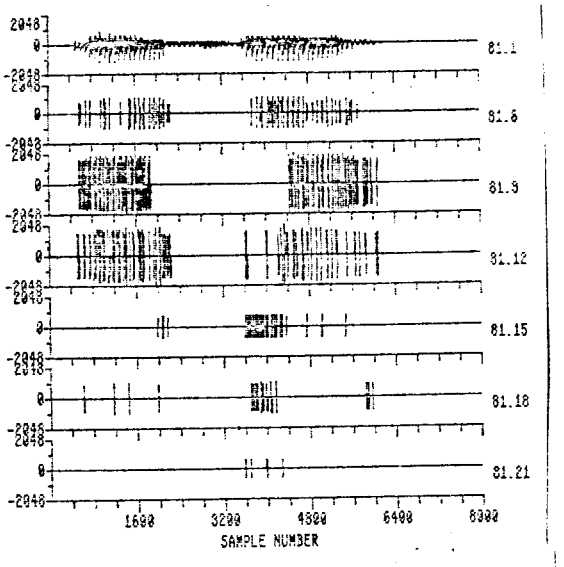
Key to Appended Plots

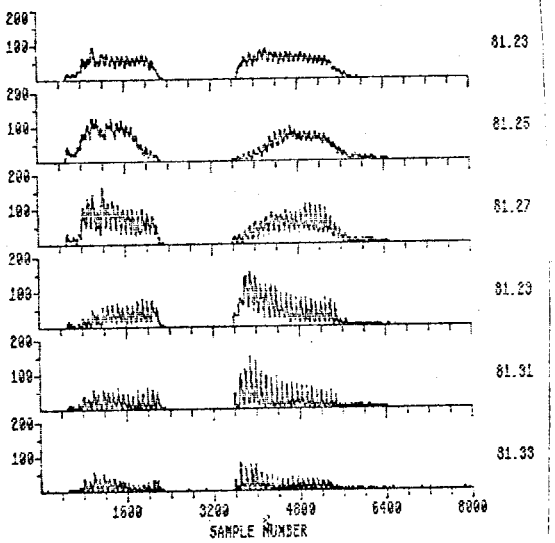
[Label numbers are in the form 81.x where x is the processor block number shown in Fig. A.3.1 on page 25 above. Each trace represents the output of the corresponding block.]

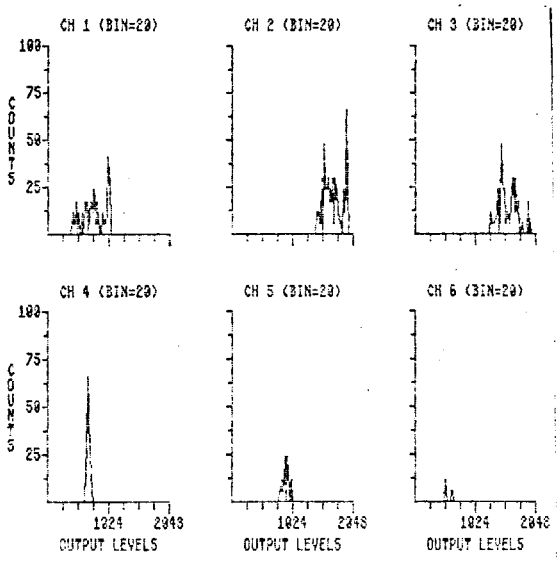
81.1	input speech signal
81.6	434- 712 Hz band output signal
81.9	712-1047 Hz band output signal
81.12	1047-1538 Hz band output signal
81.15	1538-2259 Hz band output signal
81.18	2259-3319 Hz band output signal
81.21	3319-4875 Hz band output signal
81.23	434- 712 Hz band RMS energy
81.25	712-1047 Hz band RMS energy
81.27	1047-1538 Hz band RMS energy
81.29	1538-2259 Hz band RMS energy
81.31	2259-3319 Hz band RMS energy
81.33	3319-4875 Hz band RMS energy

Samples-per-output-level histograms for each channel.

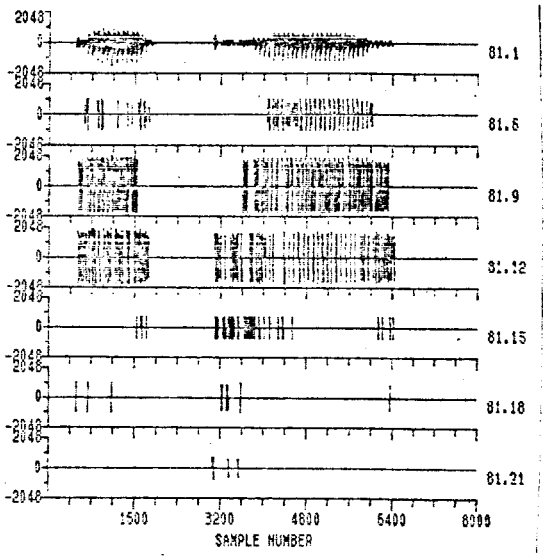
ADA

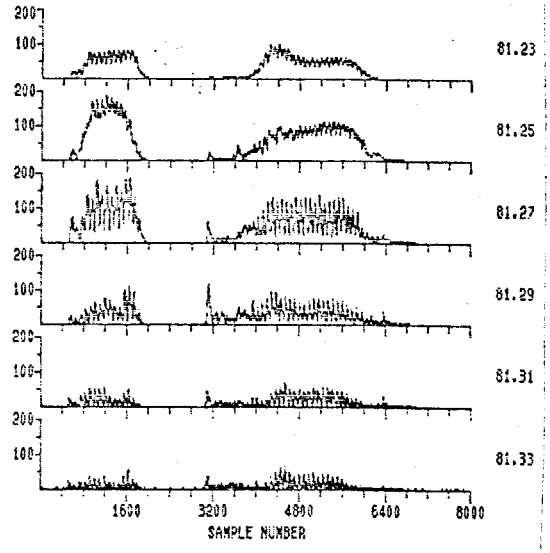

81.1
81.6
81.9
81.12
81.15
81.18
81.21
SAMPLE NUMBER

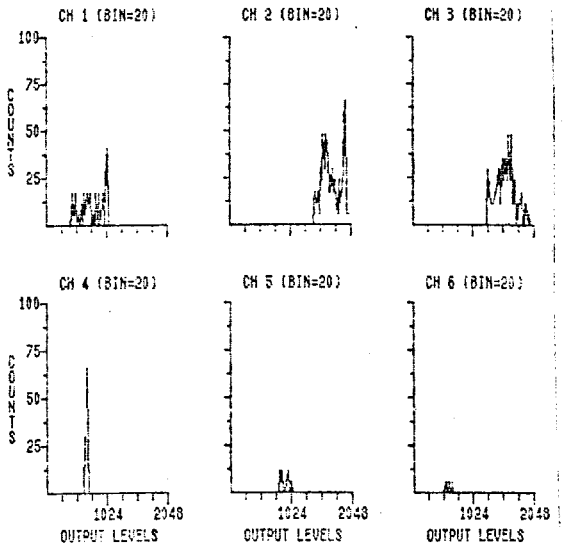

81.23
81.25
81.27
81.29
81.31
81.33
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
OUTPUT LEVELS

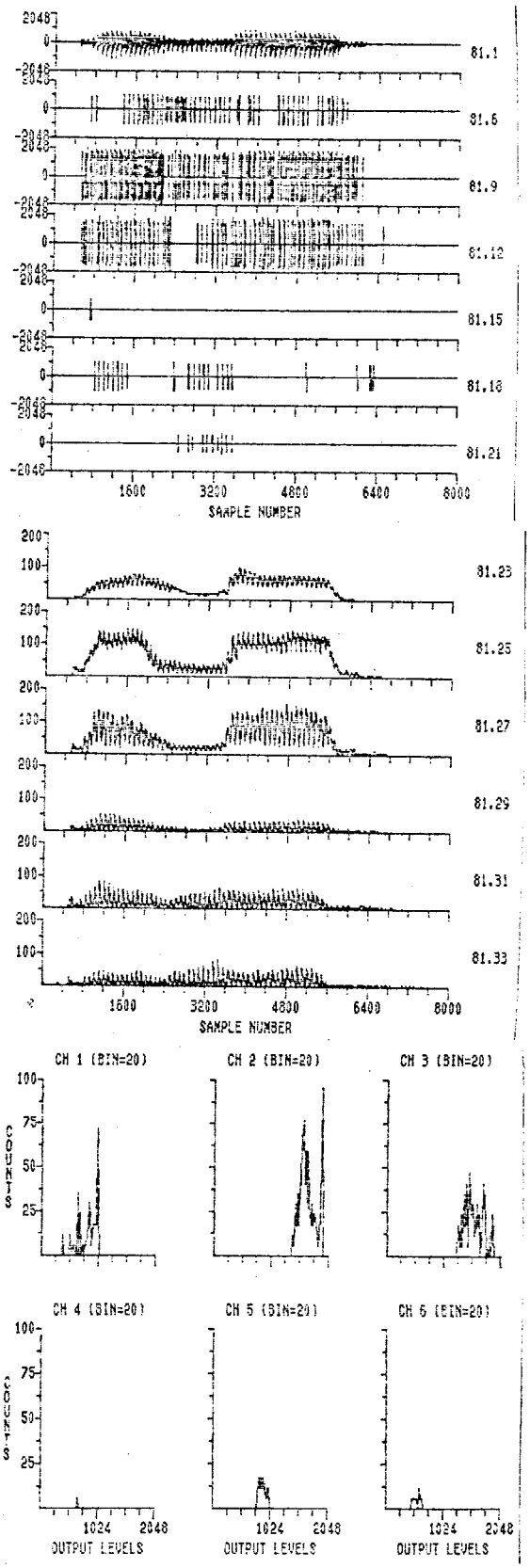
AKA


81.1
81.6
81.9
81.12
81.15
81.18
81.21
SAMPLE NUMBER

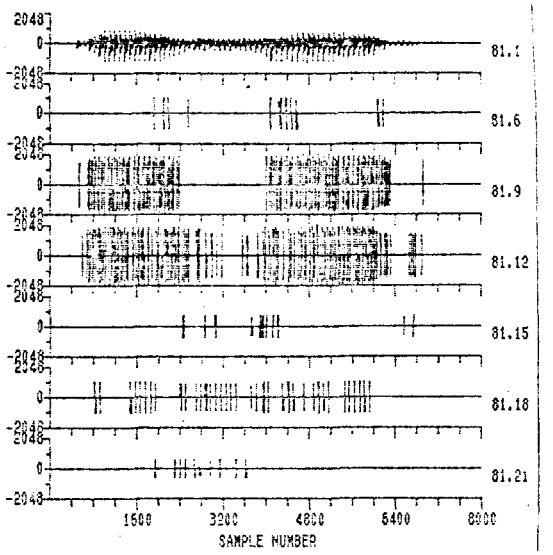

81.23
81.25
81.27
81.29
81.31
81.33
SAMPLE NUMBER

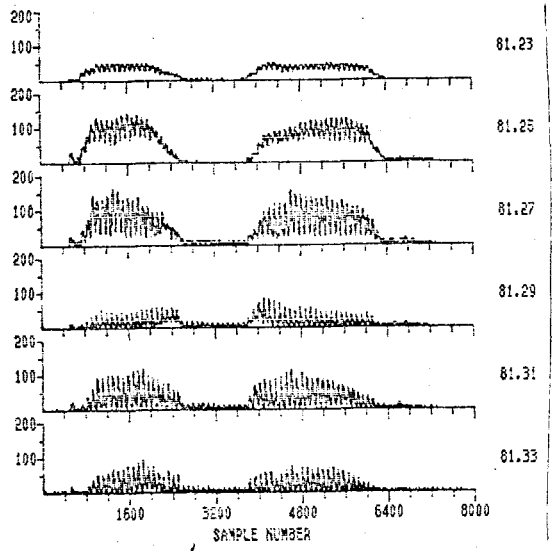

CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
OUTPUT LEVELS

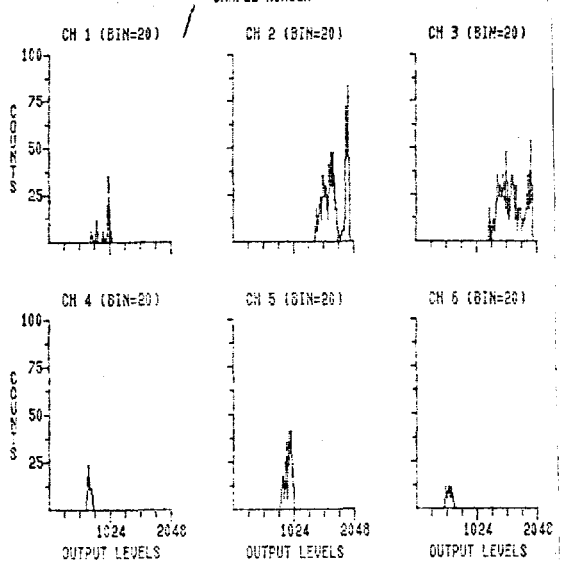
ALA


81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER
CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
100
75
50
25
1024
2048
OUTPUT LEVELS

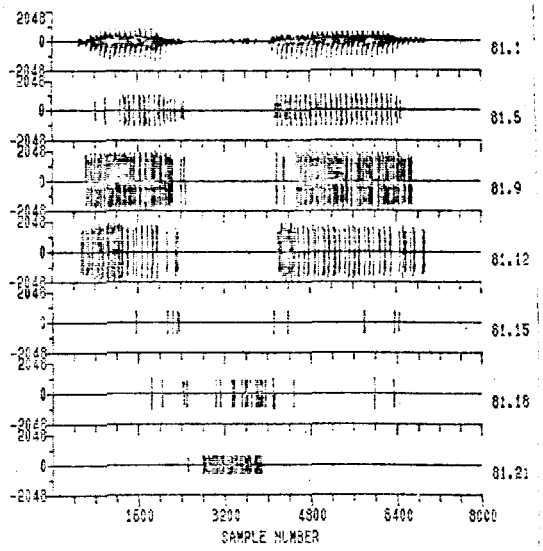
ANA

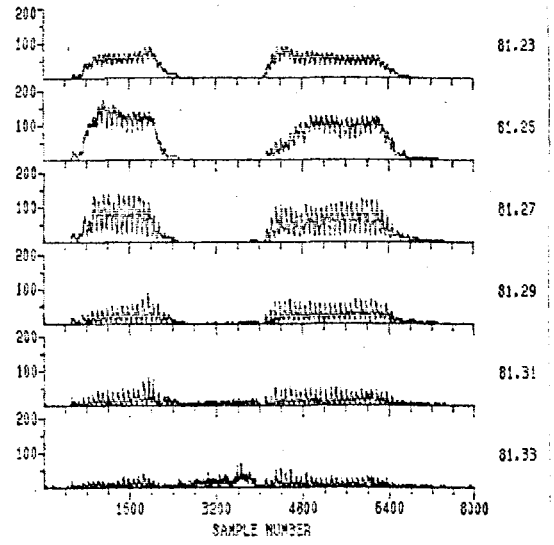

2048
0
-2048
81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

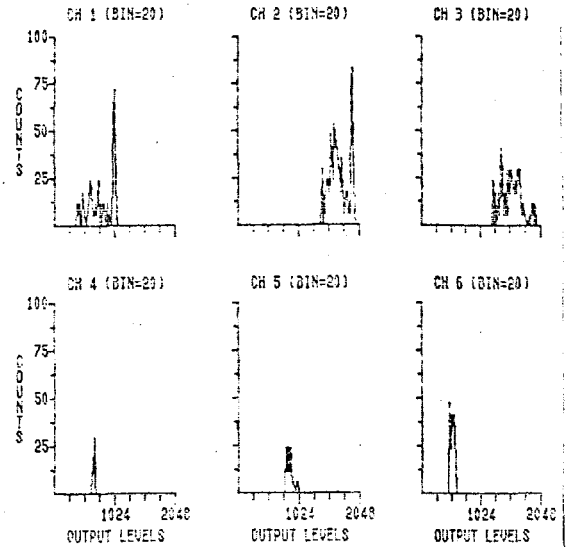

200
100
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
100
75
50
25
1024
2048
OUTPUT LEVELS

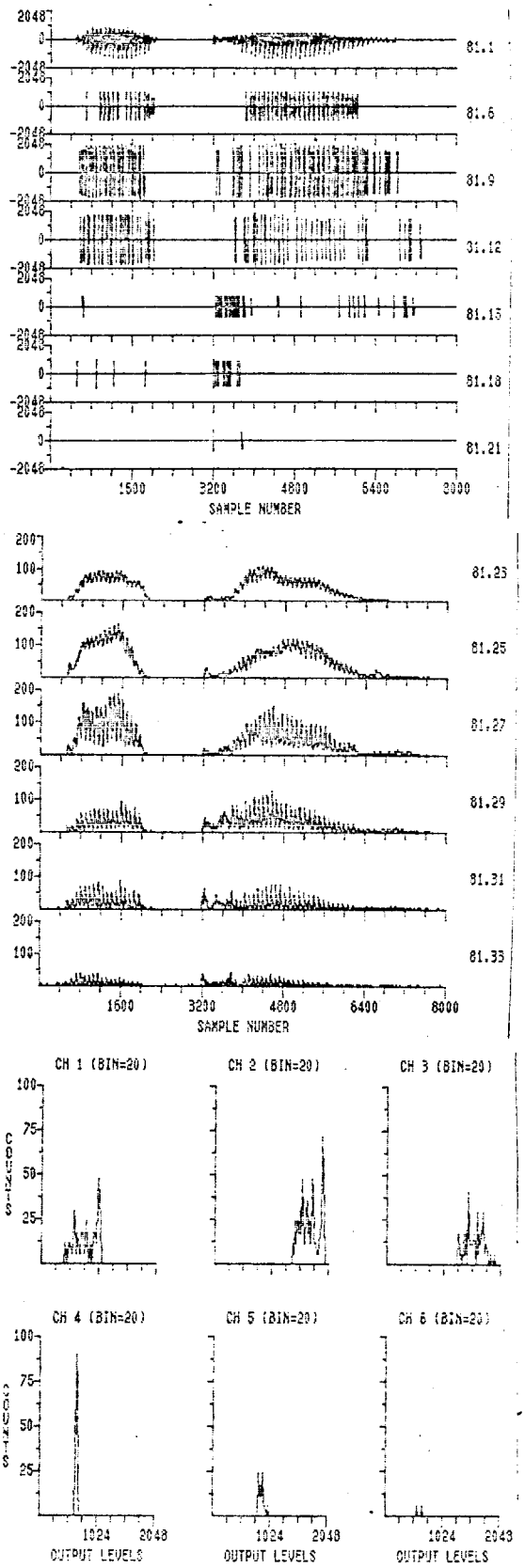
ASA


2048
0
-2048
81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER

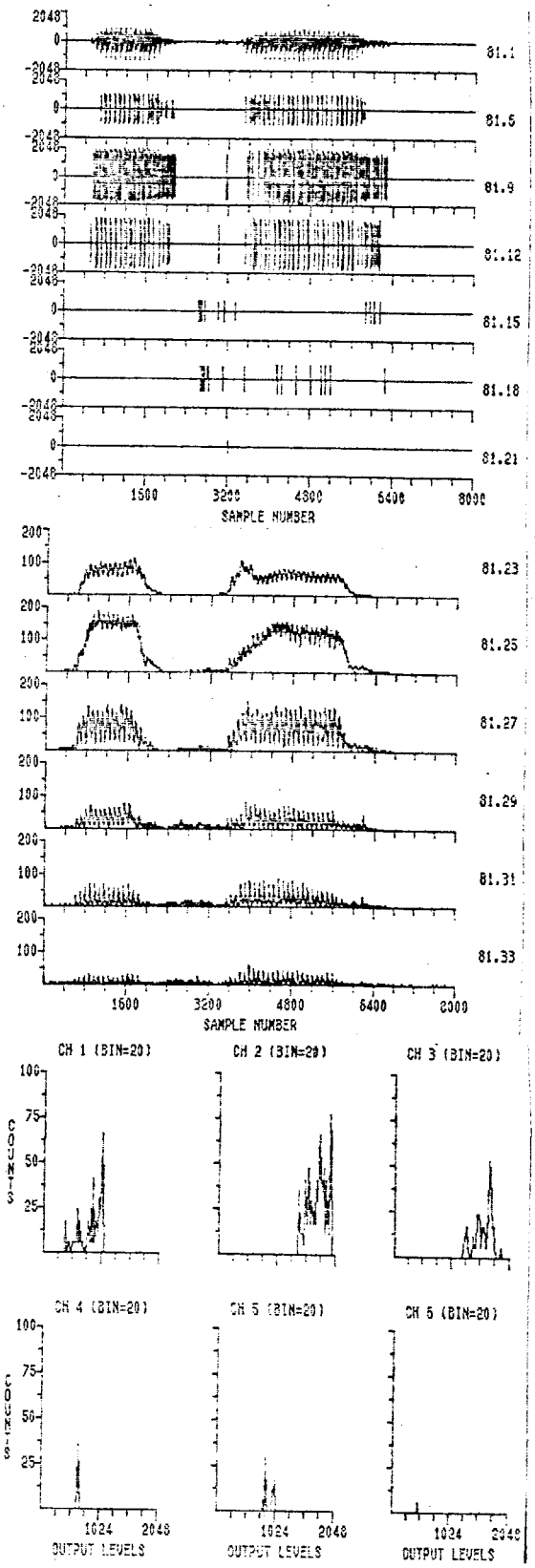

200
100
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER


CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
100
75
50
25
1024
2048
OUTPUT LEVELS

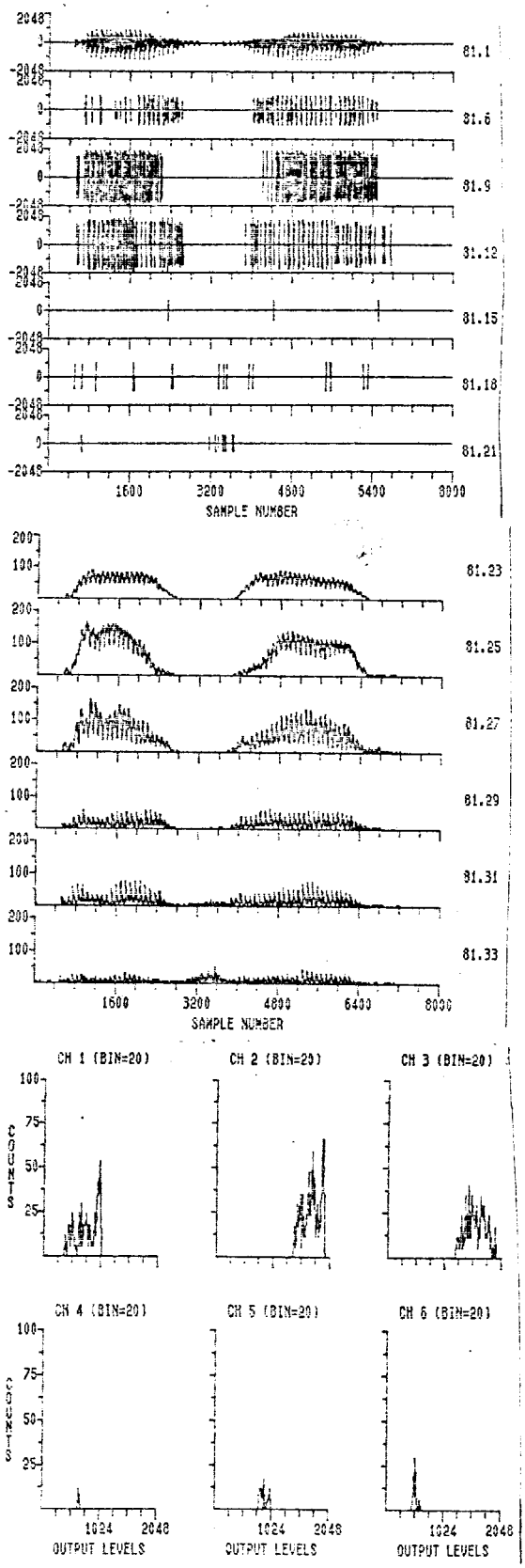
ATA


81.1
81.6
81.9
81.12
81.15
81.18
81.21
SAMPLE NUMBER
81.23
81.25
81.27
81.29
81.31
81.33
SAMPLE NUMBER
CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
OUTPUT LEVELS

ATHA


81.1
81.6
81.9
81.12
81.15
81.18
81.21
SAMPLE NUMBER
81.23
81.25
81.27
81.29
81.31
81.33
SAMPLE NUMBER
CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
OUTPUT LEVELS

AZA


81.1
81.6
81.9
81.12
81.15
81.18
81.21
1600
3200
4800
6400
8000
SAMPLE NUMBER
81.23
81.25
81.27
81.29
81.31
81.33
1600
3200
4800
6400
8000
SAMPLE NUMBER
CH 1 (BIN=20)
CH 2 (BIN=20)
CH 3 (BIN=20)
CH 4 (BIN=20)
CH 5 (BIN=20)
CH 6 (BIN=20)
COUNTS
1024
2048
OUTPUT LEVELS