RESEARCH TRIANGLE INSTITUTE

Fourth Quarterly Progress Report

June 26 through September 26, 1984

NIH Contract N01-NS-2356

Speech Processors for Auditory Prostheses

Prepared by

Blake S. Wilson and Charles C. Finley

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 multichannel 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 on a sector-by-sector basis. Work in this quarter was directed at (1) installation and checkout of RTI software at UCSF, for simulation and testing of speech processors with implant patients; (2) installation and checkout of the RTI hardware interface at UCSF, for communications between the Eclipse computer and implanted electrodes; (3) further development of the neural component of our integrated field-neuron model of electrical stimulation by intracochlear electrodes; (4) further development of our list of experimental objectives for tests with the next patient, with emphasis on the design and evaluation of "stimulus primitives"; (5) further development of modules within the computer-based simulator of speech processors, with emphasis on the modules that convey outputs from disk files to the hardware interface; (6) preparation and presentation of a formal review at UCSF on RTI's effort during the first year of project work; (7) joint development of plans with our coworkers at UCSF for tests to be conducted with upcoming patients; and (8) joint development of plans with our coworkers at UCSF for a series of single-unit experiments, to evaluate various predictions of our integrated field-neuron model. In this report we will briefly outline our progress in the first year of project work and describe the effort listed in point 3 above. In addition, we will present a detailed example of the use and application of the computer-based simulator of speech processors for multichannel auditory prostheses. Discussion of the efforts indicated in points 4, 7 and 8 is deferred for now, but will appear in future quarterly reports.

II. Overview of First-Year Effort

The main activities of the RTI group during the first year of project effort were the following:

1. Design, build and test a hardware interface to provide a communications link between an Eclipse computer and implanted electrodes;

2. Develop and evaluate an integrated field-neuron model of electrical stimulation by intracochlear electrodes;

3. Identify and contrast the most promising approaches to the design of speech processors for multichannel auditory prostheses, the product of which is a detailed list of experimental objectives for upcoming tests with implant patients;

4. Build a computer-based simulator that is capable of rapid and practical emulation of all these approaches in software;

5. Help to establish a strong collaboration between UCSF, Duke University Medical Center and RTI, so that parallel series of tests with implant patients can be conducted in the near future at both UCSF and Duke.

In all, we are proud of the progress we have made in the first year. In addition to meeting fully all requirements of the contract work statement with the completion of tasks 1, 3 and 4, we have been able to build a powerful tool for understanding and defining the "electrical-to-neural transformer" linking the outputs of the speech processor to the inputs of the central nervous system (task 2) and we have been able to help initiate a parallel testing effort at Duke (task 5). Our primary goal for the next year of this project is to apply the tools we have developed in the first year. We expect to begin testing patients at UCSF before the end of this calendar year and to begin testing at Duke this February or March. Finally, we expect to begin the single-unit experiments (to evaluate the integrated field-neuron model) sometime during the first half of this December.

III. Development of the Frankenhauser-Huxley Axon Component of an Integrated Field-Neuron Model of Intracochlear Electrical Stimulation

Before proceeding to the details of our progress on neural modeling,a brief review of the over-all modeling strategy is presented. Our work is motivated by the necessity to better understand the complexities of intracochlear electrical stimulation in order to fully specify and design advanced speech processors for cochlear prostheses. The strategy is to mathematically model the processes associated with electrical stimulation of neural tissue and then to adapt the model to the specific situation of the intracochlear prosthesis. The resultant model is an integrated field-neuron model which allows, firstly, the computation of the potential gradients along the locus of an axon as a result of electrical stimulation and then, secondly, computes the behavior of the axon in response to that particular stimulus. The over-all strategy is more fully described in our second quarterly progress report.

The potential gradients along an axon are calculated by an iterative, two-dimensional, finite-element model of a cochlear cross-section. The cross-section includes the presence of a scala tympani bipolar electrode. The electrode represents the current UCSF bipolar electrode design, compressed into two dimensions. Resistivities linking the finite elements are defined according to published values for the resistivities of tissues and fluids appearing in the cross-section. The bipolar electrodes are defined as equipotential conductors mounted in an insulating carrier medium. Fixed voltages are assigned to the electrodes and the resultant field patterns are computed by iteration for the entire cross-section. Ultimately, the potential levels along the locus of the nerve are extracted from the final field calculation. These potential levels represent the specific physical stimulus which is applied to a discrete neuron as a consequence of electrical stimulation. Our second quarterly progress report describes in greater detail the finite-element modeling approach.

The second half of the integrated field-neuron model is a lumped-element model of a myelinated axon. Potential profiles along the course of neural elements, as calculated with the finite-element model, constitute the input to this neural model. A portion of this model is represented in Figure III.1. The neuron model comprises a section of a myelinated axon containing 19 active nodes of Ranvier. Each node is isolated from adjacent nodes by 9 myelinated segments. The ends of the modeled neuron consist of 9 myelin segments alone. In all 199 computational segments are included in the model. Nodes of Ranvier are located at segments 10, 20, ... , 180, 190. Figure III.1 shows the electrical analogues assumed for each segment of the neuron. Five segments are shown, four myelin segments and one node of Ranvier. The myelin segments are assumed to be purely passive and comprise simply a parallel combination of the myelinated-segment transmembrane capacity and resistivity. The nodes of Ranvier are assumed to be nonlinear and are decribed by the Frankenhauser-Huxley node equations for the frog. Each node is described electrically by a transmembrane capacity in parallel with both a voltage-controlled current source and a series combination of a leakage resistance and battery source. The behavior of the voltage-controlled current source is described by the Frankenhauser-Huxley equations in conjunction with the transmembrane potential for that segment. The battery sources are adjusted to provide the normal resting potential for the axon at each of the active nodes. External to the axon at each segment is a voltage source, which drives through the impedance of the extracellular medium. These external voltage sources are set to the potential levels calculated by the finite-element model. For present work with the model, parameters are selected to describe the physical attributes of the myelinated, frog axon. This allows confirmation of the predictions of the model against a broad and well founded experimental literature. Once the modeling approach has been validated, parameters will be selected to more accurately describe the actual situation within the mammalian cochlea.

The present axon model extends the work of McNeal (IEEE Trans. BME 23:329-337, 1976) by providing for realistic impedance characterization of myelin. This expansion allows computation of neural responses other than simple prethreshold responses, enables observation of propagation of spike activity, and permits study of complex, arbitrary stimuli over extended periods.

Mathematically, the electrical analogues are described by node equations written for the external and internal nodes at each model segment. The node equations constitute a system of non-linear, simultaneous differential equations which are solved iteratively in time blocks. The computational algorithms have been debugged and the model's behavior is

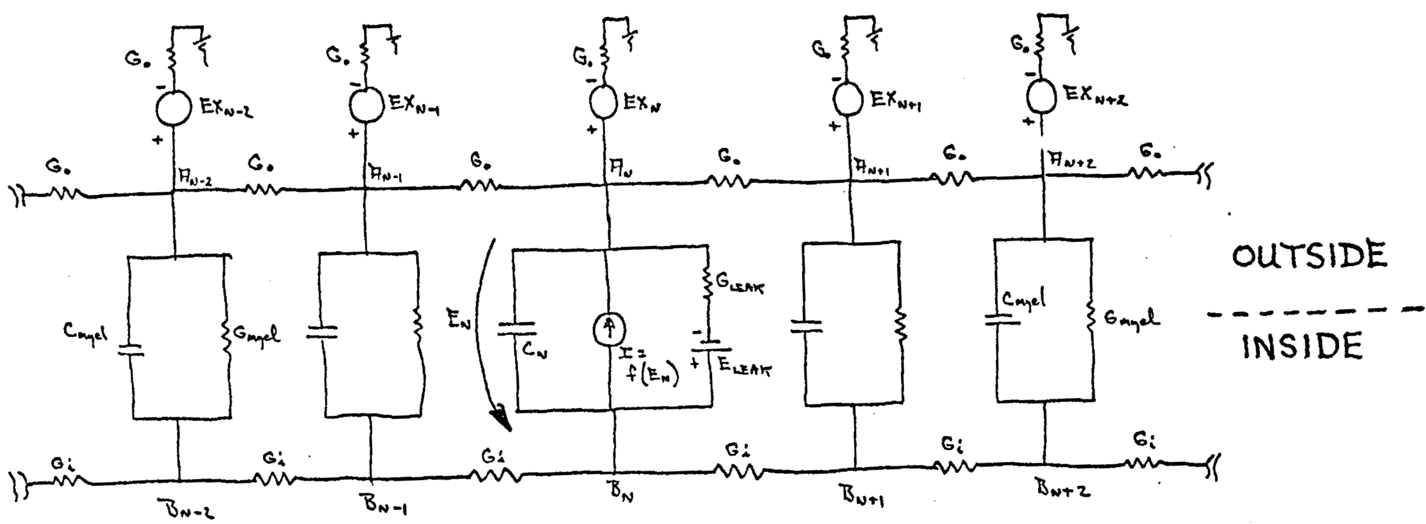

Figure III.1 Axon model showing a single Frankenhauser-Huxley node surrounded by four myelinated segments. A_N is the external segment voltage for axon segment N. B_N is the internal segment voltage. EX_N is the external voltage source for stimulation of that segment. See text for explanation.

now being explored.

Figure III.2 illustrates the response of the model to a simple voltage pulse applied to the end node of the model axon. Specifically, the external voltage stimulus is a +70 mvolt, 50 microsecond stimulus with the profile along the axon as shown in the inset of Figure III.2. The display is a perspective view of sequential time samples of the internal voltages of the axon at all 199 computational segments. The abcissa~~e~~ is the computational segment position along the axon, ranging from 1 to 199. The ordinate is the intracellular voltage referenced to the normal resting potential of -70 mvolts. Sequential time intervals receed into the background at 10 microsecond intervals. A total time span of 240 microseconds is displayed, beginning with the stimulus onset. As can be seen in Figure III.2, the 50 microsecond extracellular stimulus depolarizes the cell in the vicinity of the first node of Ranvier (segment 10). A slight regenerative depolarization can be seen at the node position at 40 and 50 microseconds into the stimulus. Then as the stimulus is turned off, the intracellular potential shifts again toward the resting potential. However, the first node is well into its regenerative phase and continues to depolarize the axon. From 60 to 240 microseconds in time, the early rising phase of an action potential is observed.

Consistent with normal axon behavior, graded stimuli reveal sub- and supra-threshold behavior of the model. Figures III.3 (a-d) show similar perspective displays of the model's response to the similar stimuli, varying only in their amplitudes. Panel (a) is the response to a 70 mvolt stimulus and is identical to the response shown in Figure III.2. Panels (b), (c), and (d) show responses to 60, 55 and 40 mvolt stimuli, respectively. Spike generation is evident in panels (a) and (b). The response shown in panel (c) ultimately produces a spike at long latency if time is extended. The panel (d) response is clearly subthreshold. Quantitative analysis to compute strength-duration and strength-latency curves is currently in progress.

As computation times are extended, more of the axon response may be observed. Figure III.4 shows a response display similar to the ones previously discussed, however with time increments of 20.0 microseconds. As time progresses, a gradual spreading of the axonal depolarization begins, ultimately causing the second node of Ranvier to depolarize regeneratively.

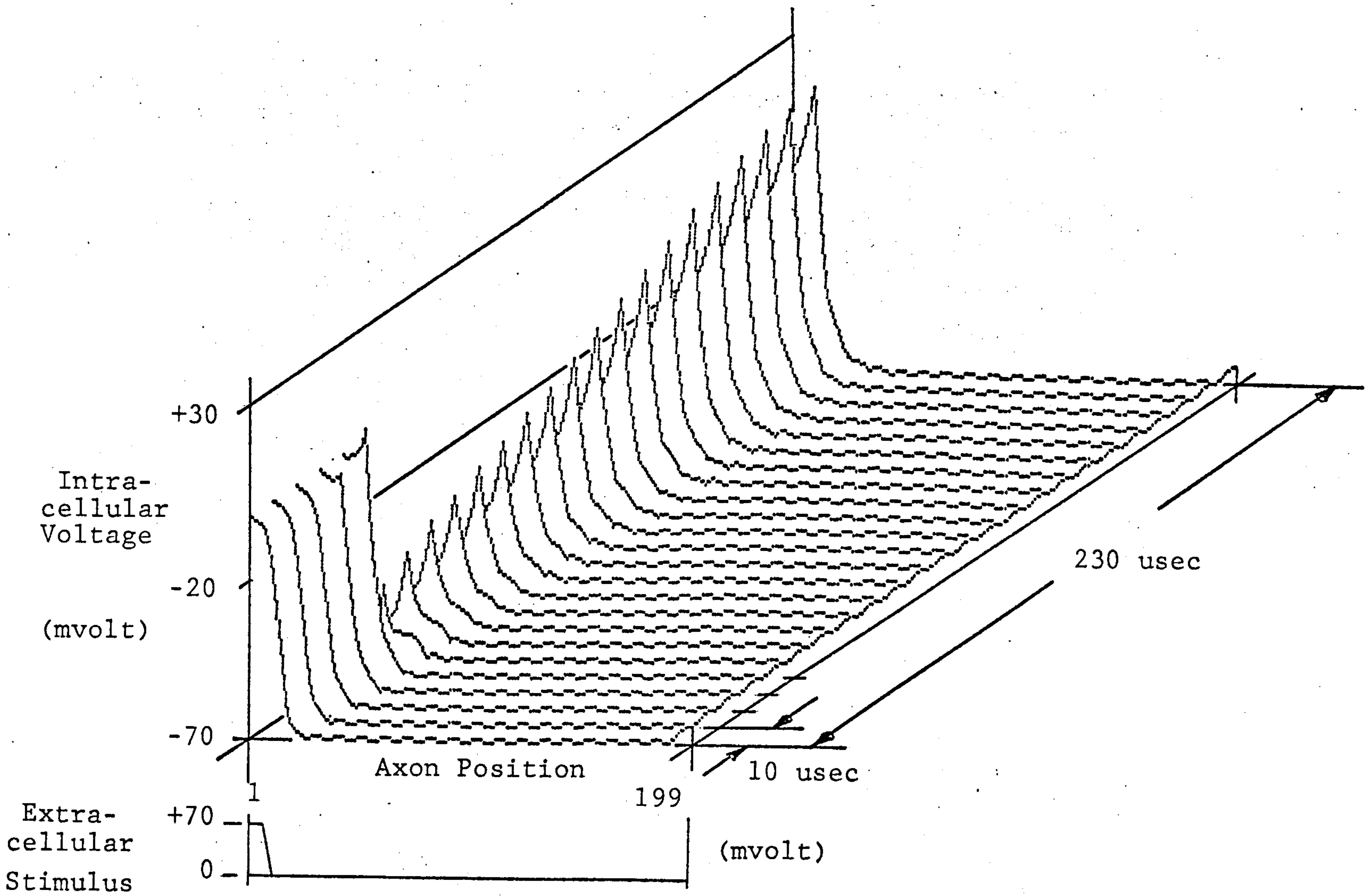

Figure III.2 Perspective display of intracellular voltages along a 19 node axon model in response to a 50 usec. external voltage pulse of 70 mvolts. Extracellular stimulus profile along axon is shown in the lower ~~figure~~ panel. See text for full explanation.

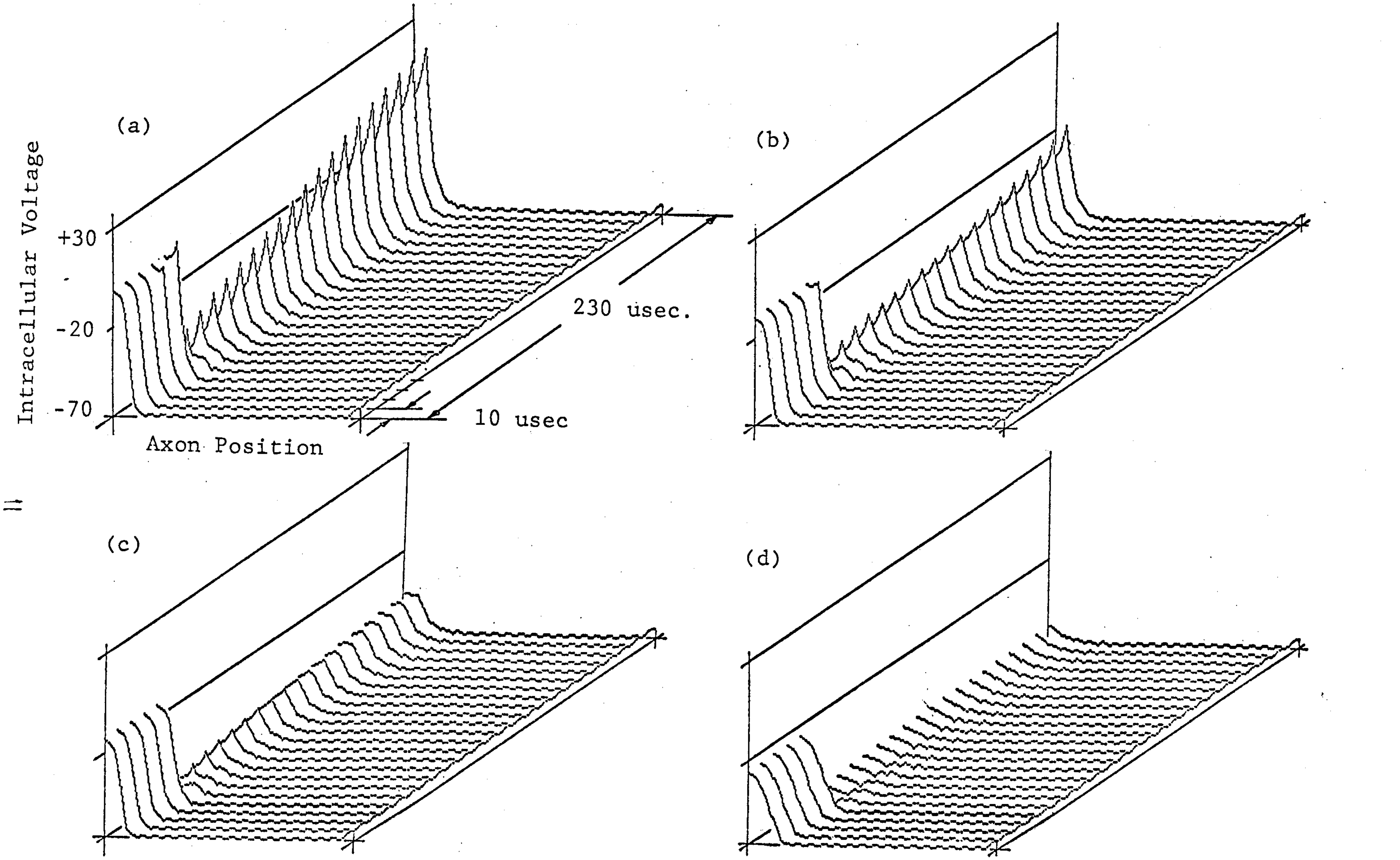

Figure III.3 Perspective displays of intracellular voltages along the axon model in response to 50 usec. external voltage pulses.
(a) 70 mvolt stim; (b) 60 mvolt stim; (c) 55 mvolt stim; (d) 40 mvolt stim.

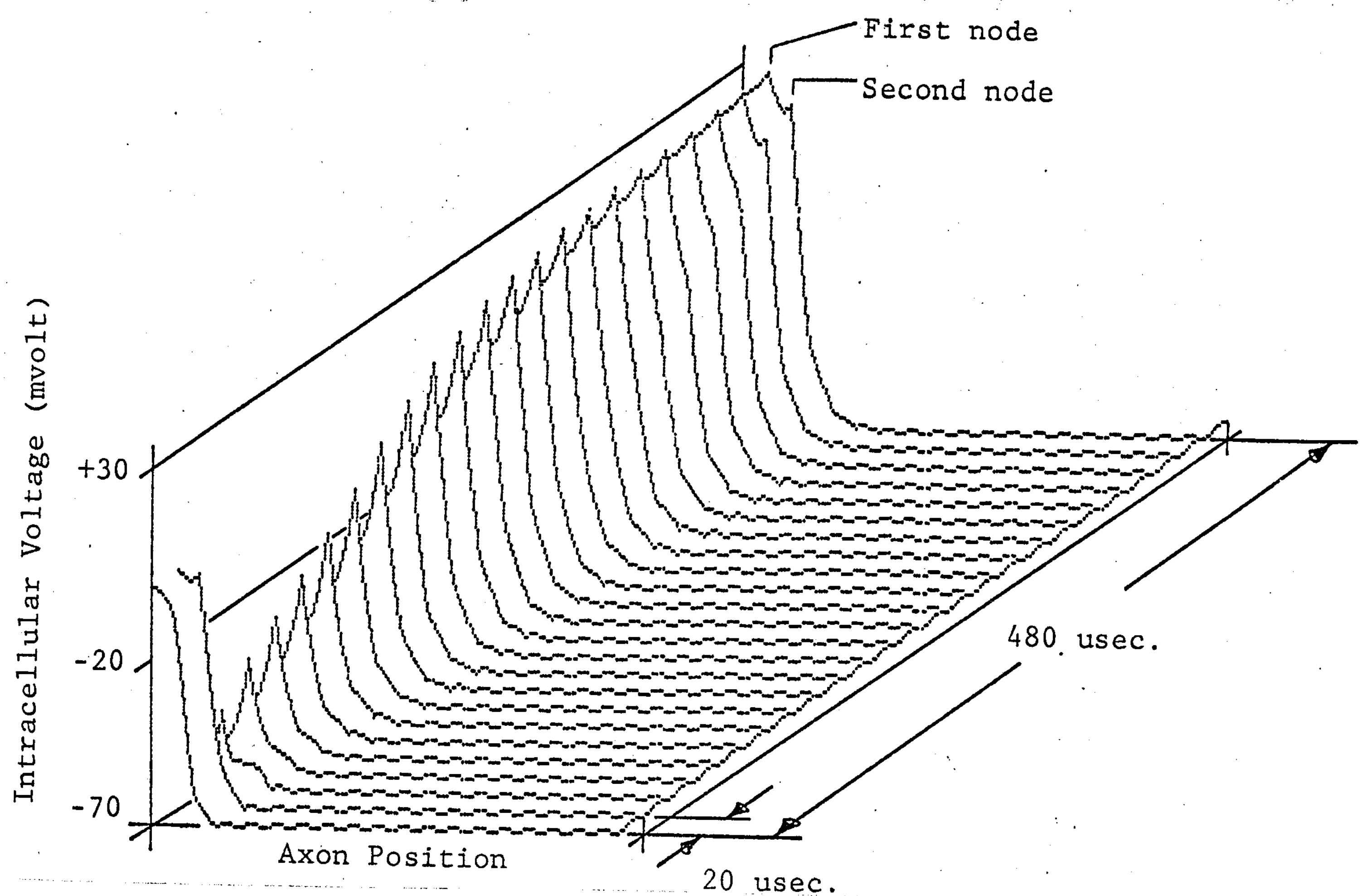

Figure III.4 Perspective display of intracellular voltages produced under the same conditions as those in Figure III.2. In the present display, however, the time increment has been doubled to 20 usec. Note recruitment of the second node to begin propagation of the spike wavefront.

Further expansion of the time display intervals to 100 microseconds reveals a spike wave front which has achieved a maximum potential level and clearly propagates along the axon, as shown in Figure III.5. Rough calculation indicates the propagation velocity of the spike front to be approximately 10 meters/second. The apparent discontinunities of the intracellular potentials along the axon, especially at the spike wave front, are typical of the so-called "saltatory conduction" of the spike from node to node (see Fitzhugh R., Biophysical J. 2:11-21, 1962).

As a final example of the qualitative behavior of the axon model, it is interesting to observe the collision of two spikes which are propagating in opposite directions along the axon. Figure III.6 shows at time intervals of 240 microseconds the collision of two spikes. Each spike was elicited at time zero with stimuli similar to the stimuli discussed above. In this case however, stimuli were applied to both end nodes of the axon simultaneously. As seen in the figure,the spikes meet in the middle of the axon, boost the intracellular potential to a peak value, then stabilize into a nonpropagating potential profile that decays approximately exponentially toward the ends of the axon. Figure III.7 shows the same collision but at an expanded time interval of 1.5 mseconds. Again the same potential profiles are observed as the spikes collide, then stabilize into a nonpropagating potential distribution. As time continues, the stabilized potential distribution suddenly dissolves into a uniform resting potential distribution. The details of such dynamics have not been explored with the model to determine ~~as to what is~~ the cause of the sustained nonpropagating distribution and its sudden decay. In any case the model does accurately predict the expected behavior for colliding spikes, specifically their complete cancellation.

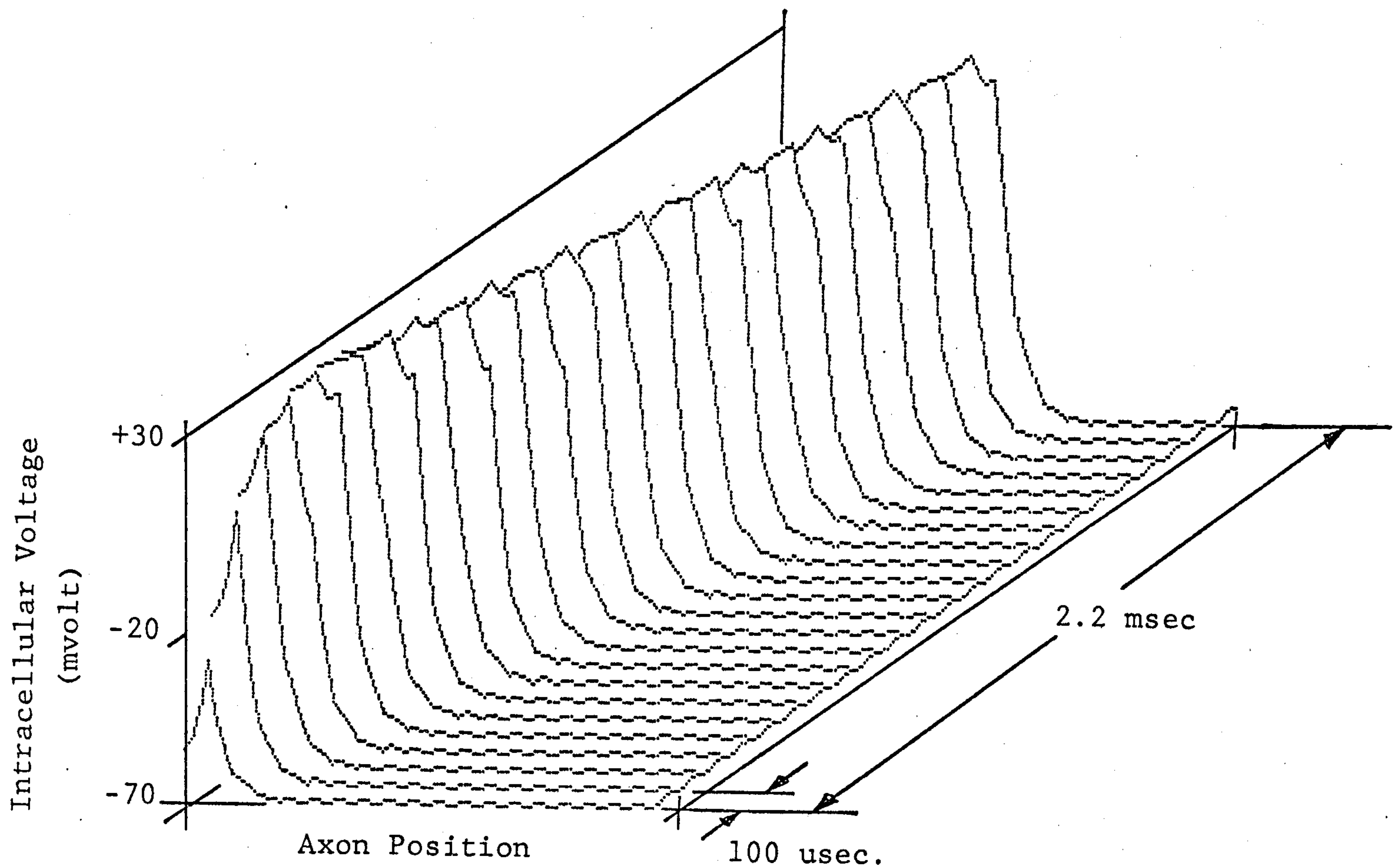

Figure III.5 Perspective display of intracellular voltage along the axon model showing a propagating spike following spike initiation.

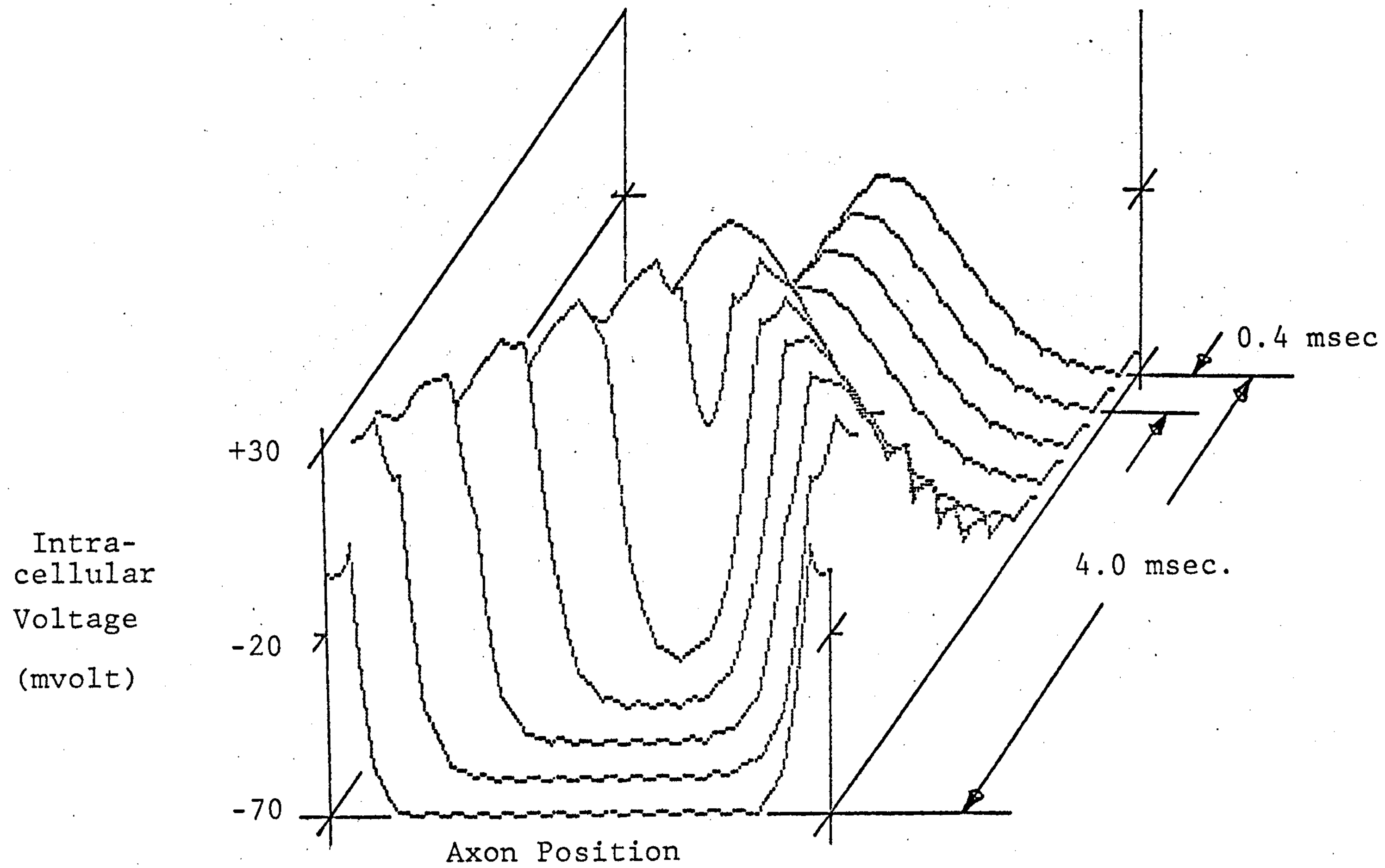

Figure III.6 Collision of two propagating spikes showing time slices of early interactions between waves.

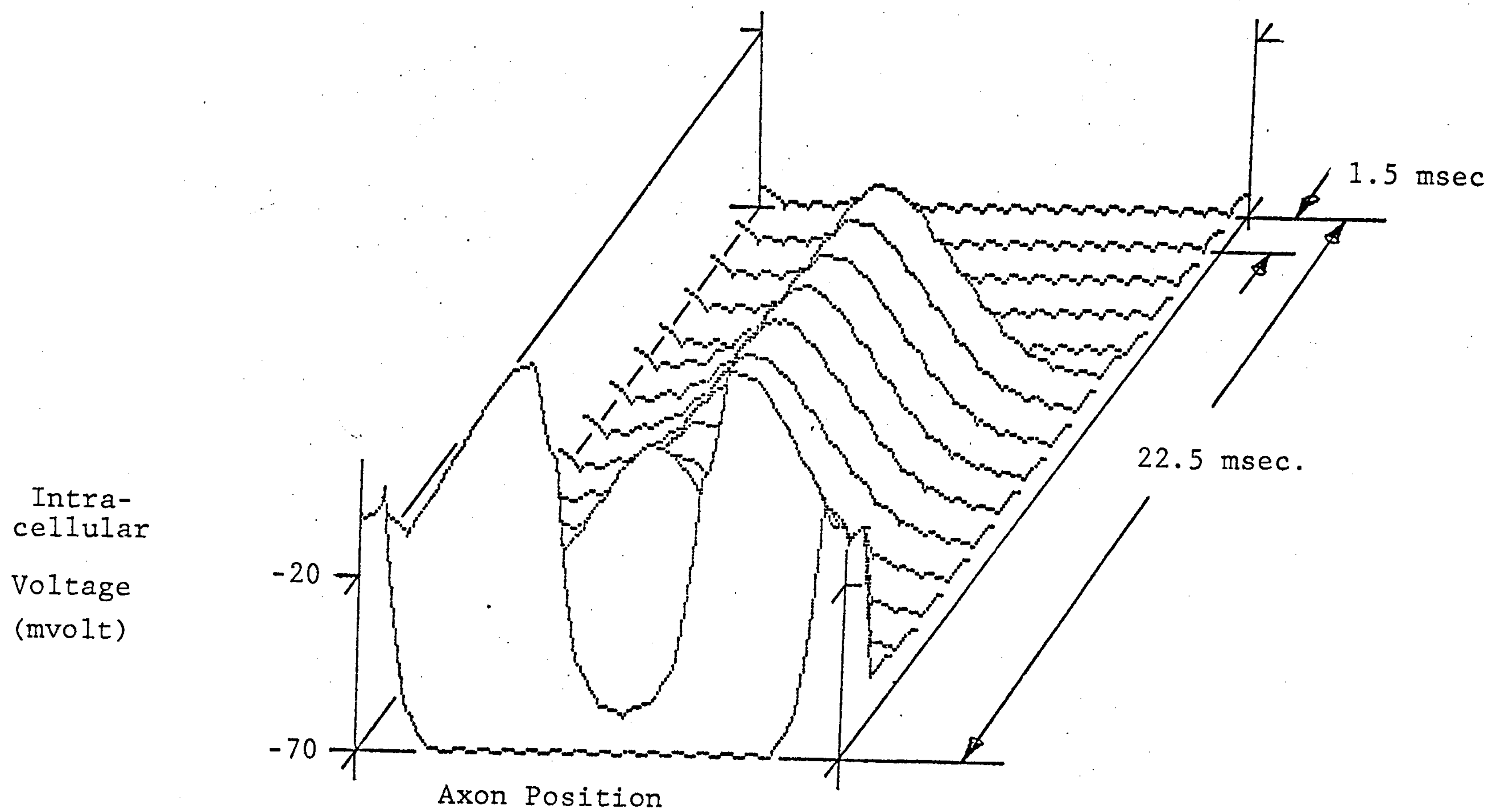

Figure III.7 Collision of two propagating spikes showing time slices of total interaction between waves, culminating in complete cancellation.

IV. Description of the Use and Application of a Computer-Based Simulator of Speech Processors for Multichannel Auditory Prostheses

In previous quarterly reports we have described the overall design of the RTI computer-based simulator of speech processors for multichannel auditory prostheses. In the present report we will present a detailed example of the use and application of this system.

First, as indicated in a previous report, the software for the simulation system includes the following programs:

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

DESIGN — program for 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 signal-processing systems;

SHOWNTEL. — 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 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.

To illustrate the use and application of these programs, we will present an example of the specification and simulation of a relatively-simple processor. The principles of specification and simulation of more-complex systems are no different from the principles indicated in the following description. Therefore we hope this description will indicate to the reader the power of the simulation system for implementing in software any of the speech processors used in current auditory prostheses.

A session in which one or more of the above programs is to be used is initiated by calling program CPEXEC. This program presents the menu of options shown in Fig. IV.1. With the exception of PREPARE, these options provide choices for calling the remaining programs in the set listed above. To specify the design of a new signal-processing system, then, the investigator enters a "1" (for "DESIGN A NEW SYSTEM"), and responds to the initial queries of the DESIGN program, as shown in Fig. IV.2. In this particular case design # 2 is selected (up to 9999 design files can be stored on the disk) and the system sampling frequency is set at 20 kHz.

Once these preliminary entries have been made, the investigator is in a position to define the network of the signal-processing system on a block-

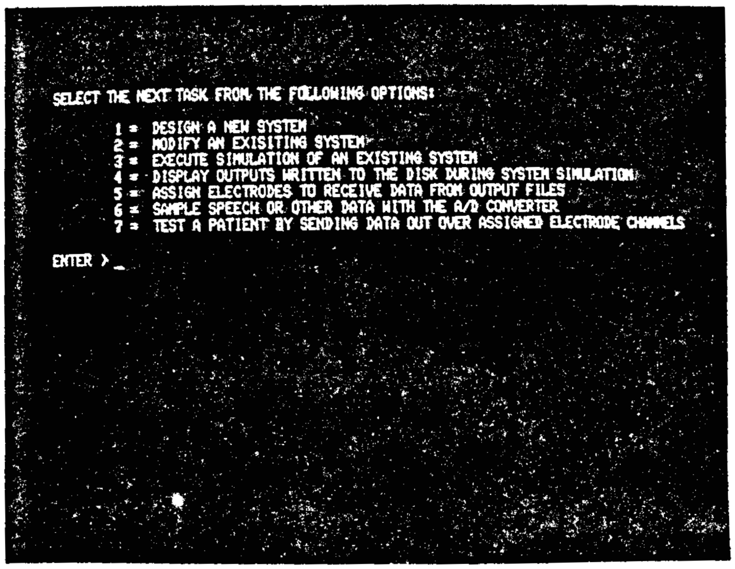

Fig. IV.1. Menu presented by CPEXEC.

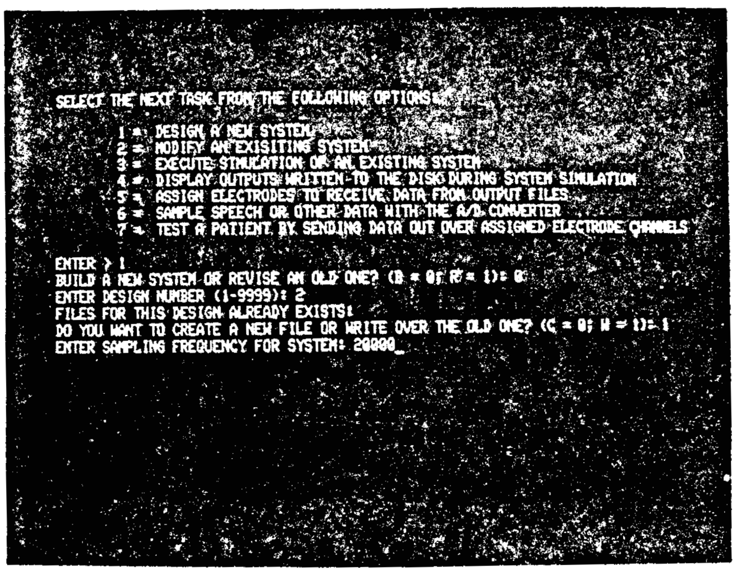

Fig. IV.2. CPEXEC menu followed by initial queries and responses of program DESIGN.

by-block basis. The system to be defined in the present example is shown in Fig. IV.3. It consists of four bandpass channels of processing in which an approximate measure of the rms energy in each band is obtained. This system

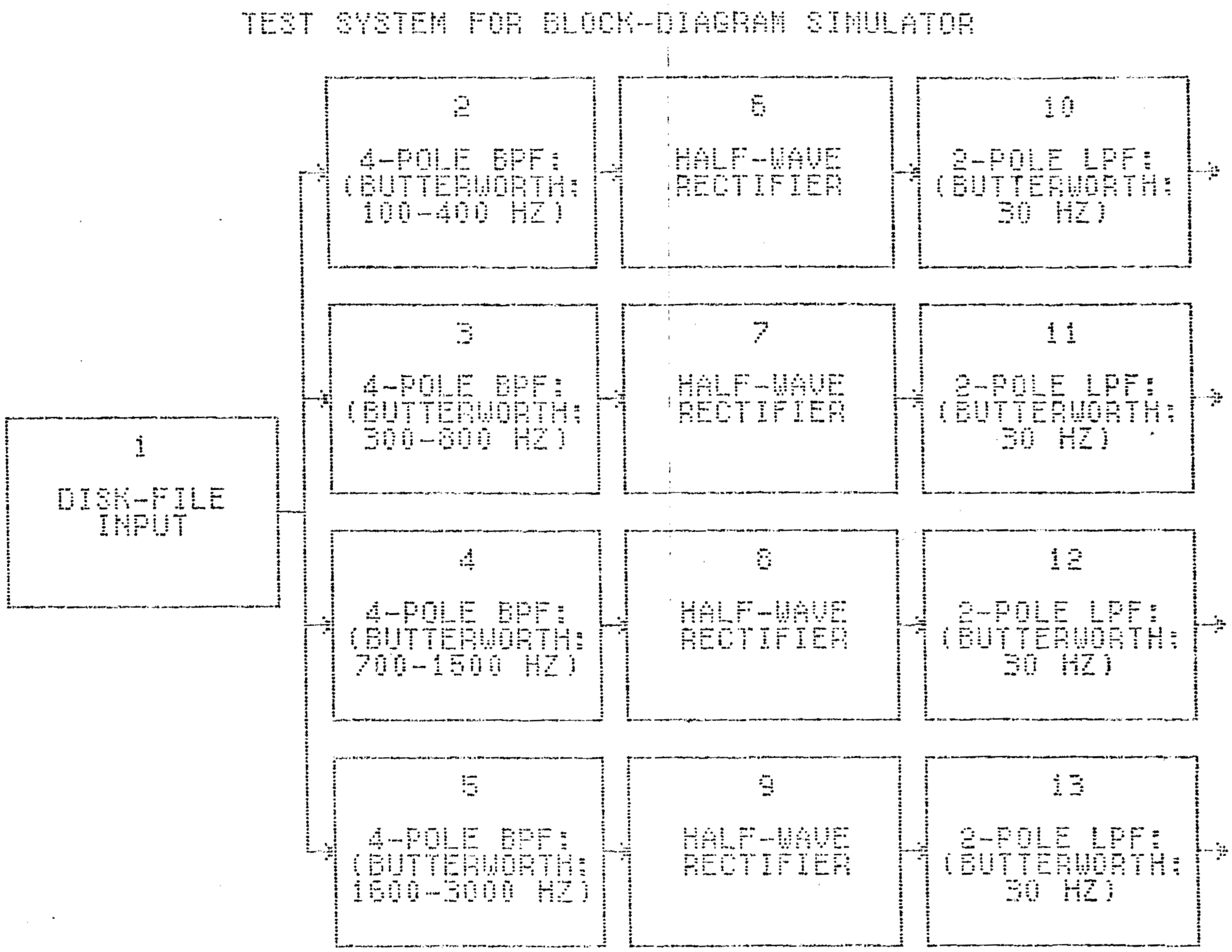

Fig. IV.3. Test system for block-diagram simulator.

is typical of the "front ends" of channel vocoders and could be used in somewhat modified form as the front end for certain types of speech processors for multichannel auditory prostheses.

The next screen presented after entering the global sampling frequency is the first of two main menus of the DESIGN program. These menus specify the functions that can be assigned to each block within the system's network of blocks. As indicated in Fig. IV.4, these functions fall into general categories. The categories shown in the first menu are those for modules that implement standard functions of digital signal processing ("DSP"), speech analysis, generation of signals ("SIGNAL SOURCE"), and various mathematical operations. The range of choices within each category is

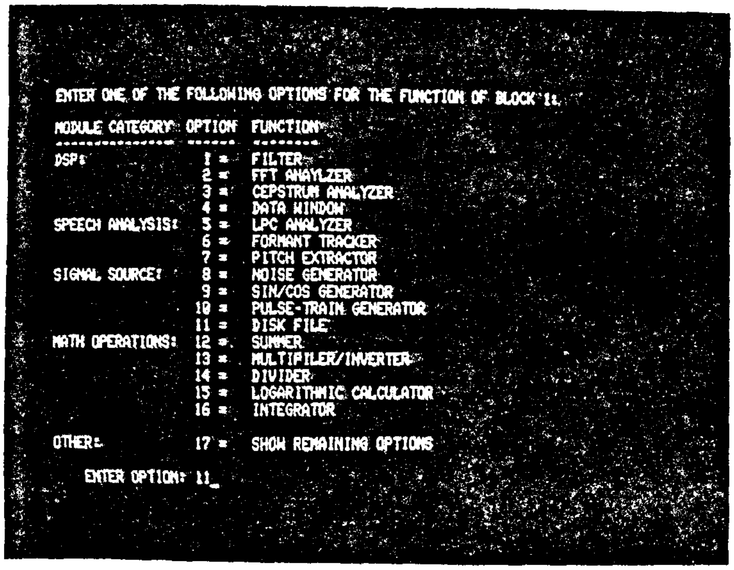

Fig. IV.4. First main menu of the DESIGN program. The option entered at the bottom indicates that the function of block # 1 is to read information from the disk and to make this information available as a signal source to other blocks in the network.

designed to encompass most of the possibilities for speech processors for auditory prostheses. For cases in which other functions must be included, the second main menu (to be described below) provides an option for adding new functions to the list.

To begin specification of the signal-processing system shown in Fig. IV.3, the function of block # 1 is entered. The number of this function is "11", which, once entered, calls a series of menus for specifying the source of information on the disk to be read by the block. The first of these menus is presented in Fig. IV.5. This menu asks the investigator to indicate when in the simulation process he or she wants to name the disk file. The "@DATA" option allows the use of a generic file naming facility in the Data General AOS operating system.

In the example, a "2" is entered to select the disk input file at the time of system specification. This entry prompts the next menu and series of queries shown in Fig. IV.6. The responses indicate that file "TUNA" is to be read by block # 1, and that this file does not have any of the special

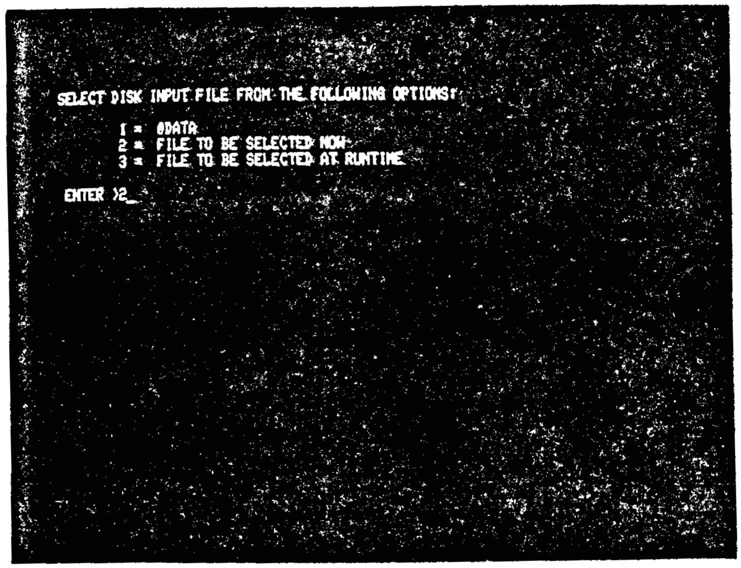

Fig. IV.5. First menu for full specification of the "disk file" option.

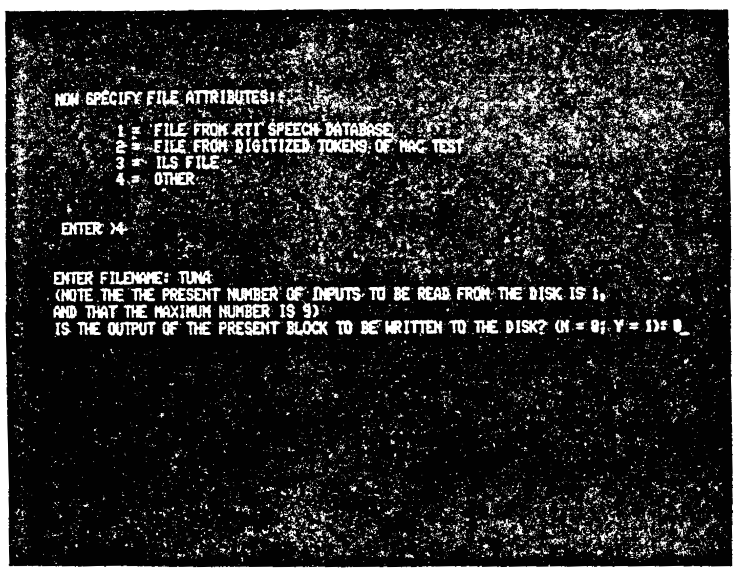

Fig. IV.6. Second menu and subsequent dialog for full specification of the "disk file" option.

headers used for files from the RTI speech data base, digitized tokens of the Minimal Auditory Capabilities Battery (the "MAC TEST"), or files generated by the Interactive Laboratory System of Signal Technology, Inc.

(option 3, the "ILS FILE"). If a response other than "4" is entered for the choices indicated in the file attribute menu, the subsequent queries for identifying the file are tailored to the structure of the file headers. Finally, once the file is identified, an option is presented to write the output of the present block to the disk. This option appears at the end of the specification sequence for each block in the system network. In general, the outputs of all blocks that feed electrode channels must be written to the disk (for subsequent handling by programs ASNELEC and TEST), and other outputs the investigator wishes to examine after network simulation must also be writen to the disk (for subsequent handling by program SHOWNTELL). The penalty associated with writing an excessive number of outputs to the disk is increased computer time required for simulation. Because file "TUNA" already exists on the disk, the option to write it back to the disk in another file is not taken for block # 1.

With the full specification of block # 1 as just described, the DESIGN program returns to its first main menu to read from the investigator the function of the next block in the network. The screen for the initial specification of block # 2 is shown in Fig. IV.7. Reference to Fig. IV.3 indicates that block # 2 is a bandpass filter, and therefore a "1" is entered in Fig. IV.7 to specify the filter function. This action in turn calls up a subroutine for filter design that allows for flexible specification of classic IIR filters. 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. In the example network, block # 2 is a fourth-order Butterworth bandpass filter with break frequencies at 100 and 400 Hz. This filter is fully specified (and implemented) with the dialog shown in Fig. IV.8. When responses have been entered for all the specification queries, the poles and zeros of the filter are computed and feedback is provided on its z-plane singularities. The typeout of singularities for the filter of block # 2 is shown in Fig. IV.9. Next, a menu is presented for display of filter characteristics or for a return to the main program (for specification of another block). This menu is shown in Fig. IV.10. As can be seen, the menu allows the investigator to call routines for the display of the steady-state, phase,

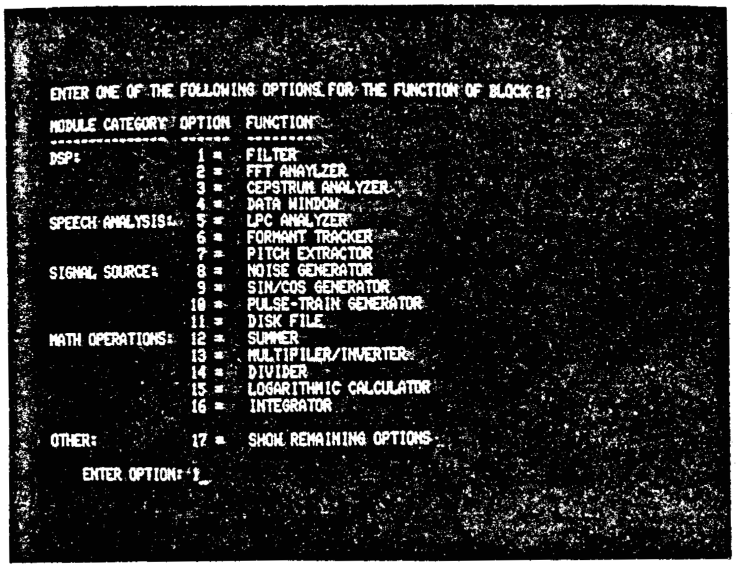

Fig. IV.7. First main menu of the DESIGN program. The option entered at the bottom indicates that the function of block # 2 is to filter an input signal.

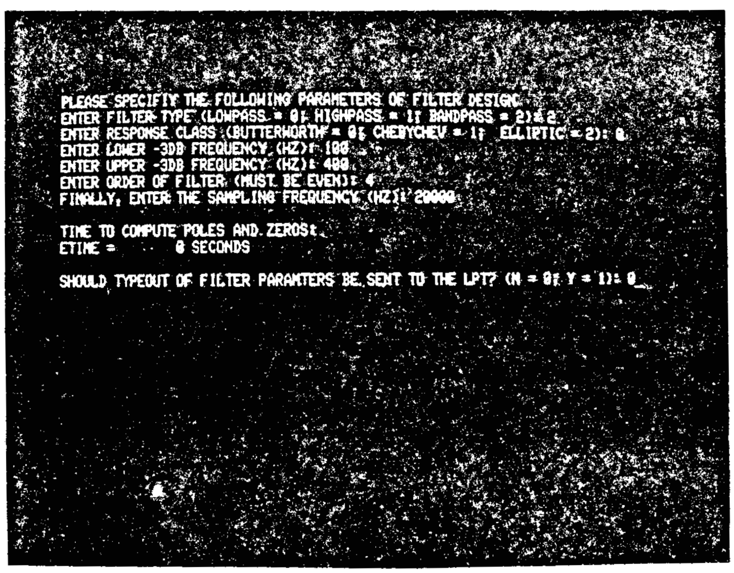

Fig. IV.8. Dialog for the full specification of a fourth-order bandpass filter with break frequencies at 100 and 400 Hz.

impulse or step responses of the filter. In addition, the filter design can

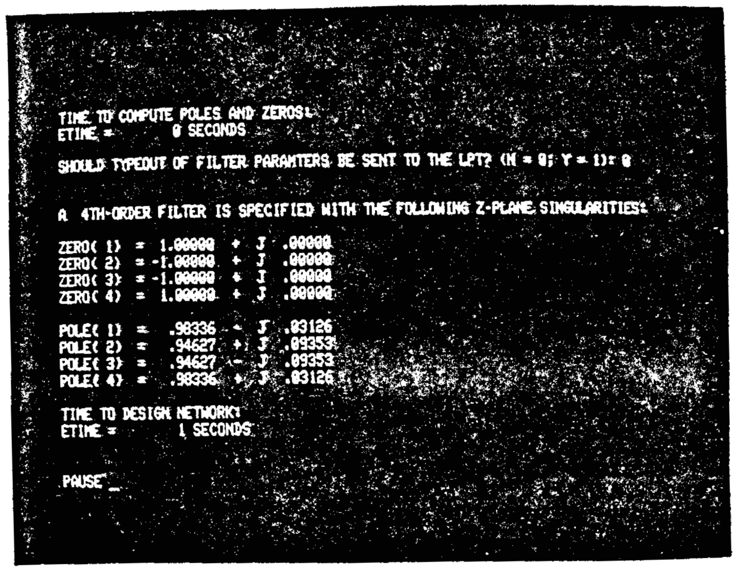

Fig. IV.9. Typeout of z-plane singularities for the filter specified in Fig. IV.8.

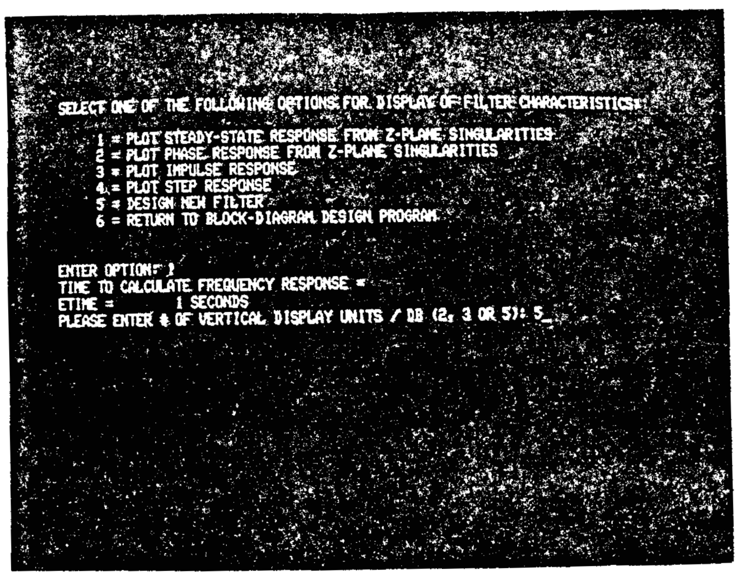

Fig. IV.10. Menu of options for the display of filter characteristics or for a return to the main DESIGN program. The option taken, along with the subsequent dialog, calls for a plot of the filter's steady-state magnitude response.

be revised by taking option 5 to "DESIGN NEW FILTER." The highly interactive nature of communication with the filter design subroutine provides the investigator with a powerful facility for specifying exactly the desired characteristics of filter response. This type of interactive communication is a general feature of the DESIGN program, and is used in the specification of all complex functions indicated in the main menus of Figs. IV.4 and IV.16.

A typical series of displays an investigator might ask the filter subroutine to produce is illustrated in Figs. IV.11 through IV.13. First, the responses to the queries shown at the bottom of Fig. IV.10 indicate that a plot of the steady-state magnitude response is desired, and that the ordinate is to have 5 display units per dB of attenuation. The resulting plot is presented in Fig. IV.11. A log-log graph of the response is drawn, so that the investigator can easily appreciate the rates of falloff in the skirts of the filter in terms of dB/decade of frequency. Finally, an option to copy the display for a permanent plot on the line printer is presented; this copy query is made for all graphs drawn by the DESIGN program.

Once the response to the copy query is entered (and the copy made if the response is a "1"), the menu for "display characteristics or return to the main program" is presented again, as shown in Fig. IV.12. This allows the investigator to examine other characteristics of the filter response or to refine the filter design before returning to the main program. In Fig. IV.12, the option to display the impulse response of the filter is taken. The dialog following the specification of the impulse response establishes various attributes of the display. For example, the time axis can be manipulated (i.e., compressed) by skipping points in the display. To assist the investigator in setting the amplification factor and y position of the trace, the program provides feedback on the maximum and minimum values in the calculated impulse response. When the responses to all queries presented in the dialog of Fig. IV.12 have been entered, the impulse response is drawn, as shown in Fig. IV.13. As before, a copy query is presented with the display so that a copy can be made if desired.

The final step in the specification of block # 2 is illustrated in Fig. IV.14. This is the same menu presented in previous figures for "display characteristics or return to the main program," but in this case the option

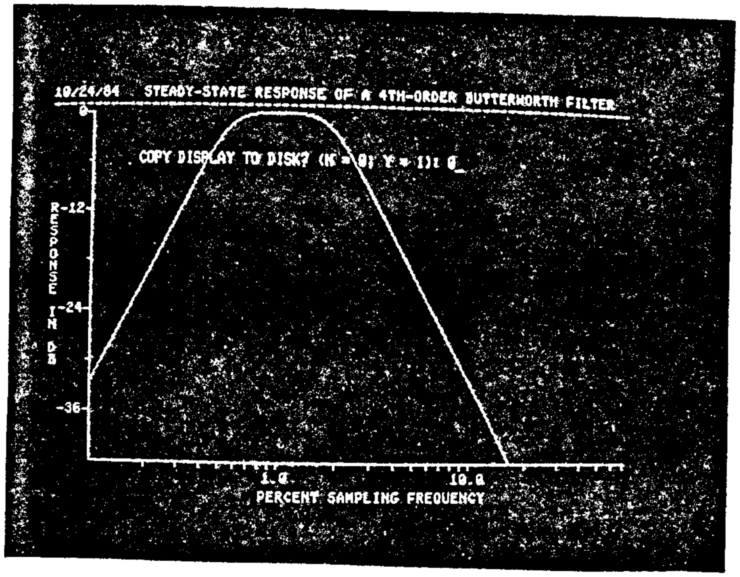

Fig. IV.11. Plot of the steady-state magnitude response of the filter specified in Fig. IV.8.

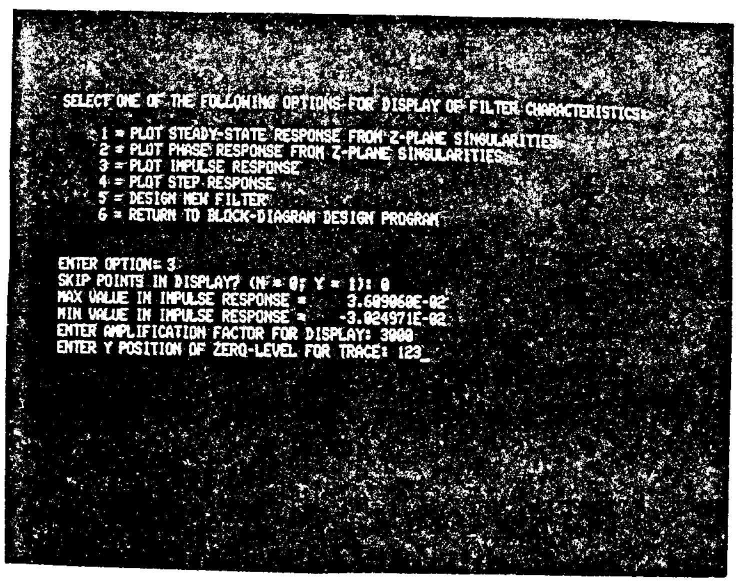

Fig. IV.12. Menu of options for the display of filter characteristics or for a return to the main DESIGN program. The option taken, along with the subsequent dialog, calls for a plot of the filter's impulse response.

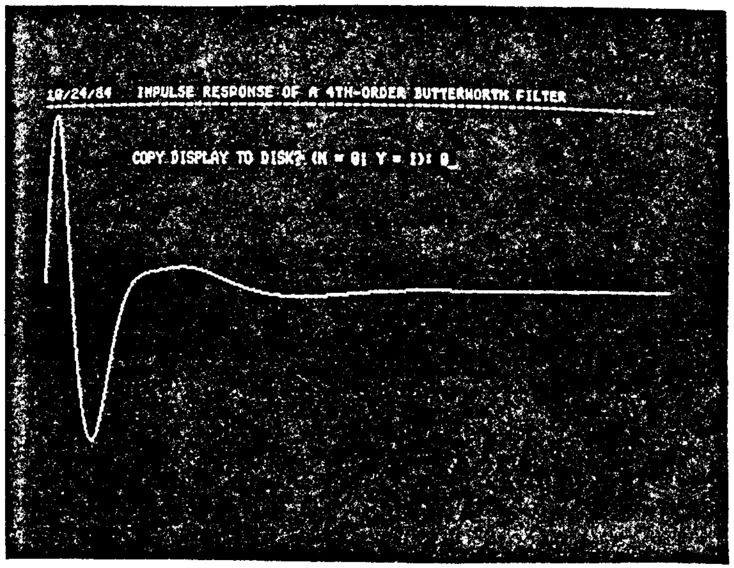

Fig. IV.13. Plot of the impulse response of the filter specified in Fig. IV.8.

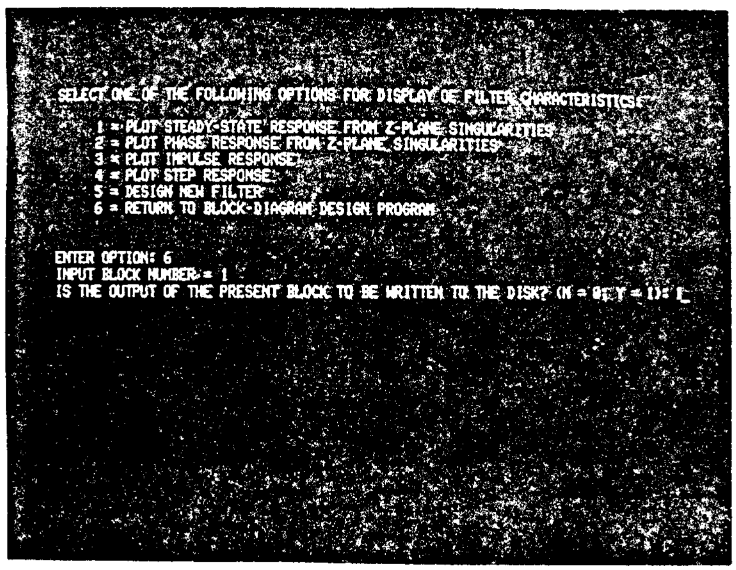

Fig. IV.14. Menu of options for the display of filter characteristics or for a return to the main DESIGN program. The option taken, along with the subsequent dialog, calls for a return to the main DESIGN program. The dialog also specifies the input to the filter and establishes that the output of the present block is to be written to the disk during system simulation.

to return to the main program is taken. Unlike the signal source function specified for block # 1, the filter must have an input from one of the outputs of another block in the network. For the system defined up to this point, the only possibility is the output of block # 1. Any other entry would produce an error message and the investigator would be asked again for the input block number. Valid entries of input block numbers establish the topology of the network in that they specify for the final system all interconnections between blocks. In general, blocks can be specified in any order (and not necessarily the left-to-right order of the present example) as long as an input can be identified for each block that requires one.

As mentioned before, the last query in the specification of all blocks is one whose response indicates whether the output (or outputs) of the present block is (or are) to be written to the disk. In the example of Fig. IV.14, the response is affirmative and therefore the output of block # 2 will be available at a later time for examination with program SHOWNTELL, or for electrical stimulation with programs ASNELEC and TEST.

Once the filter for block # 2 is fully specified as described above, the first main menu of the DESIGN program appears for initial specification of the next block in the network. Because the procedure for specifying the filters of blocks 3, 4 and 5 (see Fig. IV.3) is similar to the procedure just illustrated for block # 2, we will skip to the menu shown in Fig. IV.15, which is the screen that appears after the specification of block # 5. Thus, the investigator is now in a position to specify the function of block # 6 in the network. Reference to Fig. IV.3 indicates that the function of this block is to rectify the output of block # 2. However, rectification does not appear as an option in the first main menu of Fig. IV.15, so the second main menu is called by entering a "17" to show the remaining options. The second main menu which then appears is shown in Fig. IV.16. Options presented in this menu include those for specifying various circuit functions familiar to electrical engineers and for implementing various control functions of the DESIGN program. The control functions are listed under module category "OTHER", 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 of which is also stored on the disk; (3) identify a new user-defined rule to be added to the library of user-defined rules; (4) show the topology of the present system as defined by the

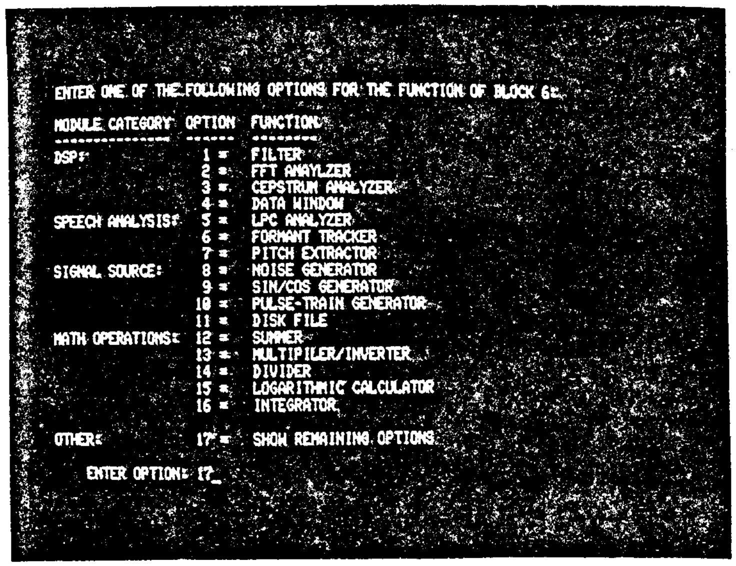

Fig. IV.15. First main menu of the DESIGN program. The option entered at the bottom indicates that the function of block # 6 is not displayed on this menu and therefore a request is made to show the second main menu, presented below in Fig. IV.16.

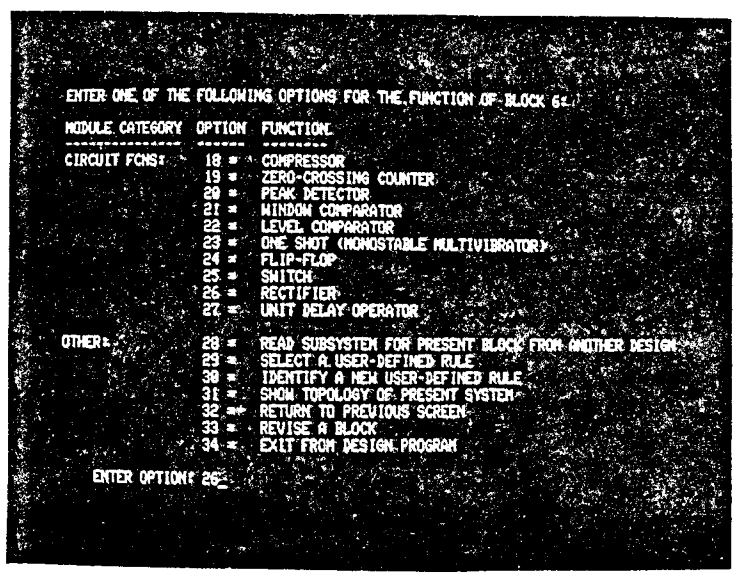

Fig. IV.16. Second main menu of the DESIGN program. The option entered at the bottom indicates that the function of block # 6 is to rectify an input signal.

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. As might be appreciated from the list just given, the first three control functions provide powerful tools for the specification of complex signal-processing systems. For example, the entire signal-processing system of Fig. IV.3 could be read as a single block into a larger and more complex signal-processing system using the command to "READ SUBSYSTEM FOR PRESENT BLOCK FROM ANOTHER DESIGN." Because many signal-processing systems can be configured from a few functional subsystems (such as the front end of a four-channel vocoder, as illustrated here), this command can save an enormous amount of the investigator's time in building different processors.

The other two options that are useful in the specification of complex systems are those for selecting and identifying a user-defined rule. These options allow the investigator to build and use functions that are not included in the main menus of the DESIGN program (Figs. IV.4 and IV.16). An example of one such function would be that of a microprocessor embedded in a signal-processing system. Specification and simulation of the logic decisions made by a microprocessor are straightforward with user-defined rules; specification and simulation of these same decisions would be very difficult without user-defined rules.

To return now to the specification of the example system, we note that the option for specifying a recitifier is "26", and is therefore entered at the bottom of the menu in Fig. IV.16. The queries presented after this entry has been made are indicated in Fig. IV.17. As can be appreciated from the short list of questions, full specification of a rectifier is much simpler than full specification of a filter. In the example, a half-wave rectifier that processes floating-point values is specified (the choice between floating-point and integer values is usually made on the basis of processing speed and the expected range of the data). Finally, as in the specification of other blocks, the input to the rectifier is identified and an indication is given on whether the output of the present block is to be written to the disk.

The procedures used for specifying the remaining blocks in the network are similar or identical to those outlined above. Once all blocks have been specified, the design data are stored on the disk by invoking option "34" in

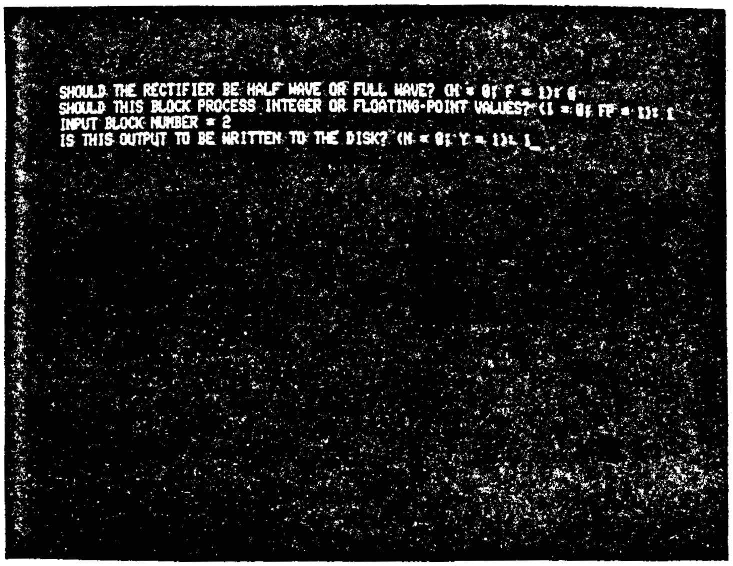

Fig. IV.17. Dialog for full specification of a half-wave rectifier that processes floating-point values.

the second main menu of the DESIGN program. This action is illustrated in Fig. IV.18. The command to exit from the DESIGN program first promts the investigator to label the design with a title and then calls into memory the program PREPARE. The function of PREPARE is to arrange the information stored in the design files in such a way that the time required for subsequent simulation of the network by program EXECUTE is minimized. Fig. IV.19 shows the screen that appears when PREPARE is doing its work. For a design of the complexity indicated in Fig. IV.3 (the example system), PREPARE requires about 10 seconds of computer time to run.

When PREPARE is finished, CPEXEC is again invoked and the investigator can call any other program in the set listed at the beginning of this section. To simulate the example network just designed, for example, an entry of "3" would be made in response to the CPEXEC menu as shown in Fig. IV.20. This entry calls program EXECUTE which first asks for the number of the design to be simulated and then asks whether the entire input file(s) or only of segment of the input file(s) should be processed. At various points in the simulation, feedback is provided by EXECUTE to inform the investigator of processing status. The two queries and initial feedback from EXECUTE are shown in Fig. IV.21.

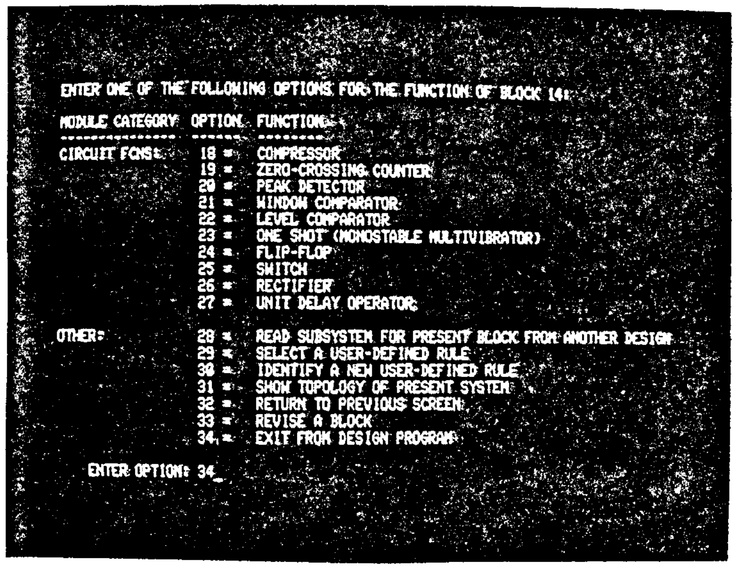

Fig. IV.18. Second main menu of the DESIGN program. The option entered at the bottom indicates that the specification of the present design is complete, and that an exit should be taken from the DESIGN program.

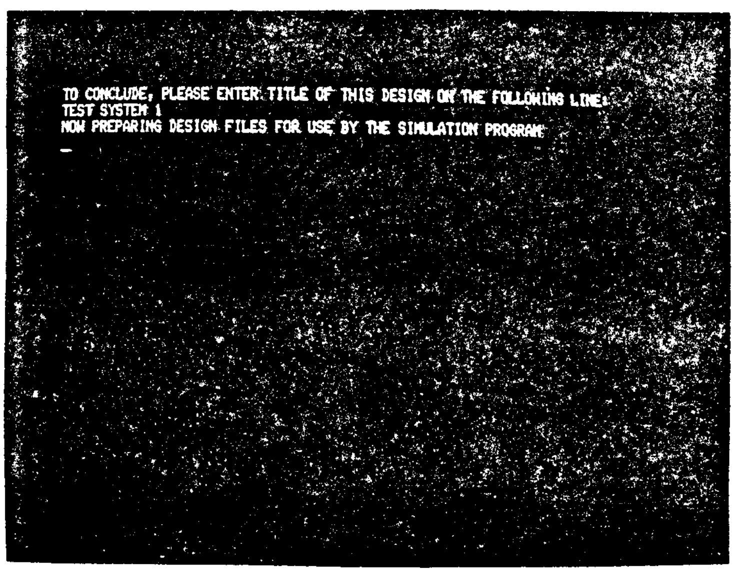

Fig. IV.19. Screen that appears during the execution of program PREPARE.

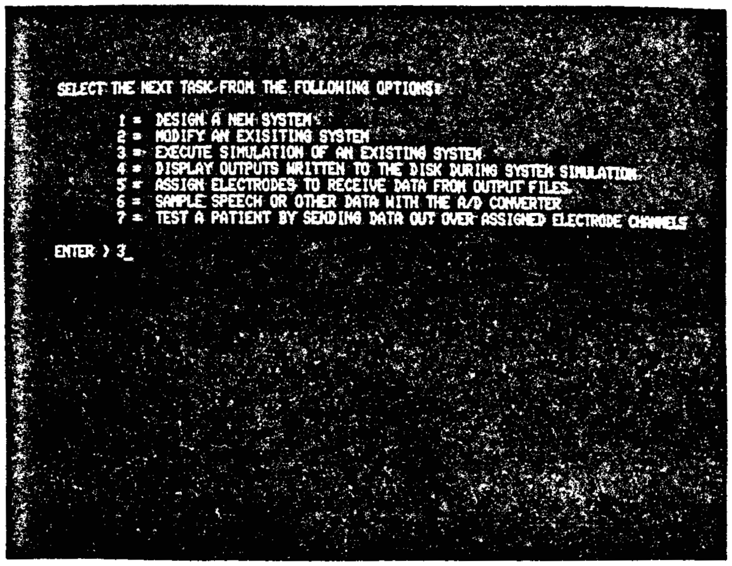

Fig. IV.20. Menu presented by CPEXEC. The option entered at the bottom indicates that the next program to be called is EXECUTE (for the simulation of a signal-processing system).

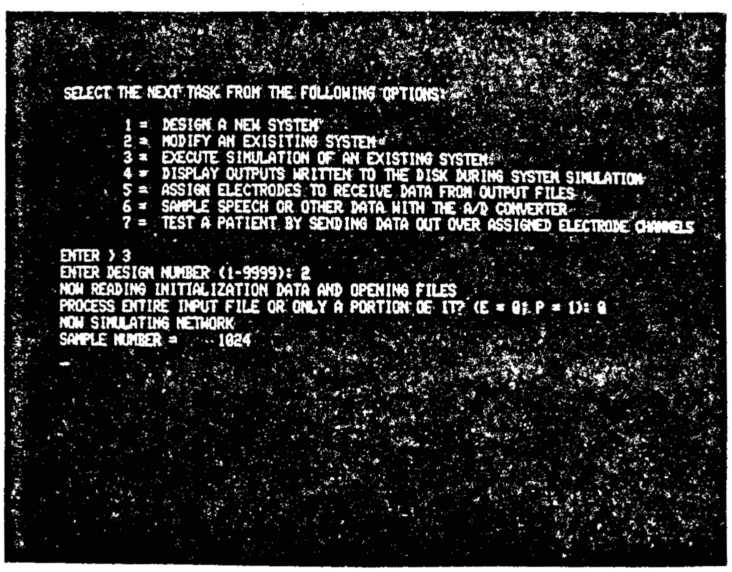

Fig. IV.21. CPEXEC menu followed by the initial dialog and feedback of program EXECUTE. The dialog indicates that design # 2 is to be simulated, and that the entire input file is to be processed.

Among the programs in the set, EXECUTE is by far the most complex in terms of the computer resources it must handle. First, data files and system design files are all stored in the extended memory of the Eclipse to minimize disk input/output operations during simulation. Next, the logic controlling the order in which blocks are simulated, the length of data segments handled in each "chunk" of processing, and number of passes through feedback loops that lack memory elements is also carefully tailored to optimize the speed of execution. Finally, disk I/O operations that must be made at "chunk intervals" are coded for rapid transfer of data in 256-, 512- or 1024-word blocks, again to speed up the simulation of complex signal-processing systems.

As indicated in Fig. IV.22, the total number of 1024-word pages used in extended memory for the example system is 14, and the total time required for processing 7168 input samples is 15 seconds. This time compares quite favorably with the performance of other simulation programs that are designed to run on general-purpose minicomputers. For example, simulation of the same system using the Interactive Laboratory System of Signal Technology, Inc. requires 280 seconds on the Eclipse. For a 10 kHz sampling rate, then, these figures correspond to 21 times real time for the RTI computer-based simulator and 391 times real time for the ILS simulator. The speed advantage of the RTI simulator is important for the present application in that it (1) allows for the evaluation of different processing strategies within single sessions with a patient and (2) allows for the processing of all tokens in the "miniMAC" test by several different processing strategies in overnight runs.

Once simulation of the network is completed, program EXECUTE closes open files on the disk and then pauses so that the investigator can examine the feedback provided during the simulation (see Fig. IV.22). When the investigator is ready to proceed, he or she strikes any key on the terminal and program CPEXEC is called. The menu for selecting the next task is then presented once again, as illustrated in Fig. IV.23. To display the output signals written to the disk during the previous simulation, an entry of "4" is made in Fig. IV.23 to call program SHOWNTELL. Figs. IV.24 through IV.28 show typical displays drawn by this program. The first display is of the input disk file "TUNA". As is evident from the waveform (see Fig. IV.24), file TUNA is a linear frequency sweep, the limits of which encompass the

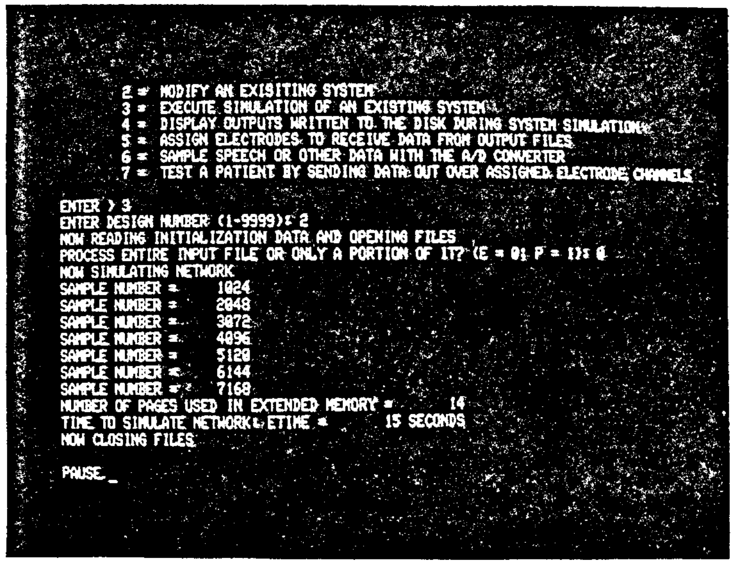

Fig. IV.22. Complete display of the feedback provided by program EXECUTE for the example system. The display indicates that the total number of 1024-word pages used in extended memory for the simulation was 14, and that the total time required for processing 7168 input samples was 15 seconds.

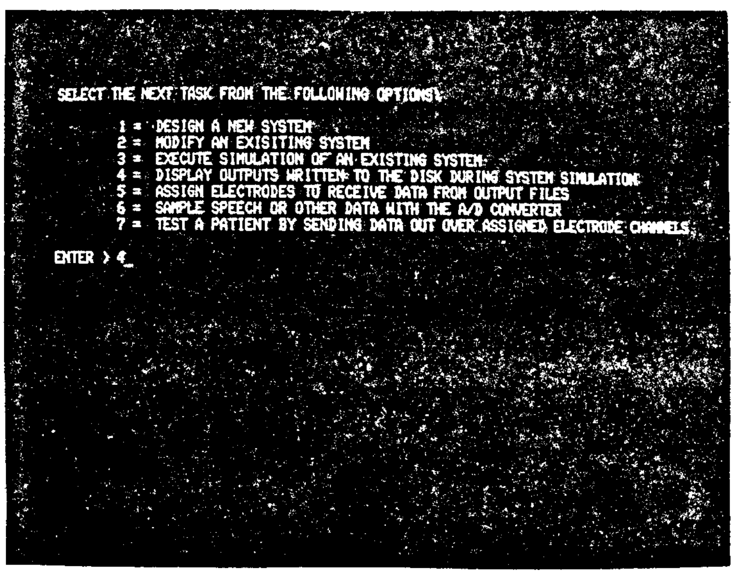

Fig. IV.23. Menu presented by CPEXEC. The option entered at the bottom indicates that the next program to be called is SHOWNTELL (for the display of outputs written to the disk during system simulation).

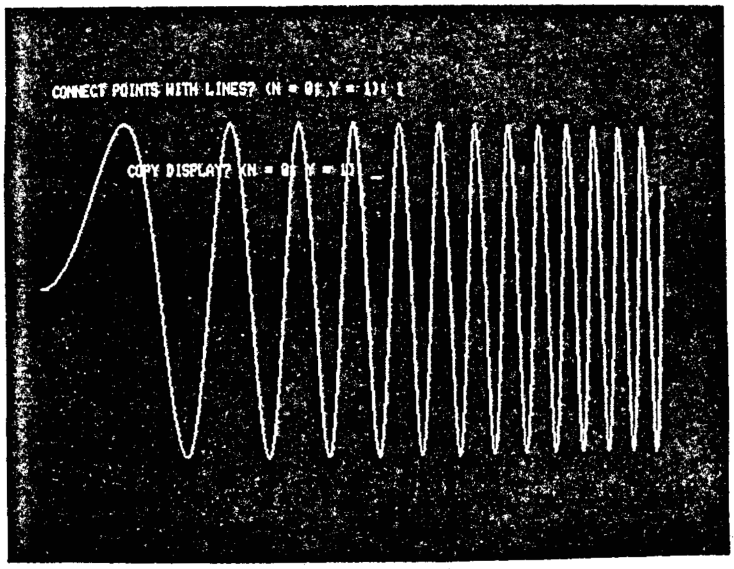

Fig. IV.24. Display of the output of block # 1 in the example network of Fig. IV.3.

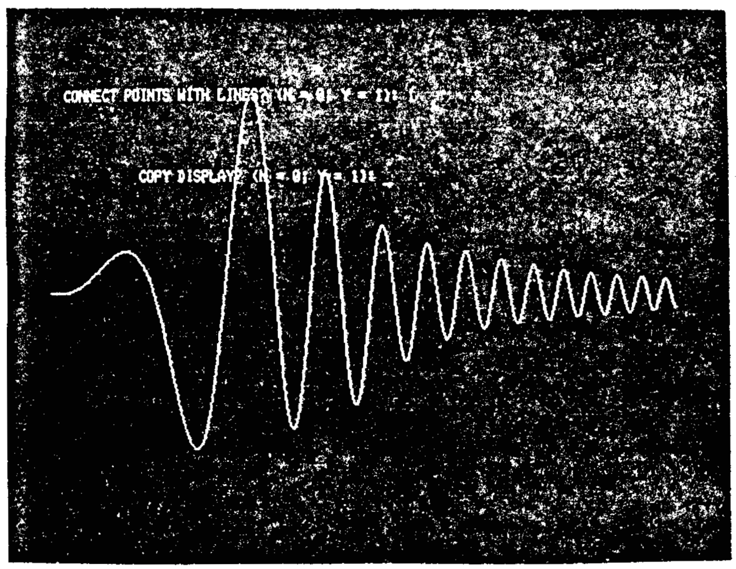

Fig. IV.25. Display of the output of block # 2 in the example netowrk of Fig. IV.3.

break frequencies of the bandpass filters in the example network.

Only the initial 512 data points of TUNA are shown in Fig. IV.24. SHOWNTELL also has facilities for displaying other portions of the file;

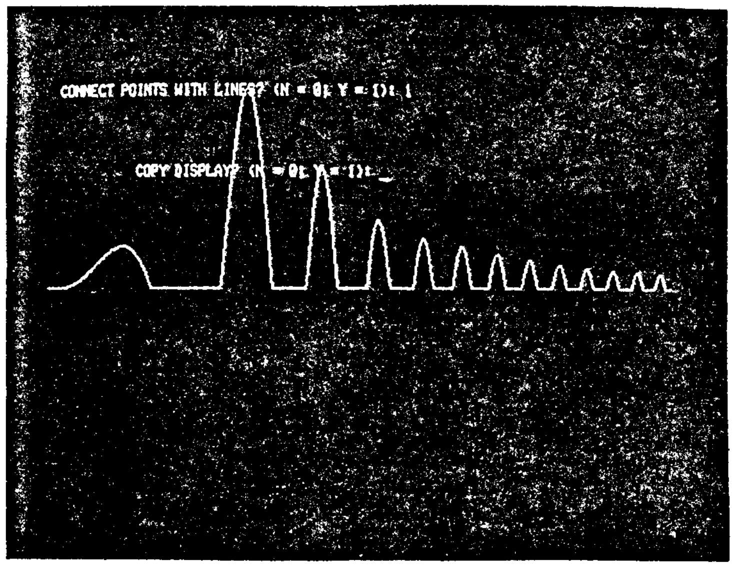

Fig. IV.26. Display of the output of block # 6 in the example network of Fig. IV.3.

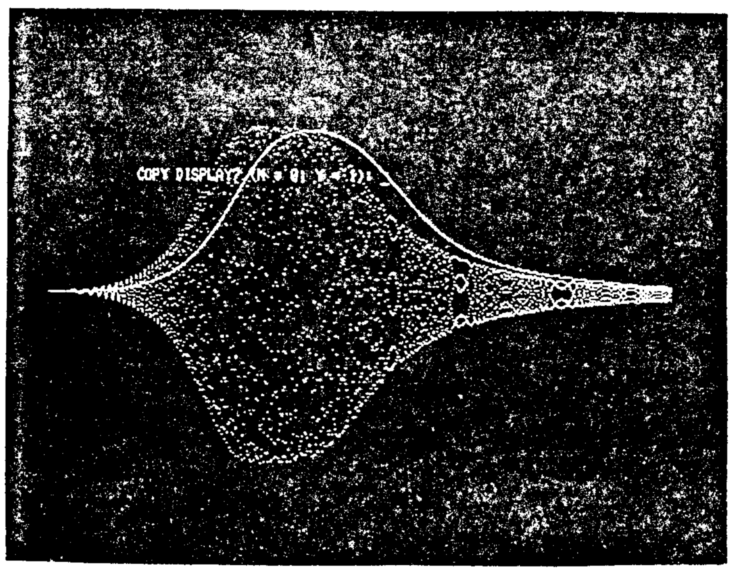

Fig. IV.27. Display of the outputs of blocks 5 and 13 in the example network of Fig. IV.3.

manipulating the number of points displayed; manipulating the y-axis scale factor of the display; interpolating or downsampling points in the display; and presenting data in multitrace and/or multioutput displays. Also, as

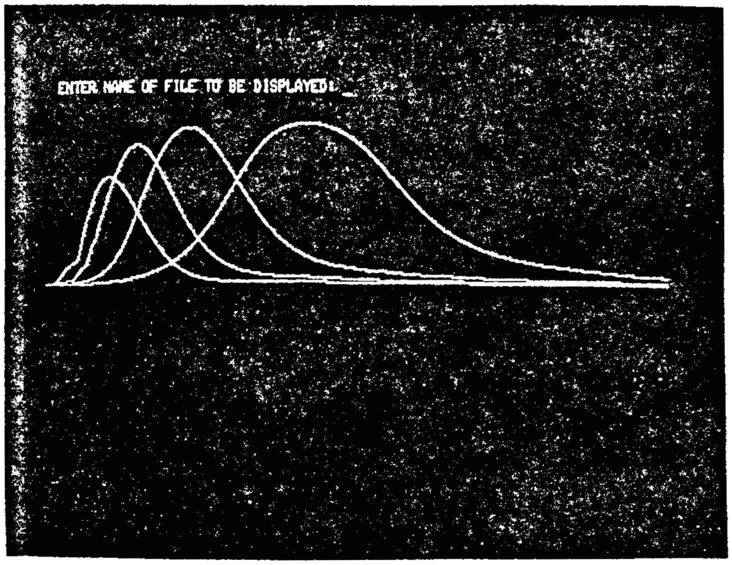

Fig. IV.28. Display of the outputs of blocks 10 through 13 in the example network of Fig. IV.3.

indicated in the initial description of SHOWNTELL on p. 17, data held in files can be presented as acoustic signals using the D/A converter.

The outputs of key signal-processing blocks in the example network are displayed in Figs. IV.25 through IV.28. Fig. IV.25 shows the output of block # 2, a fourth-order Butterworth bandpass filter with break frequencies at 100 and 400 Hz. As expected, the response of the filter is greatest when the instantaneous frequency of the input sweep falls within the filter passband.

The remaining displays are also consistent with the functions of the blocks in the network. For example, Fig. IV.26 shows the output of block # 6, which is the rectified waveform of the output shown above for block # 2. Another example is presented in Fig. IV.27. This display shows the first 2560 points of the outputs of blocks 5 and 13, where the smooth curve corresponds to the output of block 13. Clearly, the output of block 13 represents well the time course of rms energy present in the band from 1600 to 3000 Hz. The delay of output 13 relative to output 5 is a consequence of the sluggish response time (for this range of input frequencies) of the 30-Hz lowpass filter in block 13. Finally, Fig. IV.28 is a composite of the outputs of all four channels in the example network. The trace with the earliest peak is the output of block # 10; the trace with the next earliest peak is the output of block # 11; the trace with the third earliest peak is the output of block # 12; and the trace with the latest peak is the output of block # 13, as previously shown in Fig. IV.27. This progression of responses reflects the time course of frequency changes in the input sweep.

V. Plans for the Next Quarter

Our plans for the next quarter were outlined in section II of this report, "Overview of First-Year Effort." Briefly, they include patient testing at UCSF and beginning single-unit experiments, also at UCSF, to evaluate predictions of the integrated field-neuron model. In addition, we plan to install the integrated field-neuron model in the computer system of the National Biomedical Simulation Resource (NBSR) at Duke. This system consists of an AD-10 digital simulation computer (manufactured by Applied Dynamics International) connected to a VAX-11/750 host computer (Digital Equipment Co.). The AD-10 is a special-purpose, differential equation processor that can solve the Hodgkin-Huxley relations in real time. This speed of execution is about 300 times greater than that of our Eclipse computer. Thus, with a modest amount of programming effort to transform our present code for the system at Duke, we hope to enhance tremendously the utility of our integrated field-neuron model. All the necessary arrangements have been made with the staff of the NBSR so that we can use the resource free-of-charge for the present project.

Appendix 1

Abstracts for the ARO Midwinter Research Conference

"A Computer-Based Simulator of Speech Processors for Auditory Prostheses"

and

"An Integrated Field-Neuron Model of Electrical Stimulation by Intracochlear Scala-Tympani Electrodes"

A COMPUTER-BASED SIMULATOR OF SPEECH PROCESSORS FOR AUDITORY PROSTHESES.

*B.S. Wilson and C.C. Finley. Neuroscience Program Office, Research Triangle Institute, Research Triangle Park, NC 27709.

As part of a project to design and evaluate speech processors for multichannel auditory prostheses, we have developed a computer-based system for the rapid and flexible emulation of promising coding strategies in software. Use of this system allows us to make valid comparisons between many different approaches to processor design in tests with single subjects. In this way, controls are provided for inter-subject differences in pathology (i.e., differences in the densities and loci of surviving neurons and possible differences in the integrity of central auditory structures), the type of electrode array used, and apposition of individual monopolar or bipolar-pair electrodes to excitable tissue. These differences among subjects, along with differences in testing procedures among laboratories, have made interpretation of results obtained in previous tests a difficult and largely unrewarding exercise.

The software for the computer-based simulator of speech processors includes the following programs:

CPEXEC	-- executive program for managing communications between and execution of other programs in the set;
DESIGN	-- program for design of a signal-processing system, in which the user specifies the function and topology 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 signal-processing systems;
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 between our Eclipse 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.

In this presentation we will describe these programs in greater detail and provide examples of their use.

(Supported by NIH contract N01-NS-2356, "Speech Processors for Auditory Prostheses.")

AN INTERGRATED FIELD-NEURON MODEL OF ELECTRICAL STIMULATION BY INTRACOCHLEAR SCALA-TYMPANI ELECTRODES.

*C.C. Finley and B.S. Wilson, Neuroscience Program Office, Research Triangle Institute, Research Triangle Park, N.C. 27709.

One key element in the design of a cochlear prosthesis is knowledge of the "electrical-to-neural transformer" that characterizes the properties of neural responses to stimuli delivered by intracochlear electrodes. These response characteristics are dependent upon a variety of factors including the physical locations, dimensions and electrical characteristics of the implanted electrodes, as well as the survival patterns and physiological integrity of the remaining neural elements. In an effort to understand the complexities of intracochlear electrical stimulation, we have developed an integrated field-neuron model which allows computation of the potential gradients along the locus of a neural element and calculation of the resultant neural response to an electrical stimulus delivered by an electrode or electrode pair of specified geometry and placement within the scala tympani.

The potential gradients are calculated by an iterative, two-dimensional, finite-element model of a cochlear cross-section, including the presence of a scala tympani electrode pair. The electrode pair represents the current UCSF bipolar electrode design, compressed into two dimensions (Loeb et al., Med. Biol. Eng. Comp. 21:241-254, 1983). Grid points in the model are 20 microns apart and the two-dimensional sheet is assumed to be 20 microns thick. Resistivities linking the finite elements are defined according to published values for the resistivities of tissues and fluids appearing the cross-section. The bipolar electrodes are defined as equipotential conductors mounted in an insulating carrier medium. Fixed voltages are assigned to the electrodes and the resultant field patterns are computed by iteration for the entire cross-section. Finally, the potential levels at points along the locus of the VIIIth nerve elements are extracted from the final field calculation.

These potential levels are the fed into a lumped-element model of a myelinated neuron. This model is a modification of McNeal's axon model (IEEE Trans. BME 23:329-337, 1976) of resistively-linked Frankenhauser-Huxley nodes. The modified model includes myelinated axon cable properties and allows stimulation of the neural model by voltage sources located externally along the course of the neuron. Mammalian node of Ranvier characteristics are used instead of the Frankenhauser-Huxley frog nodes. Nine active nodes are included, each separated by ten myelin segments. A system of simultaneous, nonlinear differential equations are solved iteratively to calculate the model's response to any arbitary stimulus waveform.

Initial modeling results include strength-latency, strength-duration, and latent-addition curves as a function of the bipolar electrodes' physical geometry and placement within scala tympani. Experience to date shows that the integrated field-neuron model will be a productive tool in the optimization of cochlear prosthesis electrode design, in exploring basic mechanisms involved in electrical stimulation of the cochlea, and utimately in defining the "electrical-to-neural transformer" characteristics which strongly influence the design of advanced speech processors.

(Supported by NIH contract NO1-NS-2356, "Speech Processors for Auditory Prostheses.")