Digital Signal Processing Applications in Cochlear-Implant Research

Joseph Tierney, Marc A. Zissman, and Donald K. Eddington

We have developed a facility that enables scientists to investigate a wide range of sound-processing schemes for human subjects with cochlear implants. This digital signal processing (DSP) facility—named the Programmable Interactive System for Cochlear Implant Electrode Stimulation (PISCES)—was designed, built, and tested at Lincoln Laboratory and then installed at the Cochlear Implant Research Laboratory (CIRL) of the Massachusetts Eye and Ear Infirmary (MEEI). New stimulator algorithms that we designed and ran on PISCES have resulted in speech-reception improvements for implant subjects relative to commercial implant stimulators. These improvements were obtained as a result of interactive algorithm adjustment in the clinic, thus demonstrating the importance of a flexible signal-processing facility. Research has continued in the development of a laboratory-based, software-controlled, real-time, speech-processing system; the exploration of new sound-processing algorithms for improved electrode stimulation; and the design of wearable stimulators that will allow subjects full-time use of stimulator algorithms developed and tested in a laboratory setting.

More than 300,000 people in the United States suffer from a profound hearing loss that conventional hearing aids cannot restore. In many of these cases, hearing may be improved through the use of a cochlear implant, which consists of an array of electrodes that is surgically implanted in the inner ear to excite the surviving auditory neurons. When this electrode array is stimulated by devices that convert acoustic waves (such as those from speech, music, or noise) to electrical stimulating signals, some functional hearing can be restored. To convert acoustic waves to electrical signals, the electrode stimulators in the cochlear implant must perform signal-processing operations that are more complex than the frequency-dependent amplification performed by conventional hearing aids.

The work described in this article was initiated as a Lincoln Laboratory Innovative Research Program (IRP) and performed collaboratively with researchers at the MIT Research Laboratory of Electronics (RLE) and the Massachusetts Eye and Ear Infirmary (MEEI). The aim of the proposed IRP project was to use the digital speech- and signal-processing expertise available at Lincoln Laboratory to develop a facility that would enable scientists to investigate a wide range of sound-processing schemes for human subjects with cochlear implants. This digital signal processing (DSP) facility—named the Programmable Interactive System for Cochlear Implant Electrode Stimulation (PISCES)—was designed, built, and tested at Lincoln Laboratory and then installed at the MEEI Cochlear Implant Research Laboratory (CIRL). New stimulator algorithms that we designed and ran on PISCES have resulted in speech-reception improvements for implant subjects relative to commercial implant stimulators. These improvements

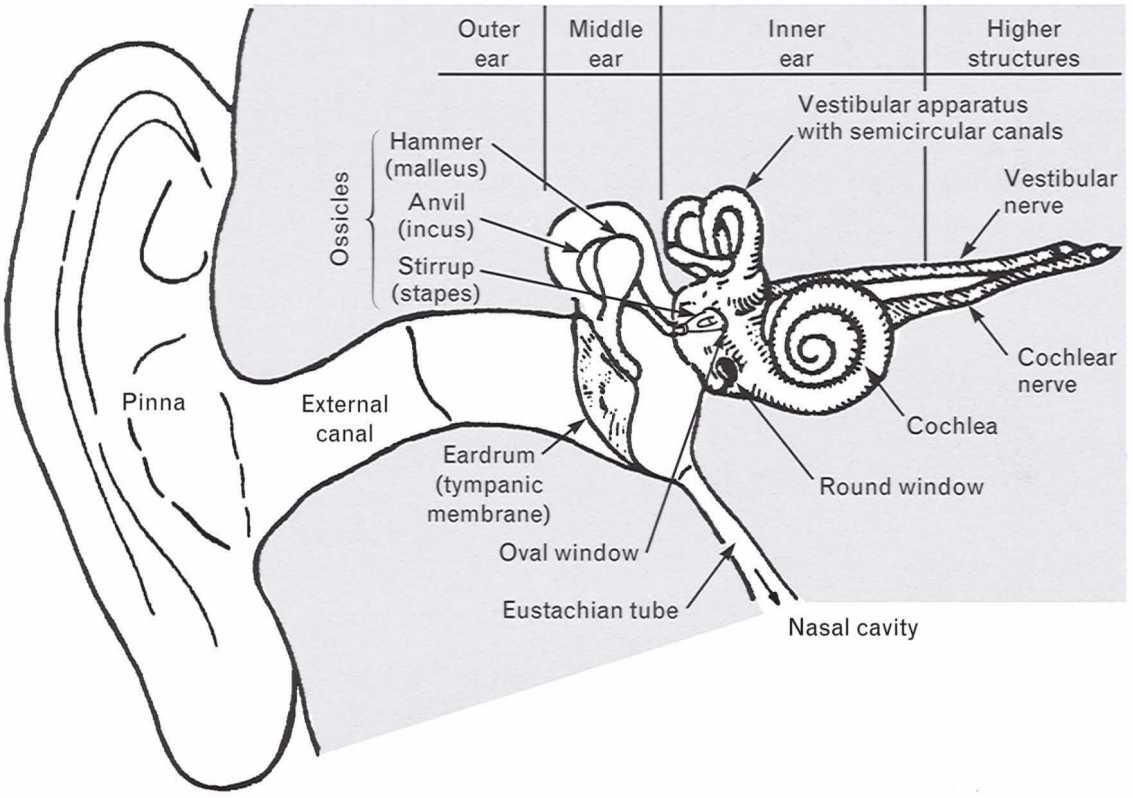


FIGURE 1. The human peripheral auditory system.

were obtained as a result of interactive algorithm adjustment in the clinic, thus demonstrating the importance of a flexible signal-processing facility.

(Note: In this article, the terms *electrode stimulator* and *implant stimulator* refer to all of the processing that converts an acoustic signal to a current source output used to drive an implant electrode. The term *processor* has been avoided because it can denote a laboratory-based computer, a microcomputer [DSP chip] within the laboratory computer, a portable analog or digital acoustic-to-current transducer, or an algorithm running in a digital signal processor [either laboratory-based or portable].)

We have used PISCES for additional applications beyond the running of new algorithms for subject interaction. Basic psychophysical measurements of cochlear-implant subjects have been performed with PISCES and, more recently, the facility has been used to explore parameter and design variations in the development of a portable, wearable stimulator device that subjects can use outside the laboratory. The facility provides powerful tools for research in cochlear-implant stimulation by enabling (1) the generation of test stimuli, (2) the analysis and display of the responses to these stimuli, and (3) the interactive control of parameter settings that affect the processing of the stimuli.

Research to extend the use of signal processing in cochlear implants and to design a new, more powerful and flexible wearable stimulator has continued beyond the IRP program by means of a National Institutes of Health (NIH) contract that has been awarded to the MIT/MEEI researchers. The particular tasks of the contract include continued work in the development of a laboratory-based, software-controlled, real-time, speech-processing system; the exploration of new sound-processing algorithms for improved electrode stimulation; and the design of wearable stimulators that will allow subjects full-time use of stimulator algorithms developed and tested in a laboratory setting.

This article summarizes the design, implementa-

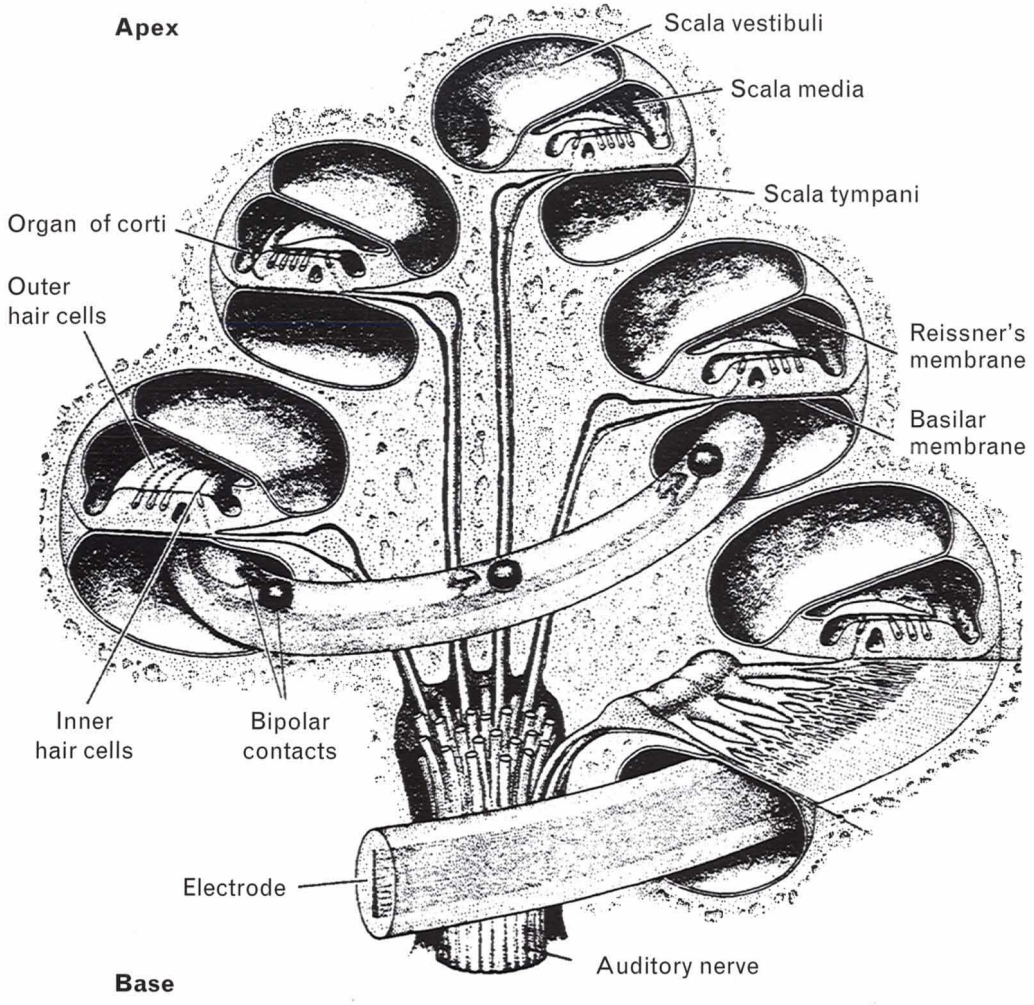


FIGURE 2. Cochlea with surgically inserted implant. (From "The Functional Replacement of the Ear," by Gerald E. Loeb. Copyright © February 1985 by Scientific American, Inc. All rights reserved.)

tion, and testing activities of the IRP project, as well as some of the work that has continued at MIT and MEEI as a consequence of the IRP. The section "Cochlear Implants for Sensorineural Deafness" describes the cochlear implant and the conditions that make it necessary; "The PISCES System" outlines the PISCES hardware and software design, as well as the modes of operation available with the present facility; "Algorithms" details the algorithms that have been implemented on PISCES and used in clinical interactions; "Clinical Results" reports the results of testing subjects using PISCES as well as an upgraded version of PISCES; and "Conclusions" discusses the consequences of the IRP effort and the subsequent follow-on work.

Cochlear Implants for Sensorineural Deafness

In the normal peripheral auditory system, sound perception begins when an incident acoustic wave causes the eardrum to vibrate [1]. As shown in Figure 1, this vibration is coupled from the outer ear to the cochlea through three small bones (ossicles) of the middle ear. The cochlea is a helical structure surrounded by bone and filled with fluid. The cochlea's round window is an opening covered by a membrane that moves outward in response to an inward movement of the stapes, or stirrup, in the oval window. Figure 2 shows that the cochlea is divided along its length (approxi-

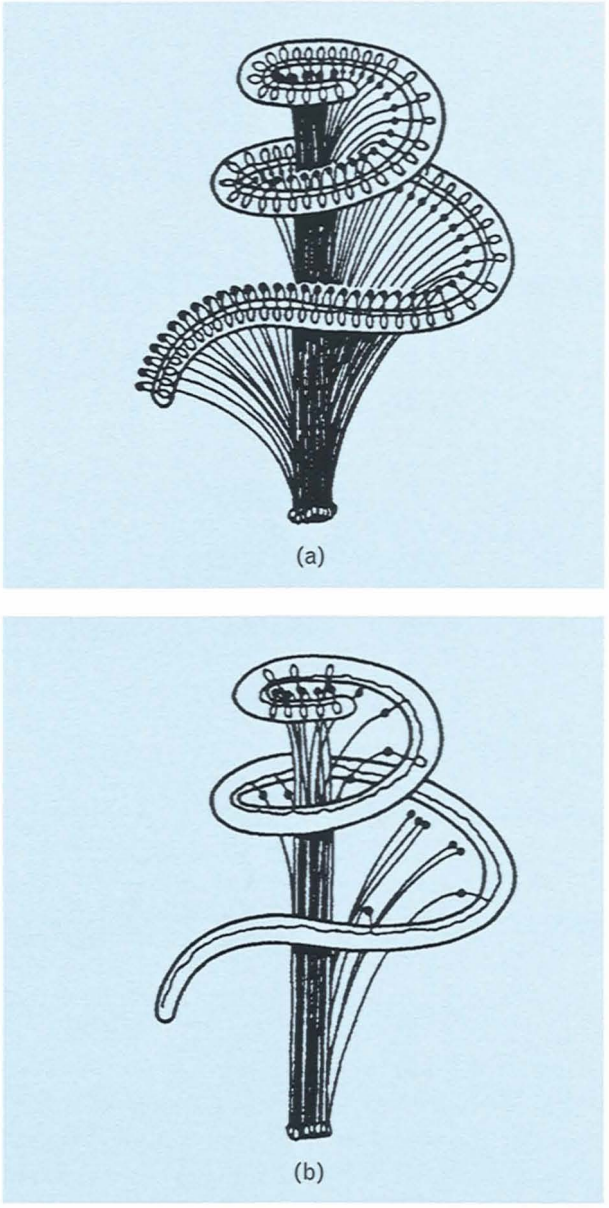


FIGURE 3. Pattern of nerve and hair cells for (a) a normal ear and (b) an ear with sensorineural impairment.

mately 32 mm) into three chambers—the scala vestibuli, the scala media, and the scala tympani—by Reissner's membrane and the basilar membrane. Because the oval window communicates with the scala vestibuli and the round window with the scala tympani, the two windows are on opposite sides of the basilar membrane, and motion of the oval window produces a pressure difference across this membrane.

For sinusoidal acoustic stimulation, the response of the basilar membrane takes the form of a traveling wave that begins at the cochlear base (the end near the oval and round windows) and moves toward the apex. Depending on the frequency of the stimulus, the position of maximum displacement of the basilar membrane will shift along the membrane's base-to-apex (longitudinal) dimension, with the highest frequencies causing maximum displacements near the base and the lowest frequencies causing maximum displacements toward the apex.

Displacement of the basilar membrane causes hair cells in the organ of Corti (see Figure 2) to release a neurotransmitter that excites the adjacent auditory nerve fibers. One of the basic cues that the brain uses to extract information from the activity of the 30,000 auditory nerve fibers is the location of active neurons on the basilar membrane—neurons that innervate hair cells near the base of the basilar membrane signal high-frequency energy while more apical neurons signal energy in the lower frequencies.

One common form of hearing impairment is called conductive because it is associated with the deterioration of the middle-ear ossicles [2] and results in signal attenuation between the acoustic input and the basilar membrane. This condition is sometimes treated with conventional hearing aids that use gain to overcome the attenuation, or with surgery that substitutes ossicular prostheses to reduce the unwanted attenuation.

Another type of hearing impairment is called sensorineural and results from the loss of hair cells and/or their associated neurons. Figure 3 illustrates the kind of differences we might expect between a normal ear and one with a profound sensorineural impairment. In the figure, hair cells are represented by open loops on the basilar membrane, cell bodies are shown as filled dots, and nerve fibers are depicted by the lines connecting the hair cells and cell bodies and joining as a bundle to represent the auditory nerve. This form of hearing loss may result from acoustic trauma caused by exposure to loud sounds; physical trauma that causes fracture of the cochlea; ototoxic drugs; disease; or a genetic predisposition to hair-cell degeneration that usually manifests itself as a person ages. In the United States alone, more than 300,000 people suffer from profound sensorineural deafness.

Because people afflicted with sensorineural deaf-

ness are not able to translate the mechanical energy of sound into the nerve signals that the brain uses to hear, conventional hearing aids that merely amplify the sound intensity are of no benefit. In many cases of sensorineural deafness, a cochlear implant may be used instead. Cochlear implants are electronic devices that, like hearing aids, consist of a microphone connected to a sound processor that is worn externally. But, rather than producing an acoustic output, the sound processor of a cochlear implant produces electrical signals that can then be sent to an array of electrodes that has been implanted in the deaf person's cochlea [3]. These electrical stimuli, delivered to the cochlea by the electrode array, are designed to excite the remaining auditory nerve fibers. The goal of these systems is to elicit patterns of nerve activity that mimic the patterns produced in a normal ear.

The insertion of a cochlear implant is a surgical procedure that varies with both the type of electrode array to be used and the place at which the device is meant to establish electrical currents and fields. Early implants comprised a single electrode in the middle ear close to the cochlea; modern implants consist of an array of electrodes snaked through the round window into the scala tympani to interact with a wide range of remaining sensory neurons [4, 5]. Figure 2 shows such an electrode assembly inserted in the cochlea. The electrode assembly comprises multiple insulated wires, each connected to a conducting contact placed near the basilar membrane with the contacts distributed along the length of the membrane. The electrodes can be excited as balanced pairs or as monopolar electrodes with a common return.

The human subjects tested as part of this research have received the Ineraid cochlear-implant system [6] (Figure 4). Manufactured by Smith & Nephew Richards Inc. of Bartlett, Tennessee, the Ineraid system consists of a microphone housed in an earhook that is worn behind the ear like a conventional hearing aid. The signal from the microphone travels down a cable to the sound-processing electronics housed in a package that is worn on a belt. The four-channel sound processor produces a set of four electrical stimuli that are sent up the cable through the earhook and connector to the electrode array implanted in the cochlea. The electrode array consists of a bundle of six

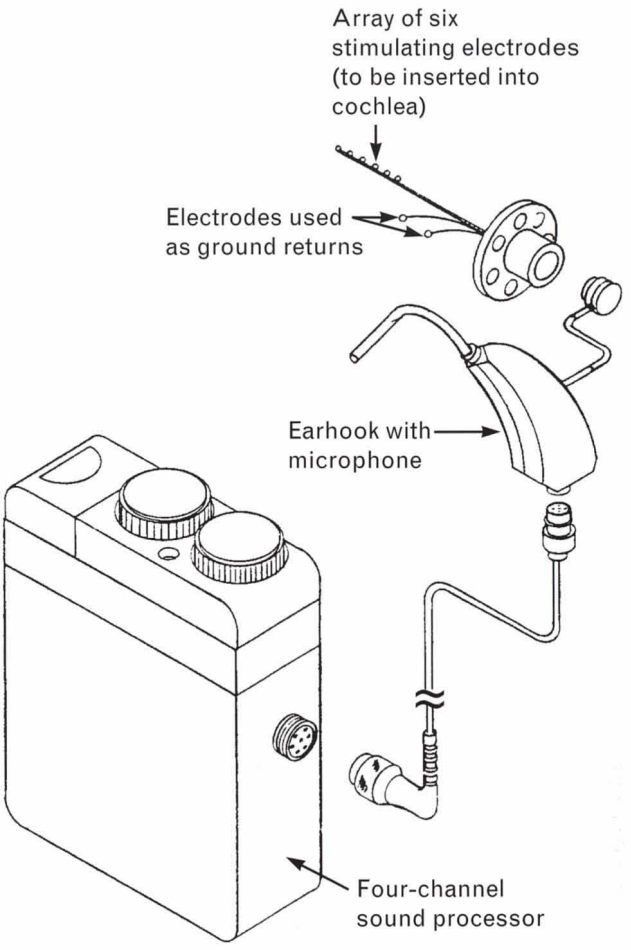


FIGURE 4. The Ineraid cochlear-implant system.

electrodes (0.4-mm-diameter balls spaced on 4-mm centers) that are inserted through the round window 20 to 24 mm into the cochlea. The array also includes two electrodes, placed outside the cochlea, that can be used as ground returns. All of the electrode leads terminate in a connector glued to the neck of a pedestal that is screwed to the skull behind the ear. The pedestal protrudes through the skin to give outside access to the electrodes. In our work at CIRL, we replaced the Ineraid stimulator with PISCES, the programmable, interactive system for cochlear-implant electrode stimulation.

The PISCES System

This section describes the PISCES hardware and software that are used to convert acoustic input to signals suitable for stimulating the electrodes of a cochlear implant. To test new stimulation strategies in labora-

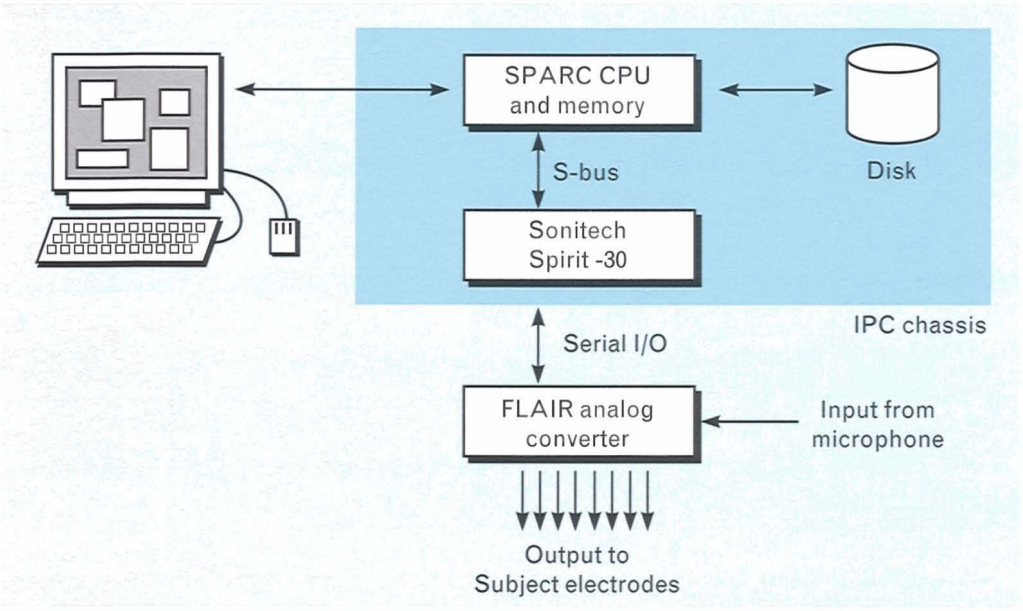


FIGURE 5. The Programmable Interactive System for Cochlear Implant Electrode Stimulation (PISCES). The system consists of a host computer—the Sun Microsystems SPARCstation IPC, a digital signal processing (DSP) board—the Sonitech Spirit-30, and an analog interface—the Flexible Lincoln Audio Interface (FLAIR).

tory experiments, we used PISCES, running the appropriate code, in place of the subject's hardware stimulator.

Hardware Configuration

A block diagram and photograph of the PISCES hardware are shown in Figures 5 and 6, respectively. The PISCES hardware includes a host computer, a DSP board, and an analog interface.

Host Computer. The host computer performs many functions. First, it serves as a general-purpose computer, performing non-real-time floating-point simulations of sound-processing algorithms. Second, it enables a user to review graphically the outputs of a given stimulation algorithm both for debugging purposes and for comparison with other algorithms. Additionally, when the host computer controls a special-purpose DSP board, the computer's tasks include the downloading of the stimulation algorithms and parameters to the DSP board, the initializing and interrupting of board processing, and the passing of data between the user, the board, and the disk file system.

After considering UNIX workstations and IBM-PC/AT compatibles, we chose the Sun Microsystems SPARCstation IPC to serve as the PISCES host computer. The Sun/UNIX expertise accumulated by the Lincoln Laboratory and MEEI personnel prior to the start of our research heavily influenced our selection of the IPC even though DSP boards were much more widely available for the IBM-PC/AT platform. The IPC is an inexpensive, 4.2-MFLOPS UNIX workstation with a 25-MHz SPARC integer and floating-point processor and spare S-bus slots for installing peripheral hardware. With regard to software, the IPC comes with SunOS (Sun's version of UNIX) and OpenWindows (Sun's version of X-windows), thus affording a multi-user, multi-tasking environment not commonly available on IBM-PC/AT compatibles.

DSP Board. Several DSP boards were commercially available for incorporation into the IPC. Each board contained a single DSP chip, fast memory, serial interfaces, and an S-bus interface to the IPC. We considered boards that contained the following DSP chips: the Motorola 56000, the AT&T DSP32C, and the Texas Instruments TMS320C30. Because the 56000 is a fixed-point processor, boards that used the 56000 were disqualified early on. Given the availability of fast floating-point processors, we chose to avoid

the complications of fixed-point arithmetic. The DSP32C was considered but disqualified because assembly-code programming of this chip requires an extensive understanding of the chip's pipelined architecture. Although we anticipated writing much of the software in the C programming language, it seemed inevitable that some assembly coding would be required, and the experience of Lincoln Laboratory personnel was that the DSP32C cannot be programmed easily in assembly language. Thus we decided on the TMS320C30—a floating-point chip that is easily programmed in assembly language. A factor that influenced our decision was that the TMS320C30 had already been employed successfully for other projects at Lincoln Laboratory.

At the onset of the IRP project in 1990, only one board vendor manufactured a TMS320C30-based product that was compatible with the IPC—the Sonitech Incorporated Spirit-30 S-bus card. (Note: Subsequently, Loughborough Sound Images introduced an S-bus-based TMS320C30 card that is quite similar to the Spirit-30.) The Spirit-30 comprises a 33-MFLOPS TMS320C30, an S-bus interface, 2 Mbytes of zero-wait-state RAM, and two serial ports. Furthermore, the Spirit-30 supports the SPOX operating system (see the following subsection, "Software Environment"), which eases the software migration from non-real-time workstation simulations to real-time DSP board implementations. Figure 7 contains a block diagram of the Spirit-30 card.

Analog Interface. To provide analog-to-digital (A/D) and digital-to-analog (D/A) capability for the TMS320C30 chip used on the Sonitech board, we designed the Flexible Lincoln Audio Interface (FLAIR) board to connect to the serial input/output port of the TMS320C30.

The FLAIR board provides a two-channel A/D input stream by using a CS5336 16-bit delta-sigma modulation converter manufactured by Crystal Semiconductor Inc. As the chip performs oversampling followed by digital filtering and downsampling, it achieves a high signal-to-noise ratio over a wide range of sampling rates. This approach eliminates the need for external anti-aliasing filters.

The D/A output from the FLAIR board is provided by Burr Brown PCM56P 16-bit converters that have a settling time of 1.5 μsec. For each pair of D/A converters, a Nippon Precision Circuits SM5813AP upsampling finite impulse response (FIR) filter chip may be engaged if desired. Located between the TMS320C30 DSP chip and the D/A converters, each SM5813AP chip can provide two channels of 1:8 upsampling and the associated digital low-pass filtering. Such upsampling and filtering prior to D/A conversion eliminate the need for a sharp analog smoothing filter. The converters can also be driven directly from the TMS320C30 DSP chip (thus bypassing the upsampling and filtering stages) to generate pulse signals

FIGURE 6. PISCES hardware.

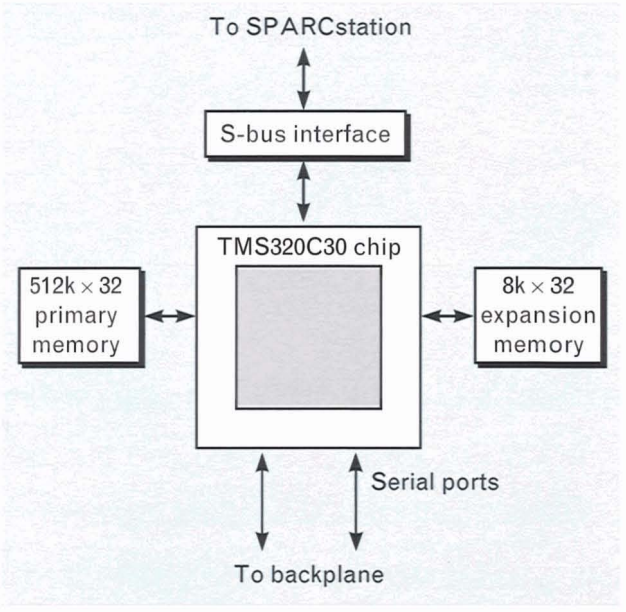


FIGURE 7. The Sonitech Spirit-30 card used in PISCES.

at the sampling interval width.

The FLAIR board assembly that is used in PISCES provides two channels of 16-bit A/D input and eight channels of 16-bit D/A output. The sampling rate, the use of upsampling or direct outputs, and the number of A/D and D/A channels are specified under software control. The hardware assembly can be easily extended to provide additional D/A outputs.

Software Environment

The development of implant-stimulation software on PISCES generally takes advantage of the three processing modes shown in Figure 8 and described below. Detailed descriptions of the actual stimulation algorithms are described in the section "Algorithms."

Step 1: Host File-to-File Mode. The host file-to-file mode allows a researcher to explore a range of signal-processing algorithms for electrode stimulation. Through the processing of test files (such as sums of sine waves, noise, chirp signals, and speech), the algorithms can be debugged and evaluated under tightly controlled conditions. All programs are written in C with the standard math and file I/O libraries. Programs are debugged with high-level debugging tools such as Sun's dbxtool—an OpenWindows-based, symbolic source-code debugger. Finally, the resulting multichannel output files are displayed with interactive waveform and spectrogram display packages such as Entropic Research Laboratory's waves+. The final result is a debugged C program that takes single-channel sampled data files as input and produces multichannel electrode-stimulation files as output.

Step 2: TMS320C30 File-to-File Mode. In the next step of the software development process, the C code is ported from the Sun host computer to the Spirit-30 board. An advantage of using the TMS320C30-based Spirit-30 board is the availability of the Spectron Microsystems SPOX operating system. Although Texas Instruments (TI) provides an ANSI C compiler for converting C code to TMS320C30 assembly code, SPOX eases the porting process by allowing the user to retain I/O with the host operating system through

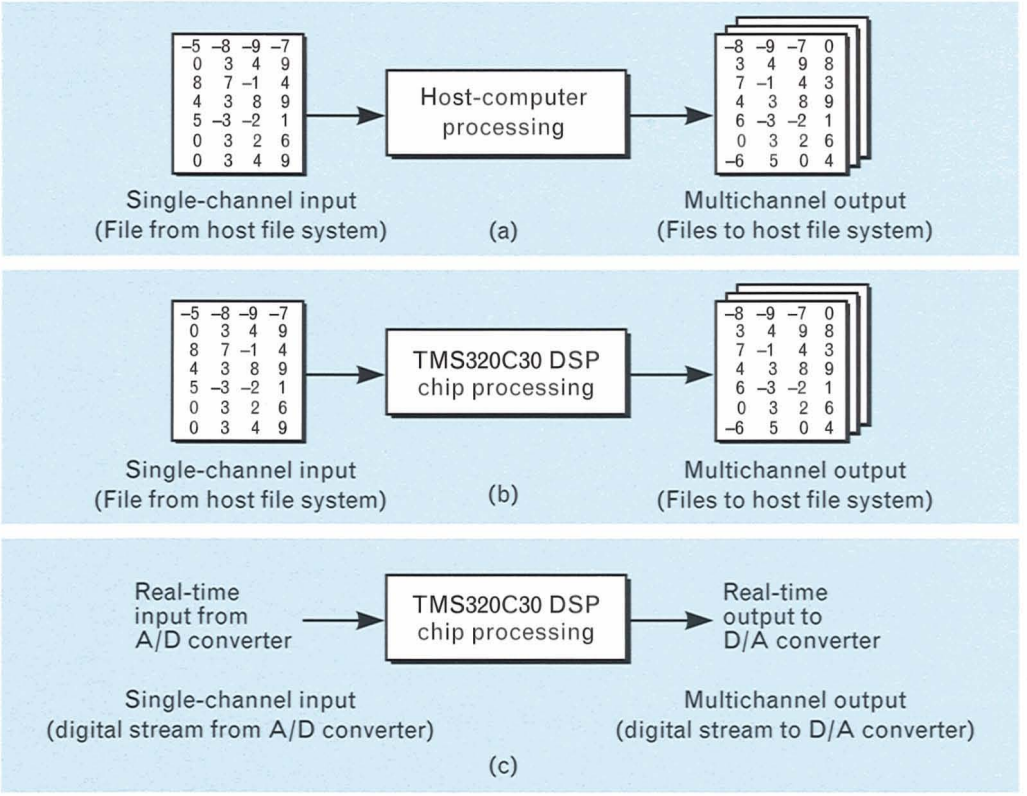


FIGURE 8. The three operating modes of PISCES: (a) host file-to-file mode, (b) TMS320C30 file-to-file mode, and (c) real-time mode.

the use of commonly used file I/O routines such as `fprintf`, `fwrite`, `fscanf`, and `fread`. In this manner, the file-to-file C program written for the Sun can be recompiled and run on the TMS320C30 with very few changes. (Careful attention, however, is required in memory management. Programmers are advised to use SPOX memory-allocation routines, which differ slightly from traditional C.) All interaction between the host computer and the TMS320C30 board required to effect transfer of data from disk to TMS320C30 is (from the user's point of view) performed invisibly. Using SPOX and comparing the output of the host file-to-file system with the TMS320C30 file-to-file system, the user can quickly identify any compiler or floating-point inconsistencies between the Sun and TI CPUs. Additionally, the TMS320C30 file-to-file mode eases the transition to real time by allowing the user to convert from C to TMS320C30 assembly language those subroutines containing critical loops that have been identified by the on-board TMS320C30 timer. Both the correctness and the effective speedup of these conversions to assembly language can be verified by using test files as input.

Step 3: Real-Time Mode. For the real-time mode, the C and assembly code that were tested in the TMS320C30 file-to-file mode are extended to support input from and output to the FLAIR board in addition to the host file system. Prior to entering this mode, the user has in-hand code that has been verified correct in the TMS320C30 file-to-file mode; thus debugging attention may be focused on the porting to the analog interface. C-callable assembly-language subroutines are available to the programmer for easy setup of the analog interface parameters (the sampling rate, the number of channels, and the specification of anti-aliasing filter), the definition of input and output memory buffers, and the initiation and termination of analog conversion.

New Features/Functions of the PISCES System

Since the installation of the PISCES system at CIRL in July 1991, several improvements and additions have been made. One improvement has been the addition of a second Sonitech Spirit-30 S-bus DSP board as well as the addition of a second FLAIR board. The two DSP boards can operate independently or they can operate as coupled synchronized processors. In the former mode, each board can run a specified stimulation algorithm and drive up to eight channels of electrical or acoustic stimulation. For applications requiring the synchronization of the two boards—for example, when a single DSP chip cannot provide the computational power to implement a desired algorithm—the two DSP boards and their FLAIR systems can be tightly coupled and synchronized so that a single processing scheme can be partitioned between the two boards.

Algorithms

This section describes two of the algorithms that we have run on PISCES to provide electrode-stimulation outputs for arbitrary acoustic inputs such as speech and music. The first algorithm described is a simulation of the Ineraid sound-processing unit that the subjects have been using in daily life. The second algorithmic implementation is a simulation based on stimulating electrodes with bipolar pulse signals whose amplitudes are derived from bandpass filter output envelopes. This algorithm, which has been studied and used by other cochlear-implant researchers, provides a totally different stimulation signal from the Ineraid stimulator waveforms. Because the pulsatile signals that stimulate each electrode do not occur simultaneously but are interleaved in time, the algorithm is called Continuous Interleaved Sampling (CIS).

Digital Simulation of the Ineraid Hardware Stimulator

The first algorithm implemented on PISCES for generating electrode-stimulation signals from sampled acoustic inputs was a digital simulation of the analog processing that is performed by the Ineraid hardware stimulator, as shown in Figure 9. In the Ineraid stimulator, the input signal is first fed to an automatic gain control (AGC) that applies dynamic range compression (DRC). The output of the AGC/DRC operation is the input to a four-channel filter bank. In the analog hardware, each continuous output waveform from the filter bank drives a voltage-to-current converter whose gain is adjusted to reflect the threshold that has been measured for the subject's corresponding electrode. In the digital implementation, the outputs of

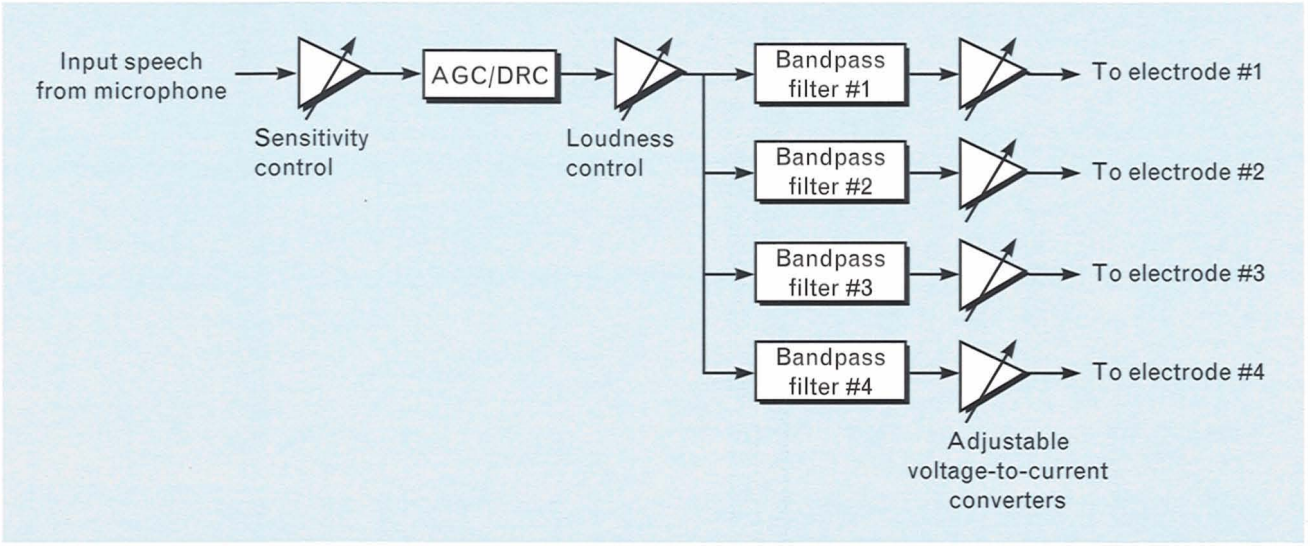


FIGURE 9. Signal processing performed by the Ineraid hardware stimulator. These processing steps, including the automatic gain control/dynamic range compression (AGC/DRC) operation, have been implemented in `cbank`—a C program that is a digital simulation of the Ineraid stimulator.

the filter bank are sent to a D/A converter, after which the output gain and voltage-to-current conversion are applied by external hardware. In both cases, the output of each voltage-to-current converter is used to stimulate one of the implanted electrodes.

The digital simulation permits a degree of flexibility that is unavailable in analog hardware. In the analog implementation, only the gain parameters can be adjusted. In the digital simulation, essentially all of the AGC/DRC and filter-bank characteristics can be specified at run time. A more subtle advantage is the ability of the sampled data implementation to simulate both AGC and a set of linear bandpass filters with the simulated components producing less nonlinear distortion than their analog hardware counterparts.

We have implemented a digital simulation of the Ineraid stimulator in a C program called `cbank`. The program has been compiled and run successfully on both a Sun SPARCstation IPC and the Sonitech TMS320C30 board. For the board, a few key subroutines were optimized in assembly language to achieve real-time performance.

We wrote and tested `cbank` with the three-step procedure outlined in the subsection "Software Environment." When running on the Sun SPARCstation, `cbank` reads a single-channel sampled data file from the host file system and produces a multichannel file as output. When running on the TMS320C30 board, `cbank` may be instructed either to read and write files from/to the host file system through SPOX, or to read and write from/to the A/D and D/A converters.

Command-line arguments and specification files allow the user considerable flexibility. These arguments and files allow a wide range of `cbank` parameters to be configured at run time rather than at compile time, thereby enhancing the user's ability to work with a subject interactively. Parameters that can be varied include the sampling rate, the number of filters in the filter bank, the filter shapes, the AGC/DRC characteristics, and the I/O type (disk versus A/D and D/A). Appendixes 1, 2, and 3 show the complete list of command-line arguments, an example of a main parameter specification file, and an example of a DRC file, respectively. The parameter specification file of Appendix 2 identifies four files, each containing coefficients for an FIR filter. These four files are read during program initialization, and the corresponding coefficients are stored in memory. In the parameter specification file, the use of an AGC (as well as the AGC attack and decay time constants) is also specified. If an AGC is used, the file of Appendix 3 will be named on the command line.

Figure 10 contains a diagram of the AGC operation used in the digital Ineraid simulation. The signal

input is multiplied by a gain factor that is determined by the signal envelope amplitude and the desired DRC characteristic. As shown by the DRC curve in Figure 11, a steady-state signal with an envelope at 0 dB would be multiplied by one, while a signal with an envelope at –20 dB would be multiplied by 10 to keep the output of the AGC at the desired constant level. The dynamic behavior of the AGC is determined by the smoothing that is applied to the envelope measurement because the measurement is used to determine the gain applied to the signal.

For signals with an increasing envelope, the smoothing time constant (called the attack time t_A) is normally short to keep the output well controlled. For a decreasing envelope, the smoothing time constant (called the release time t_R) can vary over a large range, depending on various factors. Figure 12 shows the dynamic response of the AGC for an input signal that maintains a 0-dB level for 250 msec, decreases to –20 dB for 500 msec, and then returns to 0 dB for 250 msec. The attack time is 0 msec for the two AGC outputs shown, while the release time is 250 msec for Figure 12(b) and 50 msec for Figure 12(c). Further details are contained in the section "Clinical Results."

In the digital simulation, the AGC/DRC is implemented as follows:

- For each input sample, the simulation determines whether the AGC is in attack mode or release mode.
- Depending on the mode, a new estimate of the signal envelope is calculated.
- Given this estimate, the simulation calculates an appropriate gain and applies it to the sample.

Defining $x[n]$ as the output of the A/D converter at time n, the attack-mode envelope is defined as

$$y_A[n] = \alpha_A |x[n]| + \beta_A y[n-1],$$

and the corresponding release-mode envelope is defined as

$$y_R[n] = \alpha_R |x[n]| + \beta_R y[n-1].$$

If $|x[n]| > y[n-1]$, then the AGC is defined to be in attack mode, and $y[n]$ is set equal to $y_A[n]$. On the other hand, if $|x[n]| \leq y[n-1]$, then the AGC is defined to be in release mode, and $y[n]$ is set equal to $y_R[n]$. Once the mode has been determined and $y[n]$

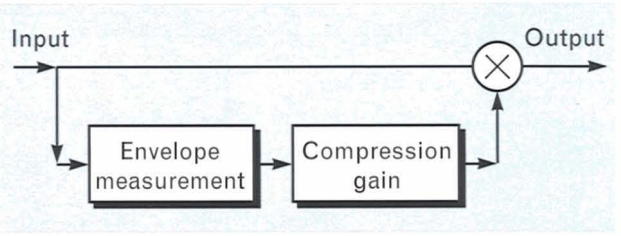


FIGURE 10. The automatic gain control (AGC) operation used in the digital Ineraid simulation.

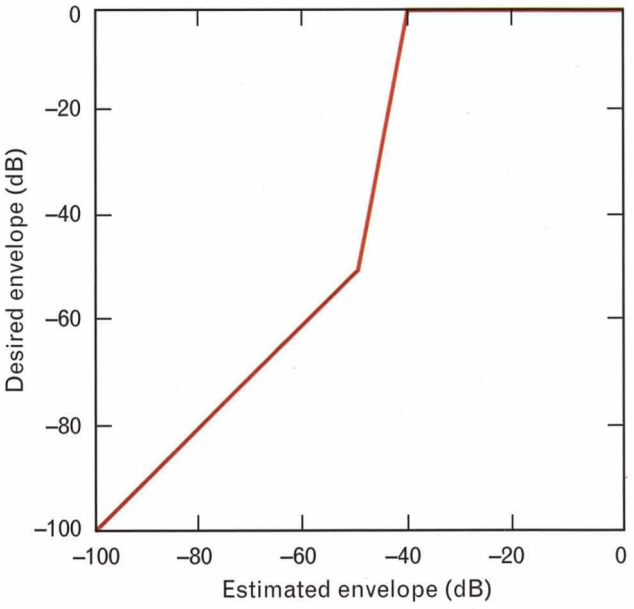


FIGURE 11. Example of DRC curve.

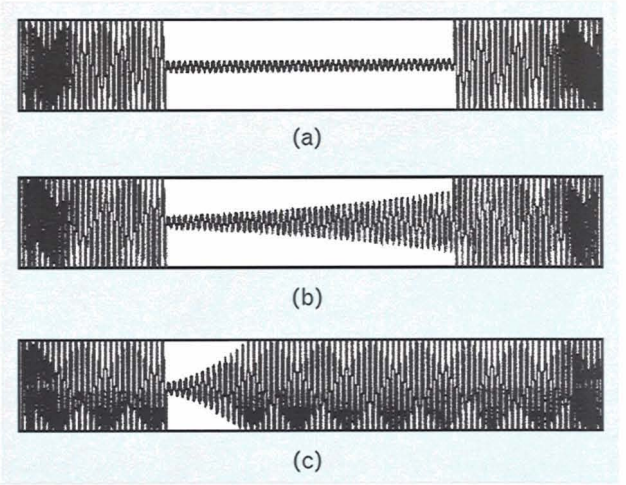


FIGURE 12. AGC dynamic-response waveforms: (a) input signal, (b) output for a release time of 250 msec, and (c) output for a release time of 50 msec. The total time duration of the waveforms is 1 sec, and the attack time is 0 msec for both parts *b* and *c*.

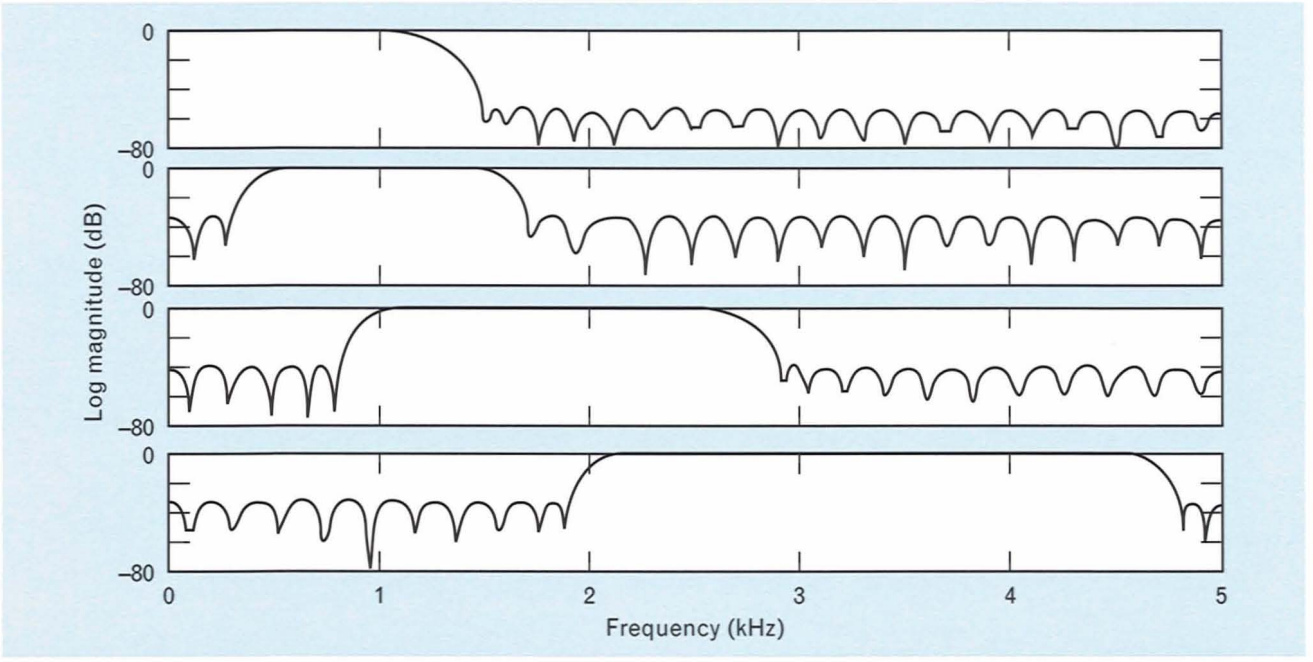


FIGURE 13. Example of `cbank` bandpass filters.

has been calculated, the value is used as an index to a lookup table to determine $g[n]$, the gain for time n. The final output of the AGC/DRC, which is used as the input to the filter bank, is

$$z[n] = x[n] \times g[n].$$

The α and β values are derived from t_A and t_R, the attack and release times, respectively, which are set in the main parameter specification file. Given an A/D sampling period τ in seconds, and t_A and t_R specified by the user in milliseconds,

$$\beta_A = e^{-1000\tau / t_A},$$
$$\beta_R = e^{-1000\tau / t_R},$$
$$\alpha_A = 1 - \beta_A, \text{ and}$$
$$\alpha_R = 1 - \beta_R.$$

The user provides the DRC function $g[\,]$ indirectly in a DRC specification file as a piecewise linear function of desired output envelopes e as a function of estimated input envelopes y. For convenience, the user specifies this function by providing the endpoints (in dB) of each linear component. The linear gains g are derived by the linear domain (as opposed to the log domain) division of e by y and are calculated as a function of y during algorithm initialization. The example DRC specification file of Appendix 3 has specified three piecewise linear regions. This specification, shown graphically in Figure 11, is linear for low envelope levels, expansive for a narrow range of intermediate envelope levels, and compressive for normal envelope values.

In the design of the filter bank for the digital simulation of the Ineraid stimulator, we employed finite impulse response (FIR) filters exclusively for three reasons. First, FIR filters can be designed to have a distortion-free, linear phase. For the case in which the filter output waveforms themselves are the stimulating signals, the linear-phase property prevents the introduction of artifacts from arbitrary and uncontrolled phase characteristics. Second, most signal-processing chips can be programmed easily to perform FIR filtering at the rate of one tap per cycle with very low overhead. Finally, when low-pass filtering is performed followed by downsampling, FIR filter outputs can be computed at the more efficient downsampled rate. As a consequence of this downsampling efficiency, FIR filters can be computationally comparable to infinite impulse response (IIR) filters.

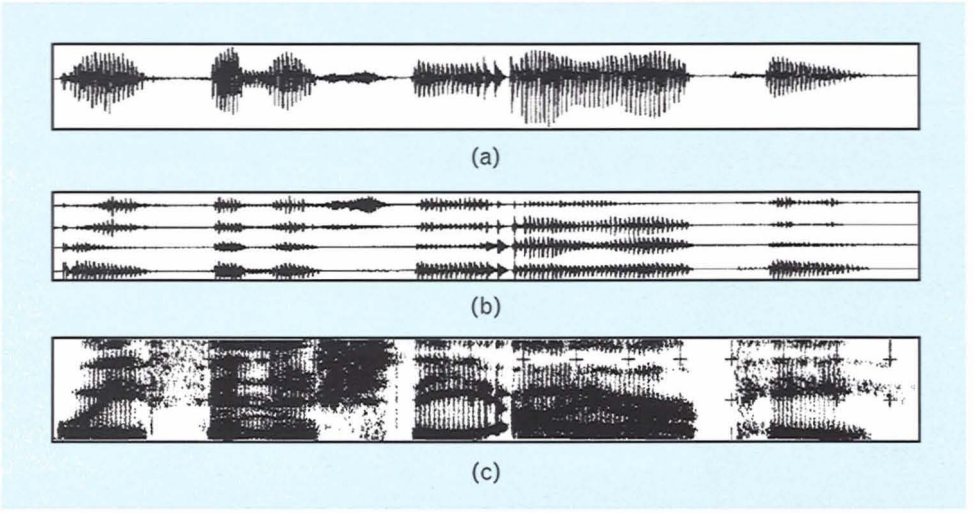


FIGURE 14. Example of `cbank`input and output: (a) waveform for input sentence: "We finished the IRP," (b) four channels of corresponding `cbank` output, and (c) wideband spectrogram of the input.

Thus we used the Parks-McClellan procedure to design sharp-transition, rectangular-frequency-response bandpass filters [7], and the Kaiser window procedure to design nonrectangular bandpass filters with responses more similar to those obtained with analog discrete components [8]. The nonrectangular filters were specified with frequency-domain points that provided the starting state for the Kaiser window procedure.

For the bandpass filter bank shown in Figure 13 and the main and DRC specification files of Appendixes 2 and 3, Figure 14 shows the output of `cbank` for a typical speech input. Bandpass filter outputs such as these were used to drive isolated voltage/current converters that provided the stimulation currents for four of the implanted electrodes.

Up to now, the Ineraid hardware stimulator has been capable of driving only four signal electrodes even though the implant assemblies provide six signal electrodes and two extracochlear electrodes that can be used as return paths for the stimulating currents. Because the `cbank` Ineraid simulation is capable of implementing an arbitrary number of channels, it is possible to explore the advantages of Ineraid-like stimulators that excite five and six electrodes, thereby more fully utilizing the capability of the Ineraid implant. We have designed five- and six-channel filter sets for this purpose.

Algorithm for Continuous Interleaved Sampling (CIS)

There are several ways to improve the Ineraid stimulator strategy. One method would be to introduce a technique for reducing the wide dynamic range of the bandpass waveform to the limited range of an electrode. Although there is some AGC processing for dynamic range and level control of the input signal in the Ineraid system, there is no control of the bandpass filter outputs. A second improvement would be to reduce the possibility of interactions that can arise when neighboring electrodes are stimulated simultaneously with continuous waveforms. Depending on their relative phase, simultaneous currents from neighboring electrodes can combine to produce stimuli that are stronger or weaker than anticipated, and these altered stimuli can create artifacts that may be misleading to a user of the Ineraid system.

The Continuous Interleaved Sampling (CIS) stimulator (Figure 15) is an attempt to overcome some of the shortcomings of the Ineraid system [9]. Instead of stimulating electrodes with continuous outputs from a bank of bandpass filters, the CIS stimulator uses pulse trains that are amplitude modulated by modified output envelopes from a bandpass-filter bank. As shown in Figure 16, the pulse outputs are interleaved in time so that the electrodes do not receive simultaneous stimulation. Rather than using the envelopes of

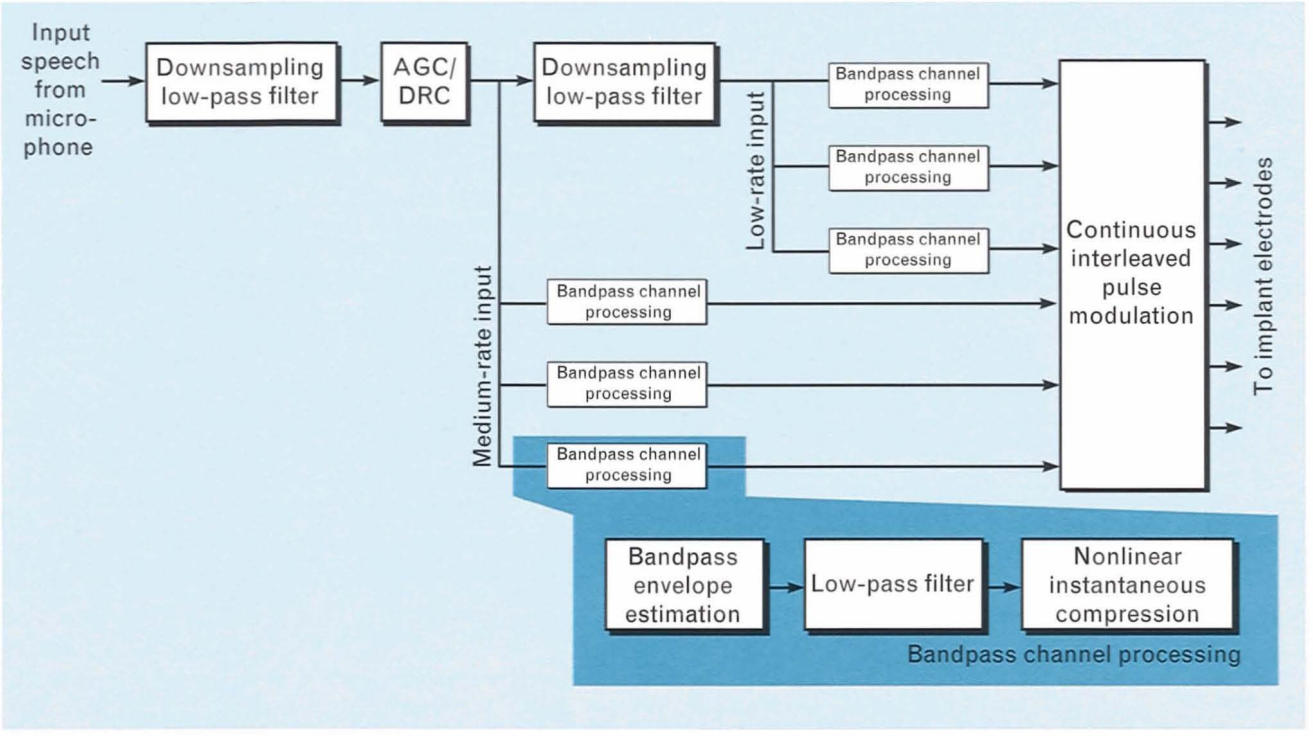


FIGURE 15. The Continuous Interleaved Sampling (CIS) stimulator.

the bandpass-filter outputs to modulate the pulse waveforms, the CIS stimulator uses compressors in each channel to map the envelope values into the dynamic range of the subject's individual electrodes. Thus, although the Ineraid and CIS stimulators both use a bank of bandpass filters to generate electrode stimulations, the CIS stimulator transforms the continuous outputs into envelope-modulated, nonoverlapping pulses. Currently, this CIS design is implemented only on PISCES; it does not yet exist as a wearable stimulator for general subject use.

We have implemented the CIS algorithm in a C program called `pbank`. As with the `cbank` program described earlier, `pbank` has been compiled and run successfully on a Sun SPARCstation IPC as well as on the Sonitech TMS320C30 board. A few key subroutines were optimized in assembly language to achieve real-time performance.

When running on the SPARCstation, `pbank` reads a single-channel, sampled data file from the host file system and produces a single multichannel file as output. When running on the TMS320C30 boards, `pbank` may be instructed either to read and write files from/to the host file system through SPOX, or to read and write from/to the A/D and D/A converters.

Command-line arguments and specification files allow each of the CIS processing blocks to be altered as required by clinical interactions and experiments. Although the AGC operation is described by the same DRC file format as in the `cbank` program, a more complex main specification file is used. The output compression curves are specified either as a series of line segments or as a table. Appendix 4 shows the `pbank` command-line argument list and Appendix 5 shows an example of a main parameter specification file. As with `cbank`, `pbank` was also designed to provide an arbitrary number of stimulation channels, making it possible to test implementations with as many as eight channels. We now describe each of the processing blocks in some detail.

Input Filtering and AGC/DRC. In PISCES, the basic system sampling rate (SSR)—i.e., the rate at which the A/D converter delivers samples of input data—determines the rate at which output samples can be delivered to the D/A converter for stimulation waveforms. Because the SSR also determines the maximum repetition rate of the pulsatile stimuli, it is typically set at a relatively high value to provide suitable

sampling of the envelope signals. The relationship between SSR and the time interval between pulses on any channel for an N-channel system can be expressed as

$$\text{Interval} = \frac{2N}{\text{SSR}},$$

where each channel's pulsatile signal requires two sample intervals (for the negative and positive excursions), and all N channels must be excited before a repeat pulse can occur. A six-channel system designed to run at an update interval of, for example, 600 μsec would require a sampling interval of 50 μsec, or an SSR of 20 kHz.

The input processing consists of two low-pass filters and the AGC, as shown in Figure 15. The two filters allow for optional downsampling that can reduce the computational load when the analog signals are processed at various points in the simulation. The first low-pass filter allows for an optional downsampling of the basic SSR. For example, an SSR of 32 kHz, which will provide a minimum pulse update interval of 375 μsec, supports a bandwidth of 16 kHz. A signal bandwidth of less than 8 kHz is sufficient for implant stimulation, so that a 50% reduction of the SSR to 16 kHz is possible. Appendix 5 lists this filter coefficient file after the BEGIN_FRONTEND line. In this case the first filter file is a low-pass filter with a passband of 8 kHz. A 2:1 downsampling of the output is shown by the 2 after the 0.0 gain value.

The AGC/DRC operation is exactly the same process described for the `cbank` program and operates on the output of the first low-pass filter to reduce the overall input dynamic range as specified by the attack and release time constants and the DRC file. For the subset of channels whose highest frequency of importance is below 2 kHz, a second low-pass filter is used to reduce the computation by allowing a downsampling for that subset of channels to 4 kHz. In Appendix 5, the file that contains the coefficients for the second filter is listed after the coefficients file of the first filter, and a 0.0-dB gain for the filter output followed by a 4:1 downsampling operation have also been specified. Figure 15 shows three upper-frequency channels driven from the first downsampling filter (running at 16 kHz), and three lower-frequency channels driven from the second downsampling filter (running at 4 kHz). The assignment of bandpass channels to run on the 16 kHz or 4 kHz sampling rate is defined by the line in the specification file before the END_FRONTEND tag (Appendix 5).

Bandpass Envelope Estimation. As in the `cbank` implementation, a set of bandpass filters is used to produce a set of signals that are then used as the basis for electrode stimulation. Unlike the `cbank` case, however, the CIS algorithm and the `pbank` implementation in particular do not use the bandpass output waveforms directly. Instead, the envelope of the waveform is measured, compressed, and then used to modulate a constant-repetition-rate bipolar pulsatile waveform.

Two envelope-measurement techniques have been implemented in `pbank`: the *rectification* estimation method and the *quadrature* estimation method. The rectification (or detection) method comprises a bandpass filter followed by a strong nonlinearity such as a full-wave or half-wave rectifier, i.e., an operation that replaces each signal sample with its magnitude value or, in the half-wave case, replaces the negative samples with zero. The output of such an operation is smoothed by a low-pass filter to eliminate spurious harmonics. The quadrature estimation method uses a second bandpass output that has been shifted 90° in phase from the original. The two signals are squared

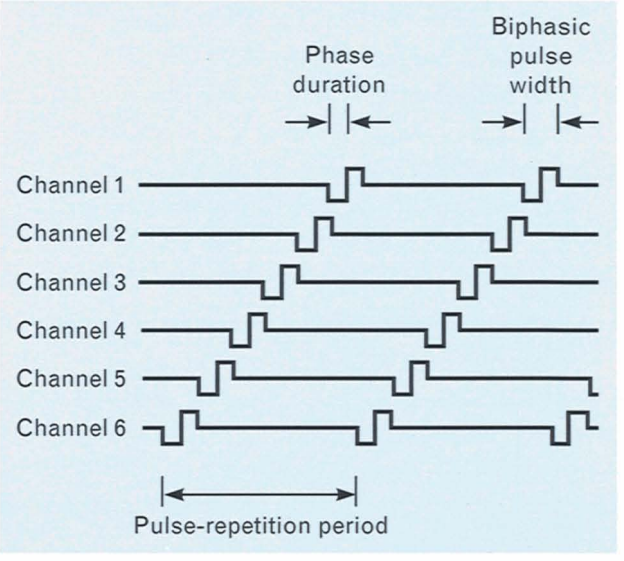


FIGURE 16. Pulse outputs from `pbank`-a C program that is a digital simulation of the CIS stimulator. Note how the pulse outputs of the six channels are interleaved in time.

and summed, and the square root taken, thus producing an estimate of the envelope. The choice of estimation procedure is specified on the `pbank` command line. The rectification estimation method is commonly used in both analog and digital systems. Unfortunately, full- or half-wave rectification generates a range of signal harmonics that may result in aliasing in the sampled data domain. Appendix 6 shows that the quadrature estimation method is somewhat more robust to harmonic distortion and, consequently, to aliasing.

The bandpass filters divide the input spectrum into six channels spaced logarithmically in center frequency and bandwidth over the range from 300 to 7000 Hz. Rectification envelope estimation requires only one bandpass filter per channel. Either of the techniques described for the `cbank` filter bank, namely, the Parks-McClellan or Kaiser window design, provides the needed flexibility. Quadrature envelope estimation requires one quadrature pair of bandpass filters per channel. Two approaches for designing such filter pairs have been used. In the first approach, a prototype low-pass filter is designed with the Parks-McClellan algorithm. Two bandpass impulse responses are obtained by multiplying the low-pass impulse response by sampled sine and cosine functions whose frequencies are at the desired center frequency of the bandpass filter. Multiplication by sine and cosine guarantees the fixed 90° phase difference [10]. A second design technique uses the eigenfilter method developed by Nguyen, which approximates arbitrary magnitude and phase responses in a minimum mean-square-error sense [11]. This technique has been used to generate quadrature filter pairs with 12-dB/octave responses. Up to now, only the Parks-McClellan frequency-shifted filters have been used with subjects.

The specification file in Appendix 5 shows that the bandpass filters have been specified in pairs for the quadrature case. (In the rectification case, the single bandpass-filter file name must be repeated to satisfy the file format.) Each of these filter pairs can be followed by a gain as specified and an additional downsampling operation.

Low-Pass Smoothing Filters. A low-pass smoothing filter is needed to ensure that the envelope signal used for the amplitude modulation of the pulse train is limited in bandwidth so that the signal can be adequately sampled by the train. With high-frequency pulse trains (higher than several kHz) for which a wideband envelope can be used with no aliasing, there is a trade-off between wideband maximal envelope variations and narrower-band envelopes that reduce harmonics of the voicing frequency. This consideration is important because the envelope signal is the only information that is provided to the subject's electrode. For the case of a full- or half-wave rectified bandpass filter output, the low-pass filtering is also needed to eliminate the out-of-band harmonics generated by the rectifier nonlinearity.

In the main parameter specification file, the files for the low-pass filters are listed for each channel, as shown in Appendix 5. As with all of the previous filter files, there is provision for a post-filtering gain as well as additional downsampling. Preceding the list of files for the low-pass filters is a line that allows for the introduction of a delay to equalize each of the channel filtering operations. This step is required because each of the channels may be implemented with different length bandpass and low-pass filters whose composite delay might be different.

The downsampling factors that are specified after each of the filtering operations must produce the same composite downsampling at the output of each of the low-pass filters. At present, the cutoff responses of the low-pass filters resemble a low-order analog filter with a cutoff frequency at about 400 Hz, thus requiring a pulse output signal of at least 1 kHz to produce an adequately sampled envelope signal.

Output Compression Mapping. The dynamic range of an electrode stimulating signal must fit a rather limited electrode dynamic range (10 to 24 dB). Envelope values are mapped into the measured dynamic range for the corresponding electrode by using specified mapping functions. Typical transformations that map from x (the envelope estimate available from the low-pass smoothing filter) to y (the mapped envelope output used for pulse modulation) are given by

$$y = A + B(x - X)^p,$$

where A, B, X, and p are dependent on subject threshold and dynamic-range measurements and the desired compression characteristic. This mapping is specified

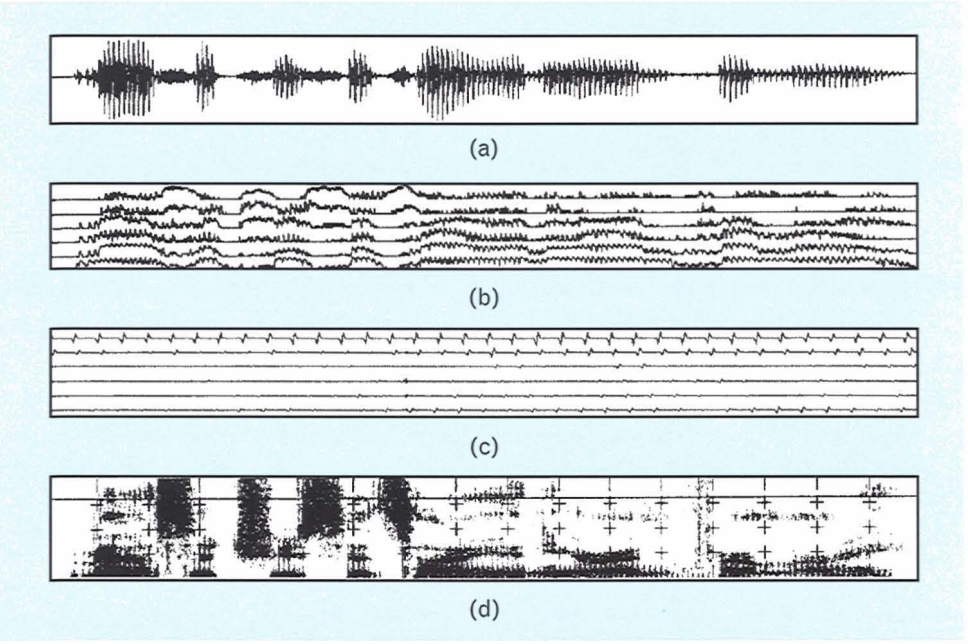


FIGURE 17. Example of `pbank` input and output: (a) waveform for input sentence: "Massachusetts Eye and Ear Infirmary," (b) estimated envelopes for all six channels, (c) close-up of the pulse output in the middle of an unvoiced fricative, and (d) wideband spectrogram of the input. This example is for `pbank` implementing a six-channel CIS simulation with the specification file of Appendix 5.

independently for each channel because each electrode's dynamic range reflects conditions in the cochlea around that electrode. In addition to the variables described above, the range of input values to be mapped into the electrode dynamic range is also a variable. For the most part, we have been using ranges from 40 to 60 dB.

The output compression mapping file is given to `pbank` on the command line as specified in Appendix 4. Additional command-line arguments tell `pbank` whether the compression mapping file is in the form of linear segments or a table.

Pulse-Modulation Waveform Output. The final pulse output waveforms for each channel are computed at the full input sampling rate. For example, a 32-kHz rate will allow a channel D/A converter to output a pulse of duration 31.25 μsec. The specification file of Appendix 5 defines the pulse sequencing as a matrix in which the activity of each channel is specified at each sampling time for one period of the output cycle. The matrix shown in Appendix 5 at a sampling rate of 32 kHz would produce the six-channel output of Figure 16 with a pulse-repetition rate of 32/12 = 2.67 kHz (biphasic pulse width of 31.25 μsec/phase). The biphasic pulse shape allows for a zero-mean output signal while retaining a narrow pulse shape. (Note: Each electrode is stimulated by a current that is proportional to the `pbank` waveform output. A zero-mean signal results in delivery of zero net charge by the electrode to the cochlea, thereby causing minimal trauma in the surrounding cochlear tissue.) The nonoverlapping pulse waveforms may reduce field interactions and, therefore, some distortion between electrodes in the cochlea. These modulated pulse trains are fed to the voltage-to-current converters, which in turn stimulate the corresponding electrodes.

Figure 17 shows the output of `pbank` for a typical speech input. Figure 17(a) shows the input waveform and Figure 17(b) displays the six envelope signals to be used for the pulse modulation. Note the considerable ripples in each of the envelope signals during the periodic portions of the waveform. These ripples indicate that the 400-Hz low-pass filters used for smoothing are still passing many pitch harmonics. The pulse output signals for an expanded portion of

an unvoiced sound (Figure 17[c]) appear to be triangular when, in fact, they are bipolar samples. The discrepancy is a result of interpolation in the plotting process.

Clinical Results

The PISCES system was installed at the Massachusetts Eye and Ear Infirmary (MEEI) Cochlear Implant Research Laboratory (CIRL) in early July 1991. With PISCES in place at CIRL, we have been able to investigate a wide range of sound-processing algorithms for 20 participating subjects using the six-channel Ineraid implant. The aims of this ongoing research have been to understand how the present stimulating algorithms provide acoustic information to implanted subjects and to use this understanding to develop new algorithms for improved speech reception.

Figure 18 contains a photograph of PISCES and the current-isolator/stimulation equipment rack. The set of D/A outputs connect to a set of isolated voltage-to-current stimulators housed in a single equipment rack outside a small sound-insulated testing room. The current outputs are available on a cable/plug assembly inside the testing room so that subjects can unplug their own Ineraid stimulator and substitute the isolated output currents produced by PISCES. In an emergency, a panic button enables a subject quick disconnection from the current drivers.

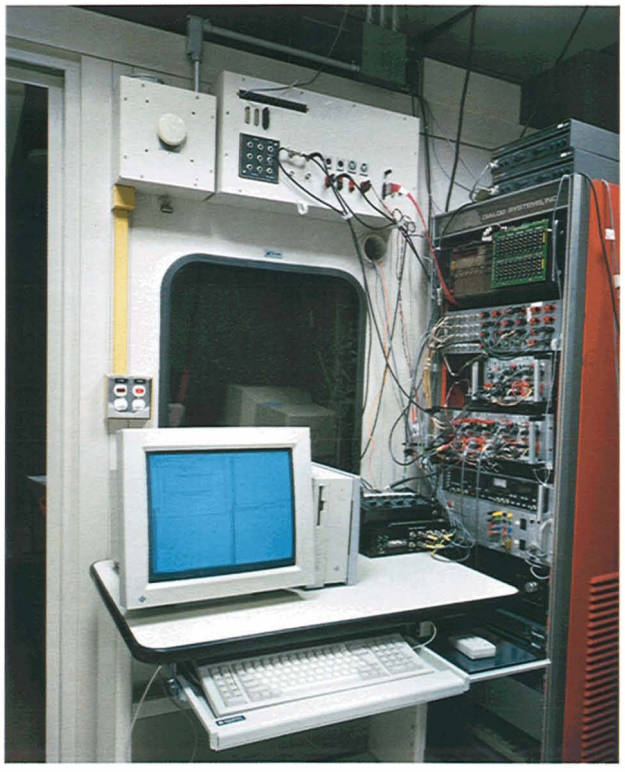


FIGURE 18. PISCES and the isolated-current-driver equipment installed at the Massachusetts Eye and Ear Infirmary (MEEI) Cochlear Implant Research Laboratory (CIRL).

When a subject is initially connected to PISCES running a stimulator-algorithm simulation, an informal conversation between the subject and the researcher can be used to gauge the gross performance of the system. Feedback from the subject provides useful information about signal levels as well as rough comparisons between the present and previous parameter settings. Indeed, the subject may identify problems in the new stimulation system, and this information may be used to fine-tune the isolator gains and other parameters. During this initialization period, the subject has the opportunity to acclimate to each new stimulator variation, thereby providing at least a small amount of learning before more quantitative testing takes place.

Before discussing the results of testing several sound-processor/stimulator variations implemented with the `cbank` and `pbank` software running on PISCES, we describe (1) some of the psychophysical parameters used to adjust processor/stimulators for individual subjects, and (2) the methods used for testing speech reception and for quantitatively evaluating the utility of sound-processing algorithms.

Psychophysical Parameters

Threshold and uncomfortable loudness level (UCL) are two psychophysical parameters that determine how a particular sound-processing algorithm should be customized for an individual subject. Threshold is defined as the minimum detectable amplitude of a particular stimulus waveform. For a waveform of increasing amplitude, the UCL is the amplitude that produces the first uncomfortable sensation. These two parameters are measured for each of the electrode pairs used by the sound-processing algorithm; together the parameters define the dynamic range of electrical stimulation for that set of electrodes.

In the case of the Ineraid hardware stimulator and

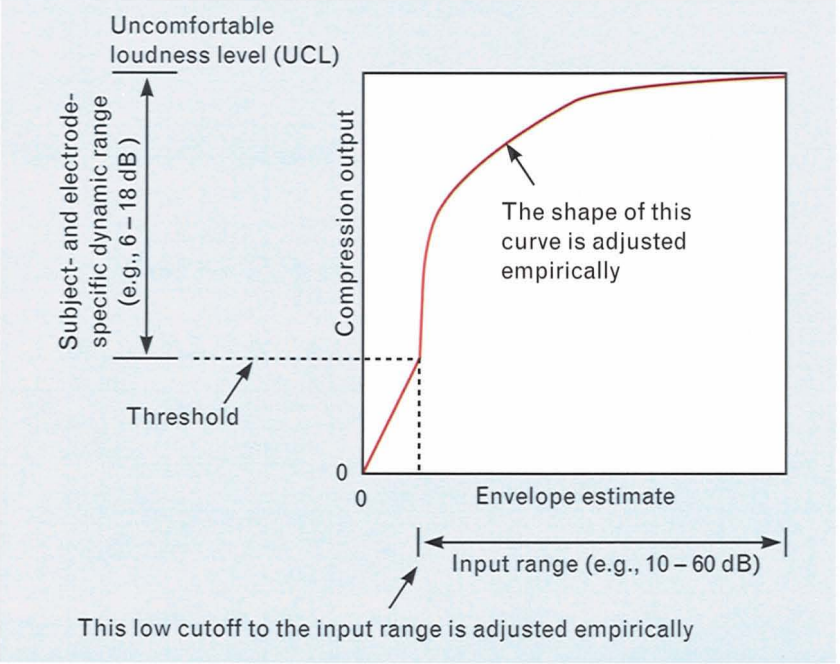


FIGURE 19. A typical output compression curve.

the Ineraid simulations implemented with `cbank`, the threshold value for each electrode influences the gain relationship across the filter channels. For example, consider a subject with uniform threshold across all electrodes and a four-channel processing scheme using bandpass-filter center frequencies of 500, 1000, 2000, and 4000 Hz. In a typical processing algorithm, the relative channel gains increase 6 dB/octave. Thus the relative channel gains of the 500-, 1000-, 2000-, and 4000-Hz channels would be 1:2:4:8, respectively. If, however, the threshold of the electrode connected to the 1000-Hz channel were twice that of the other electrodes, the relative gains would be 1:4:4:8. Consequently, a channel connected to a very sensitive (low threshold) electrode will have a lower gain on that channel than if it is connected to a less sensitive electrode.

In the case of a pulsatile, CIS-stimulator simulation implemented with `pbank` (see Figure 15), mapping the channel output to actual stimulus amplitude requires both the threshold and UCL values so that the mapping functions described in the subsection "Algorithm for Continuous Interleaved Sampling (CIS)" can be computed. The ratio between the measured values of threshold and UCL (defined as an electrode's dynamic range) is used to compute the output mapping function, as shown in Figure 19. The peak output of the envelope measurement is mapped to the UCL value, and some minimum value (usually between –40 to –60 dB relative to the peak input) is mapped to the threshold value. These mapping functions are specific to a subject, an individual electrode, and the set of waveform parameters used in the simulation.

Test Materials

Most of the quantitative speech-reception tests that are being used to evaluate the various stimulation schemes are based on measures of consonant identification: a subject tries to identify consonant sounds in vowel-consonant-vowel (VCV) settings (e.g., "asha" and "aba") as recorded by a male and a female on the Iowa Speech Perception Video Disk [12]. Random sequences of the VCV utterances are played from the video disk under control of a program that tabulates the subject's responses. The researcher can select test sets of 8, 16, or 24 consonants as spoken by either the male or female, and each consonant of a set is presented five times in random order. The subject "hears" an utterance through the stimulator simulation and

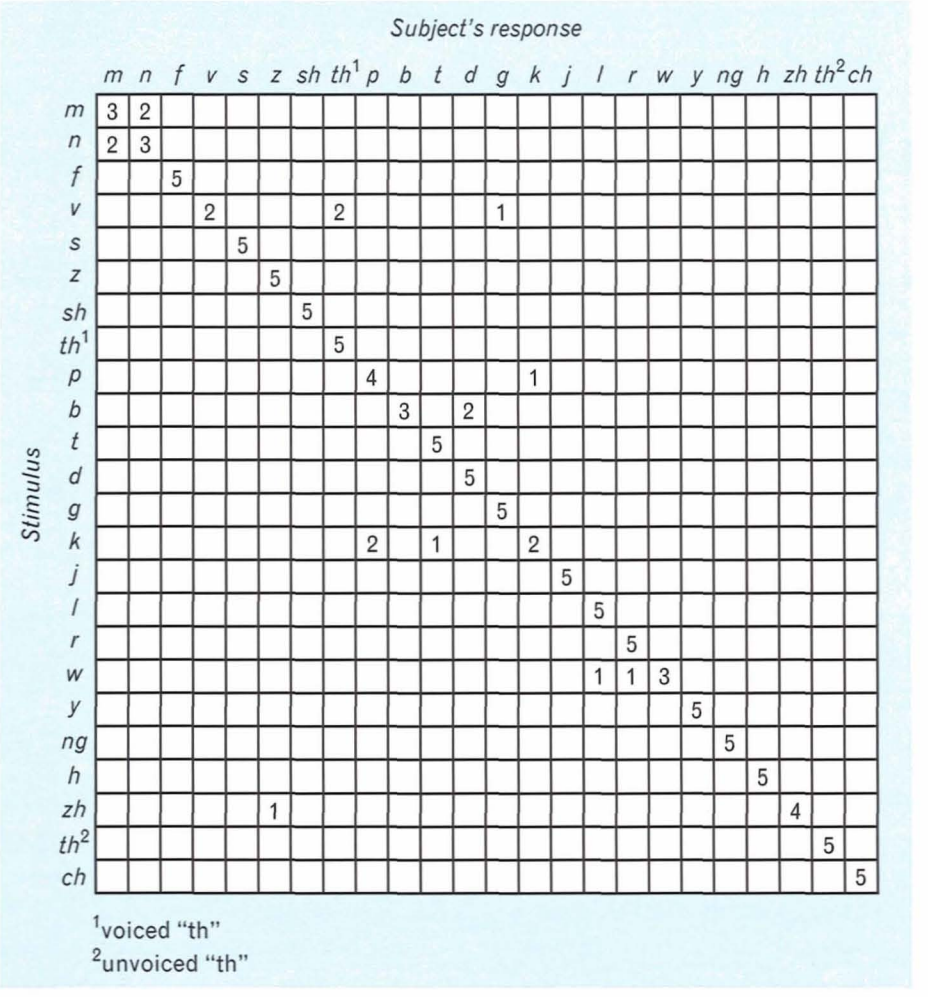


FIGURE 20. Results of a 24-consonant test as summarized in a confusion matrix. The horizontal rows in the confusion matrix represent the stimulus consonants while the vertical columns contain the consonants selected by the subject. Thus, for the ideal case in which the subject classifies every consonant correctly, all of the off-diagonal entries would be zero. The test was for 24 consonant sounds in a vowel-consonant-vowel setting with the vowel "ah" (e.g., "ama," "ana," "afa"), as spoken by a male. The 24 sounds were presented to the subject five times each (120 entries).

must identify the utterance by typing the appropriate response on a terminal. Figure 20 contains an example of the results for a 24-consonant test.

For additional testing, we have been using a set of eight vowels recorded in the carrier h_d (e.g., "had" and "hood") as spoken by both a male and a female with several exemplars of each vowel utterance. These vowel utterances are also presented to the subject in randomized order after a suitable review sequence. The basic vowel test consists of 24 items: the eight vowels presented three times each in a randomized order. These standard consonant and vowel tests allow direct comparison with results from other investigators using the same tests.

Subject Testing Results

In this subsection we describe a series of experiments that were conducted with two subjects. An experiment using `cbank` to implement a simulation of the Ineraid hardware allowed us to compare the performance of the two subjects before and after the simulation was substituted for their own hardware. The sim-

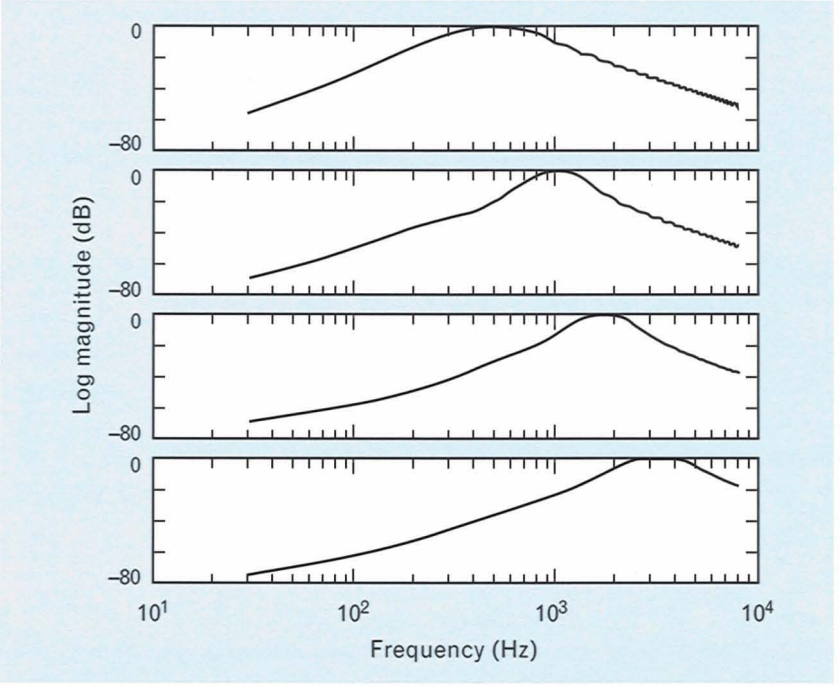


FIGURE 21. Finite impulse response (FIR) filter bank used to approximate the 12-dB/octave Butterworth bandpass filters of the Ineraid hardware processor.

ilar performance of the hardware and simulation provided evidence that PISCES was operating as expected. Taking advantage of the flexibility inherent in the `cbank` simulation of the Ineraid hardware, we also studied how the release time of the front-end AGC affects speech reception. The results of these experiments suggest that changes in the release time of the Ineraid hardware AGC could provide significantly better speech reception for some users. These results also demonstrate the utility of PISCES real-time flexibility.

As discussed in the subsection "Algorithm for Continuous Interleaved Sampling (CIS)," there are several features of the Ineraid processor algorithm that may be compromising the performance of the Ineraid system. Using CIS stimulation strategies implemented with `pbank`, we have begun a series of investigations designed to address this concern. In one series of experiments, we compared the performance of a four-channel CIS with the Ineraid strategy, and we then explored several variations of the CIS implementation that allowed us to separate the effects of parameter variations. Finally, this subsection concludes with data from an experiment that investigated a six-channel CIS implementation using `pbank`.

Ineraid Simulations. To simulate the Ineraid hardware algorithm, we used a `cbank`-based simulation with a set of FIR filters (Figure 21) that closely approximated the 12-dB/octave Butterworth filters used in the Ineraid hardware processor. The AGC parameters of `cbank` (as described in the subsection "Digital Simulation of the Ineraid Hardware Stimulator") were set to approximate the dynamic range compression of the hardware. The time constants included an attack time t_A of 0 msec and a release time t_R of 250 msec. For subjects S01 and S04, Table 1 com-

Table 1. Performance of Ineraid Hardware and Simulation for Consonant-Reception Tests

	Hardware (percent correct)	*Simulation (percent correct)*
Subject S01[1]	34.4	33.0
Subject S04[2]	83.4	83.4

[1] A 16-consonant test was used.
[2] A 24-consonant test was used.

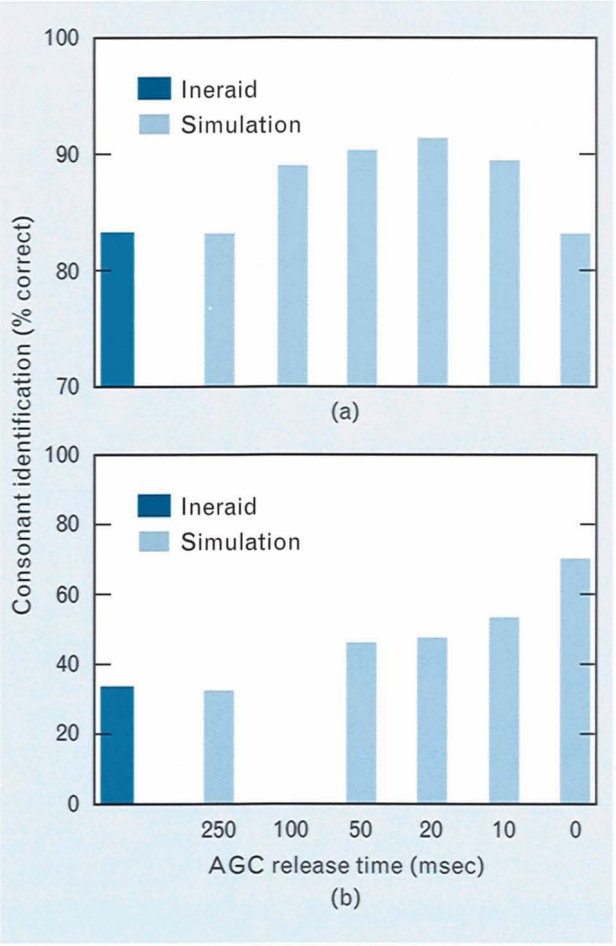


FIGURE 22. Consonant-identification scores for (a) subject S04 taking the 24-consonant test, and (b) subject S01 taking the 16-consonant test. (Note: For subject S01, no measurement was taken at an AGC release time of 100 msec.)

Table 2. Dynamic Range Measurements for Subject S04

Electrode number	*Threshold (μA)*	*UCL (μA)*	*Dynamic range (dB)*
1	32	210	16.3
2	31	235	17.7
3	33	285	18.8
4	38	280	17.3
5	40	305	17.7
6	67	300	13.0

pares the performance of the Ineraid hardware with the `cbank` simulation. Note that the overall scores for the hardware and the simulation are very close.

For the case of Ineraid processing (either in the hardware or in the simulation), the AGC operation is the only mechanism for containing the wide dynamic range variations of the input signal waveform within the narrow range for electrode stimulation. The output of the AGC is compressed in dynamic range according to the specified DRC characteristic. As a consequence, the AGC operation and its dynamic properties influence speech reception.

To examine the effect of the AGC release time t_R, we tested a set of simulations in which t_R was varied from 250 msec to 0 msec. The attack time was fixed at 0 msec (as it had been in the earlier simulation) because this instantaneous control of any increasing signal bounded the input. As the release time was shortened, we expected that weak consonants following stronger vowels would be strengthened in the manner shown in Figure 12 and, consequently, speech reception would vary. This variation can be seen in Figure 22, which presents the results for two subjects: subject S04 taking the 24-consonant test and subject S01 taking the 16-consonant test. The dark bar in each plot represents the score for the subject's own Ineraid hardware with the factory-set AGC values. The score for the Ineraid hardware and the entry for the 250 msec release time are the same data presented in Table 1.

Varying the AGC release time results in significant differences in performance for both subjects. Subject S04 moves from 83% for relatively long release times to 91% for a t_R of 20 msec while subject S01 does best with an instantaneous release time (t_R = 0 msec). These scores indicate that the optimal temporal behavior of the AGC may vary across subjects. For both subjects S04 and S01, however, release times different from those used in the hardware AGC improved the reception of consonants.

We also investigated the effect of varying the attack time t_A. In informal experiments, we confirmed the need for instantaneous control (t_A = 0 msec) of increasing signals. Otherwise, sudden increases in signal level produced sound sensations that were too loud for the subjects.

Table 3. Comparison of CIS and Ineraid Simulations

	CIS	*Ineraid*
Simulation waveform	Amplitude-modulated constant-frequency pulse	Continuous-output bandpass waveform
Channel stimulation timing	Interleaved	Simultaneous
Information source	Envelope cues only (~400-Hz bandwidth)	Waveform cues (~6000-Hz bandwidth)
Signal compression	Channel specific	On input
Psychophysics used	Threshold and UCL	Threshold

Four-Channel CIS Simulations. The performance of a four-channel CIS algorithm implemented in `pbank` with the same FIR filters as the Ineraid simulation was compared with the performance of the Ineraid system. For these experiments, we computed a set of four compression mapping curves from subject S04's data for electrodes 1 through 4 (Table 2). The curves mapped a 60-dB input range into each electrode's dynamic range. A series of experiments using these curves in a CIS simulation with subject S04 allowed us to converge to a set of gain parameters that produced a score of 92% on the 24-consonant test. This performance was not significantly different from that of the same subject using an optimum AGC setting for the Ineraid simulation. Nonetheless, subject S04 has expressed a clear preference for the CIS simulation based on his listening to conversational speech.

As discussed in the subsection "Algorithm for Continuous Interleaved Sampling (CIS)," the CIS-stimulation algorithm attempts to eliminate two disadvantages of the Ineraid processing algorithm: (1) the lack of channel-specific dynamic range compression and (2) the interaction between electrodes during simultaneous current stimulation. From Table 3 we see that, to overcome the Ineraid disadvantages, we have introduced other differences in the CIS algorithm that may have enhanced or diminished the benefits of that algorithm. Thus important performance improvements from the changes in compression technique and the lack of interaction may have been masked by the degrading effects of some of these other differences. To evaluate the impact of each of these differences, we found it necessary to compare algorithm implementations by varying only that characteristic being evaluated.

One important algorithm characteristic is the manner in which the large dynamic range of an acoustic speech signal is reduced to make it compatible with the relatively small dynamic range of the electrical signal that is used to stimulate each electrode. Figure 23 contrasts the compression methods of the Ineraid and CIS algorithms. In the Ineraid processing, an AGC operation modifies the wideband input signal, as described earlier. Threshold measurements are used for adjusting the channel gains to scale the stimulating bandpass waveforms. In the CIS algorithm, there is no AGC action on the input signal. Instead, the envelopes in each channel are mapped into the dynamic ranges of the corresponding electrodes, as defined by the threshold and UCL measures.

It is possible to compare the two methods of signal compression in the context of a standard CIS-stimulation system implemented with `pbank`. By including a front-end AGC and eliminating the mapping stage, we can create an AGC-CIS system that relies on the input AGC for its dynamic range compression in the same way that the Ineraid simulation does. With such an AGC-CIS simulation, subject S04 scored 89% on the 24-consonant test. Again this score was not significantly different from the score for the CIS or Ineraid simulations but, as before, the subject has stated a strong preference for the CIS simulation. Although

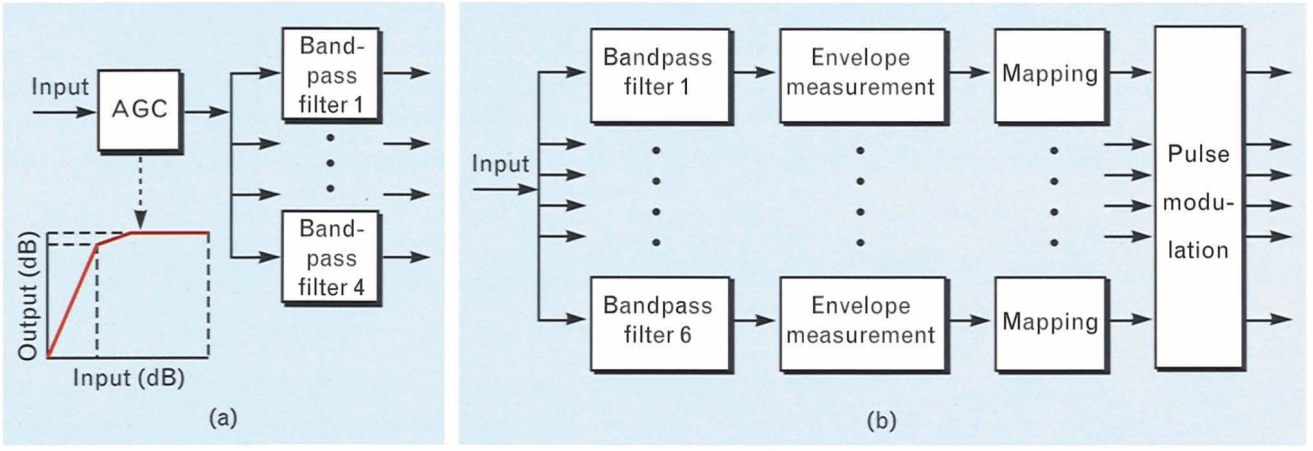


FIGURE 23. Comparison of DRC operations in (a) the Ineraid algorithm, and (b) the CIS algorithm.

there was no significant increase in speech reception for the 24 consonants, subject S04's strong preference for the CIS simulation as compared to the AGC-CIS simulation suggests that channel-specific mapping contributed to the subject's preference.

The CIS simulation also allows us to contrast interleaved with simultaneous stimulation. Recall from Figure 16 that in the interleaving process each electrode's stimulation signal occurs at a different, non-overlapping time. Instead of such interleaving, a simultaneous-output CIS (simul-CIS) algorithm can be used in which the pulse matrix in the specification file of Appendix 5 specifies that all four electrodes receive their pulsatile stimuli at the same time and are unstimulated for the rest of the update interval. Experiments with subject S04 using the simul-CIS algorithm resulted in a 24-consonant score of 85%, compared with 92% for the interleaved case. For the same output levels, subject S04 reported that the simul-CIS system produced "louder" signals than the standard CIS system. This report along with the significant decrease in score suggests that, in the CIS context, pulse interleaving results in improved reception, presumably from a lowering of the incidence of spurious interactions between electrodes.

To date, these experiments seem to indicate that (1) there are no significant differences in consonant recognition between the very different Ineraid and CIS simulations, (2) non-simultaneous stimulation in the CIS context produces better consonant recognition than simultaneous stimulation, and (3) both the compression method and the non-simultaneous stimulation used in the CIS system probably contribute to the subject's preference for that system. Because the effects on speech reception of these and the other differences shown in Table 3 deserve further exploration, this style of research continues to be a focus of our work.

Six-Channel CIS Simulations. Because the Ineraid implant consists of six intracochlear electrodes, we have been motivated to implement a six-channel CIS simulation with `pbank` to explore the effects of an increase in the number of stimulator channels. Using the filters shown in Figure 24 to divide the frequency band from 300 to 7000 Hz into six channels of equal log bandwidth, and using a `pbank` operating at 32 kHz, we produced a CIS simulation that stimulates each electrode at an update rate of 2 kHz. (Note: We allowed 4 samples of dead time in the update interval to provide an overall division of the system sampling rate by 16 pulse intervals.) The 2-kHz update rate was the same rate used by the four-channel CIS implementations described earlier. Using the psychophysical data of Table 2, we computed a set of six compression curves that mapped an input range of 60 dB into each electrode's dynamic range. With this simulation, subject S04 scored 99% for the 24-consonant test. With a similar filter design for four channels covering the same frequency extent in the CIS implementation described earlier, the subject had scored 89%. This result suggests that the number of channels has an important effect on performance.

We plan to explore the effect of increasing the number of channels in the Ineraid simulations. Differences between the Ineraid and CIS algorithms may become more pronounced as the number of channels increases because the simultaneous stimulation of a greater number of electrodes increases the likelihood of field interactions. Additional work must be also done to investigate the effects of environmental noise on both algorithms. Differences in the CIS compression scheme may result in benefits to speech reception as the signal-to-noise ratio decreases.

In closing, we would like to note that subject S04 often remains in the laboratory for hours listening to music through the PISCES system running a six-channel CIS simulation.

Conclusions

This article has summarized our ongoing effort to use current digital signal processing (DSP) techniques to enhance various areas of cochlear-implant research. Drawing on Lincoln Laboratory's expertise in digital speech and signal processing, we have designed, built, and tested an interactively adjustable implant stimulator—the Programmable Interactive System for Cochlear Implant Electrode Stimulation (PISCES). This work was performed as a Lincoln Laboratory Innovative Research Program (IRP) in collaboration with researchers at the MIT Research Laboratory of Electronics (RLE) and the Massachusetts Eye and Ear Infirmary (MEEI) Cochlear Implant Research Laboratory (CIRL). The installation of this system at MEEI/CIRL has enabled the designing and testing of new stimulator algorithms; these algorithms have demonstrated improved speech-reception performance for several subjects. In addition, the flexible nature of PISCES allows a researcher to manipulate processing schemes easily in ways that provide insight into the mechanisms responsible for subject performance. This insight should provide the basis for developing new processing algorithms that will significantly improve communication for the hearing impaired.

As part of an ongoing contract with the National Institutes of Health (NIH) Neural Prosthesis Program awarded to the MIT and MEEI researchers involved in the IRP effort, work has continued toward the design of portable implant stimulators that will provide power and flexibility for implementing the wide class

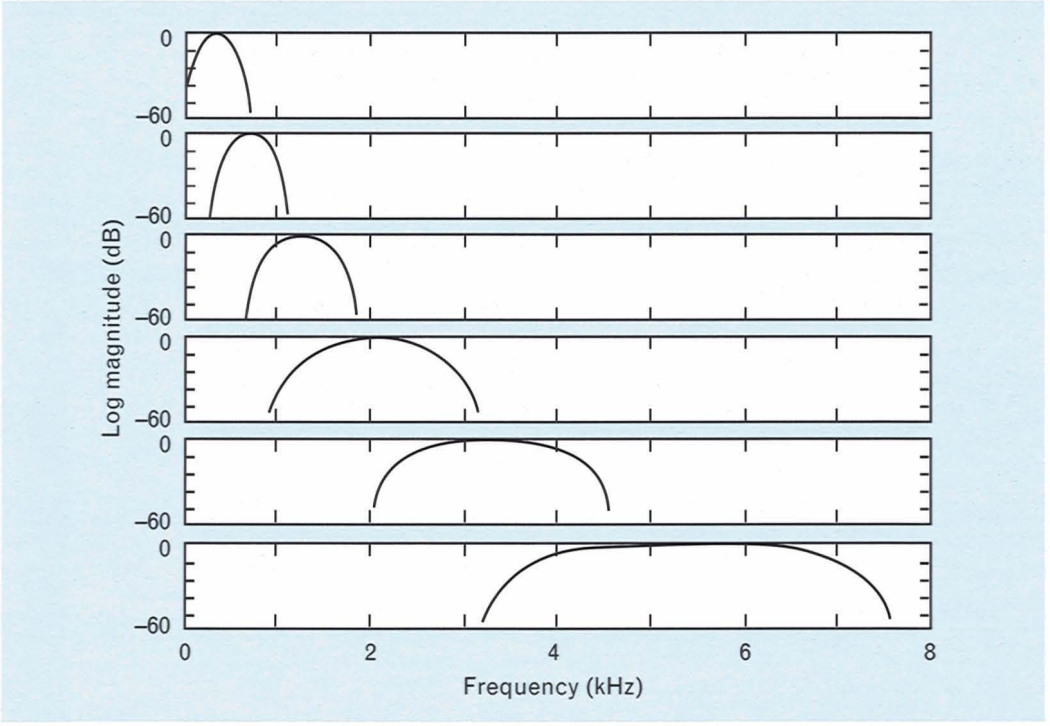


FIGURE 24. Filter bank with sharper frequency responses than 12-dB/octave Butterworth filters.

of processing algorithms that we have been exploring in a laboratory setting. These portable stimulator/processors will allow subjects full-time use of newly developed algorithms, providing the subjects with the enhanced speech reception that they enjoy in the laboratory. In addition to benefiting the individuals involved, this field use will provide valuable data concerning the long-term learning effects of subjects using new stimulation algorithms.

This model of a collaborative effort between Lincoln Laboratory and the biomedical community has resulted in a new program of funded biomedical research in the wider MIT research community. The original joint effort of Lincoln Laboratory, MIT, and MEEI has shifted to an expanded collaboration (partially as a result of the NIH contract that followed the initial IRP). The collaboration currently includes researchers and engineers from MIT, MEEI, Draper Laboratory, Research Triangle Institute, and researchers in Geneva, Switzerland, at both a research hospital and an engineering faculty. We are still hopeful that Lincoln Laboratory may be able to rejoin this activity in the future and once again provide the range of skills that enabled this effort to begin.

Acknowledgments

The authors wish to thank Joseph P. Donnelly, Ronald R. Parenti, and the rest of the IRP committee for their belief that this project was worthwhile. Alan J. McLaughlin, Lincoln Laboratory Assistant Director, Peter E. Blankenship, Associate Head of the Computer Technology Division, and Clifford J. Weinstein and Gerald C. O'Leary, Leader and Associate Leader, respectively, of the Speech Systems Technology Group, were very flexible in allowing us to proceed with this research, providing laboratory resources as required. We especially appreciate Peter Blankenship's successful efforts to help us secure follow-up support from the NIH Neural Prosthesis Program, which is currently funding the conversion of PISCES from a laboratory system to a subject-wearable device. Finally, we are grateful for the close cooperation of our colleagues at MEEI and MIT, as well as for the patience and flexibility of all our subjects.

REFERENCES

1. P.B. Denes, *The Speech Chain: The Physics and Biology of Spoken Language* (Anchor Press, Garden City, NY, 1973).
2. Working Group on Communication Aids for the Hearing-Impaired, "Speech-Perception Aids for Hearing Impaired People: Current Status and Needed Research," *J. Acoust. Soc. Amer.* **90,** 637 (1991).
3. D.K. Eddington, "Speech Discrimination in Deaf Subjects with Cochlear Implants," *J. Acoust. Soc. Amer.* **68,** 885 (1980).
4. T. Ifukube, "Signal Processing for Cochlear Implants" in *Advances in Speech Signal Processing,* eds. S. Furui and M.M. Sondhi (Marcel Dekker, New York, 1992), pp. 269–305.
5. G.E. Loeb, "The Functional Replacement of the Ear," *Sci. Am.* **252,** 104 (Feb. 1985).
6. M.F. Dorman, S. Solis, K. Dankowksi, L.M. Smith, G. McCandless, and J. Parker, "Acoustic Cues for Consonant Identification by Patients Who Use the Ineraid Cochlear Implant," *J. Acoust. Soc. Amer.* **88,** 2074 (1990).
7. T.W. Parks and J.H. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory* **CT-19,** 189 (1972).
8. J.F. Kaiser, "Nonrecursive Digital Filter Design Using the I_0-sinh Window Function," in *Proc. IEEE Int. Symp. on Circuits and Systems* (Apr. 1974).
9. B.S. Wilson, C.C. Finley, D.T. Lawson, R.D. Wolford, D.K. Eddington, and W.M. Rabinowitz, "Better Speech Recognition with Cochlear Implants," *Nature* **352,** 236 (1991).
10. G. Troullinos, P. Ehling, R. Chirayil, J. Bradley, and D. Garcia, "Theory and Implementation of a Splitband Modem Using the TMS32010," in *Digital Signal Processing Applications with the TMS320 Family* **2,** ed. P. Papamichalis (Texas Instruments/Prentice-Hall, Englewood Cliffs, NJ, 1990), pp. 221–328.
11. T.Q. Nguyen, "The Eigenfilter for the Design of Linear-Phase Filters with Arbitrary Magnitude Response," *Proc. ICASSP '91* **3,** *Toronto, 14–17 May 1991,* p. 1981.
12. R.S. Tyler, J.P. Preece, and M.W. Lowder, "The Iowa Cochlear-Implant Test Battery," Laboratory Report, University of Iowa at Iowa City, Dept. of Otolaryngology—Head and Neck Surgery (1983).

Appendix 1: Command-Line Arguments for the cbank Program

Flag	*Value Type*	*Description*	*Default*
	String	Input speech file	(no default)
	String	Output stimulation file	(no default)
-bs	Integer	Input buffer size (samples)	50
-mxf	Integer	Maximum number of filters	20
-sr	Integer	Sampling rate in Hz	10000
-sf	String	Main parameter specification file	"specfile"
-df	String	DRC specification file	"testdrc"
-v	Integer	Diagnostic verbosity level	0
-tc	(none)	Enable timing check using TMS320C30 timer	FALSE
-tr	Integer	Timer resolution in μsec	100
-aio	(none)	Use analog interface I/O (AIO) instead of file system I/O	FALSE
-ib	Integer	AIO: number of input buffers	2
-ob	Integer	AIO: number of output buffers	2
-rtm	(none)	AIO: enable real-time modification of parameters	FALSE
-rtmup	Integer	AIO: real-time update (sec)	3

Appendix 2: A cbank Main Parameter Specification File

```
VERSION=22
BEGIN_FRONTEND
# specify gain prior to AGC in dB
pregain=0
END_FRONTEND
BEGIN_FILTERS
#  format:  filter-file-name  post-gain-to-be-applied-dB
filters/0000.0800.dat   -15.4
filters/0700.1300.dat   -9.4
filters/1300.2400.dat   -4.5
filters/2300.4400.dat   -0.0
END_FILTERS
BEGIN_AGC
# values in milliseconds. first line either "enabled" or
                                                "disabled"
enabled
attack=0
release=250
END_AGC
```

Appendix 3: A cbank Dynamic Range Compression File

```
-100  -100   -50   -50
 -50   -50   -40     0
 -40     0     0     0
```

Appendix 4: Command-Line Arguments for the `pbank` Program

Flag	*Value Type*	*Description*	*Default*
	String	Input speech file	(no default)
	String	Output stimulation file	(no default)
-bs	Integer	Input buffer size (samples)	96
-mxc	Integer	Maximum number of channels	20
-sr	Integer	Sampling rate in Hz	48000
-sf	String	Main parameter specification file	"specfile"
-df	String	DRC specification file	"testdrc"
-v	Integer	Diagnostic verbosity level	0
-tc	(none)	Enable timing check using TMS320C30 timer	FALSE
-tr	Integer	Timer resolution in μsec	100
-hwave		Use half-wave rectification	FALSE
-nohilb		Use only one bandpass filter per channel	FALSE
-nocomp		No output compression	FALSE
-lcomp		Linear-interpolation output compression	FALSE
-tcomp		Table-lookup output compression	FALSE
-oclen	Integer	Number of points[1] in output compression	32
-ocname	String	Output-compression specification file	"octable"
-aio	(none)	Use analog interface I/O (AIO) instead of file system I/O	FALSE
-ib	Integer	AIO: number of input buffers	2
-ob	Integer	AIO: number of output buffers	2
-rtm	(none)	AIO: enable real-time modification of parameters	FALSE
-rtmup	Integer	AIO: real-time update time (sec)	3
-hware		AIO: hardware control mode	FALSE
-noup		AIO: no upsampling	FALSE
-fename1	String	Single-channel front-ended #1 output file	(no default)
-agcname	String	Single-channel AGCed output file	(no default)
-fename2	String	Single-channel front-ended #2 output file	(no default)
-ename	String	Multichannel envelope output file	(no default)

[1]When a linear interpolation is used, the number of points is equal to the number of line segments + 1. When a table lookup is used, the number of points is equal to the $\log_2$ of the number of table entries.

Appendix 5: A pbankMain Parameter Specification File

```
VERSION=1
BEGIN_FRONTEND
#   format:  filter-file-name  post-gain-to-be-applied-dB  downsampling-n:1
../more_atten/8k@32000.lpf.dat   0.0   2
../more_atten/2k@16000.lpf.dat   0.0   4
# all channels go thru fir #1. which go thru fir #2?
#  legend:  1:thru,0:skip
# 1-3 thru fir#2, 4-6 skip fir#2
1 1 1 0 0 0
END_FRONTEND
BEGIN_AGC
# values in milliseconds. first line either "enabled" or "disabled"
disabled
attack=0
release=250
END_AGC
BEGIN_ENVELOPE
#   format:  filter-file-name  post-gain-to-be-applied-dB  downsampling-n:1
# expecting pairs of filters, i.e. one hilbert xform pair per channel
c1@4000.cos.dat  0.0  1
c1@4000.sin.dat  0.0  1
c2@4000.cos.dat  0.0  1
c2@4000.sin.dat  0.0  1
c3@4000.cos.dat  0.0  1
c3@4000.sin.dat  0.0  1
c4@16000.cos.dat  0.0  4
c4@16000.sin.dat  0.0  4
c5@16000.cos.dat  0.0  4
c5@16000.sin.dat  0.0  4
c6@16000.cos.dat  0.0  2
c6@16000.sin.dat  0.0  2
END_ENVELOPE
BEGIN_DELAY
# the delay that we want to add between the hilbert transform and the
# low-pass filter in each channel. This number is specified in samples;
# the actual time in seconds depends on the sampling rate at the output
# of the hilbert transform. Typically, at least one of the values is
# zero; otherwise, we'd be adding artificial delay.
0 5 7 17 35 38
END_DELAY
BEGIN_SMOOTH
#   format:  filter-file-name  post-gain-to-be-applied-dB  downsampling-n:1
# if downsampling is negative, it means upsampling
# gains on output started at 12,12,6,0,0,0; now adjusted for peak output
../more_atten/400@4000.lpf.dat   12.0   3
../more_atten/400@4000.lpf.dat   12.0   3
../more_atten/400@4000.lpf.dat   12.0   3
../more_atten/400@4000.lpf.dat   12.0   3
../more_atten/400@4000.lpf.dat   12.0   3
../more_atten/400@8000.lpf.dat   12.0   6
END_SMOOTH
BEGIN_PM
# pulse modulation pattern specified as floats.
# one column per channel.  when we get to the bottom, jump to the top.
extent=12
  0   0   0   0   0 -1
  0   0   0   0   0   1
  0   0   0   0 -1   0
  0   0   0   0   1   0
  0   0   0 -1   0   0
  0   0   0   1   0   0
  0   0 -1   0   0   0
  0   0   1   0   0   0
  0 -1   0   0   0   0
  0   1   0   0   0   0
 -1   0   0   0   0   0
  1   0   0   0   0   0
END_PM
```

APPENDIX 6: THE ADVANTAGES OF QUADRATURE ENVELOPE ESTIMATION

A STRONG NONLINEARITY such as a full- or half-wave rectifier is often used to estimate the envelope of an analog signal with the output of the rectifier driving a low-pass smoothing filter to eliminate spurious harmonics generated by the process. In the sampled data domain, however, the spurious harmonics generated by the full- or half-wave rectifier cannot (generally) be removed by low-pass filtering because of the effects of aliasing. Consequently, we implement an envelope estimator that uses a pair of bandpass filters identical in frequency-response magnitude but differing by a constant 90° phase over the frequency range of interest. When the outputs of these quadrature filters are squared and summed, and the square root taken, a different estimate of the envelope results. The following discussion describes some of the differences between rectification- and quadrature-based envelope estimation.

Consider a single sinusoidal input to a bandpass filter. The output of the bandpass filter is

$$y(t) = A\cos(\omega t).$$

Full-wave rectification yields a signal whose Fourier-series expansion is

$$|y(t)| = \frac{2A}{\pi} + \frac{4A}{3\pi}\cos(2\omega t) - \frac{4A}{15\pi}\cos(4\omega t) + \frac{4A}{35\pi}\cos(6\omega t) + \ldots$$

This operation has generated a series of even harmonic terms. In the analog domain, these terms would be eliminated with a low-pass smoothing filter operation, and the result would be a term at direct current (DC) representing only the input amplitude (i.e., the envelope). In the sampled data domain, these harmonics of the input frequency at four and six times the input frequency may produce aliasing of the input frequency in the bandwidth of the low-pass filter. For example, a sinusoid at 2700 Hz will produce a sixth harmonic at a frequency of 16,200 Hz, and that sixth harmonic will, for a system sampling rate of 16 kHz, appear as 200 Hz because of aliasing effects. From the series expansion, this harmonic would be about 2/35 of the DC term, or about 6%. Higher input frequencies could produce aliasing of the fourth harmonic term at even higher levels.

For the envelope produced by the quadrature operations, the two bandpass filter outputs are Hilbert transforms of each other (i.e., the outputs are shifted by 90°), yielding

$$y_1(t) = A\cos(\omega t), \text{ and}$$
$$y_2(t) = A\sin(\omega t).$$

Next, the sum-of-squares signal is calculated as

$$y_1^2 + y_2^2 = A^2\cos^2(\omega t) + A^2\sin^2(\omega t) = A^2.$$

The squared term has no spurious frequencies, and the final square-root operation produces the constant envelope value A. Note that a low-pass smoothing operation is not even required for this simple case because no harmonics of the signal are generated.

For a more elaborate case of a two-sinusoid signal in the passband of a bandpass filter, the output (if phase offsets are ignored) is of the form

$$y(t) = A\cos(\omega t) + B\cos(\omega + \Delta)t.$$

The output of a full-wave rectifier operation is difficult to quantify for even this simple two-sinusoid case. We can, however, speculate that harmonics of the input frequencies as well as the sums and differences of the harmonics will be produced, thus resulting in aliasing for the higher-frequency inputs and the higher-order distortion terms.

If we consider the quadrature processing of the two-sinusoid signal, the outputs would be

$$y_1(t) = A\cos(\omega t) + B\cos(\omega + \Delta)t, \text{ and}$$
$$y_2(t) = A\sin(\omega t) + B\sin(\omega + \Delta)t.$$

The sum of the squares of the two bandpass-filter

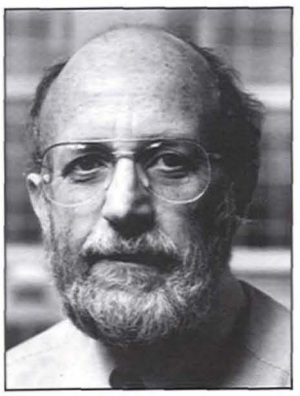
outputs becomes

$$y_1^2 + y_2^2 = A^2 + B^2 + 2AB\cos(\Delta \cdot t).$$

Note that this signal contains only constant-amplitude terms and a term at the difference frequency between the two sinusoids (in voiced speech, this difference would be the pitch frequency). These terms represent the squared envelope with no spurious frequencies that require suppression by a low-pass filter or that are susceptible to aliasing. Unfortunately, the square-root operation required to generate the envelope amplitude does generate spurious harmonics. These harmonics, however, are located only at multiples of the difference frequencies (i.e., the pitch harmonics), not the original input frequencies. In summary, for a general sum of sinusoids as an input to a pair of bandpass filters and the Hilbert envelope process, the squared output will contain only DC terms reflecting the energy of each sinusoid and sinusoidal terms at each of the possible difference frequencies. The process of taking the square root will generate harmonics of these difference frequencies, but not harmonics of the original input signals, so that there is a lower probability that aliasing, in which spurious energy is sent back into the low-pass filter range, will occur. Note that, because the envelope estimate is applied to a nonlinear compression curve before the envelope estimate modulates the output pulse train, even a perfect envelope signal that contains pitch harmonics will generate harmonics of the pitch signal.

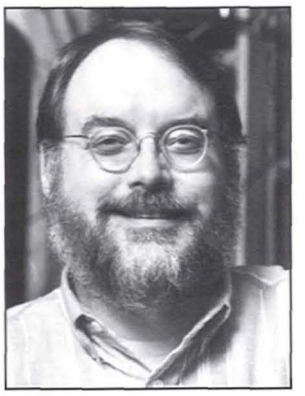
For the Hilbert envelope case in which there are only difference frequencies and some amount of difference-frequency harmonics because of the square-root operation, the envelope can be downsampled for computational savings. If the square-root distortion is ignored, then the highest difference frequency in any bandpass-filter Hilbert envelope output will be the difference between the lowest- and highest-frequency sinusoids that can be passed by that filter—a difference frequency equal to the passband width. In this case, the envelope waveform can be sampled at twice the bandwidth of the filter. If the output of the bandpass filter were a conventional amplitude-modulated signal with the full sidebands fitting within the filter bandwidth, then the modulating signal (the envelope) would be half the bandwidth of the filter. In this case, the envelope waveform could be sampled at a rate equal to the bandwidth. Because we cannot model the bandpass outputs as simple amplitude-modulated signals, we must be guided by the two-times-bandwidth rule. For the sake of being somewhat conservative, higher rates have been employed when possible.

For the case of outputs from rectification operations, there is no expectation that such signals are band limited. Conservative design principles dictate that we upsample the bandpass filter outputs before applying the rectification operations, thereby lowering the chances that aliasing, in which harmonics of the bandpass outputs are returned back into the band of interest, will occur.



JOSEPH TIERNEY
is a staff member in the Speech Systems Technology Group and a research associate at the Massachusetts Eye and Ear Infirmary (MEEI). His focus of research has been on speech coding and digital signal processing (DSP) for cochlear implants. Joe received an S.B. and an S.M. degree in electrical engineering from MIT.

MARC A. ZISSMAN
is a staff member in the Speech Systems Technology Group and a research affiliate at the MIT Research Laboratory of Electronics. His research focus has been digital speech processing, including parallel computing for speech coding and recognition, co-channel talker interference suppression, language identification, and cochlear-implant processing. Marc received an S.B. degree in computer science and S.B., S.M., and Ph.D. degrees in electrical engineering, all from MIT.



DONALD K. EDDINGTON
is the Director of the Massachusetts Eye and Ear Infirmary (MEEI) Cochlear Implant Research Laboratory (CIRL), where his focus of research has been on the electrical stimulation of the human auditory system. Don received a B.S. degree in electrical engineering and a Ph.D. degree in biophysics, both from the University of Utah.