IMPLEMENTATION OF AN AUDITORY MODEL

Dale Carnegie, Geoff Holmes, and Lloyd Smith

Department of Computer Science
University of Waikato

ABSTRACT - An auditory model has been implemented from its description in the literature. The model attempts to capture the phenomenon of auditory synchrony by detecting frequencies which dominate the output of adjacent bandbass filters. The implementation is described along with some results of processing speech with the model.

INTRODUCTION

Auditory models have recently shown promise as preprocessors for both low bit rate coding and speech recognition. These models may be used to determine how best to mask quantization noise from the human listener, or may be used to determine which spectral components of the signal are most important for coding intelligible and natural-sounding speech. Such models may be broadly classified as analytical or computational (Ambikairajah, 1989). Analytical models attempt to uncover the detailed workings of the inner ear; computational models attempt to capture relevant behaviour of the auditory system, but with an emphasis on computational efficiency rather than physiological accuracy. A model that falls into the latter category has been described by Ghitza (1987). Ghitza attempts to model speech at the auditory nerve level by capturing the phenomenon of auditory synchrony.

Auditory synchrony is the property of formants to dominate the temporal responses of bands of auditory nerve fibers (Deng, 1987). Ghitza models this phenomenon by detecting frequencies that dominate the responses of adjacent bandpass filters. The resulting "in-synchrony-bands spectrum" (SBS) provides a much reduced representation of the speech signal. The motivating assumptions for the model are, first, that the amplitude of a spectral component of a signal is the number of auditory nerve fibers that fire synchronously at the component's frequency, and, second, that nerve fiber phase responses are irrelevant (Ghitza, 1987). An "in-synchrony-band" in the model is a region of adjacent filters having the same dominant frequency, although the band is thresholded and is not considered unless the number of relevant adjacent filters is greater than or equal to the threshold. The amplitude of an SBS band, then, is the number of adjacent filters having the same dominant frequency (Ghitza used 100 highly overlapping filters covering a range from 0 to 5 kHz). The following describes an implementation of this model, and describes some preliminary results from processing speech with it.

IMPLEMENTATION

Speech input to the model is bandpass filtered at 300 - 3300 Hz, and is sampled at 8 kHz. The sampled speech is mu-law encoded, but is converted to 12 bit linear for further processing. The speech is then passed through a 128 point Hamming window, and a 128 point FFT is performed every 64 points (8 ms).

Our implementation, like Ghitza's, uses 100 highly overlapping filters, spaced evenly on a logarithmic frequency scale, although ours covers a more restricted frequency range. The center frequency of the first filter is at 208 Hz. Center frequencies are spaced at 3% intervals (on a log frequency scale), with the highest at 4 kHz. Filters below 1 kHz have a -18 dB/octave slope on both lower and upper frequency sides of the center frequency, while filters above 1 kHz have a -18 dB/octave slope on the low frequency side and a -120 dB/octave slope on the high frequency side; these shapes are intended to model the tuning curves of auditory nerve fibers. Each filter modifies the amplitudes of the FFT output according to the following formula:

$$A = B * 10^{N/20},$$

where A is the modified amplitude of the FFT spectral bin, B is the original amplitude of the FFT bin,

and N is the filter response (in dB) at the bin frequency. After modification of the FFT output, the dominant frequency is determined for each filter. The SBS amplitude of each FFT spectral component is the number of adjacent filters dominated by that frequency. Figure 1 is a diagram of the SBS analyzer.

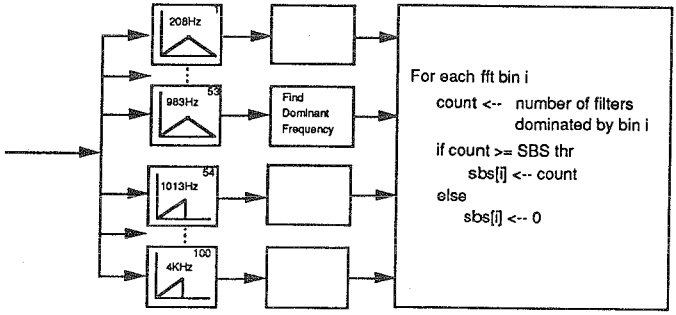


Figure 1. In-synchrony-bands spectrum analyzer.

Because each filter contributes only the value of its dominant frequency, filter gains are arbitrary and are implemented with center frequencies on a continuously rising slope (of 18 dB/octave) from 208 Hz to 4 kHz.

RESULTS

The auditory model was tested as suggested by Ghitza -- by using it to control resynthesis of speech processed by the model. The critical spectral components, as determined by the model, were passed to an inverse FFT to recreate the speech input. The inverse FFT used the original amplitudes and phases of the spectral components chosen by the model; thus, the SBS was used only to determine the critical spectral components and did not provide the base representation for resynthesis. Figure 2 is a diagram of the analysis/synthesis system as controlled by the SBS analyzer.

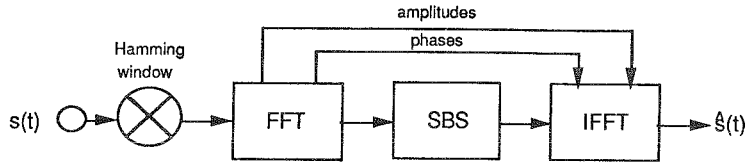


Figure 2. SBS analysis/synthesis.

The first inputs were steady state vowels. Figures 3 and 4 show the power spectra of the original vowels, AH and IY, and of the reconstructed vowels; both vowels were spoken by a male.

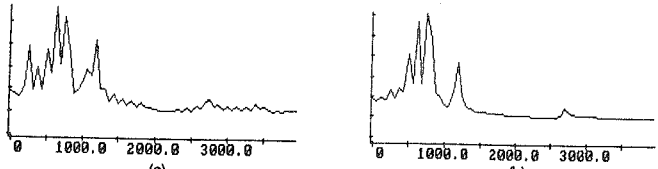
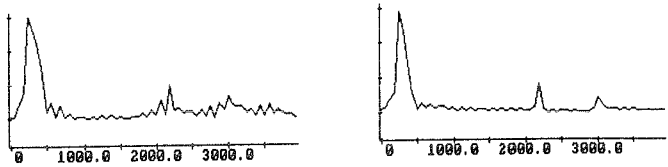


Figure 3. Power spectra of vowel AH: (a) input; (b) resynthesized after SBS processing.



(a) (b)

Figure 4. Power spectra of vowel IY: (a) input; (b) resynthesized after SBS processing.

As can be seen from the figures, both vowels are reconstructed with a high degree of spectral accuracy.

Figure 5 shows the SBS spectrum, across time, for the utterance "we were away a year ago," again spoken by a male. Ghitza (1987) did not report the SBS thresholds that he found useful; in this reconstruction we used a threshold of 10, meaning that a spectral component was used in resynthesis only if it was the dominant frequency in at least 10 adjacent filters.

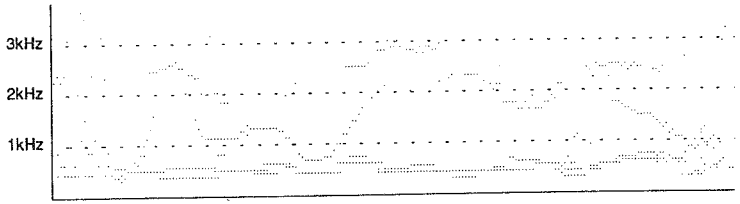


Figure 5. SBS spectrum, across time, for "we were away a year ago."

The SBS representation is, for the most part, 2 or 3 formants of the voiced portion of the utterance. Silence, at beginning and end, shows more scattered activity; most of the speech is reconstructed using 4 or 5 spectral components per frame, with little degradation in the quality of the reconstructed speech. Figure 6 illustrates the SBS response to frication in the utterance "Sally sells seashells by the seashore."

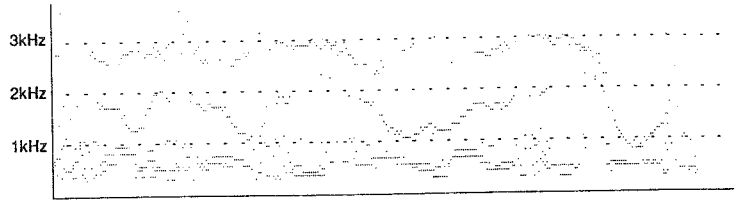


Figure 6. SBS spectrum, across time, for "Sally sells seashells by the seashore."

Ghitza (1987) reported a "tonal artifact" in speech resynthesized under SBS control, and concluded that the artifact was introduced by inaccurate reconstruction of unvoiced portions of the speech. In our implementation of the model, such a tonal artifact was not present in our reconstruction of "we were away a year ago," but was present in the reconstruction of "Sally sells seashells by the seashore." This is in agreement with Ghitza's conclusion, and it calls into question the utility of using SBS controlled reconstruction for speech embedded in noise. To test reconstruction of noisy speech, we introduced noise into test utterances in 2 ways. The first method introduced impulse noise, simulated by adding spikes at varying amplitudes into the speech sampled data. On reconstruction, the spikes were smeared into short tonal bursts.

The second method of adding noise mixed the sampled speech with white noise at several signal to noise ratios. In all cases, an objectionable tone was audible in the reconstructed speech, although the utterance was still easily intelligible; the amplitude of the added tone was clearly related to the signal to noise ratio. Resynthesis using higher SBS thresholds lowered the amplitude of the tonal artifact, but did not eliminate it. Figure 7 shows the SBS spectrum, across time, for the utterance "we were away a year ago" with white noise added at roughly 3 dB average signal to noise ratio.

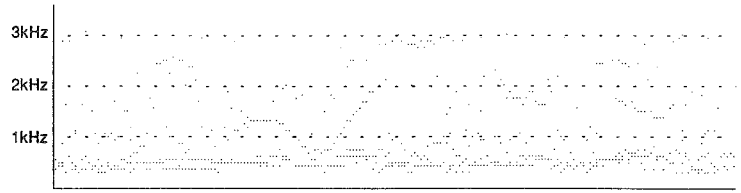


Figure 7. SBS spectrum, across time, for "we were away a year ago," with added white noise.

We did not find it necessary to preemphasize high frequencies when using filters with 18dB/octave slope (Ghitza (1987) used a 6dB/octave high pass filter on input speech). We did, however, run experiments using continuously variable slope filters, with very broad bandwidth for high frequencies. In this implementation, it was necessary to preemphasize high frequencies in order to keep high formants from being masked by low formants. The second formant of AH, for example, (figure 3a) was masked by the first formant unless the speech was differenced before SBS processing. This implementation, while hard to justify physiologically, was less sensitive to SBS threshold setting.

CONCLUSIONS

The auditory model used in these experiments simulates auditory synchrony by using a simple filterbank and counting the number of adjacent filters dominated by the same frequency. When used to control reconstruction of input speech, the model is able to greatly reduce the number of spectral components used in the reconstruction, although the reconstruction of unvoiced or noisy speech results in the introduction of some tonal artifact due to the imposition of a discrete spectrum on aperiodic input.

Modifying the filter bank to broaden filter bandwidths at higher frequencies produced a model that was less sensitive to thresholds but required high frequency preemphasis to avoid masking of high formants by lower ones.

Continuing work will experiment further with different slopes for the simulated cochlear filters and with removing the tonal artifact from reconstructed speech. This particular model was chosen for implementation because of its computational simplicity as well its emphasis on auditory synchrony; the ultimate goal will be to use it as an integral part of a low bit rate coding system.

REFERENCES

Ambikairajah, E., Black, N.D., Linggard, R. (1989) *Digital filter simulation of the basilar membrane,* Computer Speech and Language **3**, 105-118.

Deng, L. and C.D. Geisler (1989) *Responses of auditory-nerve fibers to nasal consonant-vowel syllables,* J. Acoust. Soc. Am. **82**, 1977-1988.

Ghitza, O. (1987) *Auditory nerve representation criteria for speech analysis/synthesis,* Proc. IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-35, 736-740.

IMPLEMENTATION OF AN ACTIVE COCHLEAR MODEL ON A TMS320C25

E. Jones (*) and E. Ambikairajah (**)

(*) Department of Electronic Engineering,University College Galway, Ireland
(**) Department of Electronic Engineering, Regional Technical College, Athlone, Ireland

ABSTRACT - This paper, which is the second of two papers describing the development of an active cochlear model submitted to this conference, outlines the implementation of the model on a Texas Instruments' TMS320C25 single-chip digital signal processor. The cochlear model contains both passive and active elements, in line with recent research findings in the field of auditory physiology. The passive system is operational at normal stimulus amplitudes, while the active system comes into play for low-amplitude stimuli. Tests of the implemented model have been carried out using sinewaves. This implementation could be used as a physiologically-based front-end processor for a speech recognition system.

INTRODUCTION

This paper describes the real-time implementation of an active cochlear model. The hardware consists of an analogue interface, a signal processing system based on the TMS320C25 digital signal processor (DSP), and an interface to an IBM PC to enable the PC to read the output of the DSP for plotting.

The auditory model used in this implementation consists of a cascade of digital filters, each of which has a different resonant frequency in the speech spectrum. It contains 22 sections covering the frequency range from 250 Hz to 3.5 KHz, and operates at a sampling frequency of 8 KHz. This model incorporates both passive and active elements. The passive system is operational at normal amplitudes, while the active system comes into effect at low amplitudes (Davis, 1983). Figure 1 shows a block diagram of one section of the auditory model. At each section of the model, the fluid pressure at the input to that section is converted into mechanical displacement. The pressure transfer function in the z-domain, for a single section, is given by (Ambikairajah et al., 1989; Linggard & Ambikairajah, 1986) :

$$\frac{V_o(z)}{V_i(z)} = K \frac{1 - a_0}{1 - a_0 z^{-1}} \frac{1 - b_1 + b_2}{1 - b_1 z^{-1} + b_2 z^{-2}} \frac{1 - a_1 z^{-1} + a_2 z^{-2}}{1 - a_1 + a_2} \qquad (1)$$

where V_i is the pressure input to the section, V_o is the pressure output from the section, a_0, a_1, a_2, b_1 and b_2 are digital filter coefficients and K is a gain factor. The membrane displacement transfer function is given by:

$$\frac{V_m(z)}{V_i(z)} = K \frac{1 - a_0}{1 - a_0 z^{-1}} \frac{(1 - b_1 + b_2) z^{-1}}{1 - b_1 z^{-1} + b_2 z^{-2}} \qquad (2)$$

where V_m is the membrane displacement.

The transduction of mechanical displacement to electrical energy takes place in the inner hair cells. The model of the inner hair cell used in the present work is a capacitor model, in which the input voltage corresponds to the spatially differentiated membrane displacement output of the auditory model. To detect the presence of a frequency component (or a formant when the stimulus is a speech signal), a decision algorithm is carried out on the inner hair cell outputs. If the amplitude of a component is below a certain threshold, then the corresponding section of the model is switched to the active (high-Q) state. This is accomplished by changing the digital filter coefficients for that section. Thus, implementing an active section requires no increase in computation over a passive section.

Extensive tests of the real-time model have been carried out using sinusoidal stimuli of varying frequency. Results of such tests are presented in this paper.

HARDWARE USED IN THE IMPLEMENTATION

Figure 2 shows a block diagram of the hardware used in the implementation of the active cochlear model. The hardware was constructed on a double-width wire-wrap board, and consists of three main components:

(1) an analogue-to-digital interface;
(2) a signal processing section based on the TMS320C25 fixed-point digital signal processor;
(3) an interface to an IBM PC which incorporates Static RAMs (SRAMs) which are used to pass results from the DSP to the PC.

It was decided to use a 12-bit parallel analog-to-digital converter (ADC), as opposed to a serial CODEC device, to eliminate the time necessary to convert the log-PCM output of the CODEC into the linear format required by the DSP. The ADC used is the Analog Devices AD ADC84-12, which has a conversion time of approximately 10 μs. This is connected to input port 0 of the DSP. The hardware is designed such that execution of an IN instruction by the DSP reads in the current latched input sample, and also sends a start conversion signal to the ADC (Troullinos & Bradley, 1987).

The DSP contains 544 16-bit words of internal RAM, which is used in this implementation to store coefficients and previous input/output values. For program space, external memory is used. To allow the system to operate at full speed, without wait states, the external memory must have an access time of less than 40 ns from address valid. The TMS27C291 (2Kx8) EPROM, which has an access time of 35 ns from address valid, is used. Since the instruction width of the DSP is 16 bits, parallel banks of EPROMs are used (the amount of program memory in the system is 4K words).

The third component of the system is the interface between the DSP and the PC. A digital input/output card is used to provide the interface between the PC bus and the SRAMs. At the end of each 16 msec processing frame, the DSP writes the output of the cochlear model to the SRAMs. The PC can then read these values from the SRAMs for storage, graphical display etc. To prevent bus contention, handshaking is carried out between the DSP and the PC using the DSP's BIO and XF pins. The SRAM used in this case is the MK41H68N-20, which has an access time of 20 ns from address valid.

Since the TMS320C25 uses fixed-point arithmetic, attention has to be paid to the issues of arithmetic overflow and scaling, e.g. in the case of a resonant section with a gain of greater than 1 at the resonant frequency. The TMS320C25 contains an on-board timer which can be programmed to generate a software interrupt after a certain number of machine cycles. Since the machine cycle time of the DSP is 100 ns, the timer is programmed to generate an interrupt after 1250 cycles, corresponding to a sampling interval of 125 μs.

This implementation of the active cochlea contains 22 digital filters. In order to implement an active cochlear model with more filters, a DSP with a shorter machine cycle time must be used. To minimise the number of data bus cycles, as much as possible of the data memory resides in the DSP's on-chip RAM. Since the program and data buses are multiplexed onto the same lines externally, instructions which involve an access to external data memory require an extra machine cycle for execution. This increases the number of machine cycles per filter, thus reducing the number of filters which can be implemented for a given sampling interval.

RESULTS OF THE REAL-TIME IMPLEMENTATION

The passive/active cochlear model was tested with sinusoidal stimuli. Figure 3(a) shows the membrane displacement output of the model 80 ms after the application of a sinewave of frequency 500 Hz to the input of the first filter. Figure 4(a) shows the corresponding inner hair cell output. The peak at filter number 14 is below the threshold level and, as a result, filter number 14 is switched to the active state (Figures 3(b) and 4(b)). When a filter is switched to the active state, it is allowed to remain in that state for a certain period of time, after which it is switched back to the passive state (see Figures 3(c) and 4(c)).

CONCLUSIONS

A real-time implementation of a model of the cochlea, incorporating both passive and active elements, has been described in this paper. Results obtained when the model was excited with sinusoidal stimuli were presented. The model is capable of providing increased amplification to a low-amplitude stimulus, by switching the appropriate filter to the active state. The results presented here are in good agreement with those obtained from a 22-filter model simulated on a PC using floating-point arithmetic.