A Comparison of Signal Processing Front Ends for Automatic Word Recognition

Charles R. Jankowski Jr., Hoang-Doan H. Vo, and Richard P. Lippmann

Abstract—This paper compares the word error rate of a speech recognizer using several signal processing front ends based on auditory properties. Front ends were compared against a control mel filter bank (MFB) based cepstral front end in clean speech and with speech degraded by noise and spectral variability, using the TI-105 isolated word database. MFB recognition error rates ranged from 0.5 to 26.9% in noise, depending on the SNR, and auditory models provided error rates as much as four percentage points lower. With speech degraded by linear filtering, MFB error rates ranged from 0.5 to 3.1%, and the reduction in error rates provided by auditory models was less than 0.5 percentage points. Some earlier studies that demonstrated considerably more improvement with auditory models used linear predictive coding (LPC) based control front ends. This paper shows that MFB cepstra significantly outperform LPC cepstra under noisy conditions. Techniques using an optimal linear combination of features for data reduction were also evaluated.

I. INTRODUCTION

All speech recognizers include an initial signal-processing front end that converts a noisy and/or degraded speech waveform into features useful for further processing. The front end is required to extract important features from the speech waveform that are relatively insensitive to talker and channel variability unrelated to speech message content. This first stage also reduces the data rate into later stages of the speech recognizer and attempts to decrease redundancy inherent in the speech waveform.

The vast majority of front ends are based on standard signal processing techniques, such as filter banks, linear predictive coding (LPC), or homomorphic analysis (ceps). There has also been interest in front ends based on known properties of the human auditory system. Some of these front ends are linear but with parameters that correspond to auditory properties (e.g., filter bank bandwidths increasing with frequency [1]). Most of the auditory-based front ends, however, are nonlinear since this is believed to be the case for many physiological and/or perceptual processes in the auditory system.

Manuscript received August 19, 1994; revised October 21, 1994. This work was supported by the Advanced Research Projects Agency. The associate editor coordinating the review of this paper and approving its publication was Dr. Xuedong Huang.

C. R. Jankowski is with Lincoln Laboratory and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Lexington, MA 02173 USA.

H. D. H. Vo was with Lincoln Laboratory and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Lexington, MA 02173 USA. He is now with PCSI Inc., San Diego, CA.

R. P. Lippmann is with Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02173 USA.

IEEE Log Number 9411835.

Front ends based on the auditory system have been shown to outperform more conventional signal processing schemes for speech recognition tasks [2]–[4]. Recent work has also explored the use of data reduction techniques such as linear discriminant analysis (LDA) to generate reduced feature sets. Such techniques have been shown to be successful in speech recognition tasks, especially when speech is degraded by noise or spectral tilt [5]. In these evaluations, a particular front end is typically compared against one control front end for a given speech recognition task. The control front end, recognition task, and speech corpus all differ across studies, making it difficult to compare front ends fairly.

This paper compares several front ends, including a single control front end that provides a low error rate, by evaluating the word error rate of a speech recognition system using these front ends. These front ends are based on theorized properties of the human auditory system. Techniques using optimal linear combination of features for data reduction are also evaluated. The control front end is also compared with another conventional front end, and it is shown that the choice of the control is important when conducting front end evaluations.

II. FRONT ENDS

One conventional front end, three outputs from two auditory models, and data reduction algorithms based on optimal linear combination of input features are compared in this paper. This section briefly describes the front ends and data reduction techniques. The indicated references provide more detailed information. As this section will show, all front ends begin with a linear filter bank or with energies from the output of such a filter bank, with filter spacings and bandwidths increasing with frequency. This characteristic is based on knowledge about the behavior of the basilar membrane in auditory systems. The differences between the front ends are primarily due to the signal processing that occurs after this first stage.

A. Mel Filter Bank Cepstra

The control front end is a mel filter bank (MFB) based cepstral transformation, which has been shown to outperform other conventional signal processing methods [1]. The speech waveform is first multiplied by a 20-ms-long Hamming window every 10 ms, and a discrete Fourier transform (DFT) is computed for each windowed waveform segment. In the
frequency domain, a vector of log energies is computed from each waveform segment by weighting the DFT coefficients by the magnitude frequency response of a filter bank. The center frequencies of the filters are spaced equally on a linear scale from 100 to 1000 Hz and equally on a logarithmic scale above 1000 Hz. Above 1000 Hz, each center frequency is 1.1 times the center frequency of the previous filter. Each filter’s magnitude frequency response has a triangular shape that is equal to unity at the center frequency and linearly decreasing to zero at the center frequencies of the two adjacent filters.

Each vector of log energies is then processed by an inverse cosine transform [1], creating a vector of MFB cepstral coefficients. The cepstral coefficients are then used as input features to the speech recognizer. On a SPARCStation 2 workstation, the MFB cepstral front end operates in roughly one third real-time at a data rate of 100 frames/s.

B. Seneff Auditory Model

The first auditory front end described in this work is a model proposed by Seneff [3]. The first stage is a bank of 40 time-domain infinite-impulse-response linear filters that are carefully designed to match physiological data on the response of a cat’s basilar membrane to acoustic stimuli [6]. The second stage of Seneff’s front end models the transduction stage of signal processing in the inner ear or the translation of basilar membrane motion into auditory nerve firing patterns. An efficient approximation to a half-wave rectifier, followed by an algorithm modeling short-term adaptation, a low-pass filter, and an automatic gain control, perform this transformation.

The third stage of Seneff’s model has two branches. The “mean rate” branch simply processes each of the channel outputs from the second stage with a low-pass filter. This filtered output models the average firing rate of the auditory nerve fibers corresponding to a given channel. The second “synchrony” branch uses a “generalized synchrony detector” to measure the extent that a channel’s second stage output is periodic with the characteristic period of that channel. (This period is $f/2$, where $f$ is the center frequency of the channel’s first stage linear filter.) Experiments from auditory physiology demonstrate that higher auditory processing centers in the brain make use of both rate and synchrony information [7], [8]. For all evaluations with Seneff’s model, there are two separate sets of results: one for the mean rate branch and another for the synchrony branch, both computed at a rate of 100 feature vectors/s.

On a SPARCStation 2 workstation, Seneff’s model operates in approximately 40 times real time. This is primarily due to the time domain nature of all processing stages in the model, which are quite computationally expensive when compared with the frequency domain techniques used in the MFB cepstral front end. Time-domain processing is necessary, however, due to the nonlinearities in the model.

C. EH Auditory Model

A second auditory front end is the ensemble interval histogram (EIH) model developed by Ghita [2]. The EH model begins with a physiologically based linear filter bank much like the first stage of Seneff’s model. A bank of “level crossing detectors” then processes each of the filter bank channels. Each level crossing detector has an amplitude threshold and records the times when the first stage channel output both crosses the amplitude threshold and is increasing. These level crossing events simulate the firing of auditory nerve fibers. At any given time, the times corresponding to the last 20 such crossings are recorded, with the constraint that no crossings occurring more than 40 ms before the current time are stored.

Each channel in the second stage has 12 level crossing detectors, corresponding to 12 logarithmically spaced amplitude thresholds. Different thresholds model the fact that auditory nerve fibers have different sensitivities to stimuli. Defining $bt$ to be the time between adjacent crossings for one channel, each detector records the frequencies corresponding to these times using the relation $f = 1/kt$ and accumulates this data into a frequency histogram. These histograms are then combined across levels into a single histogram for each channel. An EIH is obtained by combining these histograms across all channels. An EIH can be calculated at any desired sampling rate: One hundred samples/s was chosen for this study in order to have one consistent sampling rate for each front end.

On a SPARCStation 2, the EIH model runs in approximately 120 times real time. Approximately 40% of the total computation time is devoted to first-stage linear filtering, and another 40% is used to upsample the first-stage filter outputs by a factor of eight before the level crossing detectors. This upsampling is necessary to obtain adequate frequency resolution at high frequencies.

D. Data Reduction Techniques

Techniques are also used to reduce the dimensionality of the feature vectors produced by the front ends, with the goal of reducing recognition error rate across test conditions by concentrating relevant information into a small number of features. This section describes the two data reduction techniques used: principal components analysis and linear discriminant analysis.

Principal Components Analysis: Principal components analysis (PCA) is a linear transformation on an input feature space, producing a modified feature space, according to

$$
\vec{\xi}' = \hat{A} \vec{\xi},
$$

where

- $\vec{\xi}_i$: ith input feature vector
- $\vec{\xi}'_i$: corresponding transformed feature vector
- $\hat{A}$: transformation matrix.

$\hat{A}$ is determined such that the individual elements of all $\vec{\xi}'_i$ are uncorrelated, i.e., the covariance matrix of the set of transformed feature vectors should have nonzero elements only on the main diagonal. Computing $\hat{A}$ is relatively straightforward; the rows of $\hat{A}$ are the eigenvectors of the covariance matrix for $\vec{\xi}_i$ [9]. $\hat{A}$ is then applied to every training and testing speech frame to create feature vectors that are passed to the recognition system. Reducing feature vector dimensionality is accomplished by including only the eigenvectors correspond-
ing to the largest $N$ eigenvalues of the covariance matrix, where $N$ is the number of output features desired.

**Linear Discriminant Analysis:** LDA is a linear transformation on the input feature space, as is PCA, and the same method is used to reduce the dimensionality of the feature vector. While PCA uncorrelates the input features, though, LDA maximizes some measure of class separability. Fukunaga [10] provides more details on this procedure. Because LDA utilizes a measure of separability between classes of input data, each input feature vector must first be associated with a class before the transformation matrix can be calculated. LDA is thus a supervised procedure. As with PCA, the transformation matrix is applied to all training and testing speech frames before the recognizer.

Hunt and Lefebvre [5] used a signal processing scheme called integrated mel-scale linear discriminant analysis (IMELDA), which applied LDA to the raw and delta outputs of a mel-scale filter bank and showed significant error rate reduction in noise and with spectral shaping. These results were with a digit task, using the orthographic transcription as the class for the LDA procedure.

This paper used a different technique to generate class data. A hidden Markov model (HMM) speech recognizer first processed the speech used to train the LDA front end. During recognition, for each word the HMM system produces a maximum likelihood state sequence or a most probable mapping from input frame to HMM state. Starting with one cluster for each HMM state, leader clustering [11] was then applied to reduce the number of clusters. Each cluster was then considered a class for the purposes of training the LDA. Using the phonetic labels from the TIMIT database as classes was also attempted, but the results were poor, and no further work was performed using this procedure.

**III. ISOLATED WORD EXPERIMENTS**

This section describes a collection of experiments that were performed with the various front ends, using an isolated-word speech recognizer.

**A. Experimental Conditions**

**Database and Recognizer:** All isolated-word experiments used the TI-105 isolated word database [12], which has a vocabulary of 105 aircraft command words. Eight speakers (five male and three female) spoke five training tokens of clean speech and two testing tokens of clean speech for each vocabulary word. The words were not carefully designed to be maximally confusable as in [1] but still represent a large variety of acoustic-phonetic environments and correspond to a realistic isolated-word recognition task.

An isolated-word HMM recognizer was used that modeled each word as a sequence of eight states. While in each state, the probability density function for an observation vector was a multivariate normal distribution with a diagonal covariance matrix. This covariance matrix was shared across all HMM states across all words, and mean vectors were estimated separately for each state. A diagonal and tied covariance matrix was chosen because of the relatively small amount of training data available and based on past experience with the TI-105 database [13]. All results shown are overall word error rates averaged across all eight speakers after each speaker was used for separate speaker-dependant training and testing.

**Front End Processing:** The outputs of the MFB cepstral, Seneff, and EIH front ends can all be viewed as "pseudo-spectra." These models all begin with linear filter banks; by plotting each channel's output versus the center frequency of the channel's first stage linear filter, a spectrum-like representation can be achieved. These pseudo-spectra are then processed to generate feature vectors for the speech recognizer.

For the MFB cepstra, Seneff, and EIH front ends, feature vectors were generated by performing an inverse cosine transform on the spectra or pseudo-spectra, generating cepstral or cepstral-like coefficients. With the Seneff and EIH models, the use of a diagonal covariance matrix in the recognition system suggested this cepstral transformation; the raw features are most likely not uncorrelated, as a diagonal covariance matrix assumes. Poli [14] has suggested that the inverse cosine transform performs a crude principal components analysis, which, as we have seen, uncorrelates the input features. Uncorrelating input features forces the properties of the data to match the constraints of the recognizer model (diagonal covariance matrix) for best possible recognition error rate.

With the MFB cepstra and Seneff mean-rate front ends, the recognizer used 12 cepstral coefficients and 13 cepstral first difference coefficients; absolute energy (the first cepstral coefficient) was not used. For the Seneff synchrony and the EIH model, 24 cepstral coefficients and 25 first difference coefficients were used. More coefficients were used for synchrony and EIH since the pseudo-spectra from these two front ends exhibit sharper peaks. For a rational transfer function with poles closer to the unit circle (and thus sharper peaks in the magnitude frequency response), the cepstrum decays more slowly [15], justifying the use of more coefficients. An alternative would have been to use the same number of cepstra for each auditory model; we felt that this might have eliminated important information from the cepstral feature vectors of the synchrony and EIH front ends that was present for both the MFB cepstral and mean-rate models.

When using techniques such as PCA and LDA, the untransformed feature vector ($\mathbf{z}_i$ in (1)) is a concatenation of standard and first difference coefficients, where a "coefficient" is the output of a front end before any cepstral transformation (e.g., MFB filter bank energies, Seneff third-stage outputs). From the transformed vector ($\mathbf{z}_i^T$ in (1)), only $N$ features are passed to the recognizer, where $N$ is the desired final feature vector size. Whenever comparing PCA or LDA to another front end, $N$ is the same as the size of the feature vector for the other front end in order to not artificially reduce the amount of data from one of the front ends.

**Noise:** The error rates of the recognizer with front ends and data reduction techniques were evaluated under noisy conditions. In real-world applications, speech recognition systems most certainly operate under noisier conditions than is typical of clean, quiet laboratory recordings. Speech "babble" was added to clean TI-105 speech waveforms using a database of
Speech recorded at different sites and under different recording conditions exhibit varied long-term spectral characteristics. Here, the difference between the TIMIT speech database [19] and a pilot corpus for the Wall Street Journal database [20] is calculated. There is considerable difference in the average spectra from the two databases; Wall Street Journal data has considerably more low and high frequency information than TIMIT. The filtering effect of a telephone channel was also simulated; this exhibits the well-known bandpass effect, with cutoffs at approximately 300 and 3400 Hz.

Multistyle Training: Most recognizers are trained only with samples of quiet, normally spoken speech. Multistyle training trains a recognizer using corrupted (e.g., noisy, filtered) speech as well as clean speech. This technique has been previously successful in recognizing normal and stressed speech [18]. When using multistyle training for each sample of clean speech in the training set, a sample at each testing SNR or linear filtering condition was also included. The testing database is identical to that of experiments performed without multistyle training.

Verifying Parameters in the MFB Cepstral Front End: Before comparing the Senneff and EIH auditory models to the MFB cepstral front end, parameters of the EIH algorithm were set to provide low error rate on samples from the test corpus. We decided to verify that the existing parameters of the MFB cepstral front end were also good choices for the test corpus and recognition system used in this study.

The number of filters in the linear filter bank was first modified to 13, 16, 24 (current), 31, and 47 filters by changing the filter spacing in both the linear and nonlinear frequency regions. The edges of the filter responses remained at the center frequencies of the adjacent filters so that filter bandwidth decreased as the number of filters increased.

The number of cepstral coefficients was also varied to be 10, 13 (current), and 16 cepstra before calculating first differences and dropping the first cepstral coefficient corresponding to raw energy. This resulted in overall feature vector sizes of 19, 25, and 31.

The HMM recognizer, using a MFB cepstral front end with these various parameters, was tested across all noise conditions and spectral variability conditions. Not surprisingly, the MFB cepstral front end responded quite differently to different noise levels and spectral variability conditions, but overall, the current choice of parameters (24 filters and 13 cepstral coefficients) provided, on average, the lowest error rate across conditions.

B. Results in Noise

Standard Training: The HMM recognizer was trained with clean speech and tested with noisy speech for the first set of experiments in noise. Fig. 2 shows the error rates of the recognizer with all front ends as a function of SNR. Considering the disparate processing methods, all front ends provide similar error rates. For clean speech and speech at 30 dB SNR, all auditory models perform similarly to the MFB cepstral front end. Below 30 dB SNR, the auditory models perform slightly better than the MFB cepstral front end. Here,
the difference between the auditory models and the MFB cepstral front end is from 0.6 to 4 percentage points (depending on the SNR), which all exceed the binomial standard deviation of the MFB cepstral error rate for the testing SNR (0.3 to 1.0%). The error rate of all front ends increases considerably at very high noise levels, making the usefulness of the system questionable.

Multistyle Training: A second set of experiments evaluated the effectiveness of multistyle training on all front ends with noisy speech. In these experiments, training is performed using speech at all SNR’s, instead of only using clean speech samples. Fig. 3 shows the error rate average across all front ends both without and with multistyle training. The individual front ends exhibited trends similar to that seen in Fig. 2. For SNR of 24 dB or higher, multistyle training results in a slight error rate reduction. For SNR of 18 dB or lower, multistyle training provides substantially lower error rates (by 1 to 16 percentage points). These results are consistent with [18], where multistyle training provided substantial reductions in error rates.

Data Reduction Techniques: Several experiments tested the effectiveness of data reduction techniques on all front ends with noisy speech. The first experiments used PCA. For PCA, multistyle speech was used to both derive the PCA transformation matrix and train the HMM speech recognizer. PCA provided minimal improvement in noise as compared with standard multistyle training.

In another set of data reduction experiments, LDA was applied to the mel filter bank outputs. The recognizer achieved lowest error rate when no clustering was performed on the input states; the raw states from the HMM recognition were used as classes. The recognizer also performed best when both the LDA transformation matrix was derived, and the speech recognizer was trained with multistyle speech data. Even so, LDA performed worse than MFB cepstra for all SNR’s.

C. Results with Spectral Variability

Standard Training: Fig. 4 shows the recognition results for all front ends trained in clean speech and tested with the six spectral variability conditions. Differences between the error rates of the front ends were extremely small (never more than 1% difference between the error rates for best and worst front ends). For the conditions with low baseline error rates (e.g., “Clean” and “90° Head Shadow”), the MFB cepstra outperforms the auditory models by a small amount. For other conditions, the synchrony and EIH outputs slightly outperform the MFB cepstra front end, which shows slightly better results than the mean-rate output.

Multistyle Training: Experiments were conducted using multistyle speech for training. Multistyle training resulted in consistent but small reductions in error rate for every linear filtering condition, with a decrease in error rate of as much as 0.4 percentage points.

Data Reduction Techniques: Principal component analysis was applied to multistyle training data. Unlike the PCA results in noise, the spectral variability results show a small (maximum of 0.2 percentage points) but consistent improvement using PCA. LDA was also used with the filtered speech for
clean speech and the head shadow conditions, MFB cepstra is better, although LDA performs better for all other conditions (by as much as 1.4 percentage points). This is in contrast with the results of LDA experiments in noise, which showed LDA performing worse than MFB for all SNR’s. For the more difficult spectral variability conditions—as indicated by error rate with clean speech—data reduction techniques led to reduced error rate.

IV. COMPARISON OF FFT-BASED AND LPC-BASED CONTROL FRONT ENDS

Other sites have obtained large reductions in error rates with auditory models. For example, Ghizta at AT&T Bell Laboratories showed significant error rate reduction (e.g., approximately 20% error rate with EIH as opposed to approximately 65% with a more conventional front end at 12 dB SNR) with an isolated word recognition task using the EIH model [2]. Carnegie-Mellon University (CMU) also reported significant improvement (e.g., approximately 60% error rate with an auditory model versus almost 100% for LPC cepstra at 10 dB SNR) using both the mean-rate and synchrony branches of Senef’s auditory model with a continuous speech database and recognizer [4].

These previous studies show significantly larger reduction in error rate with auditory models than was presented in this paper. There could be several reasons for the differences. First, this work used an FFT-based control front end, whereas both Ghizta and CMU used a LPC based cepstral front end as the control. The CMU front end also uses a bilinear transform to approximate a “mel-like” frequency warping [21]. LPC-based front ends do not, however, typically perform well in noisy environments.

The CMU experiments were conducted with a speaker-independent paradigm; Ghizta’s experiments and those in this paper are speaker dependent. In addition, baseline error rates in this work are significantly lower than in these previous studies. We have seen that as the baseline error rate increases, the differences between the conventional front end and auditory models increase. It was noted that at such high baseline error rates, it is questionable how useful a recognition system is. Finally, Ghizta and CMU also added artificially generated noise, whereas we used recorded speech babble.

To evaluate the difference in control front ends (LPC based cepstra as opposed to FFT filter bank based cepstra), we acquired the software for the CMU front end and compared our MFB cepstral front end to CMU LPC-based processing. We used the same database and recognizer as we have used for all other isolated word experiments and tested with both speech babble and artificially generated white noise to study the effect of the noise type. Three different LPC orders (equal to the number of predictor coefficients) were used: 8, 14, and 18.

Fig. 5 shows the results of the evaluation, where the error rate of the recognizer with the MFB cepstral front end in both white noise and speech babble is compared with the best performing LPC-based front end (with 18 predictor coefficients). The MFB cepstral front end significantly outperforms the LPC front end, especially in noise. This is not terribly surprising since finding the poles of a vocal tract system function using LPC with noisy speech can be interpreted as finding the roots of a moderate order polynomial with noisy coefficients, which are known to have high variability. Performance for both front ends was better with the speech babble than with the white noise.

These results suggest that the use of a poorer performing control front end might have caused other studies to find much more reduction in error rate with auditory models than we have found. An auditory model might perform better than an LPC-based front end as other studies have shown but perform very similarly to and only slightly better than a better performing control such as MFB cepstra, which we show here.

V. DISCUSSION

We have conducted experiments to test the effectiveness of several signal processing schemes as front ends to automatic speech recognition systems. Front ends were evaluated with an isolated word database in clean speech, speech degraded by "babble" type noise, and speech processed with linear filters simulating "real-world" acoustic transformations.

With additive speech babble as noise, auditory models can slightly reduce recognition error rate relative to MFB cepstra. With MFB cepstral error rates ranging from 0.5 to 26.9% depending on SNR, auditory models performed as much as 4 percentage points better. We see large changes at quite high baseline error rates, but here, the usability of the recognizer is questionable. With speech degraded by linear filtering, where MFB cepstra showed error rates ranging from 0.5 to 3.1%, the EIH and Senef synchrony auditory outputs provided only slight (maximum of 0.4 percentage points) improvement.

Multistyle training was quite effective with both noise and spectral variability for all front ends. Principal component analysis (PCA) provided little error rate reduction across conditions. Linear discriminant analysis (LDA) on mel-scale filter bank outputs, using clustered HMM states as class, performed worse than MFB cepstra in noise but better than MFB cepstra for the more difficult spectral variability conditions.
Cepstral coefficients derived from a simple linear predictive coding (LPC) model perform significantly worse than MFB cepstra when a recognizer is trained in clean speech and tested in noise. As we have seen, this is theoretically justified; therefore, the same result would be expected for continuous speech systems. This result is likely the primary reason that other studies show much more reduction in error rate with auditory models than we show here; two other studies use an LPC-based front end as the control.

Future work in the area of front end evaluation might focus on continuous speech since current speech recognition systems are primarily targeted for continuous speech tasks.

VI. CONCLUSION

Front ends based on the human auditory system perform comparably to, and can slightly reduce the error rate of, an MFB cepstral-based speech recognition system for isolated words with noise and some spectral variability conditions. The magnitude of the reduction in error rate is small, though, relative to the required increase in computation time.

We have seen the importance of having a suitable control front end when testing auditory front ends. When training with clean speech and testing with noisy speech, LPC cepstra is not suitable; it performs significantly worse than MFB cepstra. Much of the difference that others have found between the error rate of recognizers with auditory models and more standard front ends can perhaps be explained by the lack of a high performing control front end.

As described, the first stage of the control MFB cepstra front end is a bank of linear filters with bandwidths increasing with frequency above 1000 Hz. MFB cepstra, therefore, can be viewed as an approximation to the linear first stage of the Sen- eff and EIH auditory models. The primary difference between the MFB cepstral, Seneff, and EIH front ends is therefore the nonlinear processing that occurs after the filter bank. One could interpret the results in this paper to then imply that speech processing based on these nonlinear effects in the human auditory system cannot provide appreciable speech recognition performance improvement. One might also argue that linear techniques are sufficient to code the relevant information necessary for high-performance speech recognition, and thus, any further improvement in overall system performance must come from the later stages of the system. We do not believe this is necessarily true. The human auditory system is sufficiently complex that years of intensive research in the areas of auditory physiology and perception have left even some fairly basic questions of auditory function unanswered. The models used here are gross simplifications, with components chosen not only to model phenomena that are currently believed to be relevant for speech recognition but also to minimize computational complexity. The methods used for converting auditory front end outputs into feature vectors were largely dictated by the structure of the speech recognition system. For a fair test of the effectiveness of auditory models, more work is necessary to obtain improved ways of incorporating features from auditory models into speech recognizers. As auditory function and speech perception are better understood, new parameters important for speech recognition should be uncovered.

ACKNOWLEDGMENT

During this study, many of those involved with the formulation and testing of the evaluated front ends were most helpful. These include D. Paul (MIT Lincoln Laboratory), S. Seneff (Spoken Language Systems Group, Massachusetts Institute of Technology), O. Ghitza (AT & T Bell Laboratories), and R. Stern and Y. Ohshima (Carnegie-Mellon University). Many in the Speech Systems Technology Group at MIT Lincoln Laboratory provided useful ideas and comments; T. Quatieri especially provided useful comments on the manuscript.

REFERENCES

Charles R. Jankowski Jr. was born in Bristol, PA, on October 19, 1966. He received the S.B. degree in electrical engineering from the Massachusetts Institute of Technology in 1988. From 1988 to 1991, he was with the Speech Group of the Artificial Intelligence Laboratory for the NYNEX Corporation, White Plains, NY. There, he maintained computer facilities and developed tools for speech recognition research and speech data collection. He returned to MIT in 1991 and received the S.M. degree in electrical engineering in 1992. He is currently in the Ph.D. program, where his research focuses on developing new features for speaker identification in harsh acoustic environments.

Richard P. Lippmann was born in Mineola, NY, in 1948. He received the B.S. degree in electrical engineering from the Polytechnic Institute of Brooklyn in 1970 and the Ph.D. degree in electrical engineering from Massachusetts Institute of Technology in 1978. His Ph.D. thesis dealt with signal processing for the hearing impaired.

From 1978 to 1981, he was the Directory of the Communication Engineering Laboratory at the Boys Town Institute for Communication Disorders in Children, Omaha, NE. He worked on speech recognition, speech training aids for deaf children, sound alerting aids for the deaf, and signal processing for hearing aids. In 1981, he joined Lincoln Laboratory and is currently a senior staff member in the Speech Systems Technology Group, where his focus of research has been in speech recognition, routing and system control of circuit-switched networks, neural network algorithms, statistics, and medical risk assessment.

Hoang-Doan H. Vo was born in 1969 in Vietnam. He arrived in the United States in 1979. He received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology in 1992.

Currently, he is with Pacific Communication Sciences, Inc. (PCS), a leading wireless communication company, in San Diego, CA.