Return to Main

The Real Cepstrum:

Applications:

Summary:

On-Line Resources:

LECTURE 13: CEPSTRAL ANALYSIS

- Objectives:
    - Introduce homomorphic transformations
    - Understand the real cepstrum
    - Introduce alternate ways to compute the cepstrum
    - Explain how we compute mel-frequency "cepstrum" coefficients

This lecture combines material from the course textbook:

> X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development*, Prentice Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-022616-5, 2001.

and information found in most standard DSP or speech textbooks:

> J. Deller, et. al., *Discrete-Time Processing of Speech Signals*, MacMillan Publishing Co., ISBN: 0-7803-5386-2, 2000.

Return to Main

ECE 8463: FUNDAMENTALS OF SPEECH RECOGNITION

Professor Joseph Picone
Department of Electrical and Computer Engineering
Mississippi State University

email: picone@isip.msstate.edu
phone/fax: 601-325-3149; office: 413 Simrall
URL: http://www.isip.msstate.edu/resources/courses/ece_8463

Modern speech understanding systems merge interdisciplinary technologies from Signal Processing, Pattern Recognition, Natural Language, and Linguistics into a unified statistical framework. These systems, which have applications in a wide range of signal processing problems, represent a revolution in Digital Signal Processing (DSP). Once a field dominated by vector-oriented processors and linear algebra-based mathematics, the current generation of DSP-based systems rely on sophisticated statistical models implemented using a complex software paradigm. Such systems are now capable of understanding continuous speech input for vocabularies of hundreds of thousands of words in operational environments.

In this course, we will explore the core components of modern statistically-based speech recognition systems. We will view speech recognition problem in terms of three tasks: signal modeling, network searching, and language understanding. We will conclude our discussion with an overview of state-of-the-art systems, and a review of available resources to support further research and technology development.

Tar files containing a compilation of all the notes are available. However, these files are large and will require a substantial amount of time to download. A tar file of the html version of the notes is available here. These were generated using wget:

```
wget -np -k -m http://www.isip.msstate.edu/publications/courses/ece_8463/lectures/current
```

A pdf file containing the entire set of lecture notes is available here. These were generated using Adobe Acrobat.

Questions or comments about the material presented here can be directed to help@isip.msstate.edu.

19: Linear Discriminant Analysis
(html, pdf)

LECTURE 13: CEPSTRAL ANALYSIS

- Objectives:
    - Introduce homomorphic transformations
    - Understand the real cepstrum
    - Introduce alternate ways to compute the cepstrum
    - Explain how we compute mel-frequency "cepstrum" coefficients

This lecture combines material from the course textbook:

> X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development*, Prentice Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-022616-5, 2001.

and information found in most standard DSP or speech textbooks:

> J. Deller, et. al., *Discrete-Time Processing of Speech Signals*, MacMillan Publishing Co., ISBN: 0-7803-5386-2, 2000.

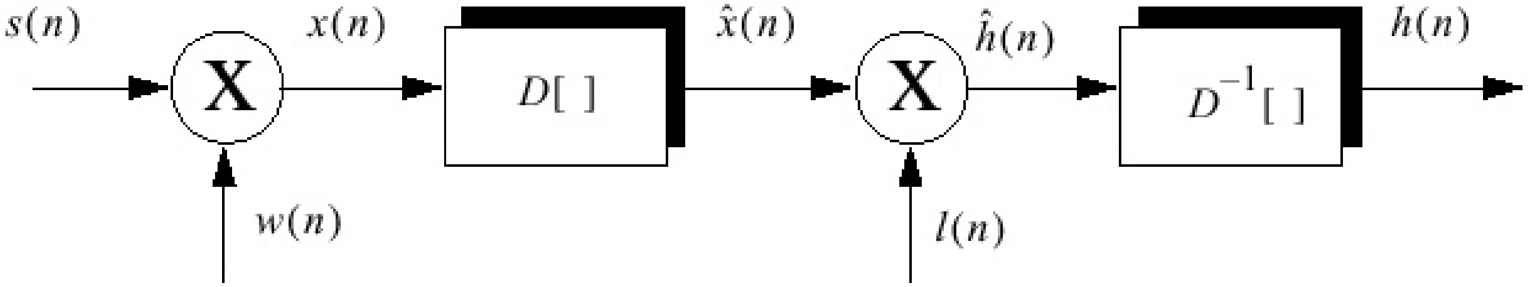
HOMOMORPHIC TRANSFORMATIONS

A *homomorphic* transformation converts a convolution into a sum:

$$x(n) = e(n) \otimes h(n)$$

$$\hat{x}(n) = \hat{e}(n) + \hat{h}(n)$$

Consider the problem of recovering a filter's response from a periodic signal (such as a voiced excitation):



The filter response can be recovered if we can separate the output of the homomorphic transformation using a simple filter:

$$l(n) = \begin{cases} 1 & |n| < N \\ 0 & |n| \geq N \end{cases}$$

Note that the process of separating the signals is essentially a windowing processing. Is this useful for speech processing?

THE REAL AND COMPLEX CEPSTRUM

The *real* cepstrum of a digital signal x(n) is defined as:

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln|X(\omega)| e^{j\omega n} d\omega$$

and the *complex cepstrum* is defined as:

$$\hat{x}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln X(\omega) e^{j\omega n} d\omega$$

where the complex logarithm is used:

$$\hat{X}(\omega) = \ln X(\omega) = \ln|X(\omega)| + j\theta(\omega)$$

$$\theta(\omega) = \arg(X(\omega))$$

Note that the real cepstrum, c(n), is the even part of complex cepstrum:

$$c(n) = \frac{\hat{x}(n) + \hat{x}(-n)}{2}$$

The word *ceps*trum was coined by reversing the first syllable in the word *spec*trum. The cepstrum exists in a domain referred to as *quefrency* (reversal of the first syllable in *frequency*) which has units of time.

THE CEPSTRUM OF POLE-ZERO FILTERS

Consider a minimum phase system with a rational transfer function:

$$H(z) = \frac{\prod_{k=1}^{Q} 1 - b_k z^{-1}}{\prod_{k=1}^{P} 1 - a_k z^{-1}}$$

Taking the complex logarithm:

$$\hat{H}(z) = \sum_{k=1}^{Q} \log(1 - b_k z^{-1}) - \sum_{k=1}^{P} \log(1 - a_k z^{-1})$$

Taking an inverse z-transform:

$$\hat{h}(n) = \begin{cases} 0 & n \le 0 \\ \sum_{k=1}^{P} a_k^{\ n}/n - \sum_{k=1}^{Q} b_k^{\ n}/n & n > 0 \end{cases}$$

It is easy to see that the cepstrum is a decaying function of time (compact). Why is this desirable?

Recalling that the real cepstrum can be computed from the even part of the complex cepstrum, the complex cepstrum can also be easily determined from the real cepstrum, c(n), as follows:

$$\hat{h}(n) = \begin{cases} 0 & n < 0 \\ c(n) & n = 0 \\ 2c(n) & n > 0 \end{cases}$$

LINEAR PREDICTION AND THE CEPSTRUM

Consider an all-pole filter:

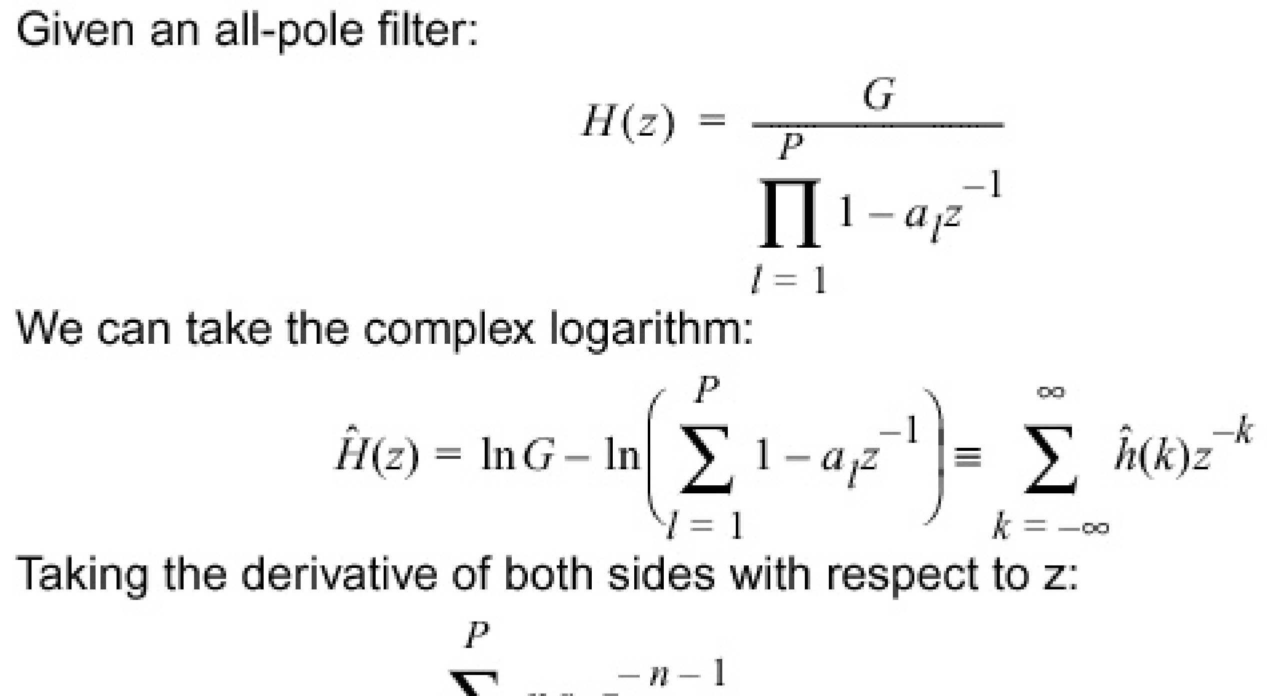
$$H(z) = \frac{G}{\prod_{k=1}^{P} 1 - a_k z^{-1}}$$

The cepstrum can be determined by the following recursion:

$$\hat{h}(n) = \begin{cases} 0 & n < 0 \\ \ln G & n = 0 \\ a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) \hat{h}(k) a_{n-k} & 0 < n \le p \\ \sum_{k=n-p}^{n-1} \left(\frac{k}{n}\right) \hat{h}(k) a_{n-k} & n > p \end{cases}$$

Note that if there are a finite number of filter coefficients, there are still an infinite number of cepstral coefficients. However, the series decays to zero and can be truncated.

The proof of this result is shown below for completeness:



Given an all-pole filter:

$$H(z) = \frac{G}{\prod_{l=1}^{P} 1 - a_l z^{-1}}$$

We can take the complex logarithm:

$$\hat{H}(z) = \ln G - \ln\left(\sum_{l=1}^{P} 1 - a_l z^{-1}\right) \equiv \sum_{k=-\infty}^{\infty} \hat{h}(k) z^{-k}$$

Taking the derivative of both sides with respect to z:

$$\sum^{P} \, {}^{-n-1}$$

Taking the derivative of both sides with respect to z:

$$\frac{-\sum_{n=1}^{P} n a_n z^{-n-1}}{1-\sum_{l=1}^{p} a_l z^{-l}} = -\sum_{k=-\infty}^{\infty} k\hat{h}(k) z^{-k-1}$$

Multiplying both sides by $-z\left(\sum_{l=1}^{P} 1 - a_l z^{-1}\right)$, we obtain:

$$\sum_{n=1}^{P} n a_n z^{-n} = \sum_{n=-\infty}^{\infty} n\hat{h}(k) z^{-n} - \sum_{l=1}^{P}\sum_{k=-\infty}^{\infty} k\hat{h}(k) a_l z^{-k-l}$$

After replacing $l = n-k$, and equating terms in z^{-n}, we obtain:

$$n a_n = n\hat{h}(n) - \sum_{k=1}^{n-1} k\hat{h}(k) a_{n-k} \qquad 0 < n \le p$$

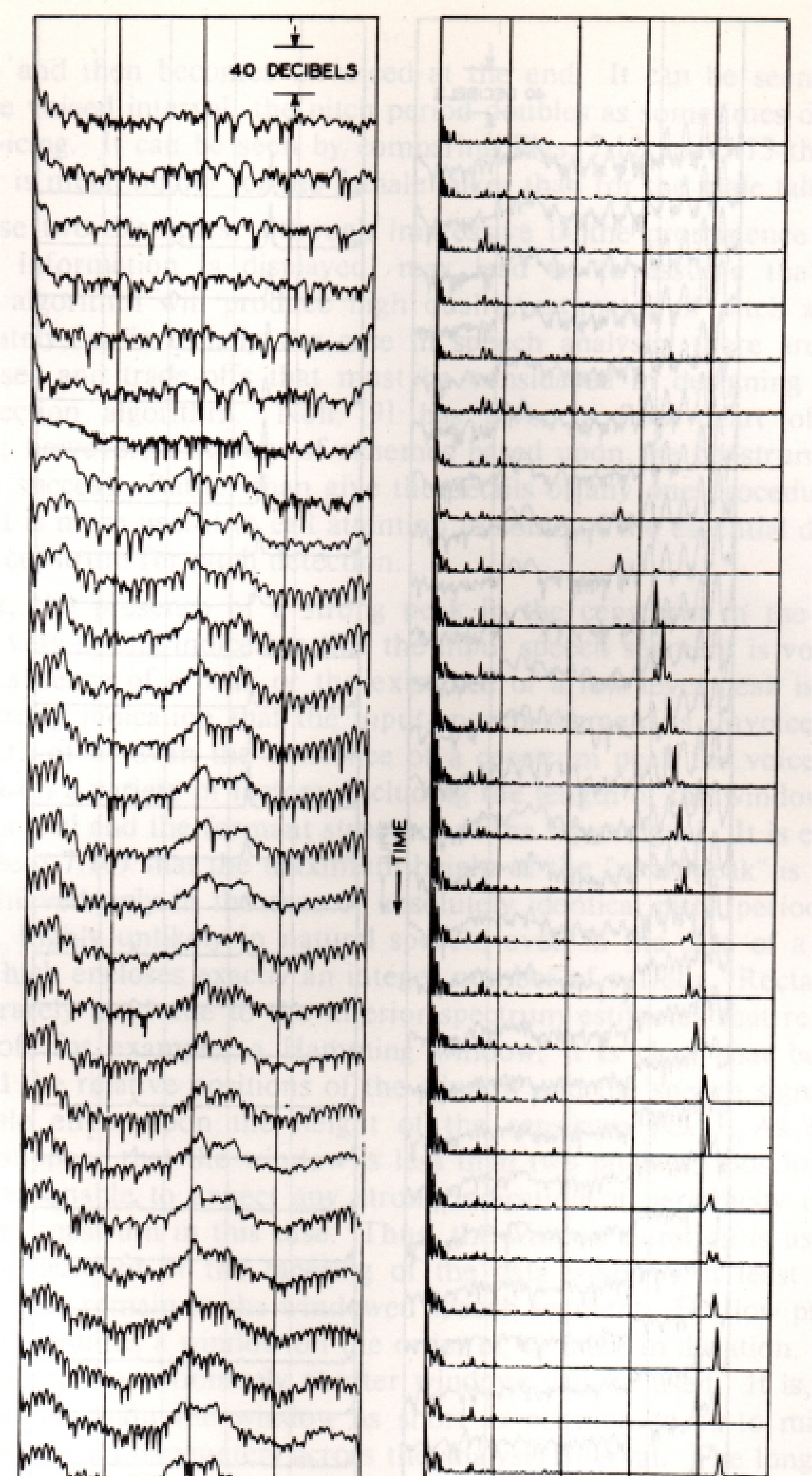
$$0 = n\hat{h}(n) - \sum_{k=n-P}^{n-1} k\hat{h}(k) a_{n-k} \qquad n > p$$

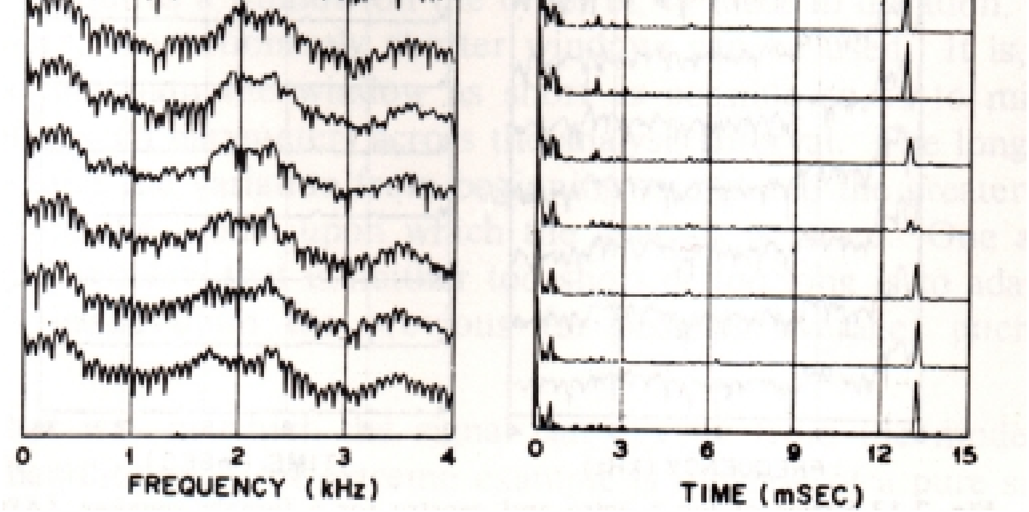
Hence, the complex cepstral cepstrum can be obtained directly from the all-pole filter coefficients:

$$\hat{h}(n) = \begin{cases} 0 & n < 0 \\ \ln G & n = 0 \\ a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right)\hat{h}(k) a_{n-k} & 0 < n \le p \\ \sum_{k=n-p}^{n-1} \left(\frac{k}{n}\right)\hat{h}(k) a_{n-k} & n > p \end{cases}$$

AN EXAMPLE OF THE CEPSTRUM FOR A VOWEL

Below is an example (from Noll) that demonstrates a typical cepstrum sequence for a vowel. The cepstrum is computed every 10 msec.





From this example, we can see two important things:

- At the onset of the vowel, where the signal is not quite periodic, the peak in the cepstrum at the fundamental frequency is not well-formed. The amplitude of this peak grows as the signal becomes more regular (periodic). The same phenomena is true for the autocorrelation function.
- It is clear that the low order coefficients of the cepstrum contain information about the vocal tract, while the higher order coefficients contain primarily information about the excitation. (Actually, the higher order coefficients contain both types of information, but the frequency of periodicity dominates.)

Hence, for speech signals, it seems the vocal tract response and the excitation signal can be separated using simple windowing in the quefrency domain.

SOURCE-FILTER SEPARATION VIA THE CEPSTRUM

An example of source-filter separation using voiced speech:

(a) Windowed Signal

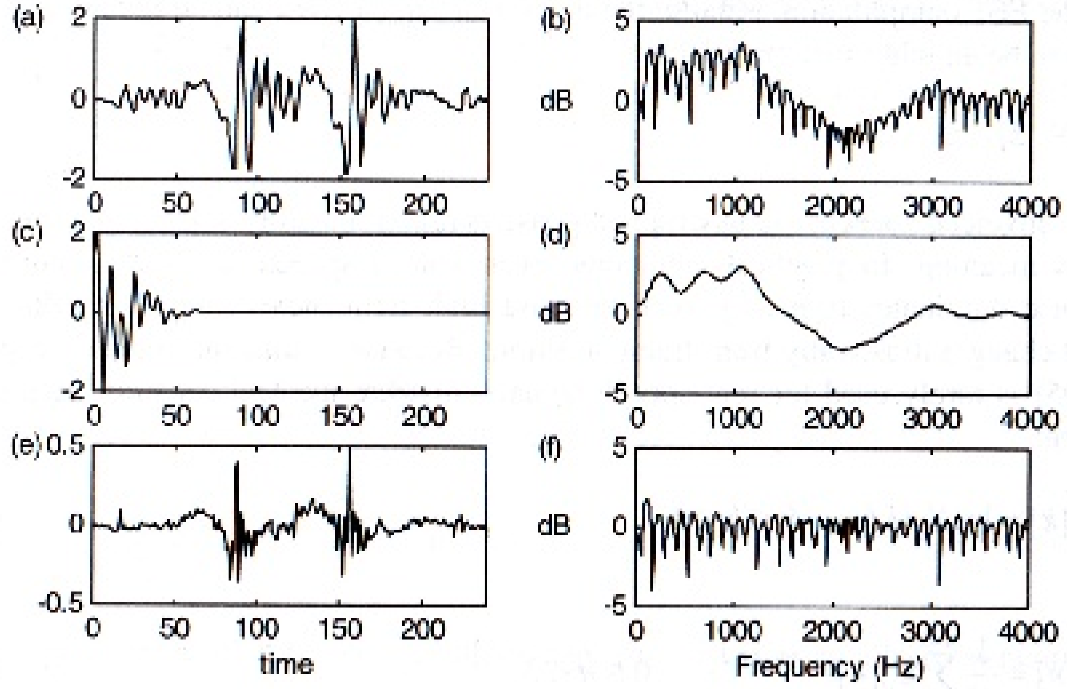
(b) Log Spectrum

(c) Filtered Cepstrum (n < N)

(d) Smoothed Log Spectrum

(e) Excitation Signal

(f) Log Spectrum (high freq.)



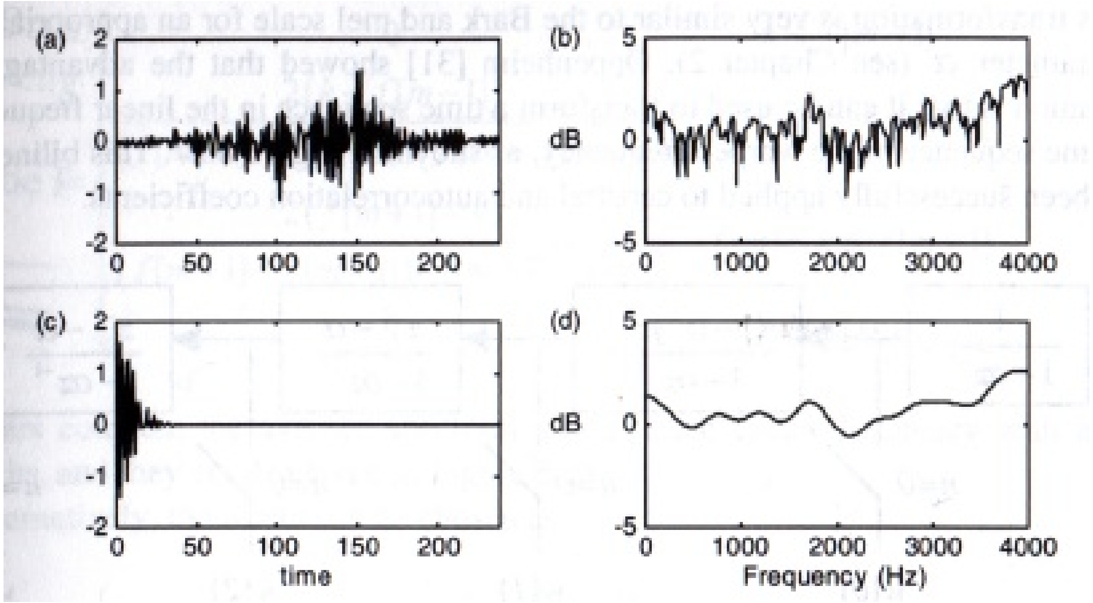
An example of source-filter separation using unvoiced speech:

(a) Windowed Signal

(b) Log Spectrum

(c) Filtered Cepstrum (n < N)

(d) Smoothed Log Spectrum



The reason this works is simple: the fundamental frequency for the speaker produces a peak in the cepstrum sequence that is far removed (n > N) from the influence of the vocal tract (n < N). You can also demonstrate this using an autocorrelation function. What happens for an extremely high-pitched female or child?

FREQUENCY WARPING USING ALL-PASS TRANSFORMATIONS

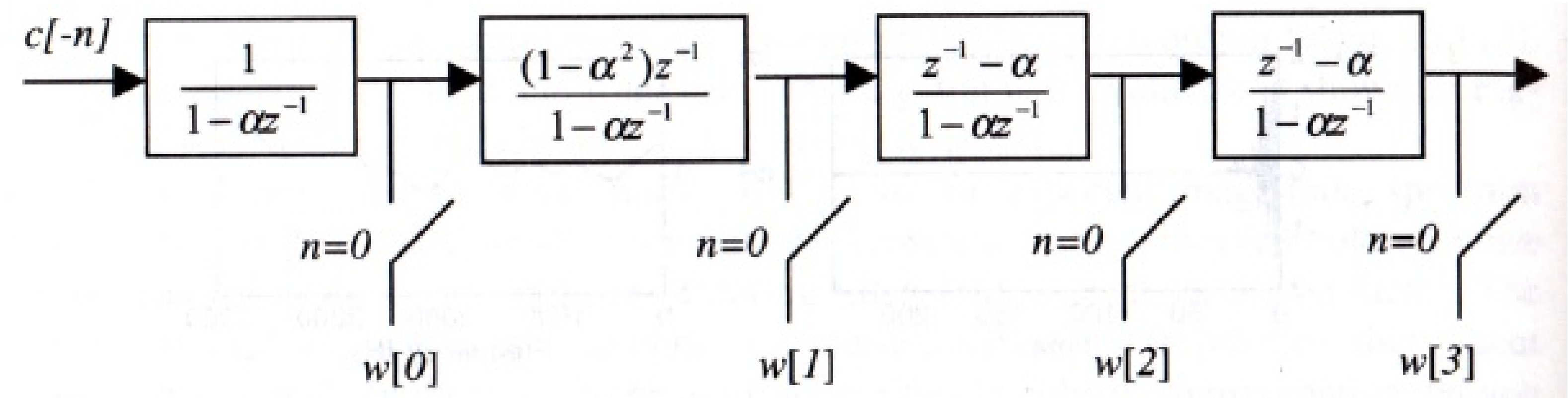
Recall the bilinear transform:

$$s = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$$

which implements a nonlinear warping of the frequency axis:

$$\Omega = \omega + 2\arctan\left(\frac{\alpha \sin(\omega)}{1 - \alpha \cos(\omega)}\right)$$

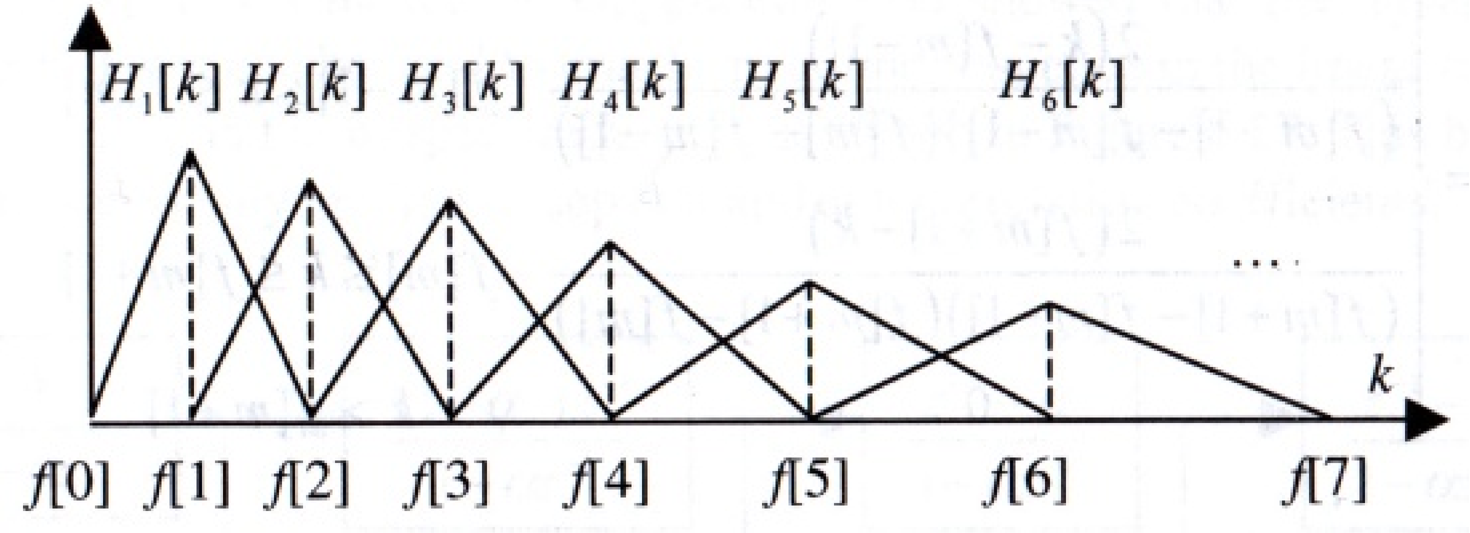
This can be implemented as a series of all-pass transformations:


c[-n]
$\frac{1}{1-\alpha z^{-1}}$
$\frac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}}$
$\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$
$\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$
n=0
n=0
n=0
n=0
w[0]
w[1]
w[2]
w[3]

The cepstral coefficients are input and the output are frequency-warped cepstral coefficients. This is an interesting way to implement speaker-specific warpings (e.g., male vs. female speakers).

MEL-FREQUENCY CEPSTRUM

Recall our filterbank, which we construct in mel-frequency domain using a triangularly-shaped weighting function applied to mel-transformed log-magnitude spectral samples:



After computing the DFT, and the log magnitude spectrum (to obtain the real cepstrum), we compute the filterbank outputs, and then use a discrete cosine transform:

$$c(n) = \sum_{m=0}^{M-1} S(m)\cos\left(\pi n\left(m - \frac{1}{2}\right)/M\right)$$

where

$$S(m) = \ln\left(\sum_{k=0}^{N-1} |X_a(k)|^2 H_m(k)\right), \qquad 0 < m \leq M$$

to compute the *mel-frequency cepstrum coefficients*. Note that the triangular weighting functions are applied directly to the magnitude spectrum, and then the logarithm is taken after the spectral samples are averaged. The resulting coefficients are an approximation to the the cepstrum, and in reality simply represent an orthogonal and compact representation of the log magnitude spectrum.

We typically use 24 filterbank samples at an 8 kHz sampling frequency, and truncate the DCT to 12 MFCC coefficients. Adding energy gives us a total of 13 coefficients for our base feature vector.

LIFTERING: WINDOWING CEPSTRAL COEFFICIENTS

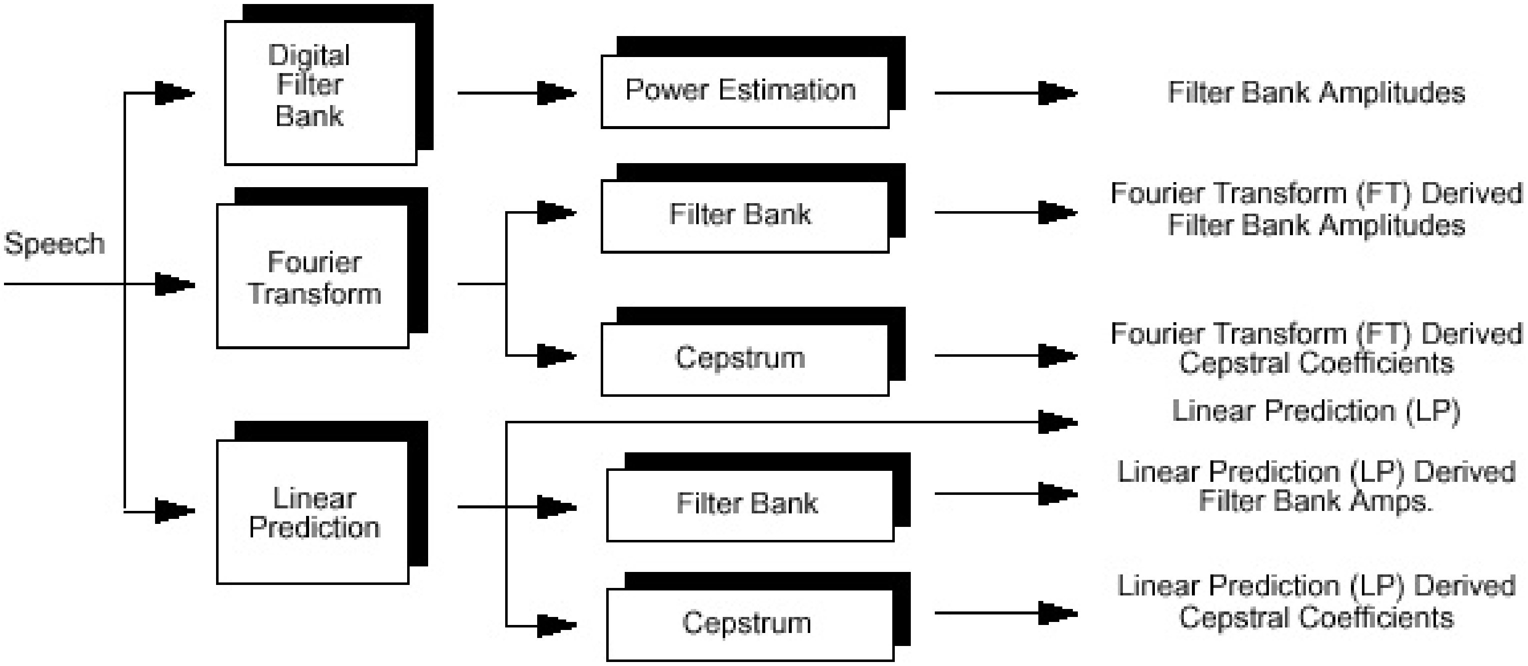
- Low order cepstral coefficients are sensitive to spectral slope, glottal pulse shape, etc.
- High order cepstral coefficients are sensitive to the analysis window position and other temporal artifacts.
- For speaker independent recognition, it is best to minimize such speaker-dependent variations in the features prior to recognition.
- We can reduce the variations in these coefficients by using a raised sine window that emphasizes coefficients at the center of the window:

$$w(n) = G\left\{ \begin{array}{ll} 1 + h\sin((n\pi)/L) & 1 \leq n \leq L \\ 0 & elsewhere \end{array} \right\}$$

- L is the number of cepstral coefficients (typically 24), and G is a constant normally designed to make the energy of the window equal to 1.

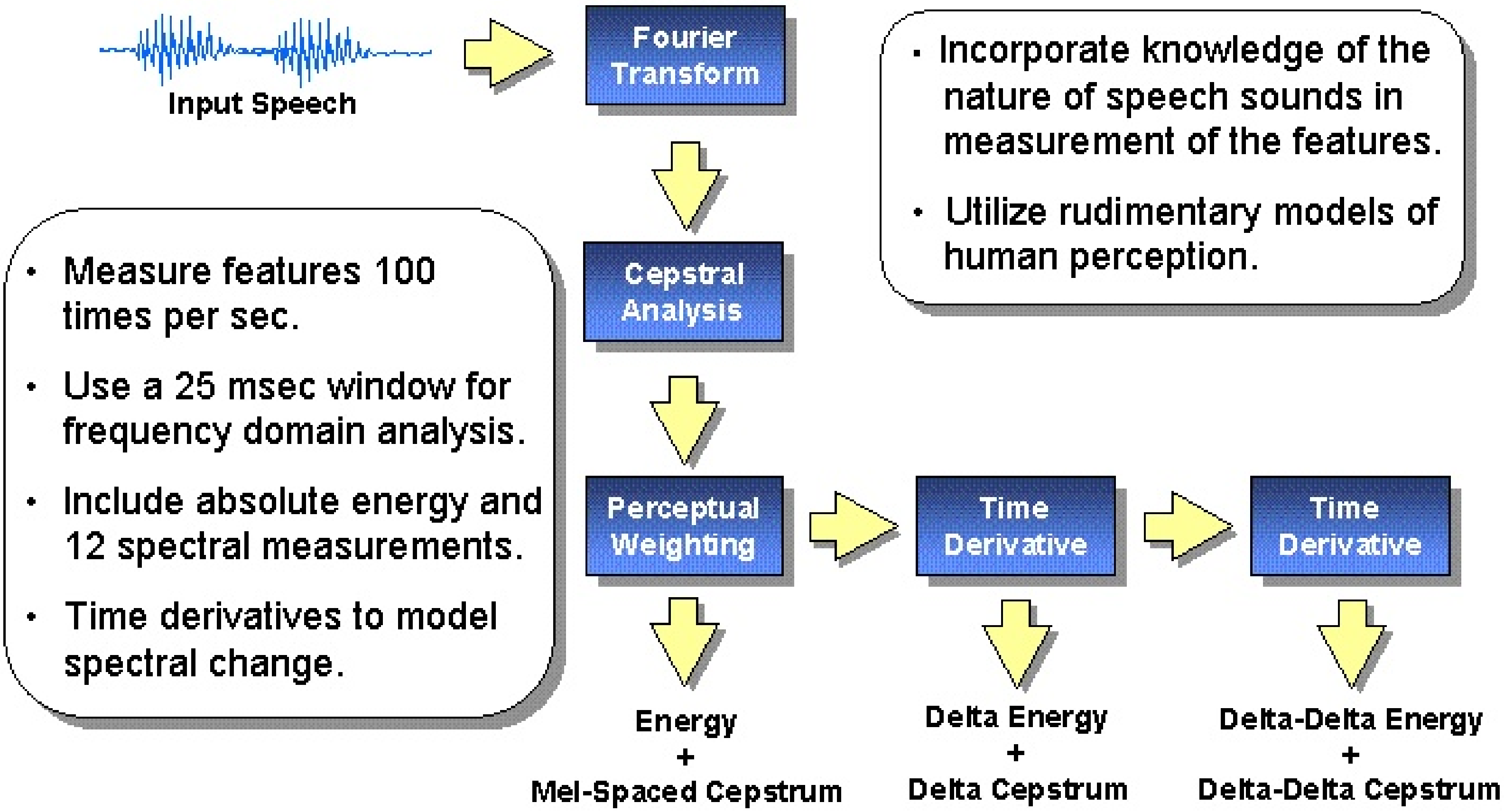
ALTERNATIVE METHODS FOR FREQUENCY DOMAIN ANALYSIS

We have now established two different ways to perform a filterbank analysis of the speech signal (temporal and spectral):


Speech
Digital Filter Bank
Power Estimation
Filter Bank Amplitudes
Fourier Transform
Filter Bank
Fourier Transform (FT) Derived Filter Bank Amplitudes
Cepstrum
Fourier Transform (FT) Derived Cepstral Coefficients
Linear Prediction
Linear Prediction (LP)
Filter Bank
Linear Prediction (LP) Derived Filter Bank Amps.
Cepstrum
Linear Prediction (LP) Derived Cepstral Coefficients

The most popular front ends are those that use cepstral coefficients dervied from the Fourier transform. Why?

A TYPICAL SPEECH RECOGNITION FRONT END



Input Speech

Fourier Transform

Cepstral Analysis

Perceptual Weighting

Time Derivative

Time Derivative

Energy
+
Mel-Spaced Cepstrum

Delta Energy
+
Delta Cepstrum

Delta-Delta Energy
+
Delta-Delta Cepstrum

- Incorporate knowledge of the nature of speech sounds in measurement of the features.
- Utilize rudimentary models of human perception.

- Measure features 100 times per sec.
- Use a 25 msec window for frequency domain analysis.
- Include absolute energy and 12 spectral measurements.
- Time derivatives to model spectral change.

Cepstral analysis

The source filter model of speech production decomposes the speech signal, s_n, into an excitation, e_n, and a linear filter, $H(e^{i\theta})$. In the frequency domain:

$$S(e^{i\theta}) = H(e^{i\theta})E(e^{i\theta}) \tag{40}$$

We wish $H(e^{i\theta})$ to represent the envelope of the speech power spectra and $E(e^{i\theta})$ to represent the fine detail of the excitation. For example, see figure 22 and figure 23. With a suitable definition of the log of a complex number ($\log z = \log|z| + i\arg\{z\}$) this may be achieved with:

$$\log(S(e^{i\theta})) = \log(H(e^{i\theta})) + \log(E(e^{i\theta})) \tag{41}$$

For most speech processing applications we require only the amplitude spectra, hence the equation is written:

$$\log(|S(e^{i\theta})|) = \log(|H(e^{i\theta})|) + \log(|E(e^{i\theta})|) \tag{42}$$

The slowly varying components of $\log(|S(e^{i\theta})|)$ are represented by the low frequencies and the fine detail by the high frequencies. Hence another Fourier transform is the natural way to separate the components of $H(e^{i\theta})$ and $E(e^{i\theta})$. This produces the cepstral analysis, shown diagrammatically in figure 28.

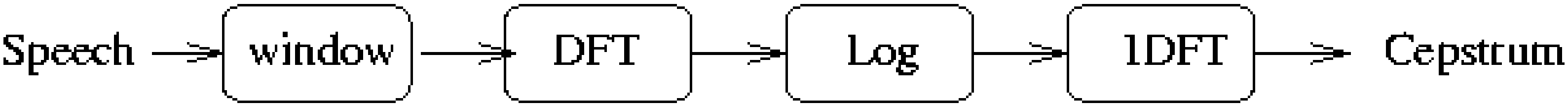


Figure 28: Cepstral analysis

For the example speech of figures 21,22,23 the resulting (real) cepstral analysis is shown in figure 29.

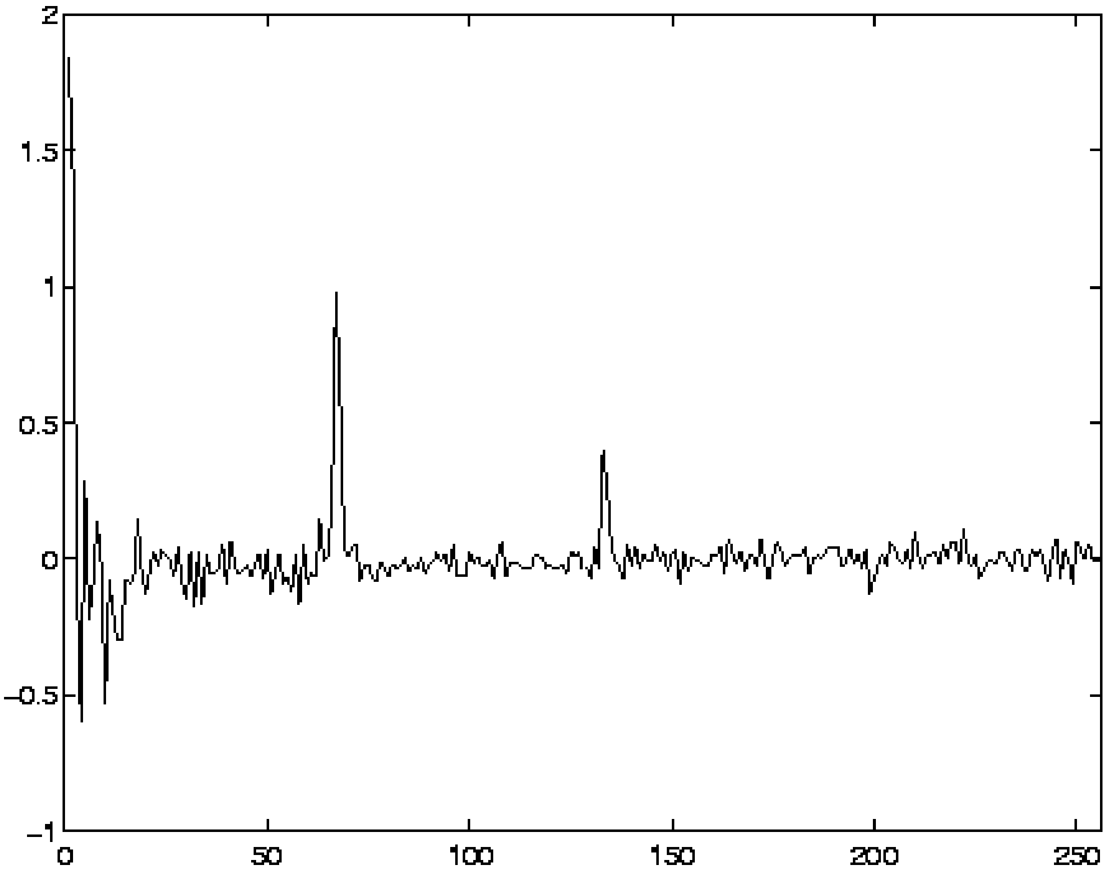


Figure 29: The full real cepstrum. Calculated using Matlab: `ifft(log(abs(fft(hamming(512) .* sig))))`

It can be seen that most of the detail occurs near the origin and in peaks higher up the cepstrum. Thus the lower numbered coefficients provide the envelope information. The remainder of the detail is mostly contained in the peaks which are separated by the pitch period (in this case about 70 sample) and provide the fine detail pitch information.

An enlargement of the few samples is shown in figure 30

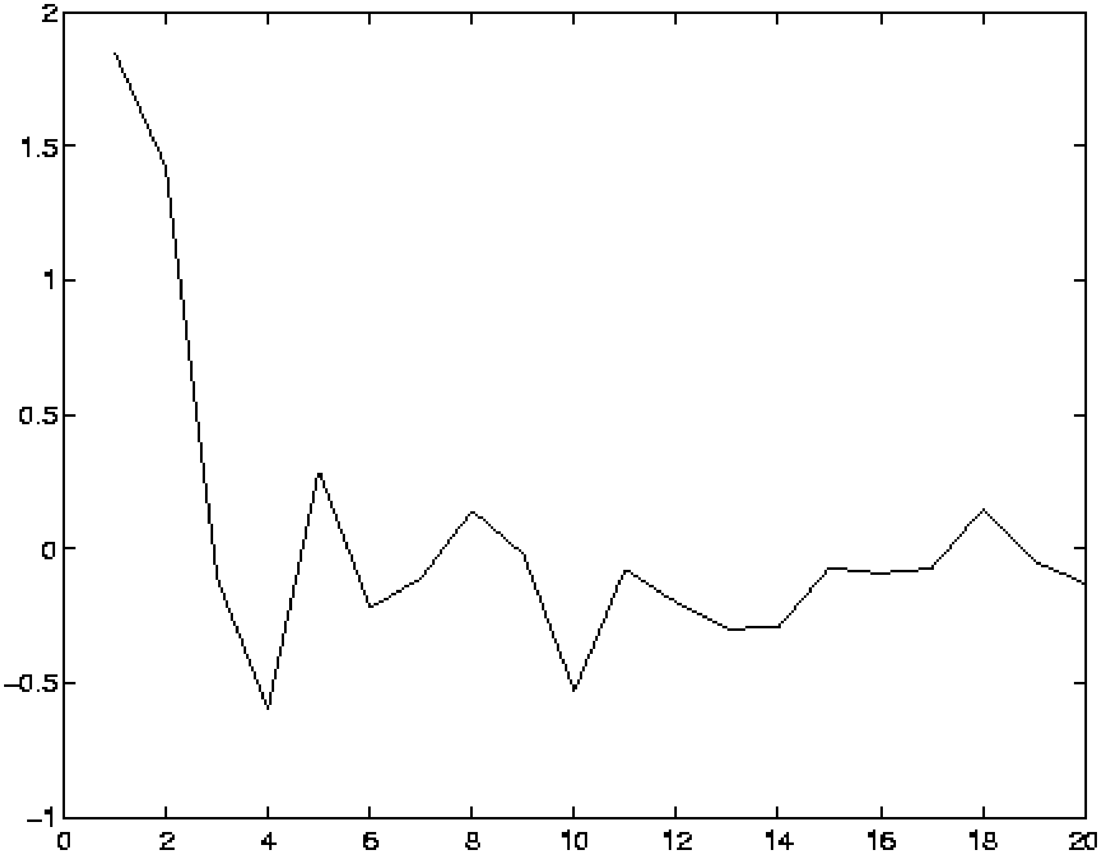


Figure 30: The first 20 cepstral coefficients

Spectral Signal Processing for ASR

Melvyn J. Hunt

Dragon Systems UK Research & Development Ltd.
Millbank, Stoke Road, Bishops Cleeve, Cheltenham GL52 4RW, UK
melvyn@dragonsys.com

ABSTRACT

The paper begins by discussing the difficulties in obtaining repeatable results in speech recognition. Theoretical arguments are presented for and against copying human auditory properties in automatic speech recognition. The “standard” acoustic analysis for automatic speech recognition, consisting of mel-scale cepstrum coefficients and their temporal derivatives, is described. Some variations and extensions of the standard analysis — PLP, cepstrum correlation methods, LDA, and variants on log power — are then discussed. These techniques pass the test of having been found useful at multiple sites, especially with noisy speech. The extent to which auditory properties can account for the advantage found for particular techniques is considered. It is concluded that the advantages do not in fact stem from auditory properties, and that there is so far little or no evidence that the study of the human auditory system has contributed to advances in automatic speech recognition. Contributions in the future are not, however, ruled out.

1. INTRODUCTION

The purpose of this paper is twofold. The first is to describe what has emerged as the “standard” acoustic analysis for automatic speech recognition and a discussion of some variants and extensions — PLP, LDA, correlation methods and variants on log-power representations — that have been found to be useful at several independent sites, especially with noisy speech. The second purpose is to discuss the extent to which the development of representations for automatic speech recognition has benefited from knowledge of the human auditory system.

Most of us who have worked in the field of speech recognition feel some pride at the progress that has been made; we may even feel amazement at how well current state-of-the-art recognizers function considering the apparently crude methods being used. Nevertheless, we have to recognize that progress in speech recognition techniques — as opposed to exploiting the possibilities of more powerful hardware — has been quite slow. Partly, this is because of the difficulty in obtaining reproducible results in the statistical estimation of recognition accuracy. This difficulty stems not simply from the large size of the test corpora needed; speech data can vary along many different dimensions, all of which can affect the performance of the recognizer. Conclusions drawn from one experiment relevant to a particular user population and application may not always be valid for a different user population or application. All that a good experiment can do is to determine performance on speech of a particular type without being able to guarantee any universal validity to the conclusions that have been drawn.

A second difficulty in making solid progress in speech recognition is the sheer complexity of a speech recognition system, especially a modern large-vocabulary system. The effect of a decision about one part of the system may depend crucially on choices made in other parts of the system. When a new method is reported to give better results than a conventional method, it is usually not evident whether this advantage would be replicated in other systems. Usually, it is also not clear whether the conventional method has been implemented optimally.

These remarks apply as much to the choice of representation of the speech signal as they do to any other aspect of a speech recognizer. In particular, it is impossible to make statements about how good particular features are in representing speech without specifying how those features will be used in the speech recognizer.

In fact, it is not even possible to draw a clear line between the representation of the speech signal (carried out in the so-called acoustic “front-end”) and the recognition process itself, especially the process used in comparing the representation of the speech to be recognized with that of specific speech sounds.

One can make a case that development of the representation of the speech signal is not where most progress has been made. In the last decade progress has arguably been concentrated in the understanding of how the distributions observed in training data are best represented and on what generalizations should be made.

Nevertheless, most people would agree that significant potential for progress still exists in finding a representation of the acoustic speech signal that best retains the information needed for effective speech recognition, especially in noise, while suppressing irrelevant information. Even without the aid of grammatical, semantic and pragmatic information, human listeners outperform today's best automatic speech recognizers [1]. In 1978, Doddington [2] reported that human listeners could achieve a recognition accuracy of 97.7% on a speaker-independent isolated-word recognition task in which the words were selected randomly from the 26,000 most common words in the Brown Corpus and no two consecutive tokens were spoken by the same speaker. This is still beyond the state-of-the-art for automatic speech recognizers. It is usually assumed that this ability is due to a superior human technique for processing sounds in general. (We cannot rule out the possibility, however, that the apparent superior abilities of humans in phonetic classification are due to higher-order active processing using implicit knowledge of the speech production process and its constraints rather than stemming from a better physiological "front-end".)

This assumption that humans possess a superior technique for analyzing speech sounds has led many researchers to attempt to reproduce characteristics of the human auditory system in the hope of obtaining improved automatic speech recognition.

Dissenters from the view that we should try to copy human hearing have sometimes argued that it is no more necessary for the front-end of a speech recognizer to model the ear than it is for an aircraft to flap its wings just because birds do. This is a false analogy. Air pre-dated both humans and birds, and each has found its own way of staying aloft in it. Speech, by contrast, did not pre-date human ears, or at least not primate ears, so it must have evolved taking account of the abilities and limitations of our hearing system. Birds' wings have not influenced the properties of air but human hearing must have influenced the properties of speech. It would be surprising if there were features in the speech signal that could contribute usefully to automatic speech recognition and yet be imperceptible to humans: if they cannot be perceived it is unlikely that they would be controlled in the speech production process. In developing automatic speech recognizers, then, there is on the face of it a definite case for studying and perhaps copying the human hearing system.

There are, on the other hand, some arguments suggesting that detailed copying of the human hearing system might not be useful. First, we do not know enough about it as a whole, and copying some parts and not others may be counter-productive. Second, we do not know much at all about how the output from the human hearing system is used to recognize speech — the human recognition system may be so different from our automatic systems that a quite different representation of the speech is needed. Finally, processing techniques that have evolved under the constraints and advantages of human physiology may be inappropriate for our current digital hardware.

The development in the choice of features for speech recognition has seen many transient fashions and apparent blind alleys. One can never be sure whether one of these methods would have proved outstanding if enough effort had been applied to it, and that may yet happen. The problem discussed at the beginning of this introduction of the difficulty of drawing general conclusions from speech recognition experiments arises here. Consequently, this paper will largely confine itself to techniques that have been found to be useful in multiple laboratories and under multiple conditions. In particular, the next section describes what has steadily evolved to being the standard acoustic analysis for speech recognition.

This paper will not deal with modulation spectrum techniques, nor with multi-stream techniques, both of which are the province of Nelson Morgan's paper [3] in this workshop.

2. THE EVOLUTION OF THE CURRENT "STANDARD" ACOUSTIC FRONT-END

From the very start of work on automatic speech recognition it was apparent that comparing speech waveforms would not be a fruitful approach: not only does the waveform lack the insensitivity to phase effects introduced by reverberation, for example, it is also sensitive to the details of the excitation of voiced and unvoiced speech in a way that would make the identification of phonetic units directly from it extremely complicated. There were some attempts to use zero-crossing rates measured from the waveform, but this was driven more by early limitations in computational power than any advantages that these features offer for speech recognition.

What was seen to be needed was a representation of the vocal tract transfer function as far as possible without influence from the details of the excitation, apart, perhaps, from its gross spectral shape, which can distinguish voiced from unvoiced sounds. The output from a bank of filters, designed originally for speech transmission in the so-called channel vocoder [4], and spanning a frequency range from 0 Hz to typically 5 kHz, provided the required properties. Computing the (log) energy output of each filter at regular intervals results in a smoothed version of the short-term speech spectrum. The channels are broad enough not to show the harmonic structure resulting from the periodic excitation of voiced speech, at least for typical male voices. The original filter-banks were analogue devices, but nowadays an FFT is normally used to simulate a filter-bank in software.

Reflecting the frequency resolution of the ear, the band-pass filters in channel vocoders were generally made to be broader at high frequencies. Most current recognizers which include a filter-bank in their front-end retain this property, usually through the so-called *technical mel-scale*, which has the center frequencies of the channels equally spaced up to 1 kHz and logarithmically spaced beyond that frequency [5].

This non-uniform channel spacing is often seen as an instance in which knowledge of properties of human hearing properties has helped automatic speech recognition. However, there is a good case that the advantage of non-uniform channel spacing stems not from the general properties of human hearing but from the specific properties of the speech signal itself. First, formant bandwidths tend to increase with increasing frequency. Moreover, unvoiced sounds, which generally have their energy and the information needed to distinguish between them concentrated at high frequencies, have much broader formant bandwidths than voiced sounds, which themselves have their energy concentrated at lower frequencies. These properties are acoustical in origin and have nothing directly to do with the resolution of the human ear.

Second, the information needed to make fine phonetic distinctions between voiced speech sounds, when measured as the contribution of each of a set of equally spaced bands, is concentrated in the lower part of the analysis range. Even with a set of bands that are logarithmically rather than equally spaced from the lowest frequency, the contribution has been found [6] to peak at around 2 kHz and decline uniformly beyond that frequency. This non-uniform distribution of information seems likely to stem from the fine control that is available on the lower formants, which itself stems from the acoustic and physiological properties of the speech production process.

The advantage of non-uniform spacing of channels in a recognizer front-end appears likely to be due to its provision of non-uniform weighting to different parts of the speech spectrum, reflecting physiological and acoustic constraints on speech production, rather than its replication of the frequency resolution of the ear.

If this view of the source of advantage for non-uniform frequency resolution in a speech recognizer is correct, then the debates on whether one should use the mel scale or a bark scale and whether the technical mel scale is a good enough approximation [*e.g.* 7] will have been unnecessary.

LPC [8], like the channel vocoder originally developed solely for speech transmission, was an early rival to the filter-bank as a front-end for speech recognizers. The computation needed for an autocorrelation-method LPC analysis is less demanding than that for the FFT used in a simulated filter-bank. LPC was originally motivated on purely acoustic grounds as a least-squares fit to the waveform generated by an unbranched acoustic tube, which behaves like an all-pole filter, and which is a good model of the vocal tract, at least for unnasalized vowels. For speech recognition purposes, it may be better to view it as a smooth parametric fit to the short-term power spectrum using a special match criterion in the spectrum domain. We will return to this point in Section 3 on PLP. Conventional LPC, unlike PLP, no longer seems to be a strong contender for front-end processing.

Returning to filter-banks, there is a problem with using the output of the filter-bank directly in a speech recognizer. Typically, recognition systems use Euclidean distances to estimate log probabilities in their spectral comparison process. For Euclidean distances to correspond to log probabilities, as they should if the recognition process is to work well, samples of the parameter set should have a multivariate Gaussian distribution about the mean for the corresponding speech sound, be uncorrelated and the parameters should be scaled such that their variances about the mean for that speech sound are all the same. (We will discuss what we mean by "speech sound" in this context in Section 4 on LDA.) For a speech signal, samples of the log energies in adjacent channels of a filter-bank do indeed have something close to a multivariate Gaussian distribution, but they are highly correlated. Fortunately, it turns out that the cosine transform of the log energies from the filter-bank, *i.e.* the mel-scale cepstrum, produces coefficients that are very close to being uncorrelated; that is, the cosine transform is close to a principal components analysis. Cepstrum coefficients can consequently be weighted such that Euclidean distances become good indications of log probabilities.

Various weighting schemes have been used in DP-based template-matching recognition schemes. Some [*e.g.* 9, 10] made the weight an empirically chosen function of the index of each cepstrum coefficient, while others have been statistically derived, from the total variance of the cepstrum coefficients [11] or from their average within-class variance [12]. For HMMs, one could in principle include a full covariance matrix in the metric used for each state of each reference model. In this case, it would not matter if the features representing the speech spectrum were correlated, and the transformation into the cepstrum would be unnecessary. However, the use of individual covariance matrices is computationally expensive and there is almost never enough training data to estimate the matrices accurately enough. Instead, the cepstrum is normally used and the weights applied are usually $(1/\Phi_i^2)^{-½}$, where Φ_i^2 is the within-class variance of the i'th cepstrum coefficient estimated for that state or basis function. Sometimes, the individual variance estimates are replaced by "trained grand variances", in which the within-class variance statistics are pooled over all states or basis functions [13]. This is equivalent to the weighting scheme used in [12] and resembles the approach used in LDA.

The cepstrum is also widely used in speech processing to deconvolve the periodic voiced excitation signal from the effects of the vocal tract [14], but in this case the cepstrum is derived from a uniformly spaced, higher resolution log spectrum, not from a non-uniform mel-scale filter bank. Truncating the mel-scale cepstrum does effectively provide a further smoothing of the spectral representation, but this is not the primary motivation for using the mel-scale cepstrum. Moreover, the property of the cepstrum of being the optimum transform for removing harmonic structure from a spectrum does not strictly apply to the mel-scale cepstrum. The primary motivation, namely the approximation of the cepstrum to a principal components analysis, allowing a weighting scheme to be effective, has a secondary useful property: since a truncated set of principal components provides the most compact representation of the variance in the spectrum that it is possible to get using linear transformations, the truncated cepstrum is also effective in providing a compact representation of the spectrum.

There is a price to be paid for applying a linear transformation such as the cepstrum to the log power spectrum when attempting to recognize speech in noise. To a good approximation, speech and an interfering noise signal can be treated as simply additive in the power spectrum before taking logs (and the log spectrum of the combined signal can then be quickly obtained by table lookup). The effect on the mel-scale cepstrum of adding noise to a speech signal is much more complex. Consequently, when Varga and Moore [15] successfully demonstrated a technique for simultaneously modeling a speech signal and a multi-state noise signal the representation in which the phonetic comparisons were made was the mel-scale log power spectrum. Compared with other techniques being used at the time, performance in very high noise was very good, but performance in lower noise, especially in speech-independent tests was less good than alternatives employing an appropriate linear transformation such as the cepstrum.

Most current representations of speech for speech recognition augment the mel-scale cepstrum coefficients with a measure of their rate of change over time [16] (the "delta-cepstrum") and often also the "acceleration" of the cepstrum coefficients (the "delta-delta-cepstrum"). In principle, it would be possible to compute the rate of change from pairs of adjacent frames and the acceleration from three consecutive frames. However, this simple frame differencing is not a robust estimator of the rate of change, which is instead normally estimated from 5 or 7 consecutive frames with the values weighted to give the best straight-line fit to the sequence. The acceleration is then derived from sequences of these values.

Because changes in overall level and the application of linear distortions to the speech signal have the effect of adding constants to the log spectrum and hence to the cepstrum, the dynamic cepstrum parameters are in principle unaffected by such changes. This makes them relatively more effective for speech recognition over channels subject to unpredictable changes of these kinds.

The technique generally known as RASTA [17, 18] takes this advantage a step further and applies "leaky" integration to the delta-cepstrum to regenerate something close to the original static cepstrum but with a running short-term normalization applied to the spectrum. This, however, is more the province of Nelson Morgan's paper.

We have now arrived at what is probably the standard current acoustic representation for speech recognition: the mel-scale cepstrum augmented with the delta-cepstrum and often the delta-delta cepstrum. The succeeding sections of this paper look at some variants and additions to this representation that have been found to be effective in several independent research centers.

3. PLP

PLP, *Perceptual Linear Prediction* [19], can be directly attributed to a single researcher, Hynek Hermansky. However, what is arguably its central idea, namely fitting the parameters of an all-pole spectrum to a mel-scale spectrum (rather than to the uniform spectrum as in conventional LPC analysis), rests on a technique described by Makhoul [20], which he called *selective LPC*. In LPC, the parameters of the all-pole fit are derived from the autocorrelation function of the speech waveform. Since the autocorrelation function is the Fourier transform of the power spectrum, it is possible to derive a pseudo-autocorrelation function from a distorted power spectrum, such as a mel-scale spectrum, and hence make an all-pole fit to the distorted spectrum. PLP makes an all-pole fit to the output of a mel-scale filter-bank, with the order of the all-pole fit being lower than the number of channels in the filter-bank. Cepstrum coefficients are then normally derived from this smoothed spectrum.

Notice that the theoretical justification for LPC, namely that the spectrum being fitted was generated by an all-pole filter, no longer applies once the spectrum has been distorted onto the mel scale. Instead, the all-pole fit in PLP has to be seen as an empirical technique fitting a smooth parametric curve to the mel-scale spectrum using a fitting criterion that emphasizes discrepancies more in the high-energy part of the spectrum than in the low-energy parts. The use of a mel-scale filter-bank to represent the spectrum has two potentially beneficial effects. First, the smoothing produced by the filter-bank reduces the influence on the all-pole fit of irrelevant spectral fine-structure such as pitch harmonics, and second it reduces the weight given in the fit to the higher frequencies in the spectrum.

In addition to the use of a mel-scale spectrum, PLP has two other perceptually motivated features. First, it applies a loudness sensitivity curve to the spectrum, reflecting the reduced sensitivity of the ear at the low and high ends of the speech spectrum; and second, it takes the cube root of the power in the spectrum before computing the pseudo-autocorrelation function. This second property is motivated by the observation that the neural firing rates in the ear tend to correspond to the cube root of the incident acoustic energy. At least in this author's experience, however, these additional features contribute rather little to the behavior of a PLP representation.

4. LINEAR DISCRIMINANT ANALYSIS

Cepstrum coefficients for speech are approximately uncorrelated. This is also approximately true for delta-cepstrum coefficients and delta-delta-cepstrum coefficients, both among themselves and across the sets. These properties are only approximate, though, and one might of course want to try using other acoustic features that do happen to be correlated with each other. Also, the cepstrum orders coefficients according to their contribution to the total variance in the signal and, as we have seen, truncation of the cepstrum retains a high concentration of the total variance in the reduced set. However, what one really wants to retain is the ability to distinguish between phonetic units, and that is not necessarily the same thing. Moreover, cepstrum truncation does not help in deciding what mix of static, delta and delta-delta coefficients should be retained. These are the motivations for using linear discriminant analysis (LDA) [21] to derive the acoustic representation.

Although the use of LDA was proposed in 1979 [22], its first reported use does not seem to have been until 1988 [23, 24]. One of those first reports was motivated precisely by an attempt to explore alternatives to the cepstrum in the form of two quite different representations generated by an auditory model that had to be combined.

LDA finds the linear transformation that diagonalizes the matrix $W^{-1}B$, where W is the within-class covariance matrix (*i.e.* the pooled covariance of individual samples about their own class means) and B is the between-class covariance matrix (*i.e.* the covariance of the class means).

This begs the question of what constitutes a "class". Ideally, one might want the class to be a phonetic unit of some kind.

Back in 1979, we were far from using phonetic units. Instead, the classes were simply taken to be the frames of the whole-word templates constructed by time-aligning and averaging together the training examples of each word. The within-class covariance matrix is then obtained by observing and pooling the covariance of the frames of the training examples about the frame of the corresponding template to which they have been aligned. The between-class covariance matrix is simply the covariance of the template frames. Nowadays, a similar procedure is used, with a class mean being a multivariate Gaussian representing a state in a sub-word model or a mixture component in such a model.

The transformed features obtained by multiplying the original features by the matrix of eigenvectors of $W^{-1}B$ are uncorrelated in their average within-class variation and all have the same average within-class variation. Their total variation is also uncorrelated. They are ordered according to a measure of discriminating power, the F-ratio, which is the ratio of between-class to within-class variance. In fact, it can be shown that under a set of assumptions to be listed below LDA provides the optimum discriminating power for a given number of features.

Under the following conditions, LDA can be shown to be the optimal transformation for pattern classification using Euclidean distances: (1) the original features have identical multivariate normal distributions about their own class means, and (2) the class means themselves have a multivariate normal distribution. Although they are not unreasonable as approximations, these assumptions are not in general strictly valid for speech. Moreover, LDA gives a recipe for optimal discrimination between Gaussian basis functions in models not between words in a language, which is what we are trying to achieve. Nevertheless, LDA is found to provide a useful transformation of an original set of features.

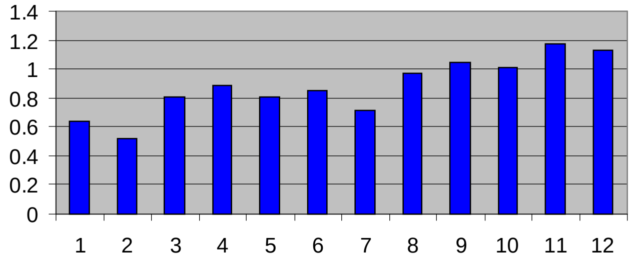
LDA is particularly useful when, for computational or storage reasons, the dimensionality of the feature set has to be reduced. The higher-order transformed features usually have very low F-ratios and, as would be expected from this, their omission from the recognition calculations makes little or no difference to recognition accuracy. Indeed, for reasons that are not well understood, in many though by no means all cases, omitting the last few transformed features has been found to *increase* recognition accuracy.

A modification to the derivation of the LDA transformation can confer robustness on the representation. Hunt and Lefèbvre [25] described a technique in which the recognition models were built from undegraded speech but the W covariance matrix was derived by observing the distribution about the means in the recognition models not only of undegraded speech samples but also of samples degraded with additive noise and linear distortion. The resulting transformation was found to be just as effective as a transformation derived only from undegraded speech in recognition tests with undegraded speech, yet performance with noisy and linearly distorted speech was much improved. There is evidence that in deriving the LDA transformation sometimes it may be worth exaggerating the degree of variation (in gain, for example) expected to be encountered in the recognition process so as to reduce further the weight given to this feature.

One common set of input features to the LDA consists of mel-scale cepstrum coefficients, delta-cepstrum and delta-delta-cepstrum. An alternative is simply to present sequences of, say, five frames of cepstrum coefficients to the LDA as the feature set and let the LDA decide how best to combine them into dynamic and static features. Experience on which of these alternatives works better seems to vary between laboratories. The extension to much longer sequences of frames is in the province of Nelson Morgan's paper and will not be discussed here.

Some approaches have been described that attempt to improve the effectiveness of the linear transformation. One [26] began with a standard LDA representation and applied further rotation and scaling by an iterative method progressively improving the discrimination between confusable pairs of spoken letters. Another, a *heteroscedastic* transformation [27], loosens the assumption that all within-class covariance distributions are identical to the assumption that they just share the same principal axes. A third [28] uses a maximum mutual information criterion. Although promising results have been reported for these variants, they cannot yet be said to be established techniques.

Figure 2.



Average variances of cepstrum coeffiecients C1 to C12 in the high-speed condition divided by corresponding values in the stationary condition

5. CEPSTRAL CORRELATION METHODS

For speech recognition in noise Mansour and Juang [29] proposed replacing the Euclidean distance normally used in spectral comparisons with a correlation metric applied to sets of cepstrum coefficients (excluding C0). (Actually, they proposed several alternative metrics, but they all shared the same spirit.) Geometrically, this amounts to replacing the distance between points in space represented by sets of cepstrum coefficients by the cosine of the angle between the two vectors defined by the two sets of cepstrum coefficients. In this new metric, the norm of the cepstrum coefficient vector is effectively irrelevant; only the relative values of the cepstrum coefficients in a set matter.

This is another example where a technique can be viewed as part of the spectral comparison metric or part of the spectral representation itself. Something close to the correlation metric can be produced by normalizing all sets of cepstrum coefficients to a constant value and then using Euclidean distances [30].

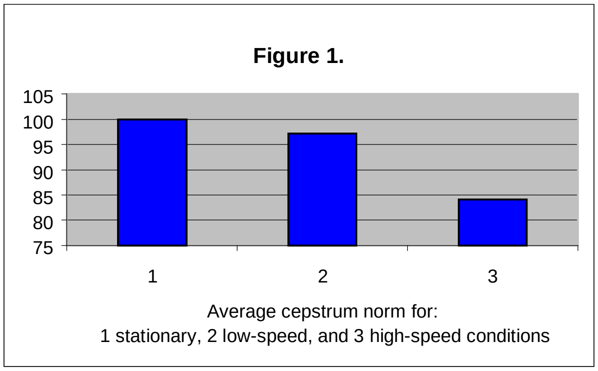


Figure 1.

It is easier to see why this process might be useful by thinking of the effect of noise in the spectral rather than the cepstral domain. Additive noise with a fairly smooth spectrum close to the average spectrum of voice speech will tend to fill in the valleys between the formants in the spectrum of voiced speech, but it will not shift the locations of the formants. It is as though the contrast in the spectrum is reduced, but the overall pattern is otherwise largely unchanged. This reduction in the dynamic range of the spectrum reduces the absolute values of the coefficients of the corresponding cepstrum, but to a first approximation it preserves their relative values.

Of course, this method does depend on the noise spectrum being flat, which is often not the case in real applications. Test results obtained exclusively with artificial added noise are not necessarily any guide to performance in the real world.

To illustrate this, we measured the variance of cepstrum coefficients C1 to C12 about their means for 14 speakers who made recordings in a car with a far-field microphone in three conditions: (1) stationary with the engine running, (2) low-speed highway driving and (3) high-speed highway driving. These three conditions result in three virtually non-overlapping SNR bands. Figure 1 shows that the cepstrum norm does indeed decrease as the SNR decreases. However, Figure 2 shows that with this real noise, which has more power concentrated at low frequencies than speech has, the decrease is not uniform over the different cepstrum coefficients. When the variance of the coefficients averaged over all the speakers in the high-speed condition is compared with the corresponding values in the stationary condition, the change is not found to be uniform. Indeed, the change in the higher coefficients is in the opposite direction from that expected. Consequently, with this real noise the average effect of decreasing SNR is not simply to reduce the cepstrum norm but also to change the angle of the cepstrum coefficient vector.

6. ALTERNATIVES TO LOG POWER

The log power representation of the speech spectrum is clearly attractive because of its gain-invariance properties and of the approximately Gaussian distributions that it provides. It presents, however, one major problem. Since, as *x* tends to zero, *log(x)* tends to minus infinity, the function is very sensitive to small values of *x*. In the spectrum, this means that there is most sensitivity to those parts with lowest power, *i.e.* to those parts where the SNR is normally worst.

This property of the log function makes the results of spectral subtraction of estimated background noise sensitive to small errors in the noise estimate and has led to some complex procedures for spectral subtraction [31].

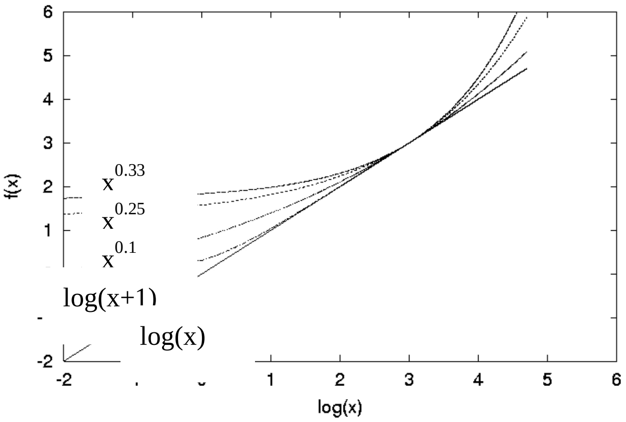
One well established technique for dealing with problems associated with small values in the spectrum is to apply a lower bound to them (so-called “masking” [32]) or to replace *log(x)* with *log(x + c),* where *c* is a small constant. This function behaves like *log(x)* for *x>>c* but limits, of course, to *log(c)* rather than going unbounded as $x \to 0$. For $x >> c$, $log(x + c) \approx log(c) + x/c$ (see box). Thus *log(x + c)* moves smoothly from a function that is limited at *log(c)* and changes linearly with *x* (*i.e.* more slowly than *log(x)*) when *x* is small to *log(x)* when *x* is large. In fact, it behaves like a “soft” mask on the spectrum.

Hermansky and Morgan's *J-RASTA* [33] exploits this behavior neatly by applying the RASTA technique of time-differencing followed by leaky integration to *log(x + c)* rather than to *log(x).* For *x* large, where the SNR is normally high, the technique behaves like the original RASTA formulation, normalizes away linear spectrum distortions, and slow gain changes. For *x* small, however, where the SNR is normally low, we are in the linear region of *log(x + c),* where the differencing/reintegration technique removes steady additive noise [18].

Another motivation for adding constants to the argument of the log function in speech recognition is to simulate the effect of steady additive noise in models that have been trained on quiet material but are to be used in the recognition of noisy speech. Gales and Young’s PMC [34] is a somewhat more sophisticated extension of this technique.

While the justification for masking techniques is purely mathematical, the exploration of another departure from the log function in representing the speech power spectrum, namely the cube root representation, took its original motivation from perceptual data, where auditory nerve firing rates had been reported to be proportional to the cube root of the acoustic power. It was found that the cube root representation could give better recognition results with noisy speech than the

Figure 3. Aligned log and power functions plotted on a log scale.



conventional log representation [35], though performance with quiet speech was worse.

Root power functions, of the form $\sqrt[n]{p}$, constitute a family of functions, with the limit as $n \to \infty$ approximating the log function apart from a scaling factor and additive constant to (see box). Further experimental work [36] has shown that although $n = 3$, *i.e.* the cube root, may be optimal among this family of functions for very low SNR, the optimal value of *n* increases uniformly as the SNR improves. Other work [37] has shown that it is not necessarily optimal to use the same value of *n* for the test and the training material: when using models trained in the quiet and testing with noisy speech it was found to be better to have a smaller value of *n* for the training material than for the test material

Figure 3 suggests that this behavior can be understood in purely signal processing terms. In the figure, root-power functions are shown relative to the log function, having first been "normalized" in value and slope to the log function at a particular point. It can be seen that as *n* decreases the root power functions have increasingly compressed outputs relative to the log function for small input values. The behavior is similar to that observed for the function *log(x + c)*, which is also shown in the figure. Thus, root power representations are similar to masked log representations, with lower values of *n* corresponding to higher values of the mask, *c*. In light of this, it is not surprising that the optimal value of *n* increases as the SNR improves, nor indeed that in recognizing noisy speech with models trained in the quiet it is better to use a smaller value of *n* for the training material (similar to adding a small offset to the spectrum, simulating steady noise, before taking logs) than for the test material.

Root-power functions do not share the elegant level-invariance properties of the log function. For this reason, carefully normalizing the overall level of the signal would be expected to be particularly important, and indeed the performance of root-power representations has been shown experimentally [37] to depend strongly on the normalization method.

Although root-power methods were originally motivated by auditory properties, they can, it seems, be best understood in purely signal processing terms.

Some Properties of the Log and Root-Power Functions

$\log(1+x) = x - x^2/2 + x^3/3 \ldots$ (1)

$(1+x)^a = 1 + ax + a(a-1)x^2/2! + a(a-1)(a-2)x^3/3! \ldots$

as $a \to 0$, $(1+x)^a \to 1 + ax - ax^2/2! + 2ax^3/3! \ldots.$
$= 1 + ax - ax^2/2 + ax^3/3 \ldots. = 1 + a.\log(1+x)$

Rewriting (1+x) *as* y, *we get*

$\log(y) \to (y^a - 1)/a$ *as* $a \to 0$

or, in terms of root powers, writing $a = 1/n$
$\log(y) \to n(\sqrt[n]{y} - 1)$ *as* $n \to \infty$

Also,
$\log(c+x) = \log(c) + \log(1+x/c)$
which, from equation (1),
$= \log(c) + x/c - x^2/2c^2 + x^3/3c^3 \ldots$
which, for $x \ll c$
$\cong \log(c) + x/c$
i.e. *linear in* x

7. DISCUSSION AND CONCLUSIONS

This paper has tried to describe and provide motivations for the most common acoustic representation for speech recognition and some variants that have been found to be useful in several research centers, especially with noisy speech.

The philosophical case for taking what we know about the human auditory system as an inspiration for the representation used in our automatic recognition systems was set out in the Introduction, and it seems quite strong. Unfortunately, there does not seem to be much solid empirical evidence to support this case. Sophisticated auditory models have not generally been found to be better than conventional representations outside the laboratories in which they were developed, and none has found its way into a major mainstream system. Certainly, there are successful approaches and features that are generally felt to have an auditory motivation—the use of the mel-scale, the cube-root representation, and PLP. However, this paper has sought to show that they have no need of the auditory motivation, and their properties can be better understood in purely signal processing terms, or in some cases in terms of the acoustic properties of the production process. Other successful approaches, such as LDA and the correlation metric, made no pretense of having an auditory basis.

There might be a parallel with approaches to speech recognition in general. Up to as recently as a decade ago there was a widespread feeling that the only way to achieve really effective large-vocabulary continuous speech recognition would be to understand how humans recognized grammatical sequences of continuously spoken words. Now we have large-vocabulary continuous speech recognition based purely on statistical methods, and the need to discover and copy what humans do is felt to be much less.

And yet... despite the present evidence, it seems that we must surely have much to learn from what the human auditory system does in representing the speech signal, and perhaps what the human brain does in implicitly understanding speech production. For example, harmonic structure in voiced speech is nothing but a nuisance to be smoothed out in our spectral representations; yet speech with a breathy excitation or artificially excited with broad-band noise, which should be ideal for our artificial representations, is, at least to this author's ears, much less clear than speech generated with normal periodic voiced excitation. Speech produced with a fundamental frequency of 300 Hz, whose resulting harmonic separation presents real problems for our current spectral representations, especially in noise, seems to our ears to be no harder to understand than speech produced with a typical adult male fundamental frequency of 100 Hz. Some support for this subjective impression comes from a recent report [38] that human difference limens on the frequency of the first formant did not depend on F_0 over a range from 100 Hz to 370 Hz.

It seems probable, then, that although the study of human hearing and speech processing has contributed little to automatic speech recognition so far, it does have potential to do so in the future. Nevertheless, it remains this author's view that we have more to learn from studying and modeling human speech production than we have from studying and modeling speech perception.

8. ACKNOWLEDGEMENTS

The author is grateful to John Bridle and Nathan Price for help with this paper.

9. REFERENCES

1. R.P. Lippmann, "Speech Recognition by Machines and Humans", *Speech Communication*, Vol. 22, No. 1, 1997.

2. G.R. Doddington & B.M. Hydrick, "High Performance Speaker-Independent Word Recognition", *Acoustical Society of America and Acoustical Society of Japan Joint Meeting*, Honolulu, Hawaii, Nov.1978. Abstract in *J. Acoust. Soc. Am.*, Vol.64, supplement No.1, Fall 1978.

3. N. Morgan, "Temporal Signal Processing for ASR", *Proc. IEEE International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Keystone Resort, Colorado, December 12-15, 1999.

4. H. Dudley, "Remaking Speech", *J. Acoust. Soc. America*, Vol. 11, pp. 169-177, 1939.

5. S.B. Davis & P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Trans. Acoust., Speech & Signal Processing*, Vol. ASSP-28, pp.357-366, Aug. 1980.

6. H.J.M. Steeneken & T. Houtgast, "Mutual Dependence of the Octave-Band Weights in Predicting Speech Intelligibility", *Speech Communication*, Vol. 28, No 2, June 1999, pp. 109-124.

7. S. Umesh, L. Cohen & D. Nelson, "Fitting the Mel Scale", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-99*, Phoenix, Arizona, March 1999, Vol. 1, pp. 217-220.

8. J.D Markel & A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag, Berlin, 1976.

9. K. Paliwal, "On the Performance of the Quefrency-Weighted Cepstral Coefficients in Vowel Recognition," *Speech Communication*, 1982, Vol. 1, pp. 151-154.

10. B.H. Juang, L.R. Rabiner & J.G. Wilpon, "On the Use of Bandpass Liftering in Speech Recognition", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-86*, Tokyo, April 1986, pp. 765-768.

11. Y. Tohkura, "A Weighted Cepstral Distance Measure for Speech Recognition", *Proc. IEEE Int. Conf. on Acoustics, Speech & Signal Processing, ICASSP-86*, Tokyo, April 1986, pp. 761-764.

12. M.J. Hunt & C. Lefèbvre, "Distance Measures for Speech Recognition", *National Research Council of Canada, National Aeronautical Establishment Aeronautical Note*, AN-57, March 1989.

13. D.B. Paul "A Speaker-Stress Resistant HMM Isolated Word Recognizer", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-87*, Dallas, 1987, Vol. 2, pp. 713-716.

14. A.M. Noll, "Cepstrum Pitch Determination", *J. Acoust. Soc. America*, Vol. 47, pp. 293-309,1967.

15. A.P. Varga & R.K. Moore, "Hidden Markov Model Decomposition of Speech and Noise", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-90*, Albuquerque, 1990, pp. 845-848.

16. S. Furui, "Comparison of Speaker Recognition Methods Using Static Features and Dynamic Features", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 29, No. 3, pp. 342-350, June 1981.

17. H. Hermansky, A. Bayya, N. Morgan, P. Kohn, "Compensation for the Effect of the Communication Channel in Auditory-like Analysis of Speech", *Proc. 2nd European Conference on Speech Communication and Technology, Eurospeech 91*, Genoa, Italy, 24-27 September 1991, pp. 1367-1370.

18. H.G. Hirsch, P. Meyer & H.W. Ruehl, "Improved Speech Recognition using High-Pass Filtering of Subband Envelopes", *Proc. 2nd European Conference on Speech Communication and Technology, Eurospeech 91*, Genoa, Italy, 24-27 September 1991, pp. 413-416.

19. H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech", *J. Acous. Soc. America*, Vol. 87, pp. 1738-1752.

20. J. Makhoul, "Spectral Linear Prediction: Properties and Applications," *IEEE Trans. Acoustics, Speech, Signal Processing*, pp. 283-296, June 1975.

21. N. Campbell, "Canonical Variate Analysis – a General Formulation", *Australian Journal of Statistics*, Vol. 26, pp. 86-96, 1984.

22. M.J. Hunt, "A Statistical Approach to Metrics for Word and Syllable Recognition", *J. Acoust. Soc. America*, 1979, Vol. 66, pp. S535-536.

23. M.J. Hunt & C. Lefèbvre, "Speaker Dependent and Independent Speech Recognition Experiments with an

Auditory Model", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-88*, New York, April 1988, Vol. 1, pp. 215-218.

24. L.R. Bahl, P.F. Brown, P.V. de Souza & R.L. Mercer, "Speech Recognition with Continuous Parameter Hidden Markov Models", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-88*, New York, April 1988, pp. 40-43.

25. M.J. Hunt & C. Lefèbvre, "A Comparison of Several Acoustic Representations for Speech Recognition with Degraded and Undegraded Speech", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-89*, Glasgow, Scotland, May 1989, pp. 262-265.

26. C. Ayer, M. J. Hunt, & D. M. Brookes. "A Discriminatively Derived Linear Transform for Improved Speech Recognition", *Proc.* ", *Proc. 3rd European Conference on Speech Communication and Technology, Eurospeech 93*, Berlin, September 1993, Vol. 1, pp. 583-586.

27. N. Kumar & A.G. Andreou, "Heteroscedastic Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition", *Speech Communication*, Vol. 26, No 4, December 1998, pp. 2 83-297.

28. K. Demuynck, J. Duchateau & D. van Campernolle, "Optimal Feature Sub-Space Selection Based on Discriminant Analysis", *Proc. 6th European Conference on Speech Communication and Technology, Eurospeech 99*, Vol. 3, pp. 1311-1314.

29. D. Mansour & B.H. Juang, "A Family of Distortion Measures Based Upon Projection Operation for Robust Speech Recognition," *IEEE Trans. on Acoustics, Speech and Sig. Processing*, Vol. 37, No. 11, Nov. 1989, pp. 1659-1671.

30. C. Bateman, D. K. Bye & M. J. Hunt, "Spectral Contrast Normalization and Other Techniques for Speech Recognition in Noise", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-92*, San Francisco, March 1992, Vol. 1, pp. 241-244.

31. P. Lockwood & J. Boudy, "Experiments with a Non-Linear Spectral Subtracter (NSS), and Hidden Markov Models for Robust Speech Recognition in Car Noise Environments", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-92*, San Francisco, March 1992, Vol. 1, pp. 265-268.

32. D.H. Klatt, A Digital Filter Bank for Spectral Matching", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP*-79, Washington DC, April 1979, pp. 573-576.

33. H. Hermansky & N. Morgan, "Rasta Processing of Speech", *IEEE Trans. on Speech & Audio Processing*, 1994, Vol. 2, pp. 587-589.

34. M.J.S. Gales & S.J. Young, "An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-92*, San Francisco, March 1992, pp. 233-236.

35. P. Alexandre & P. Lockwood, "Root Cepstral Analysis: a Unified View – Application to Speech Processing in Car Noise Environments", *Speech Communication*, Vol. 12, pp. 278-288, 1993.

36. J. Tian & O. Viikki, "Generalised Cepstral Analysis for Speech Recognition in Noise", *Proc. IEEE/Nokia/COST Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, May 25-26, 1999, Tampere, Finland, pp. 87-90.

37. P. Lockwood & P. Alexandre, "Root Adaptive Homomorphic Deconvolution Schemes for Speech Recognition in Noise", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, ICASSP-94*, Adelaide, South Australia, April 1994, pp. 441-444.

38. F. Karlsson & A. Eriksson, "Difference Limen for Formant Frequencies Discrimination at High Fundamental Frequencies", *Proc. 6th European Conference on Speech Communication and Technology, Eurospeech 99*, Vol. 4, pp. 1687-1690.

INTERVIEW: Ben Gold
INTERVIEWER: Andrew Goldstein
PLACE: Berkeley, California
DATE: 15 March, 1997

Goldstein: Could you tell me something about your education and early career?

Gold: I graduated from Brooklyn Polytechnic in 1948 with a Ph.D. in Electrical Engineering, went to work for some small company in Manhattan for two years, then moved over to Hughes Aircraft Company in Culver City, California for three years. Since 1953 I have been an MIT Lincoln Lab employee, although now I am retired. I probably have always been interested in something to do with signal processing, although I didn't always call it by that name.

The important early work in digital signal processing came in a heavy flurry of activity in the early 1960s by Charlie Rader, myself, and several other people. I worked closely with Charlie, so I'm thinking in those terms. Before that I had been doing work which probably falls under the label of artificial intelligence. In those days we called it "pattern recognition." I designed a Morse code translator, which involved some signal processing but not very much. Then I got interested in the speech area. One day in late 1959 or early 1960 I found myself at Bell Labs talking to a gentleman named John Kelly, who is now gone, and he was describing something called the Pitch Detector to me. He also talked about vocoders. I had heard of vocoders, but I hadn't been very aware of what it was all about. Kelly inspired me to look very carefully at the problem of finding the fundamental frequency of human speech.

I was at Lincoln at that time, had already built the Morse Code Translator, and I came back and started working on this pitch problem. I did it just out of curiosity. In those days at Lincoln Lab you could almost pick and choose what you wanted to work on, up to a certain point. Working on pitch detection was considered okay, but my boss wanted to turn it into something "useful"

Goldstein: Who was that?

Gold: That was a fellow named Paul Rosen. He said, "Well, if you have such a good pitch detector, shouldn't we build a vocoder that includes that pitch detector?" So that put me onto vocoders. Now, a vocoder, among other things, has many filters in it, and once you get into filters you are into signal processing. It was frustrating at the beginning because although the computer was capable of doing a good program for pitch detection, it really wasn't capable of simulating an entire vocoder. It was just too complicated. Nobody knew how to do filters on computers. So we were poised when we realized that, "Hey, maybe there's a way to do this." We got very excited, and we started doing a lot of work very quickly over a period of two or three years. That's where most of the work came from.

Goldstein: I've been trying to track down the origins of digital filtering. I heard from someone that they had digital filters available to them in the early '50s. I'm not sure how to reconcile that statement with what you're telling me now

Gold: Even earlier, people use the computer to process signals in a way that could be called digital signal processing. I think the difference was that people like Kaiser, Rader, myself, and maybe a few other people saw the beginnings of what you might call a formal field of study. We produced an awful lot of stuff very quickly. We were vaguely aware of some of the seismic work, but it wasn't the same thing. Sure, they could program certain things, but they didn't have for example the notion of a finite impulse response filter, or an infinite impulse response filter. They certainly didn't have any notion at the time of the FFT, which was really kind of a bombshell. That was what created the digital signal processing as a real field, because it has so many possible applications. I still feel that the period starting with some of Kaiser's work, some of our stuff, and working its way up to the divulgence of the FFT, from '63 to '67, was when the whole thing exploded and became a real field.

Goldstein: You said you started to drift into speech. Can you tell me about how that happened?

Gold: Yes. I was interested in problems of this sort. I was 16 when I went to see the New York World's Fair. At the fair they had an exhibit of the Voder. It was fascinating. Here was a machine that kind of spoke. Not very well, but it spoke. I was more interested in some of the other things. There was an exhibit where if you won a lottery you could make a long distance call to anybody in the country through the Bell System, and everyone could listen in. I got a real kick out of that, but that didn't lead to anything later. The Voder, on the other hand, I always remembered.

I think what really happened was more complicated, and let me try to trace it out. As I mentioned, I came to Lincoln in 1953 and was in a communications theory group. I worked there for a year, and then I got a Fulbright Fellowship and went to Rome, Italy for a year. I came back in '55 to the same group which now had a different name. It was called pattern recognition, and was one of the first groups in artificial intelligence. The group leader was a fellow named Oliver Selfridge He changed the subject matter of the group drastically, and that's why I worked on this Morse Translator, which was a form of pattern recognition. The group did well for a few years, but then the division heads at Lincoln Labs started to be unhappy with the group. Four people from the group were just arbitrarily transferred to Paul Rosen's group. A year later, the pattern recognition group was disassembled completely and it disappeared. We four engineers were left to work in another group. One of them left right away, but I stayed.

Goldstein: You were one of the four who were at first transferred before the rest of it was dissolved, then?

Gold: My interpretation is that they took the four best engineers and decided "we don't want the other people at all." When you got transferred to another group in those days at the laboratory, the boss didn't just come over to you and say, "Well, do this." He let you drift around and maybe pick up stuff from what other people are doing in the group, and he gave you carte blanche for a year or so. I had gotten interested in work that a fellow named Jim Forgie had been doing on speech recognition in a different group. I think probably my Morse Code work made me feel that this would be an interesting area to work on.

When I came to work at Lincoln, they had big IBM computers, but they didn't have the kind of computers we're used to now or that we got used to a few years later. In other words, as an engineer for quite a while I never felt that I needed a computer. If you needed to solve a mathematical equation, the programmer would do it for you. So until about 1958, I had nothing to do with computers. The Morse Code work I did in pen and paper, and engineers built the device. I had nothing to do with the hardware.

Around 1958 I began to feel that I had to learn something about these computers. I actually went to a school run by IBM for three weeks, and I discovered it was easy. I got interested. Then I discovered that at Lincoln labs there was a Computer Group who were pretty pioneering in the field of computer design. In particular there was a guy named Wes Clark, who was a great computer designer. Eventually I learned that they had built an enormous computer which really could do work in signal processing like speech recognition and pitch detection. Eventually I got to work with the TX-2. That really was the start of it

Goldstein: I still don't understand the difference between the kind of digital filtering that you started to work on, and the techniques that were already in the toolbox of engineers. You said that the seismic people were doing different things, but what was it that you needed to do that wasn't available in terms of digital filtering?

Gold: Let's go back to the vocoder, because this is a good example. Here was one way that the work that I did operated in terms of practical use. First I started off with the notion that just because it's a nice thing to do I'd like to build a better pitch detector. From talking to Kelly, I felt that I could.

Anyway, I programmed something on the TX-2 computer that turned out to be a good pitch detector. But how to prove it was tricky, because in order to prove that you have a good pitch detector you need a vocoder. You need to excite the vocoder with the result of the pitch detector. We didn't have a vocoder, but Bell Labs had a vocoder. So we actually took a 2-track tape down to Bell Labs. One track had speech on it, the other track had the results of my pitch work recorded as pulses, which were sort of in synchronicity with the speech. That 2-track tape was fed into the Bell Labs vocoder and we could hear and record the output.

We brought the recording back and played it to our boss, Paul Rosen, and he said, "It sounds great. Let's build a vocoder." So that got us into what you might call analog signal processing. We didn't know how to build digital vocoders, so we actually built an analog vocoder and didn't do anything with the pitch detector except run programs.

After a while, we were able to test our own vocoder with our program pitch detector. It was slow: to analyze two seconds of speech took the computer about two minutes, 60 to 1 real time. So here is what we had to do. We had to take the speech down to the computer, run the speech in through the computer, run the pitch detector program, record, make our 2-track recording, and bring it upstairs to where the vocoder was. It was pretty slow. So we kept saying, "Wouldn't it be nice if we could program the vocoder on the computer?" So we went back to Bell Labs and visited Jim Kaiser. There may have been other people there, but he's the one I remember. He said he knew how to build certain digital filters. That was just what we needed. We said, "My God, this is fantastic." We could actually build digital vocoders. So we started furiously looking around for literature, and I found there was a book. Have you ever heard the Radiation Laboratory Series?

Goldstein: Sure

Gold: It's in a volume published in 1947 called Theory of Servomechanisms by James, Nichols and Phillips. I had looked at that book many times, but there was one chapter that I had totally ignored. It was a chapter on sample data control systems. That chapter was written not by the main authors but by a gentleman named Hurewicz, who was a mathematician. He wasn't interested in our kind of digital filters at all, but he spelled out the theory in such a way that it could be used directly. It was a revelation.

Here we were in 1963, and that chapter hit me like a bombshell. I practically memorized every word. Maybe a year or two earlier, Charlie Rader had come into the group. I had showed him how to use TX-2. He learned very fast. He and I both got very excited about it. It turned out that we were still pretty raw and didn't really know much. People brought up with doing analog work, analog filtering, found it very difficult to change their mindset to start thinking in terms of digital filters. At the time it didn't seem right. How could a filter be digital if it's analog? It didn't make any sense. But little by little we brainwashed ourselves.

There was another person I wanted to mention, a fellow who worked at Lincoln. This is kind of a sad story. This is a guy named Joe Levin, who was working on seismic detection. Lincoln at that time had a large program to monitor the Kennedy-Khrushchev Test Ban Treaty in '63. The tests were underground, and so there was a big effort in distinguishing underground explosions from earthquakes. Joe Levin was a staff member working on that subject. He was a really smart guy who knew a lot of things. Fortuitously, we told him about our visit to Bell Labs and he said, "Listen, I am teaching a course in control theory, so I know all about these things, and I'd be happy to give you guys a few lectures." So he gave us four or five lectures, and it helped a tremendous amount.

We got to the point where the three of us felt that we knew enough to write a really good paper on the subject. But then he got killed while he was driving on Massachusetts Turnpike. So Charlie and I wrote the paper, and it was a good paper. It was one of the early papers. At that point we were in the field, and despite the complaints of our bosses, who really didn't see the big picture, we kept working on it without getting fired.

Goldstein: Was the pressure from above ever serious?

Gold: Well, at Lincoln things were really pretty hands-off, but we had been doing all this vocoder work, we had published it, and we had been making an impact. My boss saw that we had kind of been neglecting it. One day he met me in the hall and he said, "Why the hell aren't you guys working on vocoders?!" I mean, he yelled at me. My response was, "We will, we will." But we kept working on digital signal processing. We did a little more work on vocoders, and the two fields eventually came together.

Goldstein: It sounds like the field was emerging, and you needed to coalesce.

Gold: Yes. By the way, other people were involved. Jim Kaiser was definitely involved. There was a gentleman named Hank McDonald, although I don't even know if he's still living. He was Jim's boss, and he actually didn't do much research, but he was very interested in pursuing the area and encouraged Jim to work on it

Goldstein: At the time, did you feel any need to define the field that you were working in. It was sort of in between a few areas, it was not very well defined. Did you feel any personal need to have it defined, and if so, how did that process work?

Gold: It worked in a strange way. Like all processes. Let me mention a few other characters. In 1966, there was a gentleman named Ken Stevens who was a speech guy on the faculty at MIT. He had gotten to know me through my vocoder work, and called me up one day to ask me if I like to spend a year at MIT as a visiting professor. After a little bit of discussion with my bosses at Lincoln, it worked out. Charlie and I thought at the time that we had enough material to write a book on vocoders. But this DSP stuff kept coming along, and by the time I got to MIT and started teaching a course, it was basically a DSP course. So the emphasis had shifted slowly.

Another thing happened. While I was at MIT, a guy who used to be my first boss at Lincoln, Bill Davenport, asked me if I would make contact with Al Oppenheim. Al at that time was a tenure track assistant professor. He had graduated several years before. He was doing work on something that he called homomorphic filtering. Al looked me up, and we exchanged information. He told me what he was doing, I told him what I was doing.

We became friends, and at one point we went down to Bell Labs. They deserve a lot of credit, there's no doubt. We spoke to a fellow named Bruce Bogert. Now Bruce was not a DSP guy, particularly. He was interested in a lot of different things, and one of his interests again was earthquakes. He had come up with the idea of cepstrum. Now it turns out that Oppenheim's homomorphic filtering was also the idea of cepstrum. They are almost synonymous. The two of them went at it and really had a great discussion, and they both saw good stuff in what the other person had been doing. So that encouraged Oppenheim to continue work in that.

The other thing that happened was Oppenheim got very interested in what Charlie [Rader] and I were doing. And just around that time the FFT hit. And it was actually an interesting story about how it hit. I was teaching this course, and it was mainly a course on digital filtering the Z transform, different kinds of filters. There was a lot of stuff along the lines of applications to vocoders. I had a TA, a teaching assistant, named Tom Crystal, who was still a graduate student. Well, a lot of MIT students spend time at Bell Labs. One day, during the 1966-67 academic year, when the course was nearly finished, he brought a little document to me after class. It was by Cooley and Tukey. At that time it hadn't been published as a paper, as a journal article, but simply as an internal memo.

I can tell you my reaction. After the first few paragraphs the hair stood up on my head. I said, "This is unbelievable, and I know that this is very, very important." The rest of the paper, I couldn't understand at all. It was all mathematics, and I was just not good at that. It was quadruple sums, really complicated stuff. It was just algebra, but it was very hairy. So I asked Charlie, who is better at that than me, and Tom Stockham, to ÒtranslateÓ, because I knew it was important. They came up with some wonderful, wonderful ways of looking at it, which I think actually sparked the whole field.

At that point, given Oppenheim's work, given the FFT, and given the stuff on digital filters, we said, "There's a book here," and Charlie and I sat down. I had written a fair amount of stuff already for my class, and we just sat down, we said "we're going to write a book, we're going to get Oppenheim to write a chapter, going to get Stockham to write another chapter." Charlie wrote the chapter on the FFT, and that was our book.

Goldstein: What was Stockham's particular area of expertise?

Gold: Stockham was an assistant professor and a good friend of Oppenheim on faculty at MIT. His advisor was Amar Bose, and he was interested in the kind of things that made Bose rich, like the impulse response of a room. He was doing that kind of work, but he was paying attention at that time to what was going on in DSP. He conceived idea of high speed convolution, which was a way of filtering using FFTs. It was a breakthrough paper, and that was the main topic of his chapter in our book. By that time Charlie and I had written four or five papers on the subject, and we were deeply, deeply into it. Of course by that time the world was deeply into it. There was a lot going on.

Goldstein: In the early '60s or late '50s was any desire or interest in defining this field:

Gold: I'd say that we were interested in defining the field. From my point of view, once I understood Hurewicz's chapter in The Theory of Servomechanisms, I felt that this was already a field. Just a few years later I actually offered to teach a DSP course. We had done quite a lot of work on quantization effects, on different kinds of filtering. Here was a whole field that we just called digital filters. The reason it became more than that was because of the FFT. We knew that there was such a thing as a discrete Fourier transform, but it seemed much too "clugey," because you need n2 operations. But if you do it with nlogn, it makes a whole world of difference

Goldstein: How did the FFT open the field up beyond digital filtering. What things became possible?

Gold: If you look at the DFT, the FFT is simply a way of doing the DFT, but it makes looking at the DFT very interesting. With the

DFT you can, for example, define a filter bank, or you can define individual filters. There are enormous connections between different kinds of digital filters and different kinds of ways of dealing with the DFT. So the whole thing becomes a unified field. Things that really weren't possible to compute were now computable. I think that was probably the most significant point. You can compute things like Hilbert transforms, filters with complex parameters rather than real parameters, and things that you just couldn't do in the old system.

For example, a fellow named Bob Lerner at Lincoln had spent an enormous amount of time and money just building an analog delay line for audio frequencies. Well, that's completely trivial on the computer. The field now not only had a theoretical basis, it had a computational basis. I think that's why it really prospered, because you could do anything. As computers got faster, it turned out that things that took an IBM computer the size of a room could be done on a chip.

Goldstein: Did that computational capacity change the focus from theoretical work to application work?

Gold: No, I would say that computation and theory became very strongly integrated so that you could do both at one time, and that had a tremendous effect on how the field grew. Because you could try something out and you could actually see what happened on the computer very quickly, and that gives you insights that you couldn't get just with paper and pencil.

Goldstein: That's similar to digital filters in the beginning of the '60s where you were able to try different vocoders without having to actually build them.

Gold: That's right, and we did. It was pretty slow compared to what we can do now, but it was fast enough that we felt it was really worth doing.

Goldstein: Let me step back for a second, because we moved past your pitch detector machine. Can you put the development of the pitch detector in the context of the technology that was available at the time? What did you want to do that was different than earlier pitch detectors and what tools did you have available?

Gold: This short-lived pattern recognition group still had some very interesting notions. One of the notions that Oliver Selfridge advanced he called Pandemonium. He always had tricky names for things. He thought this was some sort of paradigm for how the brain operates. His idea was there are many, many independent modules in the brain, and they all go their own way, doing what they like to do, but somehow, in solving a pattern recognition problem, they get together and produce a good answer. What this says, in terms of engineering, is perhaps a single algorithm isn't sufficient to get a particular result. Maybe you need several algorithms which are quasi-independent.

This inspired me when I worked on pitch detection. I put together six little elementary pitch detectors. I had a method which was really nothing more than a histogram, a probability estimate of what these different pitch detectors told me about what they thought pitch was. The combination of these six elementary detectors led to a single detector which was better than any of them. It was quite good. So that was an important background for my work on pitch detection, but not particularly for the DSP. It was really for the pitch detector

Goldstein: Could you tell me about the class that you were teaching in DSP? Was this the first instruction they had at MIT in this area?

Gold: This was probably the first time anywhere. It was 1966-67, and they just made an announcement. I had a fairly large class, maybe 20 people. One of them was Larry Rabiner. He was a grad student at the time, and that year while I was there Larry and I shared an office. We got to be friends. He was a very smart guy. By the time he graduated in 1967 he already knew a lot of stuff on speech and on DSP. He was the one guy who I really remember well. There are other people who took my class who have done okay, but he became a star. Al Oppenheim did not take the class; he just came and spoke with me, about the time the class was finishing. Tom Crystal sat in on it. He was a TA, and he has done fairly well. I forget where Tom is now, but I know that he was a committee chairman for IEEE, probably for the Signal Processing Committee.

Goldstein: So the class was intended for graduate students?

Gold: The class was definitely intended for graduate students, and I think there were only graduate students in it. This was very new stuff. It wasn't that it was terribly difficult. I think what was hard about it was the fact that your mindset had to change. The idea that you could, with a computer, do a filter, which had always been a coil and a capacitor, seemed very strange. It was very strange to me, and I think to many people.

Goldstein: When you started working with digital filters, you had to be conscious of things like quantization effects. Were there design considerations from the analog world that you could forget about?

Gold: Well, the basics about digital filters can be summed up as sampling and quantization. Sampling is a very basic thing, and what it tells you is that whatever filter you build, its frequency response is periodic in frequency. That's very different than analog filter. The question becomes, "What do I do about that? How do I handle it?" The way you handle it, most of the time, is that at the very end, you build an analog filter to get rid of all those periods and save only the main period.

[End of tape 1, side a]

Gold: Accuracy is a very key question in both analog and digital filters. In the analog filter world, what people used to do first was what's called the approximation problem. That is, you find a mathematical function that does the kind of filtering you expect. That's the least of your worries in the analog domain. What you really worry about is how to build it. There's been a whole slew of volumes on techniques for building better analog filters that are less sensitive to perturbations in the parameters, because you can't build a coil that's exact. You can come pretty close digitally. But in the analog world what you try to do is develop a structure that is less sensitive to variations. So that's a whole big field, and occupied many people at Bell Labs and other places through the '20s and '30s and '40s.

Now, when you come to build a digital filter, first of all you have a sampling problem, which we mentioned, and that is not terribly difficult to get around. The quantization problem is now becoming less difficult, but it was quite a problem. You could think of quantization as just noise, and you don't want a noisy filter. The question is how much noise for how much word length. Charlie and I at the very beginning actually worked out some theoretical results. We said, "You've got to do it this way, and these are the answers, here are some numbers." But even that isn't enough, because depending on the structure of your digital filter you have more or less sensitivity to the parameters in the digital filter just as you do in the analog domain. So we had to figure out better structures. A lot of people did a lot of work on that, and that went on for maybe 10-15 years.

Goldstein: Can you relate the work you did on quantization issues to that done by Widrow? Were you after the same issues?

Gold: It's my impression that Widrow did a lot of work on what later came to be called neural networks a long time before anyone else. But that had nothing to do with digital filters at all.

Goldstein: That's true, but he did get involved with neural networks after working on adaptive filters. His Ph.D. had been on analyzing the noise from quantization.

Gold: All I can say is that I wasn't aware of what Bernie did, and we had no connection in that sense. I may have done the very same thing that he did and I didn't know.

Goldstein: You mentioned a few names that I've heard before, people who were in this community at MIT. Could you lay out for me the social relations in that community? Was everybody on the same plane as colleagues, or were there senior people?

Gold: The people I've mentioned are Rabiner, who was a grad student, Oppenheim who was tenure track faculty, and Stockham, who was also tenure track faculty. These are the three people who were associated with MIT except for Charlie and me. We were also associated with MIT working at Lincoln Lab.

Goldstein: I've heard Tom Stockham's name come up as an inspiration. He was described as being important intellectually and also socially.

Gold: Tom was a good friend of Al Oppenheim. Larry Rabiner was not as close. I'm not sure if he even knew Al Oppenheim at the time, but he knew me because of the class. Charlie and I worked together, and so Charlie knew everybody that I knew. I know that Al and Tom were really good friends, because Al bought Tom's house in Lexington. So Tom, Al, and I became quite good friends. Charlie and I didn't become great friends, but we worked together for a lot of time. After a while many other people got interested. I have a very good friend, Joe Tierney, who two or three years later suddenly realized, "Hey, look what these guys did. I want to learn it too." He started really doing stuff.

Charlie, Larry and I were definitely interested in audio processing. I think Al was more of a mathematical type, and I think Tom was also interested in audio processing. But at a certain point the radar people got interested. You're talking two orders of magnitude more speed from radar compared to audio, maybe three orders, and yet the possibilities were looking so good that even radar people started fooling around with this. Eventually, a lot of DSP came out of radar.

Goldstein: When you say the radar people, do you have anyone specific in mind?

Gold: The only name that comes to mind is a fellow named Ed Muehe, and it was peripheral with him. I mean, he was interested in it. There was also a fellow named Bob Purdy. Purdy was a good radar guy. What actually happened was that in the late '60s, I came in one day to my boss and said, "Isn't this DSP stuff great. Why don't we build a computer based on it." It was quite a provocative statement, to build a whole computer. Anyway it turned out we did, and it cost a lot of money. It was a big computer. From end to end, it probably covered this entire room.

Goldstein: So it was about 25 feet square?

Gold: It was big, yes. It was called the Fast Digital Processor, the FDP, and it was built with in-house funds. Somehow my boss felt strongly enough about it that he was able to find the money. For many years they had what they call line item money, money that just came in through the Air Force or some agency like that, which was pretty automatic. Then there was other money that you had to apply for. So if something came along that the managers felt was good but not sponsorable, they'd use in-house money. The Fast Digital Processor, was built with in-house money. It cost a lot of money, and the directors got very antsy about it towards the end. So they said, "We've got to use this for something useful. Let's use it for radar." So we all became radar people. For a few years we worked on radar, and Muehe and Purdy were people that I worked with, but it was really the same people, like Charlie and me, who were pushing radar DSP.

Goldstein: Was there similar work going on elsewhere?

Gold: Oh, I'm sure there was. By that time the whole world was working on it. I'm just talking about what I know at Lincoln. I know that, for example, at Bell Labs they had built a very nice piece of software that they called BLODI, which stands for Block Diagram Compiler. It was basically a macro program where you could specify DSP blocks and have the computer assemble it and give you an algorithm. Charlie actually liked that program so much that he built something called PATSY for our computer. I don't remember what it stands for anymore. It was a nice algorithm. There was that kind of work going on with us and elsewhere.

In the late '60s, Al came to Lincoln for two years and worked on what he called the homomorphic vocoder. It was a way of using his mathematics to build really a new type of vocoder algorithm, which is now a standard. Oppenheimer is one of the great guys. After those two years he went back to MIT and organized the first really intensive graduate course on DSP. From the course he wrote his book, which became as close to a best seller as a DSP book can be. But that was already into the 1970s.

Goldstein: When you said that you suggested to your boss that you build a computer based on DSP, what do you mean by that? Based on DSP or to do DSP functions?

Gold: So it could do very fast FFTs and very fast digital filters. The thought was about parallelism. These days of course you can build huge parallel processors, but in those days it wasn't that easy. So what I proposed was four individual processors, each running in parallel, and if you structure the computer correctly it can do digital filtering and FFTs four times as fast as if you only had one processor.

Now of course nearly anything you build, as you probably know, becomes obsolete by the time you finish building it. By the time we finished building the FDP, technology had advanced to the point where we now could do the same thing with raw speed. We actually built a succession of computers which were a lot simpler, but very fast for those days, to do signal processing. We no longer used the rather awkward structure of the FDP. It was good in its time, for a few years. We wrote a paper on it, and it was nice, but it became obsolete very quickly.

Goldstein: Who was the boss who was interested in seeing it applied to radar?

Gold: I would single out Jerry Dineen, who was director of Lincoln in the '70s. Paul Rosen, who had my group leader, was an associate division head by this time, so he was my boss's boss. His boss was Walter Morrow, who is now the Lincoln Lab Director. The names I would pick were Dineen, Rosen, Morrow, and Irwin Lebow, who was sort of my direct boss. These were the people who pushed radar. One of Lincoln's big things of course is radar, computers and communication. That's what it was founded on. DSP was sort of an orphan for a while. The directors didn't see that this was anything wonderful. But later on they did. It took a few years

Goldstein: You made a comment that people needed to change their mind set to see the digital filtering world. Was that a very serious issue with some people?

Gold: I think it was pretty serious.

Goldstein: Were there some people who weren't able to readjust?

Gold: I think some people felt that, and maybe still do to some extent. At the time that Oppenheim was teaching his course, Louie Smullin, who was a microwave person, was the chairman of the EE department. Louie didn't think there was anything interesting in DSP. It was just signal processing: "We know signal processing, why get so involved?" He just didn't see it at all. Another person is Bill Siebert, who eventually integrated it into his classes. He is another professor at MIT. But at first Bill didn't think that it was that important. So, you know, things take time.

Goldstein: I'm interested in following the progress of research of this kind into functioning systems. A lot of the people never pursued that, or never really followed it. It sounds like you were a little more involved with actual systems.

Gold: We could build interesting systems using these ideas, and also could write interesting computer programs. Eventually the two things merged. When we first started working on this, we would write a program, it would be non-real time, and we would get results which would indicate how good or bad our algorithm was. We would polish it up, and then we would turn it over to a hardware man who could build it and make it run much faster than we could. Eventually the technology sort of merged so that the designer would simply write a program or make a chip. You know, the two became one in a sense.

Goldstein: When did that happen?

Gold: I'd say by the 1980s it was clear that was happening. Maybe other people more visionary could see it sooner, but I think the whole thing was really sparked by the integrated circuit revolution. Things were obviously getting faster, smaller, better, and the DSP people knew that. I think anybody could see that something that you build on a board today in five years would be on a chip. That was sort of common knowledge

Goldstein: Were you aware that any of the work that you were doing or the work that you saw going on around you was showing up in commercial systems of any kind or in you know functioning hardware?

Gold: Yes. The person who comes to mind first is a fellow named Lee Jackson. Lee worked in Bell Labs and he knew Jim Kaiser. He was a younger person, and within two or three years after we started doing our stuff he got very involved and got to be very good at it. In the late '60s he left Bell Labs and he either formed or joined a company to build digital filters. I don't know how successful it was, but he's the one who comes to mind. There may be many other examples.

Goldstein: When you sell a digital filter, is it just software that runs on a computer that your client already owns?

Gold: No, probably in those days it was real hardware. It was a special purpose piece of hardware that did only that, that you couldn't program or anything. It was faster and smaller.

Goldstein: Who was using these?

Gold: Good question. I don't know if the company went out of business or is still in existence. My guess is they didn't really sell a lot. I think chips came along and the whole thing became a different game. I never paid much attention to what happened commercially and how much money people made. My guess is that the integrated circuit revolutionÑit wasn't really a revolution, it was an evolutionÑkept getting better and better. All sorts of digital devices were being fabricated and sold in many different ways by people like Intel and other places. I think that DSP devices were just part of that game. When people had to do something, and it needed some digital filters, they put in some digital filters. There was no big deal anymore. It was just another component. You could probably program it in most cases. It fell into the whole area of integrated circuit technology.

Part of it was still theoretical. In fact, at a certain point I more or less crawled out of the field and just got back into speech. There were all these professors with their graduate students, and they were doing stuff that on the one hand was just too advanced for me because it had lots of math, and on the other hand was sort of a waste of time. It was just being done to get a thesis out.

Goldstein: That's interesting. When did that start? When did you start to have that feeling?

Gold: I'd say that by the mid-seventies I was entirely involved in speech work, building vocoders, analyzing them, getting more involved in perception, which is something I am still doing now, but not really doing theoretical work in DSP. That was really just a few years of stuff for me.

Goldstein: I see

Gold: It was a few years of theoretical work, then a few years of work in which I was pretty heavily involved in computer design for DSP, and after that just drifting back into speech.

Goldstein: Were you involved when linear predictive coding became an important issue?

Gold: We got involved very quickly. What happened there was interesting. Remember I mentioned that we were doing radar work? One of the reasons that we were doing radar work was that the funding for speech work had dried up, and that was one of the reasons directors ordered us to do radar work, "because we can give you money for that." All of a sudden LPC came along, just another bombshell.

Goldstein: Tell me when.

Gold: I'd say very late '60s, early '70s, probably going into up to the mid-seventies. In any case, we jumped into that pretty quickly. We had a fellow named Ed Hofstetter who was actually an old microwave person. He wasn't that old, but he came from microwave. He got interested in speech, and got interested in programming, something he had never done, and he got very good at it. He was also a good mathematician. When LPC came along, he was one of the first to really pick up on it and understand it. He said, "I could write a real time program using the FDP." At that time nobody could do real-time computation on a general purpose computer to do LPC.

Goldstein: The FDP?

Gold: The Fast Digital Processor. It was a fast computer. He actually programmed the first real-time LPC on the FDP. So that got us back into the speech area, and we actually did quite a bit of work on LPC.

Goldstein: Why did that get you back into the speech area?

Gold: Well, because LPC caused the funding switch to open again, and we got money, and were able to work on stuff. Just before that there had been a flurry of work on just DSP theory, and new filter structures had been invented, and Hofstetter, among others, also programmed those on the FDP. It was very useful to have a fast computer, because the ability to do things in real time turns out to be an important issue.

Goldstein: Can you tell me how you became aware of LPC and just re-create that sequence?

Gold: Either Schroeder or Atal came down to MIT and gave a lecture. We attended, and it was something that seemed very odd to us, because it was a whole different way of looking at how you build a speech processor. But after a little bit of fussing and thinking about it, we realized this was very powerful, and we got into it. I'm trying to remember whether people in our group did any theoretical work on it. I know that building the first real time LPC was a nice innovative thing, but in terms of any theoretical work, I don't think the people at Lincoln contributed. I learned about LPC, but I didn't contribute to any of the theory.

Goldstein: You said it struck you as a strange way to build a speech processor, but LPC wasn't necessarily limited to applications in speech, was it?

Gold: No, it came out of auto-regressive analysis, which was used in the statistical domain a lot

Goldstein: Right

Gold: But it had never occurred to me that it could be used for speech. That was the nugget, the fact that somebody, either Schroeder or Atal at Bell Labs thought, "Hey, this might be a new speech processor."

Goldstein: I see. So in their lecture they said that?

Gold: Yes. They probably spent a year or two working quietly at Bell Labs building something, programming something, and then they were ready to announce it. When they did, everybody in the world picked it up. And they are still picking it up.

Goldstein: The way you've been talking up to now, it makes it sound like digital signal processing was synonymous with digital filtering. I don't want to say something like that unless I've checked it. Is that the way it was used back then?

Gold: That's the way I thought of it. Until the FFT came along, I didn't see any particular use for these Z transforms and all the theoretical stuff aside from programming, building, and analyzing filters. Now, it's true that when you build something like a vocoder the major part of the computation is definitely filtering, but you also do other things. You do rectification, and you do low pass filtering in addition to band pass filtering. You have to do down sampling. There are several other functions that take place, but they all seem very, very trivial. We could have done those functions ten years ago on computers. They were easy, but filters were hard. So I'd say filters were the heart and soul of DSP.

Goldstein: You said that by the mid-seventies you had gotten out of digital signal processing and were concentrating on speech.

Gold: Well, I got out of it in the sense that I didn't sit down and try to do new theoretical things. I paid attention to the field. I used these things all the time as a matter of course.

Goldstein: That's what I wondering: how somebody could be doing speech without these tools.

Gold: Oh, no. I use them to this very day. I'm always using DSP things, I'm always thinking in terms of cepstra or Hilbert transforms, things like that, and thinking of them in a digital way. But this is all textbook stuff now.

Goldstein: Could you tell me highlights from say the mid-seventies up to the present day of new techniques that arrived that one could use in applications such as the ones you are interested in?

Gold: Well, LPC was new.

Goldstein: But LPC in the late '60s.

Gold: And LPC really can't even be done analog. It's by definition a digital process. What you're doing is saying, "I predict this sample on the basis of n previous samples." So it's automatically digital. LPC is certainly a big thing. There have been some nice things. I'm thinking of the different ways you could look at auto-regression, like maximum entropy ideas. There have been many things along those lines that have to do with the mathematics of linear prediction, but none of them really are in the same category as just the invention of LPC itself, or the FFT, or the fact that you could build digital filters.

Goldstein: It sounds to me like you're saying that applications have been absorbing these major pillars since the late '60s.

Gold: What I'm saying is the reason that it's such an important area is because of integrated circuits and the ability to combine theory and computation. You can realize things and do it all, and connect it with theory very strongly.

Goldstein: Yes, I've heard you clearly.

Gold: That just couldn't be done until integrated circuits and things like the FFT came along.

Goldstein: Thank you very much.

Pagemaster: History Center Staff (sloan-project@ieee.org)
URL: ww.ieee.org/organizations/history_center/sloan/oral_histories/bgold_transcript.html
Last modified: 13 Sep 1999 10:39 AM

ASSR HOME