Lecture 05:
[Rabiner] Acoustic Feature Extraction for Speech Recognition

DEEE725 음성신호처리실습
Speech Signal Processing Lab
Instructor: 장길진
Original slides from:
Lawrence Rabiner, Mark Hasegawa-Johnson (UIUC),
Dan Jurafsky, Sarita Jondhale

Introduction

- Spectral analysis is the process of defining the speech in different forms of parameters for further processing
  - Short term energy, zero crossing rates, etc.

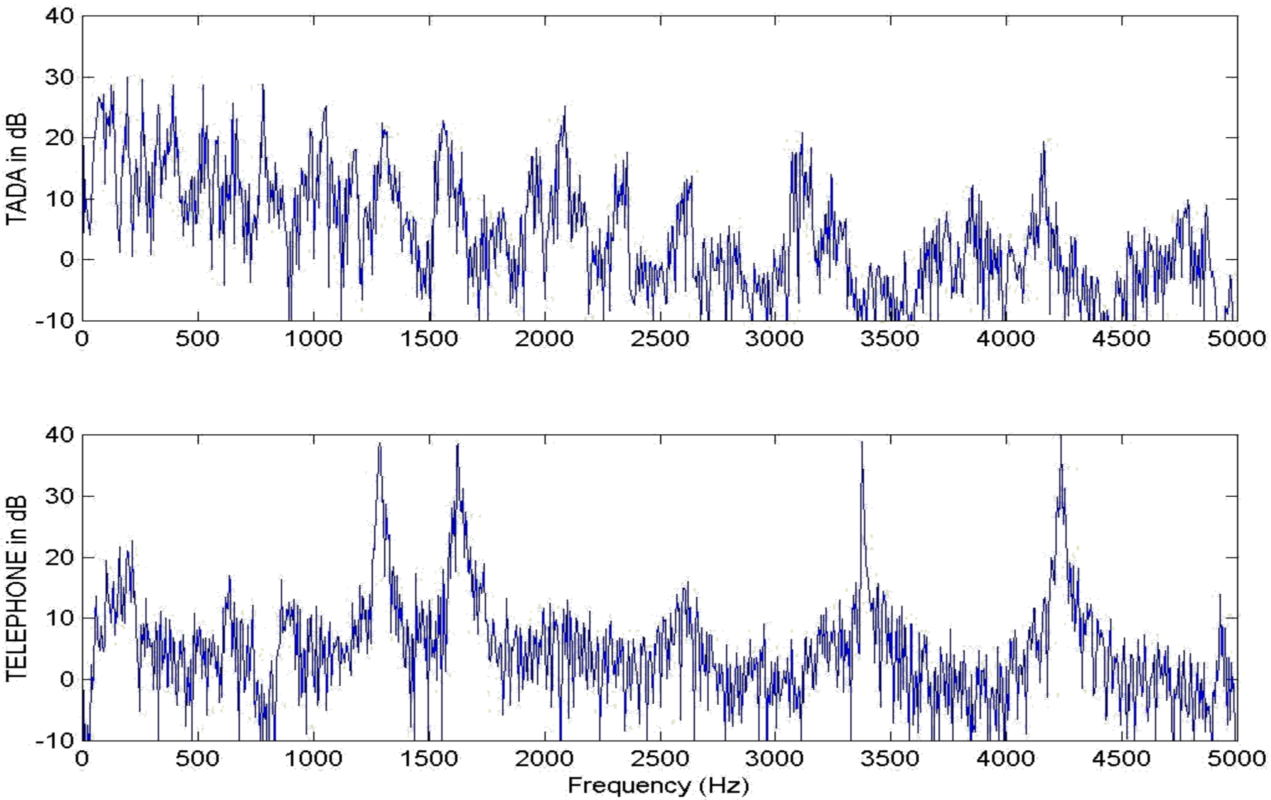
- Methods for spectral analysis are therefore considered as core of the signal processing front-end in a, especially, speech recognition system

Acoustic & Auditory Features for Speech Recognition

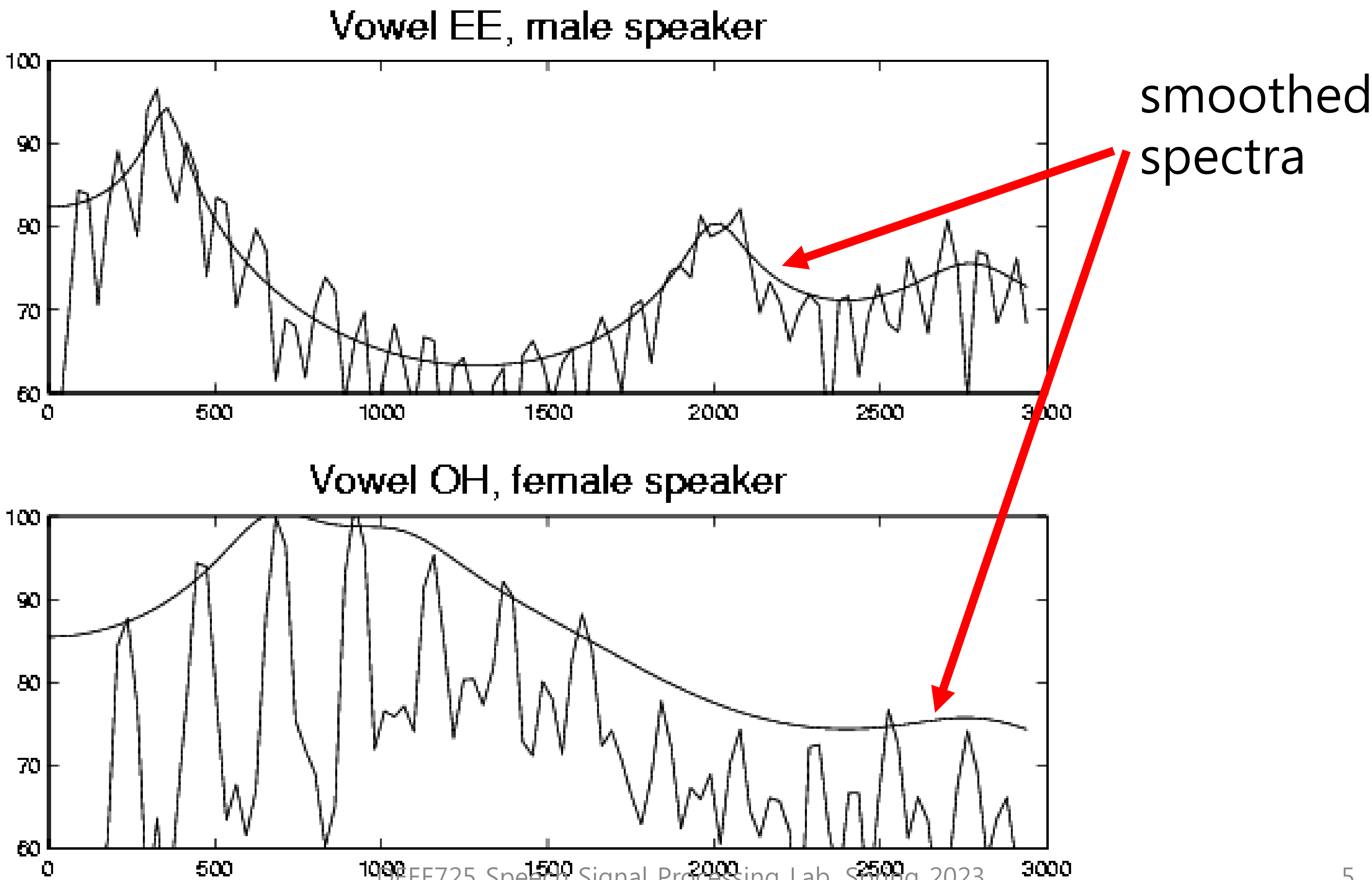
- **Log spectral features:**
  - **log FFT, log filterbank energy**
  - **cepstrum, MFCC (mel-frequency cepstral coefficients)**
- Time-domain features
  - (log) energy, zero crossing rate, autocorrelation
- Auditory model based features
  - auditory spectrogram, correlogram, summary correlogram

Log Magnitude STFT

$$\log\left|S_n(e^{j\omega})\right| = \log\left|\sum_{m=0}^{L-1} s(m)w(n-m)e^{j\omega m}\right|$$



The Problem with FFT: Euclidean Distance ≠ Perceptual Distance



Perceptual Distance

- Note 1:
  - Human auditory system is in logarithmic scale
  - If the power ratio are the same, $\frac{X_1(\omega)}{Y_1(\omega)} = \frac{X_2(\omega)}{Y_2(\omega)}$, human listeners perceive that difference is the same
    - Numbers in the volume meters of audio devices changes the output speaker gain exponentially
- Note 2:
  - Most of the language information is contained in the **envelope (smoothed spectra)**
  - Spectral fine structures (local shapes) account for pitch (F0), and are **to be ignored for speech recognition**

Some Solutions

- To logarithmic scale
  - Products of ratios ➔ Sum of Euclidean distance of log spectra

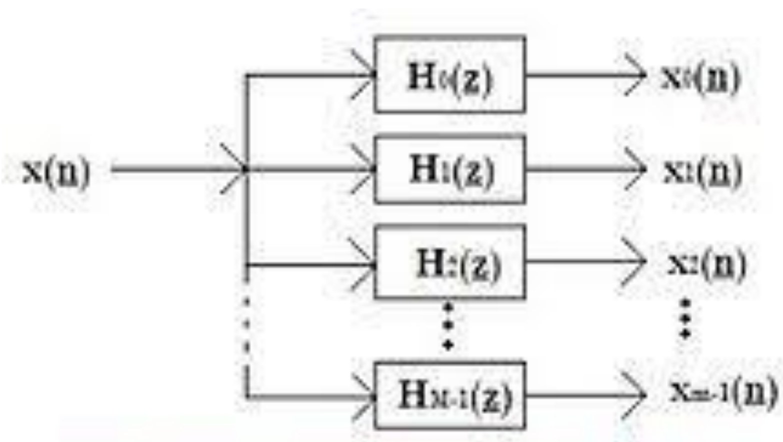
$$(d_2)^2 \quad = \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\log S_1(\omega) - \log S_2(\omega)\right|^2 d\omega$$

- Spectral smoothing
  - Filterbank energies rather than raw DFT magnitudes

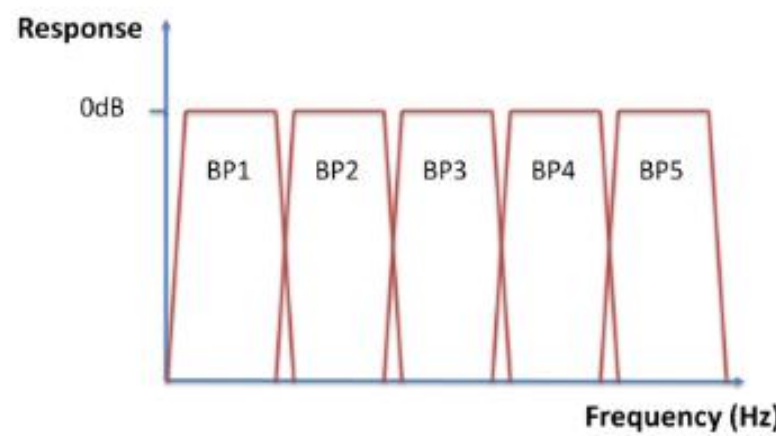
The Bank of Filters Front-end

- One of the most common approaches for processing the speech signal is the bank-of-filters model
- This method takes a speech signal as input and passes it through a set of filters in order to obtain the spectral representation of each frequency band of interest.
  - The band pass filters coverage spans the frequency range of interest in the signal

$$s_i(n) = s(n) * h_i(n) = \sum_{m=0}^{M_i-1} h_i(m)s(n-m), \quad 1 \leq i \leq Q$$



Multidimensional Analysis Filter Banks



The Bank of Filters Front end Processor

The bank-of-filters approach obtains the energy value of the speech signal considering the following steps:

- **Signal enhancement and noise elimination**
  To make the speech signal more evident to the bank of filters.

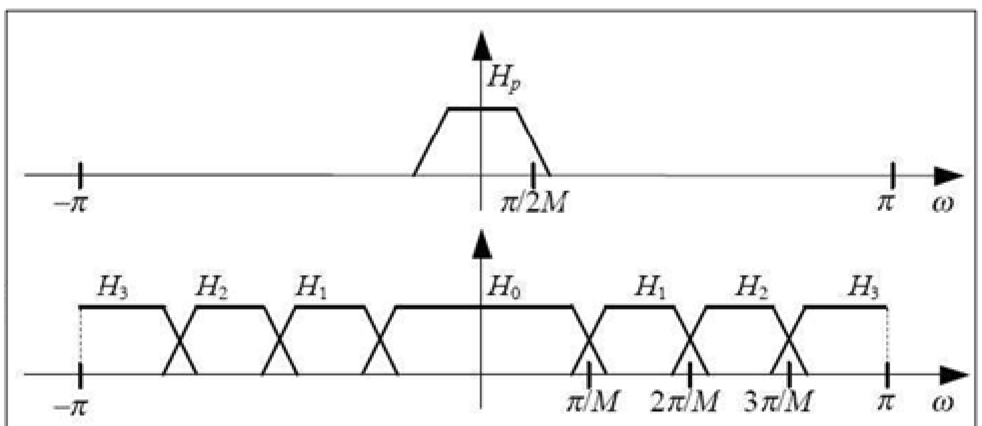
- **Set of bandpass filters**.- Separate the signal in frequency bands. (**uniform/non uniform** filters)

uniform filterbank

- The most common filterbank is the **uniform filter bank**
- The center frequency, f_i, of the i^{th} bandpass filter is defined as

$$f_i = \frac{F_s}{N} i, \quad 1 \leq i \leq Q$$

  - where *Fs* is the sampling rate, *N* is the number of uniformly spaced filters required to span the frequency range of the speech
  - *Q* is number of filters used in bank of filters



Nonuniform LOGARITHMIC filterbank

- The criterion is to space the filters uniformly along a logarithmic frequency scale.
- For a set of Q bandpass filters with center frequncies f_i and bandwidths b_i, $1 \le i \le Q$, we set

$$b_1 = C$$

$$b_i = \alpha b_{i-1}, \; 2 \le i \le Q$$

$$f_i = f_1 + \sum_{j=1}^{i-1} b_j + \frac{(b_i - b_1)}{2}$$

The most commonly used values of α=2

where C and f_1 are arbitary bandwidth and the center frequency of the first filter and α is the logarithmic growth factor

Nonuniform FIR Filter Bank Implementations

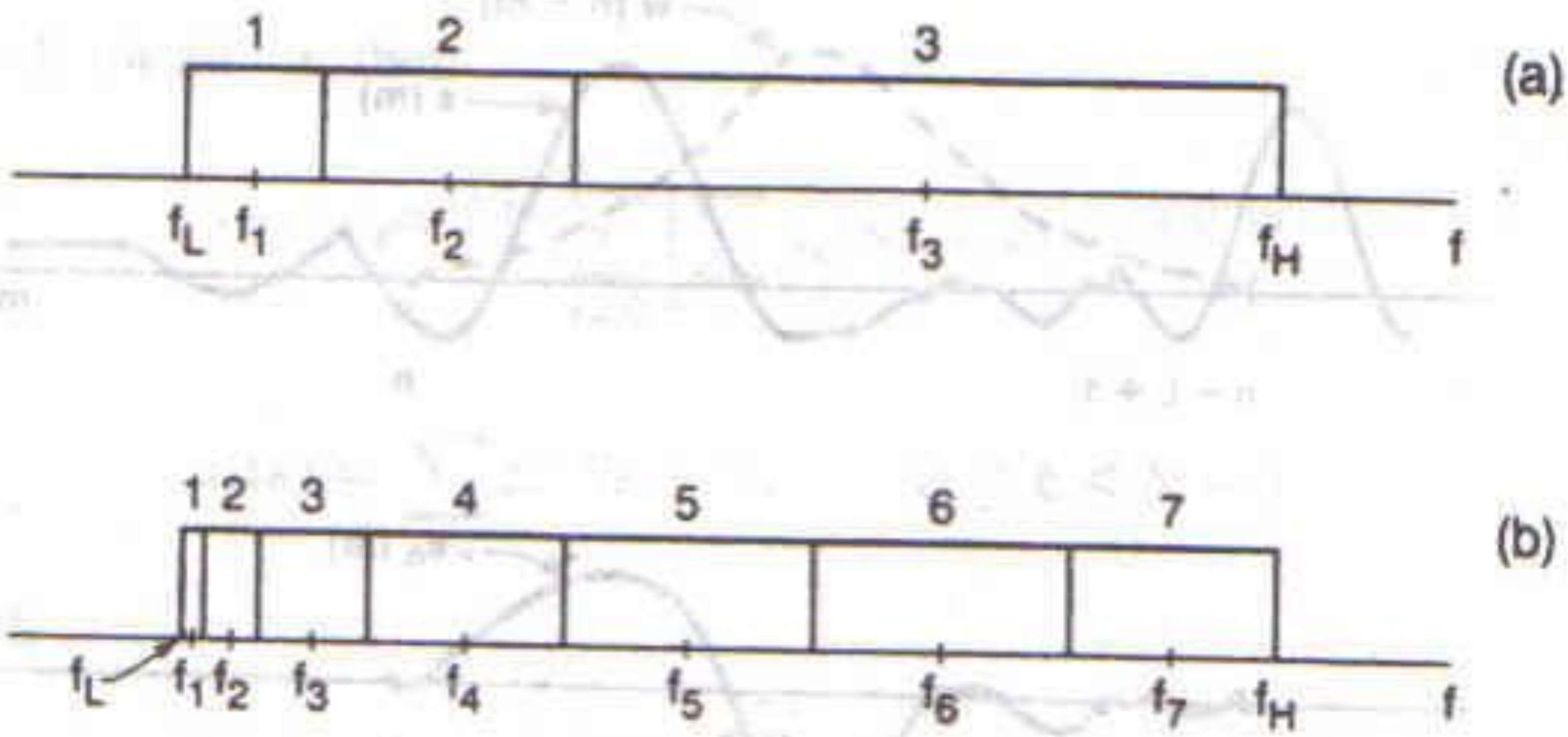
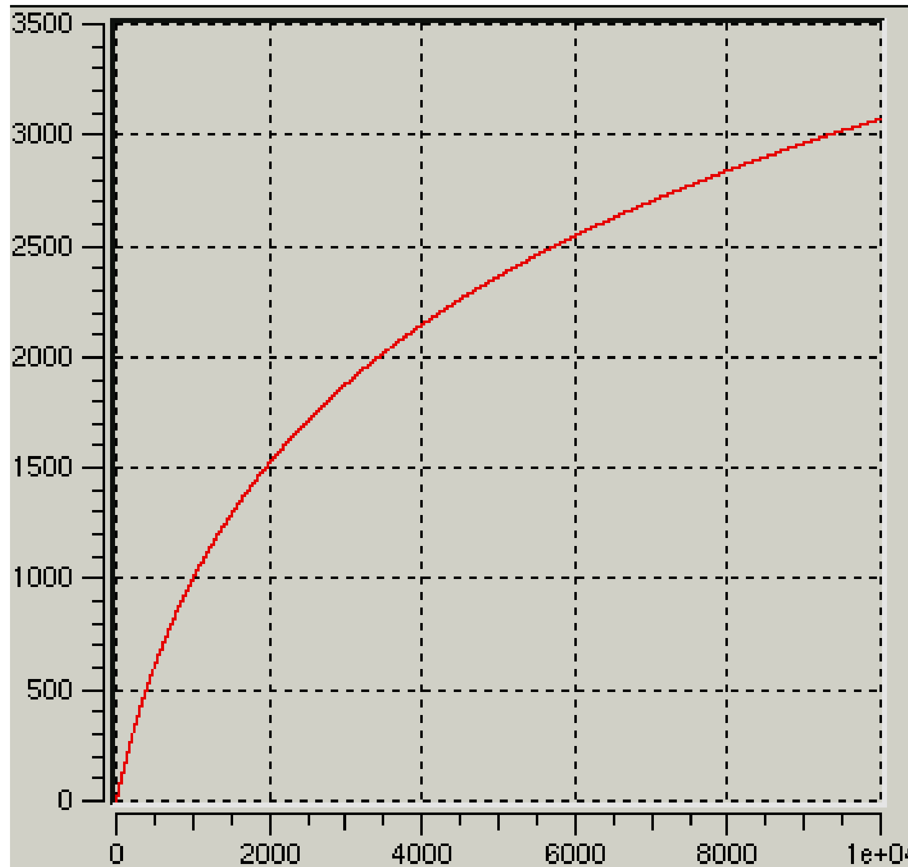


Figure 3.18 Two arbitrary nonuniform filter-bank ideal filter specifications consisting of either 3 bands (part a) or 7 bands (part b).

The Mel Frequency Scale: Humans Can Distinguish Tones 3 Mel Apart

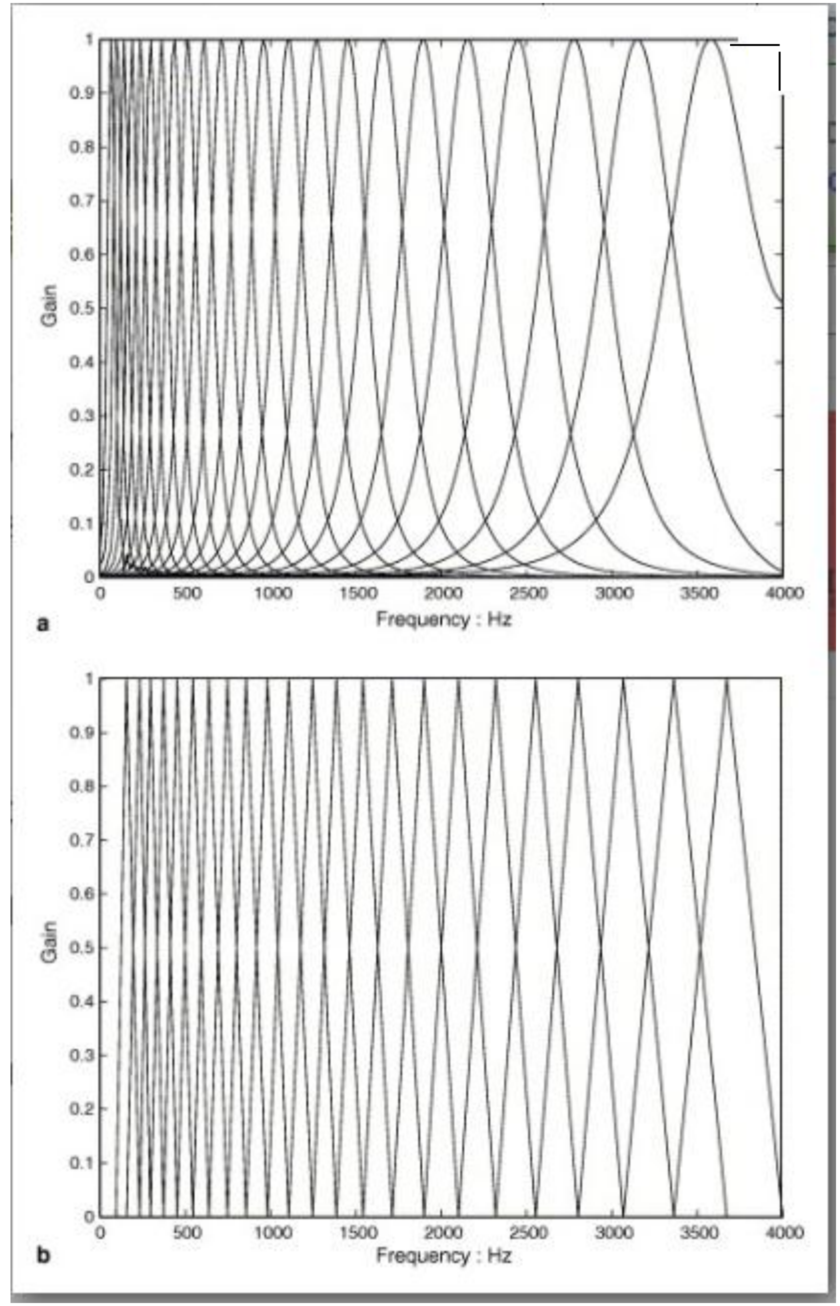
$$m = 1127.01048 \log(1 + f/700).$$



$$f = 700(e^{m/1127.01048} - 1).$$

Mel Filter Bank

- Gaussian filters (top), Triangular filters (bottom)
- Frequencies in overlapped areas contribute to two or more filters
- The lower frequencies are spaced more closely together to model human perception
- The end of a filter is the mid point of the next
- Warping formula: warp(f) = arctan|(1-a^2) sin(f)/((1+a^2) cos(f) + 2a) where -1<=a<=1|

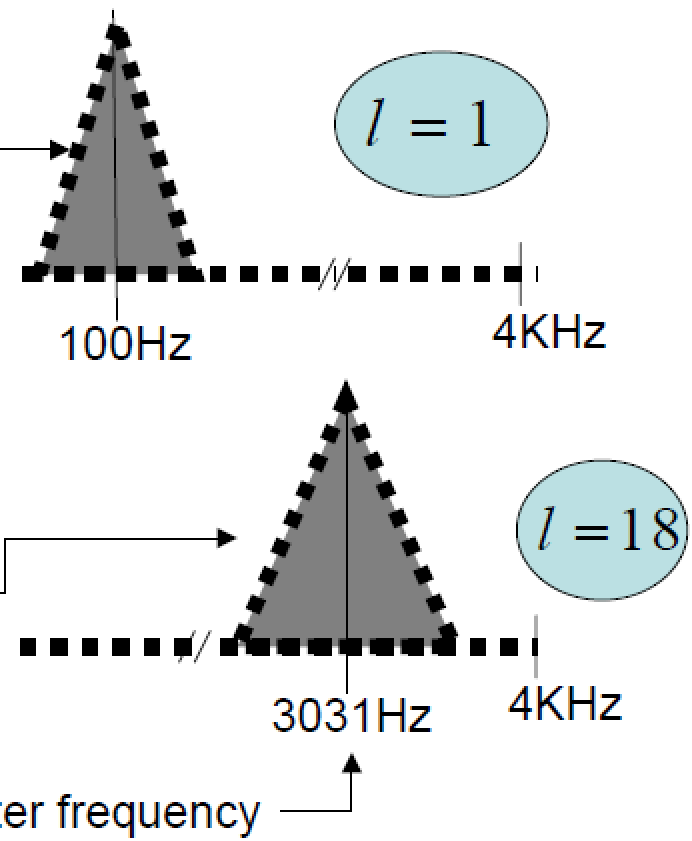


Mel Frequency Table

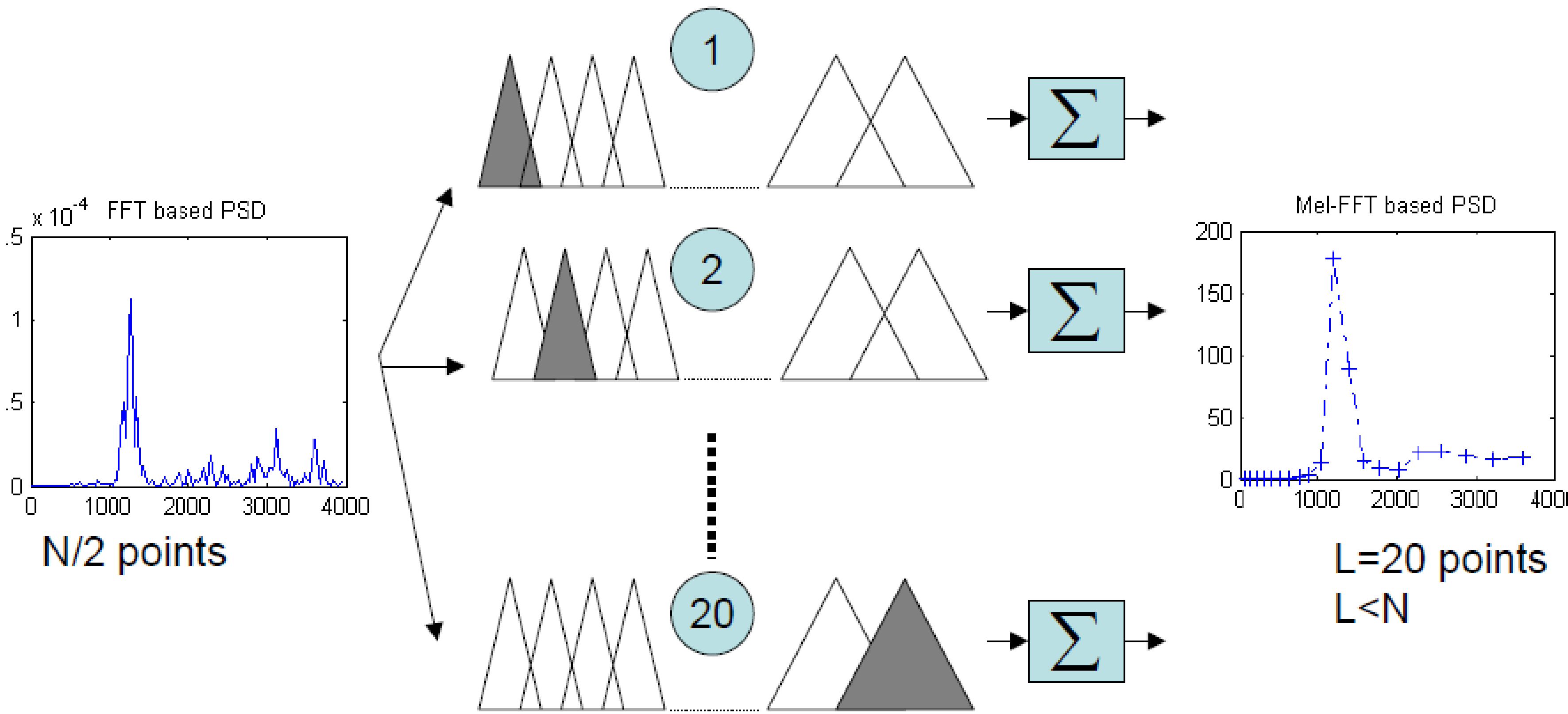
Index	Bark Scale Center Freq. (Hz)	Bark Scale BW (Hz)	Mel Scale Center Freq. (Hz)	Mel Scale BW (Hz)
1	50	100	100	100
2	150	100	200	100
3	250	100	300	100
4	350	100	400	100
5	450	110	500	100
6	570	120	600	100
7	700	140	700	100
8	840	150	800	100
9	1000	160	900	100
10	1170	190	1000	124
11	1370	210	1149	160
12	1600	240	1320	184
13	1850	280	1516	211
14	2150	320	1741	242
15	2500	380	2000	278
16	2900	450	2297	320
17	3400	550	2639	367
18	4000	700	3031	422
19	4800	900	3482	484
20	5800	1100	4000	556
21	7000	1300	4595	639
22	8500	1800	5278	734
23	10500	2500	6063	843
24	13500	3500	6964	969

$M_l(k)$

$$l = 0, 1, \ldots L-1$$
$$k = 0, 1, \ldots, N/2$$

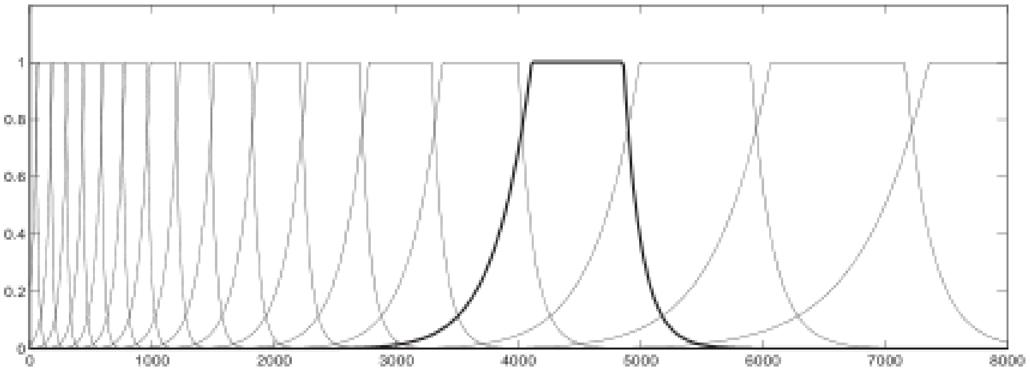


Mel Filter Bank



Multiply the power spectrum with each of the triangular Mel weighting filters and add the result -> Perform a weighted averaging procedure around the Mel frequency

Critical Band Analysis



- The bark filter bank is a crude approximation of what is known about the shape of auditory filters.
- It exploits Zwicker's (1970) proposal that the shape of auditory filters is nearly constant on the Bark scale.
- The filter skirts are truncated at +/- 40 dB
- There typically are about 20-25 filters in the bank

$$C_k(\omega) = \begin{cases} 10^{1.0(\Omega-\Omega_k+0.5)} & , \Omega \le \Omega_k - 0.5 \\ 1 & , \Omega > \Omega_k - 0.5 \\ & , \Omega < \Omega_k + 0.5 \\ 10^{-2.5(\Omega-\Omega_k-0.5)} & , \Omega \ge \Omega_k + 0.5 \end{cases}$$

$C_k(\omega)$ is a weight of the k filter at frequency ω

Ω_k is a centre frequency of the filter k

$k = 1, 2, \ldots, K.$

Critical Band Formulas

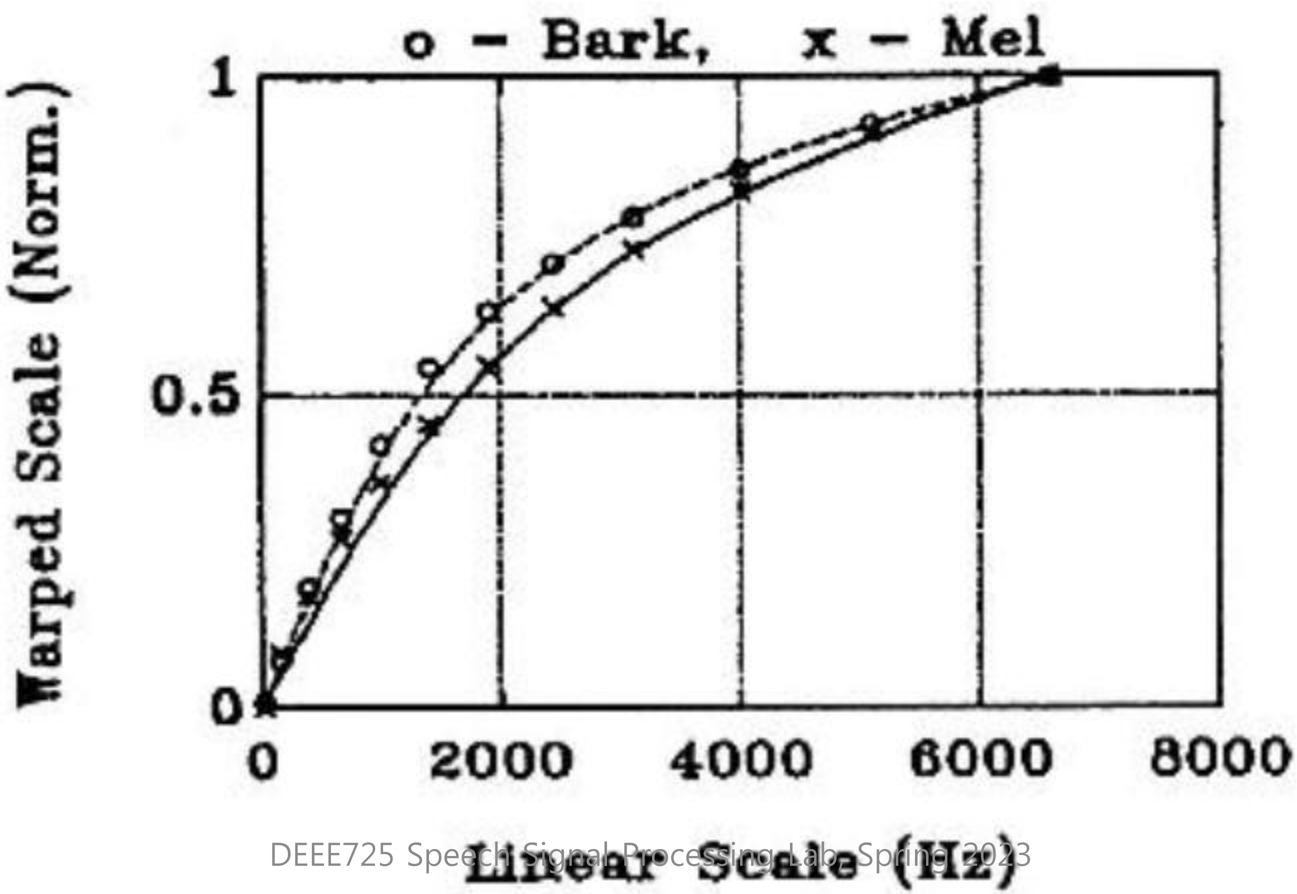
$$b(2\pi f) = 6 \log \{ f/600 + [(f/600)^2 + 1]^{0.5} \}$$

for $f \gg 600\text{Hz}$,

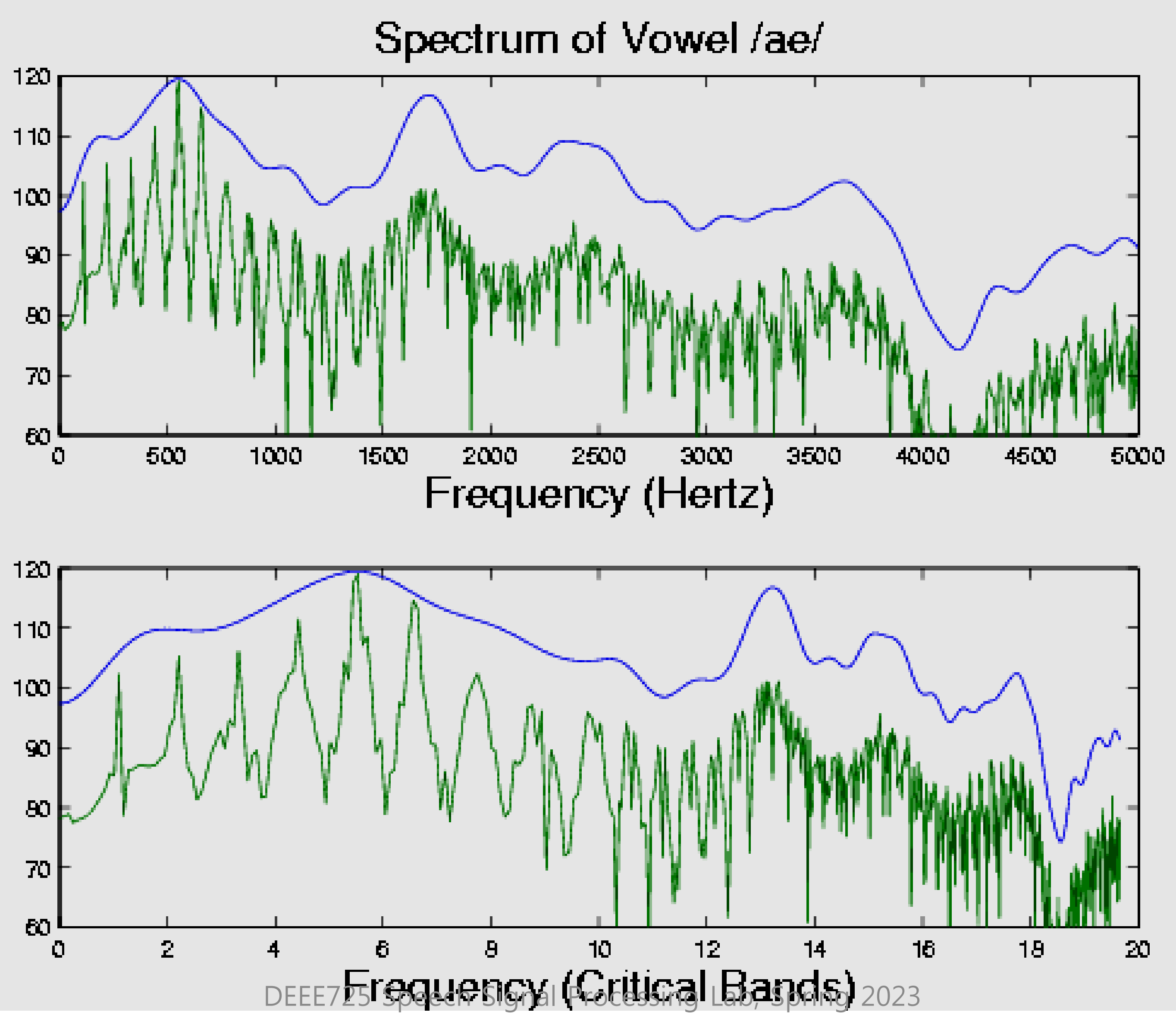
$$b(2\pi f) \approx c_1 \log f + c_2$$

Frequency Warping

- Audio signals cause cochlear fluid pressure variations that excite the basilar membrane. Therefore, the ear perceives sound non-linearly
- Mel and Bark scale are formulas derived from many experiments that attempt to mimic human perception



Bark-Scale Warped Spectrum



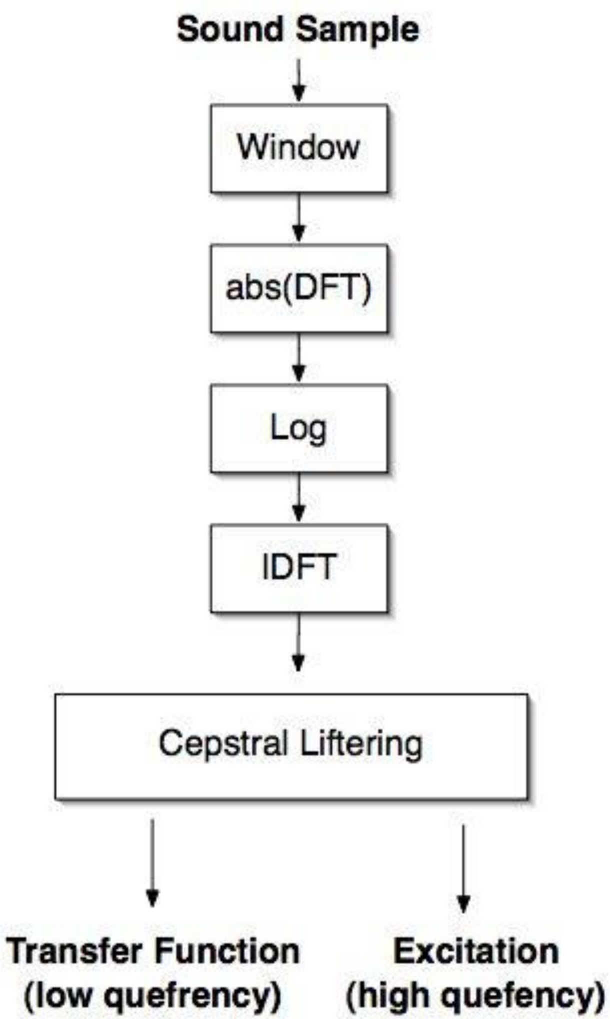
MFCC: mel-frequency cepstral coefficients

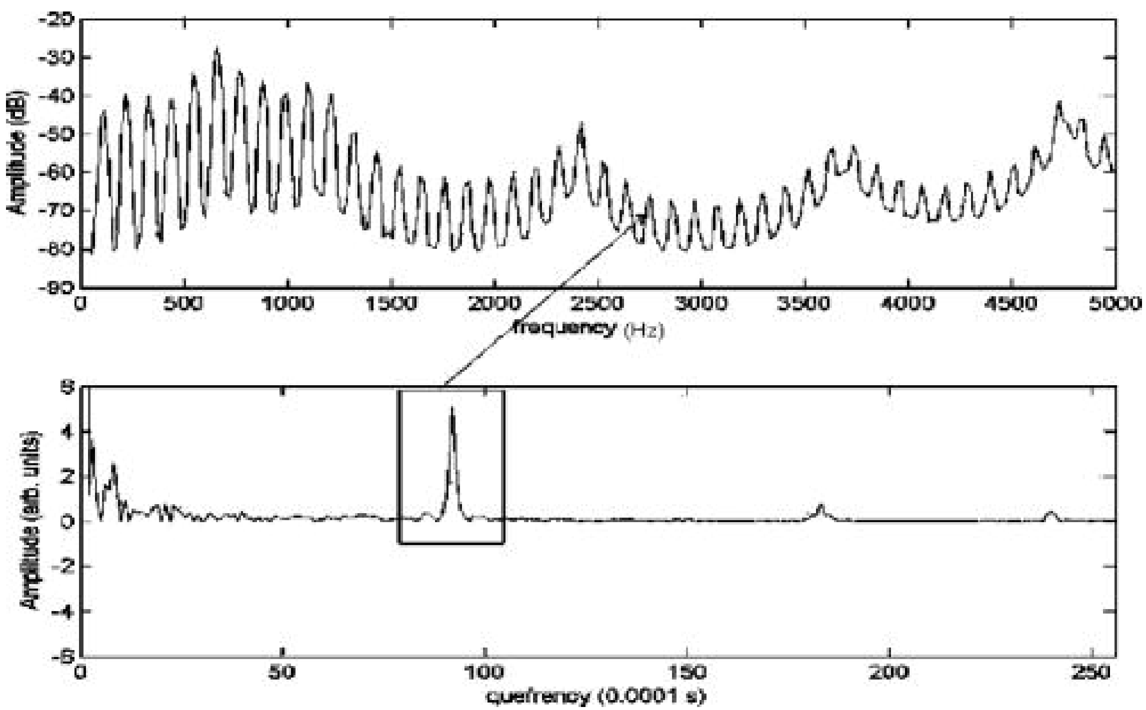
CEPSTRUM
MFCC

Cepstrum

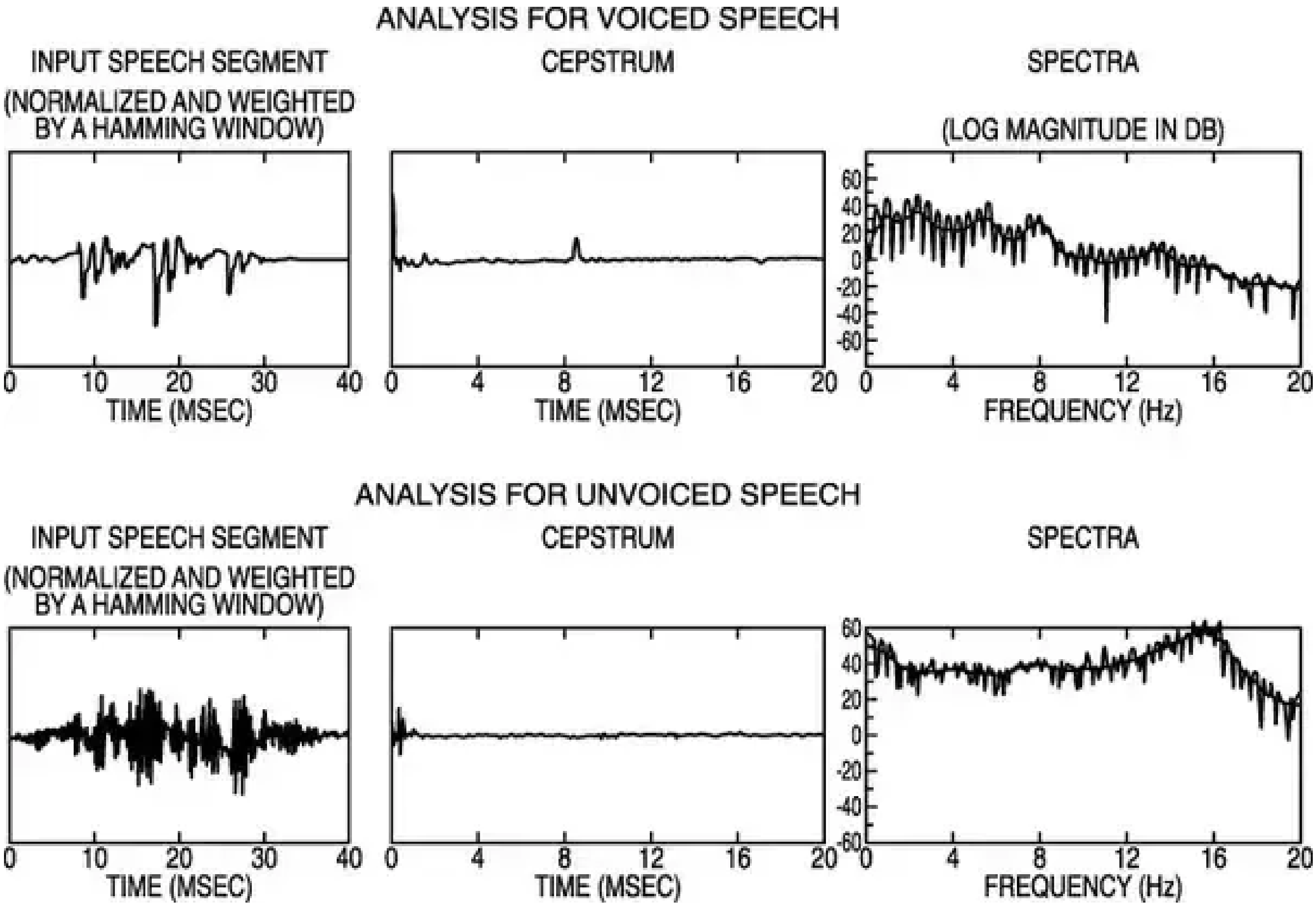
- **History** (Bogert et. Al. 1963)
- **Definition**
  Fourier Transform (or Discrete Cosine Transform) of the log of the magnitude (absolute value) of a Fourier Transform
- **Concept**
  Treats the frequency as a "*time domain*" signal and computes the frequency spectrum of the spectrum
- Envelope and Pitch Segregation
  - Vocal track excitation (E) and harmonics (H) are multiplicative, not additive.
  - The log converts the multiplicity to a sum
    $\log(|X(\omega)|) = \log(|E(\omega)||H(\omega)|) = \log(|E(\omega)|)+\log(|H(\omega)|)$
  - E moves slow, H exhibit fast oscillating harmonics
    - The envelope shows in the lower part of the Cepstrum, the pitch shows up as a spike in the higher part

Cepstrum Pipelining and Results

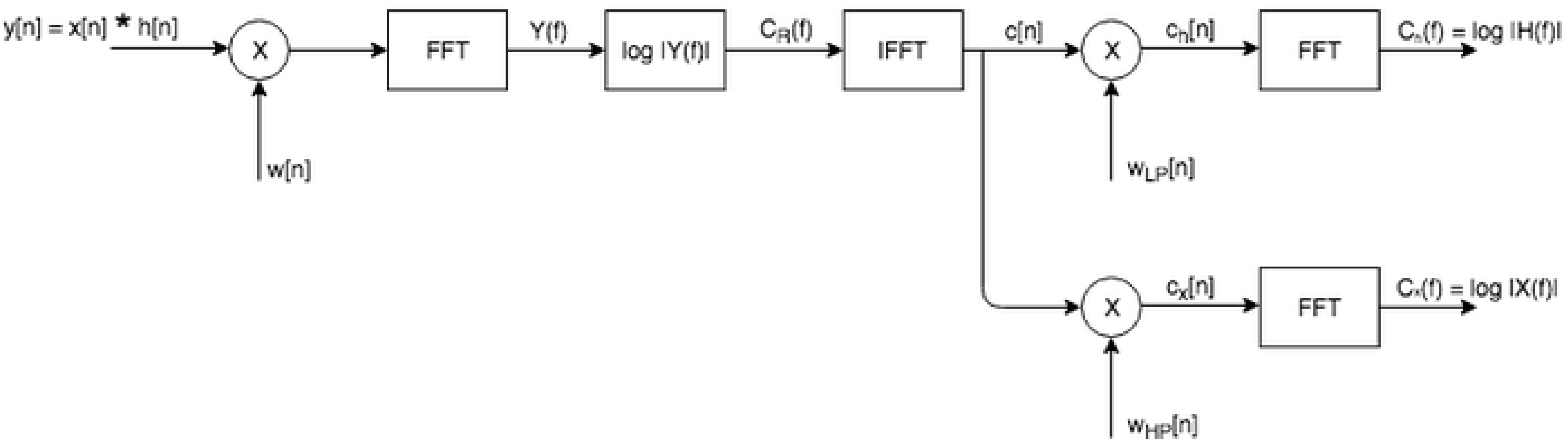


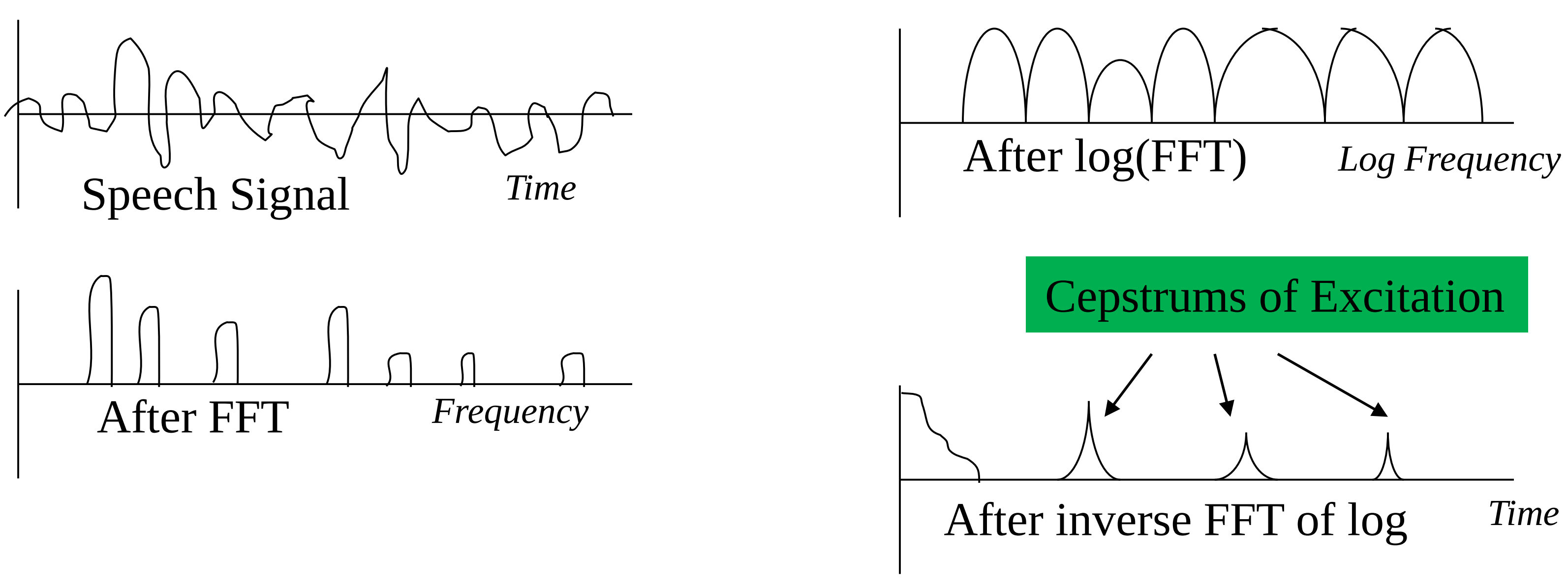


Cepstrum: Voiced vs Unvoiced



Cepstral analysis



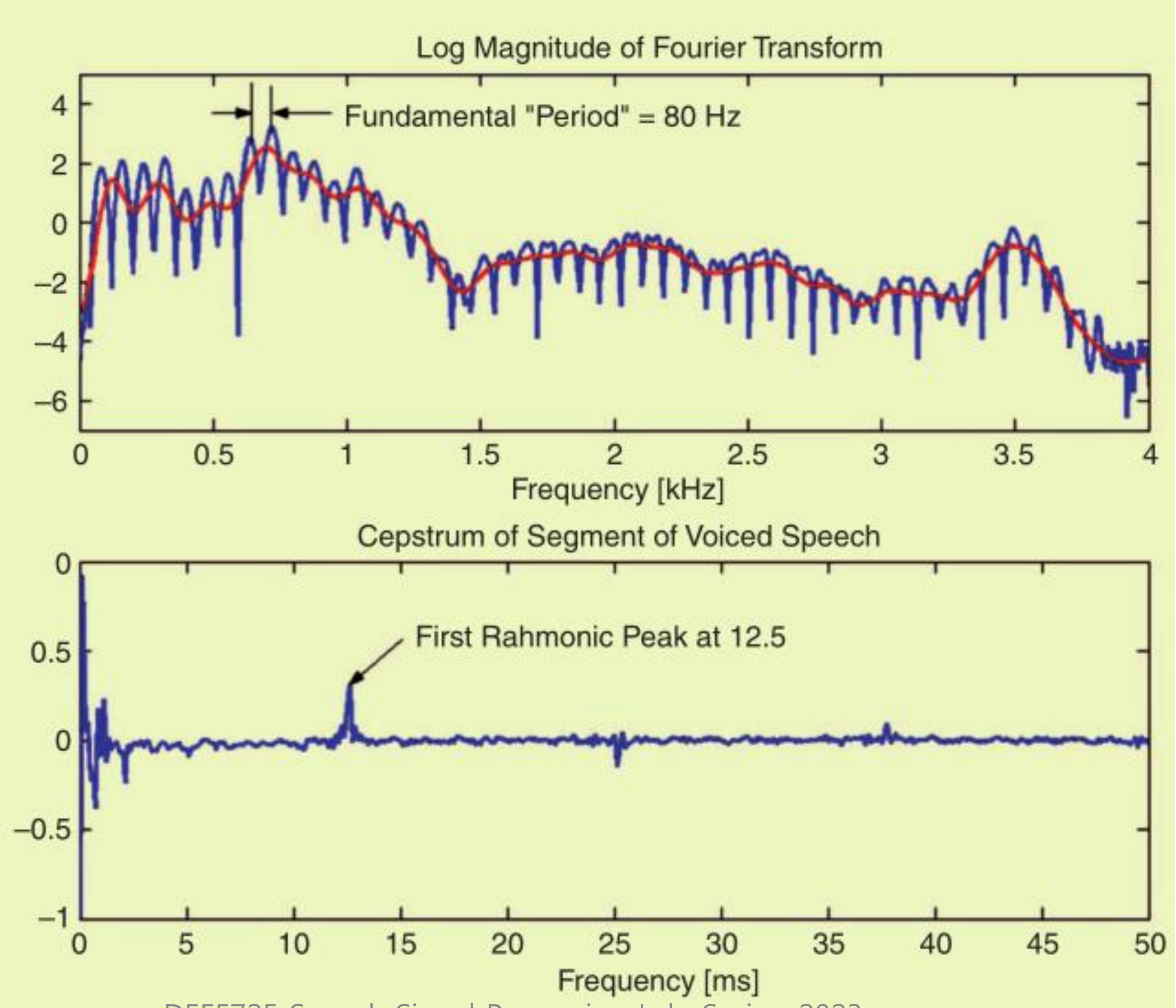


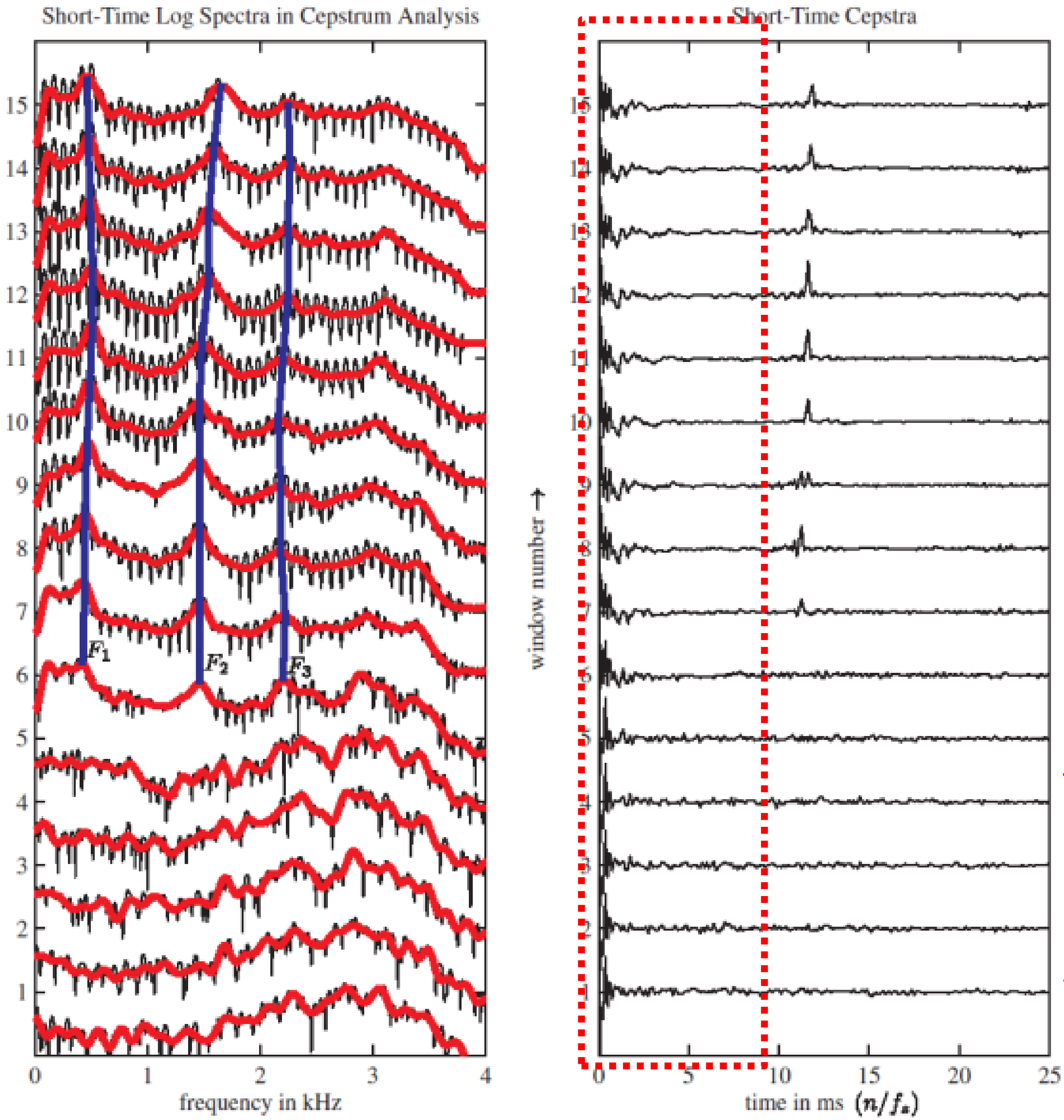
Terminology

Cepstrum Terminology	Frequency Terminology
Cepstrum	Spectrum
Quefrency	Frequency
Rahmonics	Harmonics
Gamnitude	Magnitude
Sphe	Phase
Lifter	Filter
Short-pass Lifter	Low-pass Filter
Long-pass Lifter	High-pass-Filter

Note the flipping of the letters – example Ceps is Spec backwards

Cepstrum and Pitch



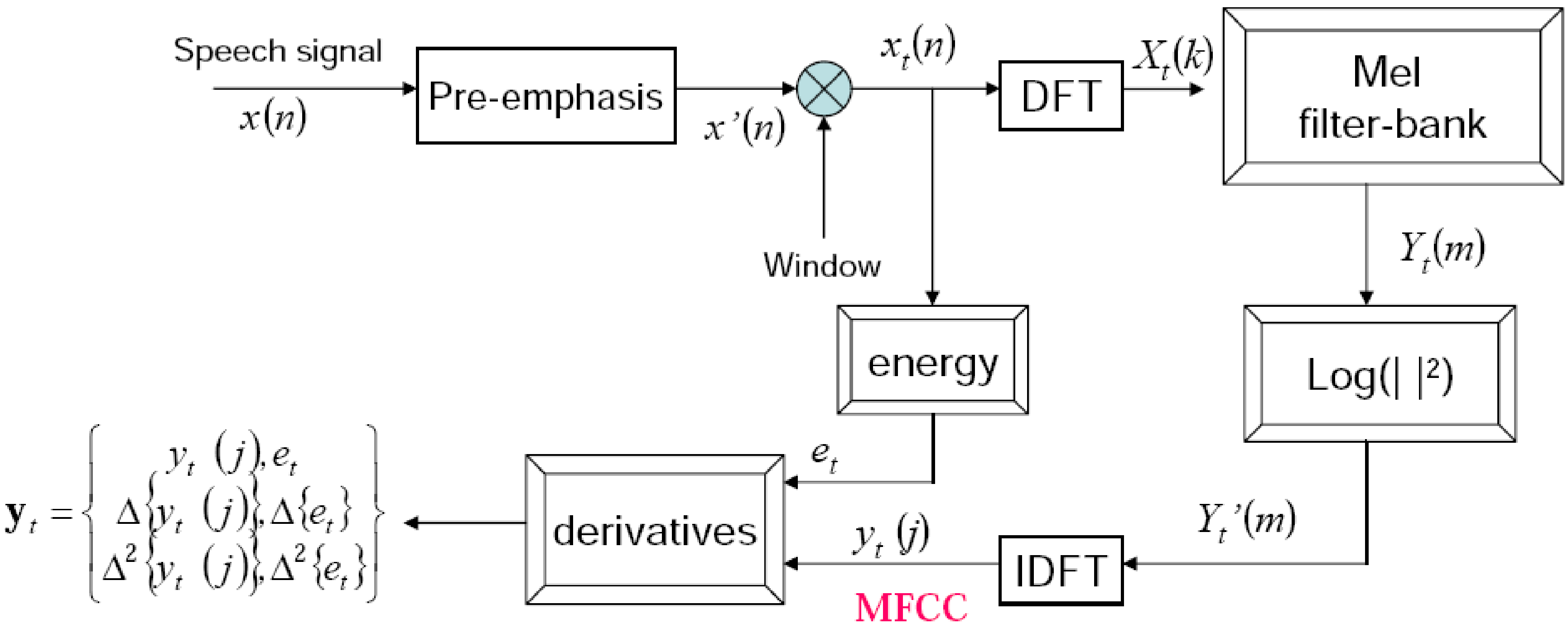


Take only first *N* (usually 12) coefficients ➔ Spectral envelope ➔ features for speech recognition

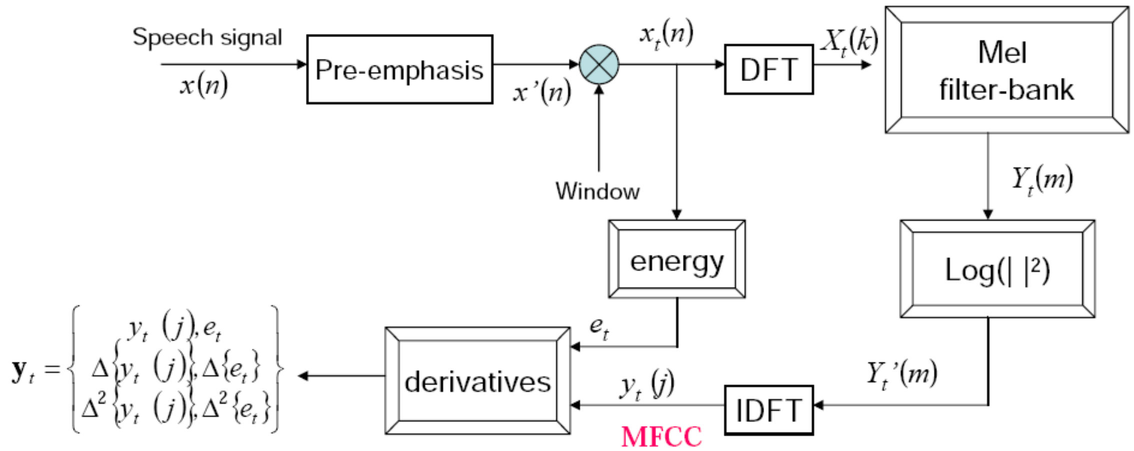
[Rabiner & Schafer, 2007]

MFCC

- Mel-Frequency Cepstral Coefficient (MFCC)
  - Most widely used spectral representation in ASR



MFCC Steps



- **Preemphasis** deemphasizes the low frequencies (similar to the effect of the basilar membrane)
- **Windowing** divides the signal into 20-30 ms frames with less than 50% overlap applying Hamming windows to each
- **FFT** of length 256-512 is performed on each windowed audio frame
- **Mel-Scale Filtering** results in 20-40 filter values per frame
- **Discrete Cosine Transform** (DCT) further reduces the coefficients to 12-14 (or some other reasonable number)
- The resulting coefficients are statistically trained for ASR

Note: DCT used because it is faster than FFT and we ignore the phase

Using DCT

- The cepstrum requires Fourier analysis
  - But we're going from frequency space back to time
- So we actually apply inverse DFT

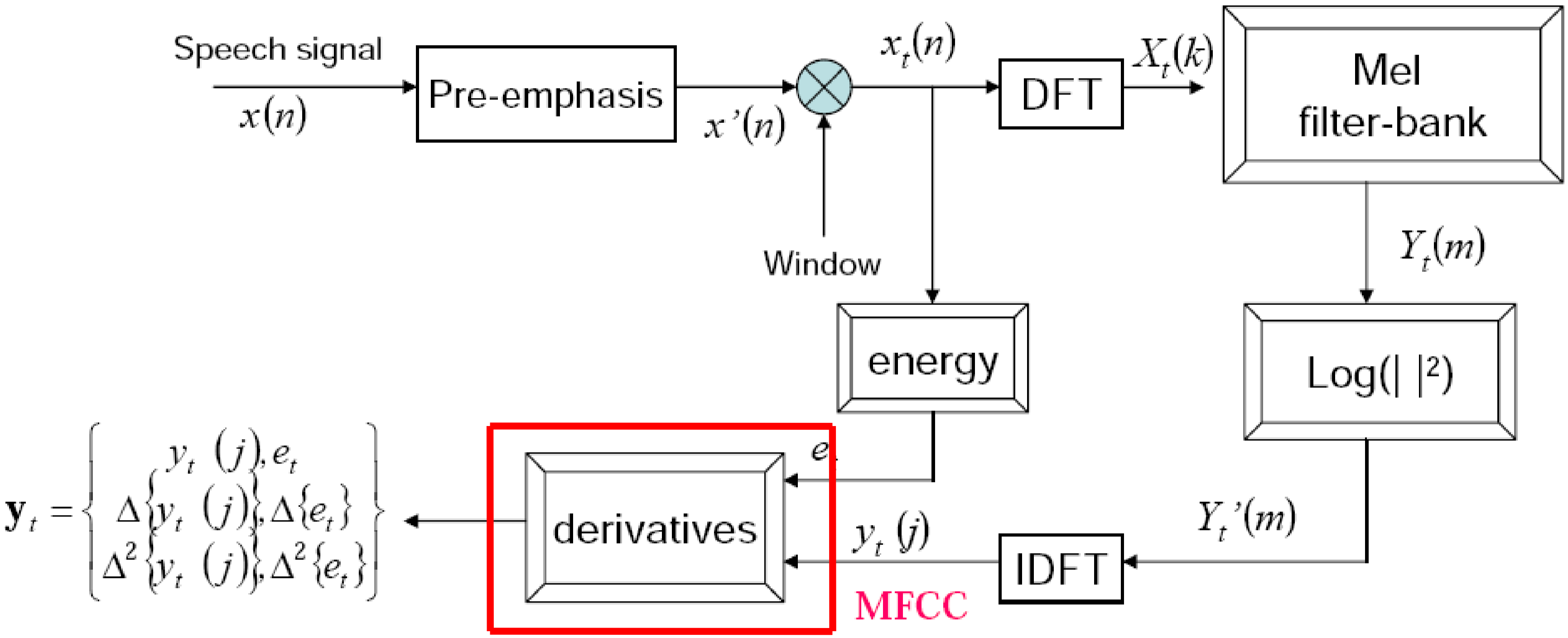
$$y_t[k] = \sum_{m=1}^{M} \log(|Y_t(m)|) \cos(k(m-0.5)\frac{\pi}{M}), \text{ k=0,…,J}$$

- Details for signal processing view:
  - Since the log power spectrum is real and symmetric, inverse DFT reduces to a Discrete Cosine Transform (DCT)

Another advantage of the Cepstrum

- DCT produces highly **uncorrelated** features
- We'll see when we get to acoustic modeling that these will be much easier to model than the spectrum
  - Simply modelled by linear combinations of Gaussian density functions with diagonal covariance matrices
- In general we'll just use the first 12 cepstral coefficients (we don't want the later ones which have e.g. the F0 spike)

MFCC


Speech signal
$x(n)$
Pre-emphasis
$x'(n)$
Window
$x_t(n)$
DFT
$X_t(k)$
Mel filter-bank
$Y_t(m)$
Log(| |²)
$Y_t'(m)$
IDFT
$y_t(j)$
MFCC
energy
e_t
derivatives
$\mathbf{y}_t = \begin{Bmatrix} y_t(j), e_t \\ \Delta\{y_t(j)\}, \Delta\{e_t\} \\ \Delta^2\{y_t(j)\}, \Delta^2\{e_t\} \end{Bmatrix}$

Dynamic Cepstral Coefficient

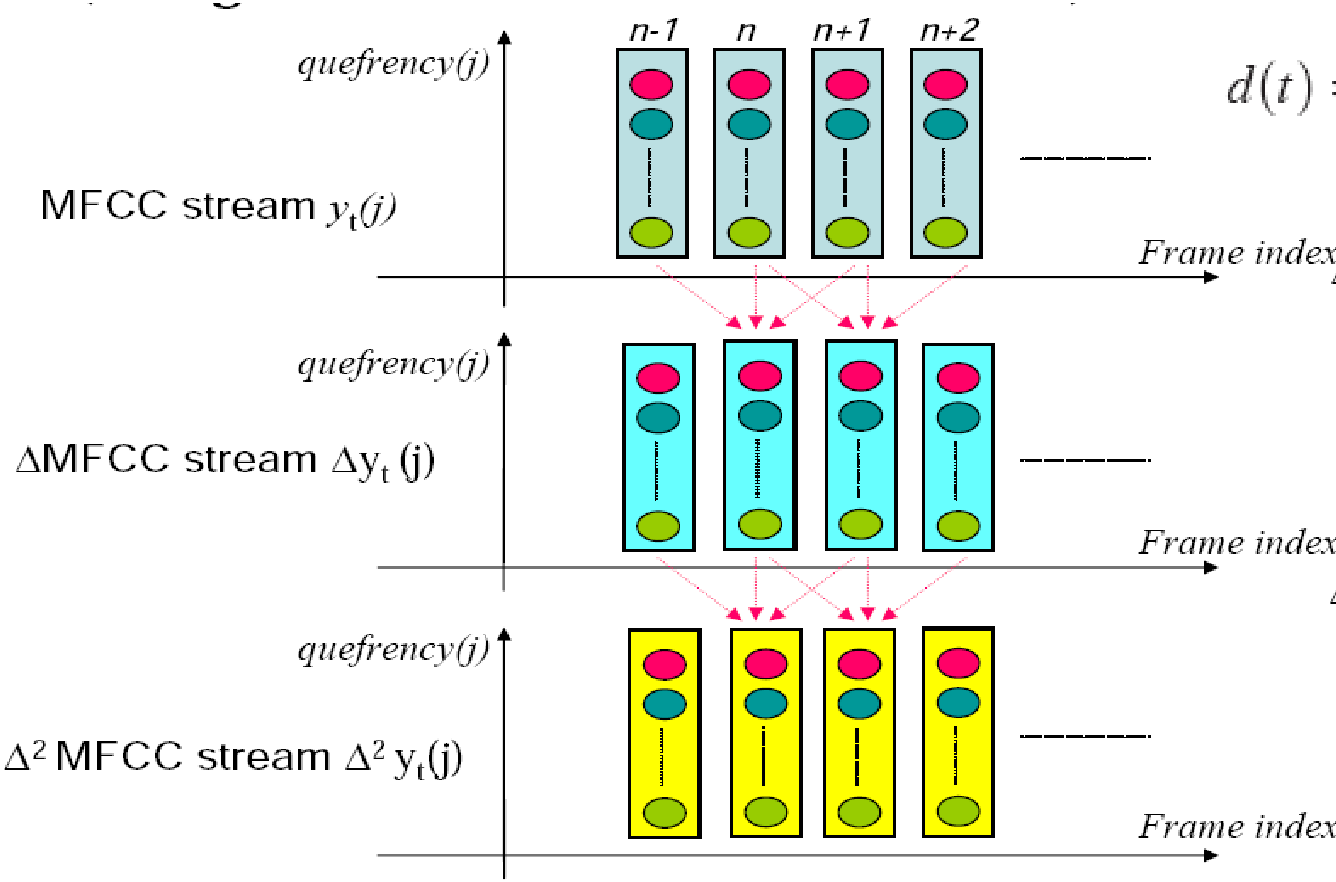
- The cepstral coefficients do not capture energy
- So we add an energy feature
  - 12 ➔ 13 dim

$$Energy = \sum_{t=t_1}^{t_2} x^2[t]$$

- Also, we know that speech signal is not constant (slope of formants, change from stop burst to release).
- So we want to add the changes in features (the slopes).
  - We call these **delta** features: 13 ➔ 26 dim
  - We also add **double-delta** acceleration features: 26 ➔ 30 dim

Delta and double-delta

- Derivative: in order to obtain temporal information



$$d(t) = \frac{c(t+1) - c(t-1)}{2}$$

$$\Delta y_t(j) = \frac{\sum_{m=-p}^{p} m \bullet y_{t-m}(j)}{\sum_{m=-p}^{p} m^2}$$

$$\Delta^2 y_t(j) = \frac{\sum_{m=-p}^{p} m \bullet \Delta y_{t-m}(j)}{\sum_{m=-p}^{p} m^2}$$

Typical MFCC features (from HTK)

- Window size: 25ms
- Window shift: 10ms
- Pre-emphasis coefficient: 0.97
- MFCC:
  - 12 MFCC (mel frequency cepstral coefficients)
  - 1 energy feature
  - 12 delta MFCC features
  - 12 double-delta MFCC features
  - 1 delta energy feature
  - 1 double-delta energy feature
- Total 39-dimensional features

Why is MFCC so popular?

- Efficient to compute
- Incorporates a perceptual Mel frequency scale
- Separates the source and filter
- IDFT (DCT) decorrelates the features
  - Improves diagonal assumption in HMM modeling

MFCC Enhancements

Resulting feature array size is 3 times the number of Cepstral coefficients

- Derivative and double derivative coefficients model changes in the speech between frames
- Mean, Variance, and Skew normalize results for improved ASR performance

Derivative Filter

$$\left(\sum_{d=-D}^{d=D} d * x[frame + d]\right) / \sum_{d=-D}^{d=D} d^2)$$

Mean Normalization

```
/* Note: Java */
public static double[][] meanNormalize(double[][] features,  int feature)
{   double mean = 0;
    for (int row: features)=0; row<features.length; row++)
    {   mean += features[row][feature];    }
    mean = mean / features.length;

    for (int row=0; row<features.length; row++)
    {  features[row][feature] -= mean;  }
    return features;
}   // end of meanNormalize
```

Normalize to the mean will be zero

Variance Normalization

```
/* Note: Java */
public static double[][] varNormalize(double[][] features, int feature)
{   double variance = 0;
    for (int row=0; row<features.length; row++)
    {  variance += features[row][feature] * features[row][feature];  }
    variance /= (features.length - 1);

    for (int row=0; row<features.length; row++)
    {   if (variance!=0) features[row][feature] /= Math.sqrt(variance);  }
    return features;
}  // End of varianceNormalize()
```

Scale feature to [-1,1] - divide the feature's by the standard deviation

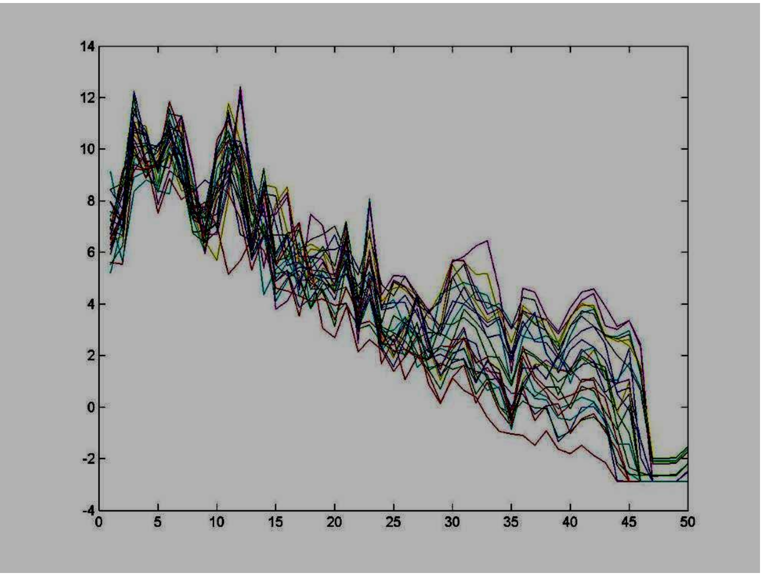
Skew Normalization

```
public static double[][] skewNormalize(double[][] features, int feature)
{  double fN=0, fPlus1=0, fMinus1=0, value, coefficient;
   for (int row=0; row<features.length; row++)
   {  fN += Math.pow(features[row][feature], 3);
      fPlus1 += Math.pow(features[row][feature], 4);
      fMinus1 += Math.pow(features[row][feature], 2);
   }
   if (momentNPlus1 != momentNMinus1) coefficient = -fN/(3*(fPlus1-fMinus1));
   for (int row=0; row<features.length; row++)
   {  value = features[row][column];
      features[row][column] = coefficient * value * value + value - coefficient;
   }
   return features;
}  // End of skewNormalization()
```

Minimizes the skew for the distribution to be more normal

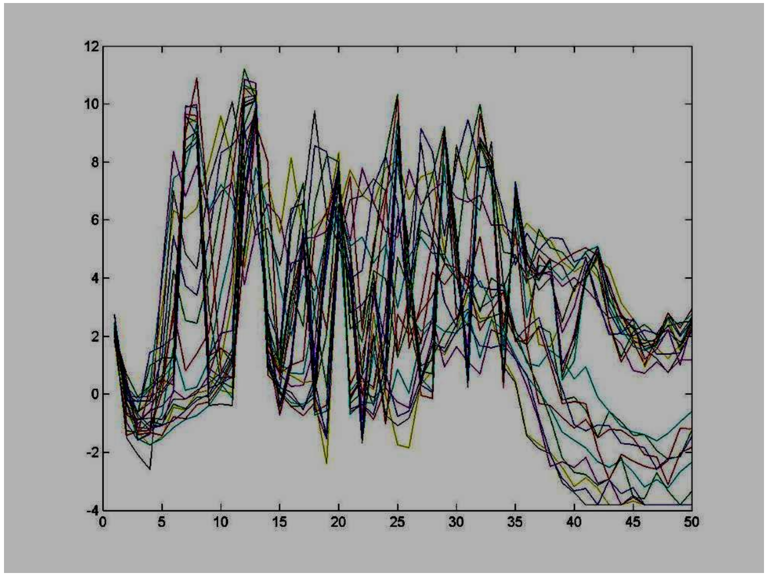
Mel-Scale Spectra of Music

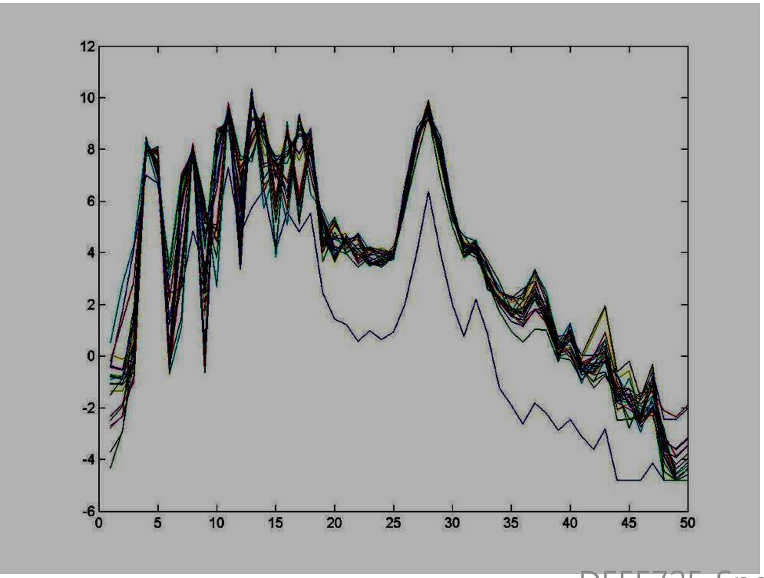
(Petruncio, B.S. Thesis University of Illinois, 2003)



Piano

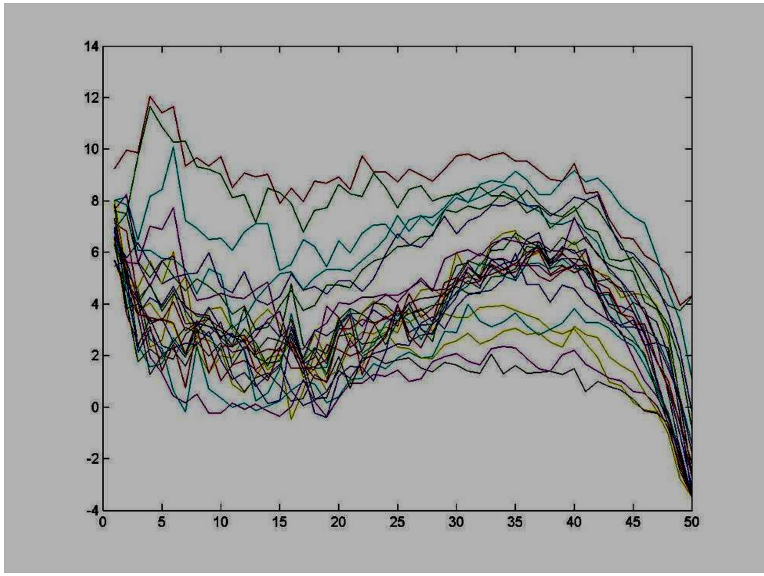
Saxophone





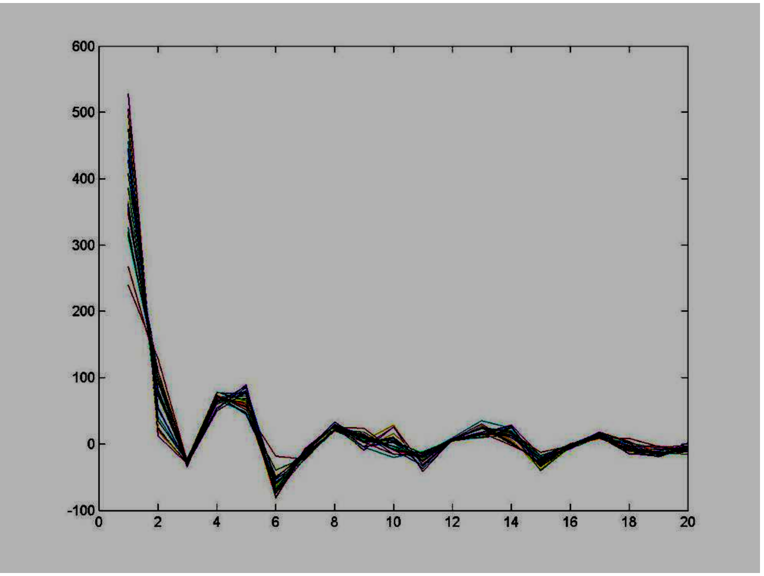
Tenor
Opera
Singer

Drums

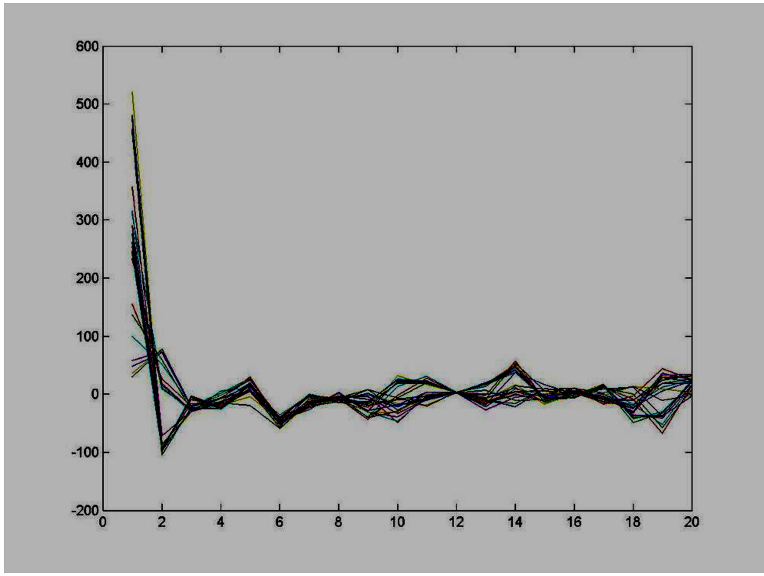


MFCC of Music

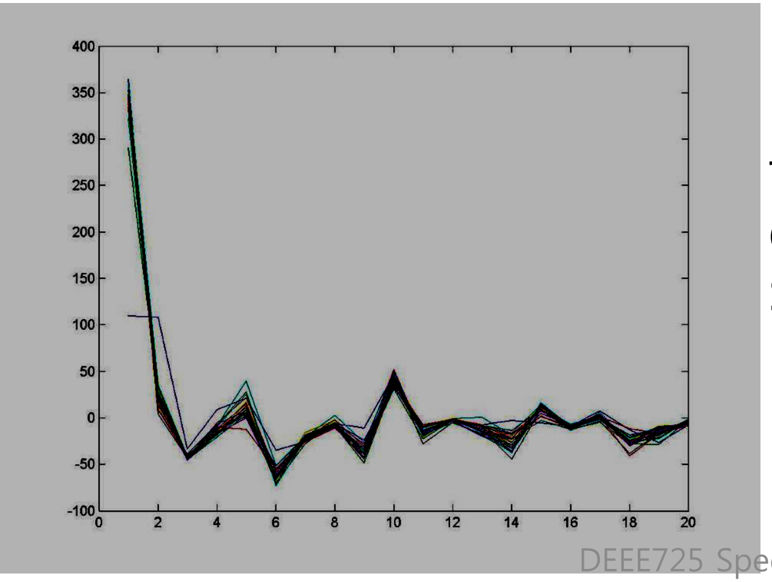
(Petruncio, 2003)



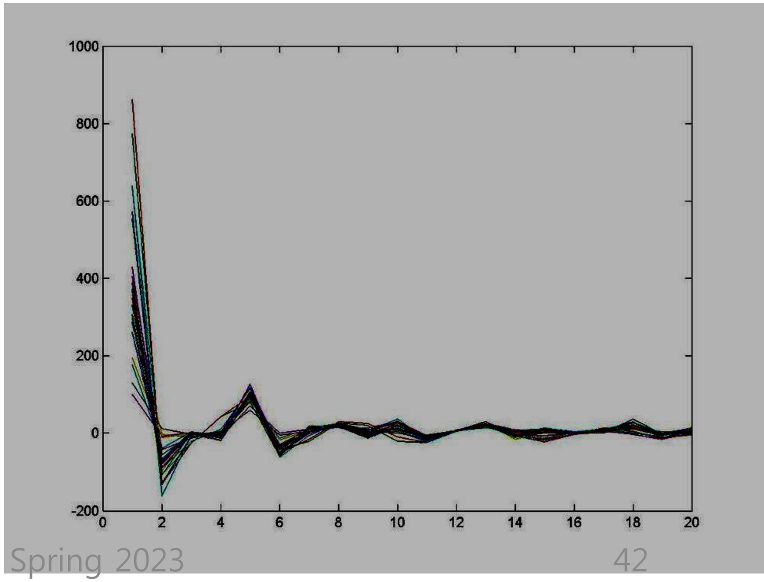
Piano



Saxophone



Tenor
Opera
Singer



Drums

Cepstrum-Based Features

- Average(log(energy)) = c[0]
  - $c[0] = \int \log|X(\omega)|d\omega = ½ \int \log |X(\omega)|^2 d\omega$
  - Not the same as log(average(energy)), which is $\log \int |X(\omega)|^2 d\omega$

- Spectral Tilt: one measure is -c[1]
  - $-c[1] = -\int \log|X(\omega)|\cos(\omega)d\omega \approx$ HF log energy – LF log energy
- A More Universally Accepted Measure:
  - Spectral Tilt = $\int (\omega-\pi/2) \log|X(\omega)|\, d\omega$

- Spectral Centrality: -c[2]
  - $c[2] = -\int \log|X(w)|\cos(2w)dw$
  - $c[2] \approx$ Mid Frequency Energy ($\pi/4$ to $3\pi/4$) – Low and High Frequency Energy (0 to $\pi/4$ and $3\pi/4$ to π)

Summary

- Log spectrum, once/10ms, computed with a window of about 25ms, seems to carry lots of useful information about place of articulation and vowel quality
  - Euclidean distance between log spectra is not a good measure of perceptual distance
  - Euclidean distance between windowed cepstra is better
  - Frequency warping (mel-scale or Bark-scale) is even better
  - Fitting an all-pole model (PLP) seems to improve speaker-independence
  - Modulation filtering (CMS, RASTA) improve robustness to channel variability (short-impulse-response reverb)
- Time-domain features (once/1ms) can capture important information about manner of articulation and landmark times
- Auditory model features (correlogram, delayogram) are useful for recognition of multiple simultaneous talkers

ELEC747 Speech Signal Processing

Gil-Jin Jang

END OF
ACOUSTIC FEATURE EXTRACTION FOR SPEECH RECOGNITION