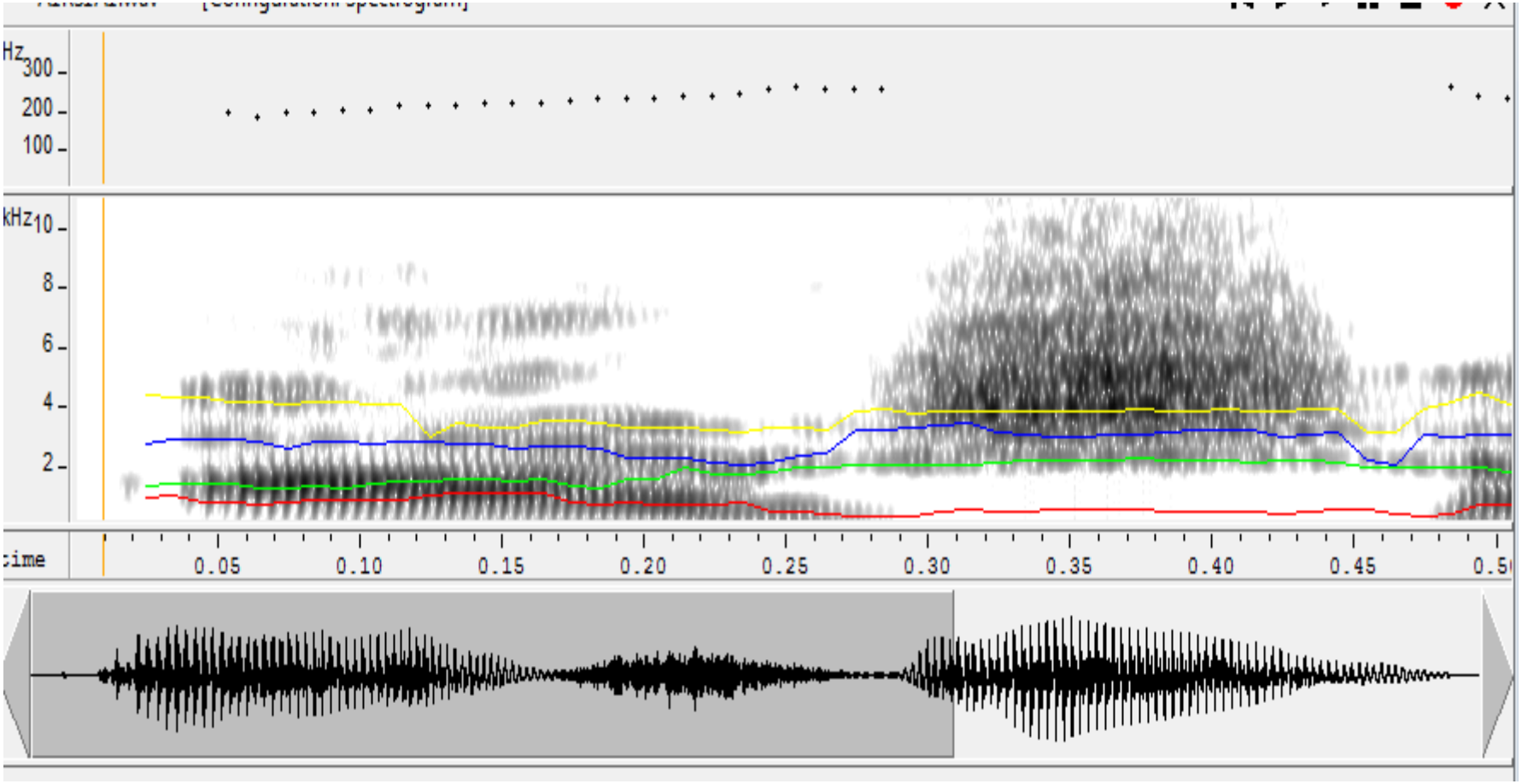
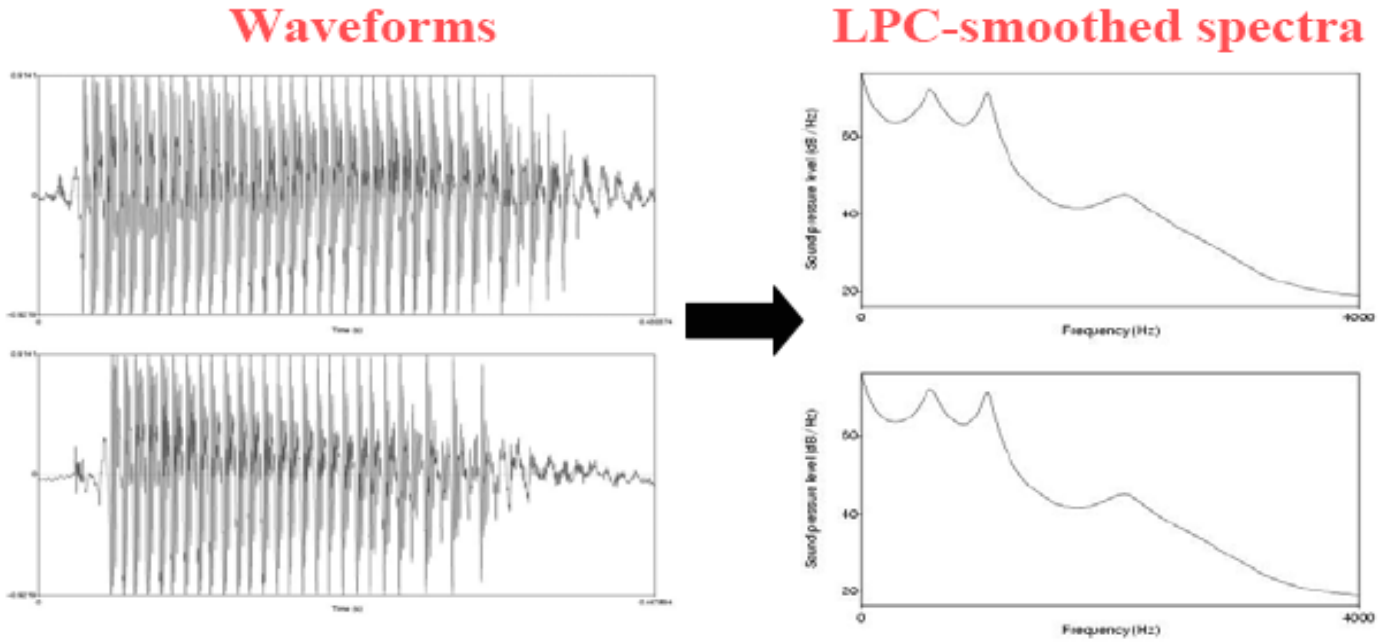
Features Extraction


Hz
300
200
100
kHz
10
8
6
4
2
time
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

Why do we need feature extraction?

- Acoustic speech signal varies over time. Can't compare two waveforms

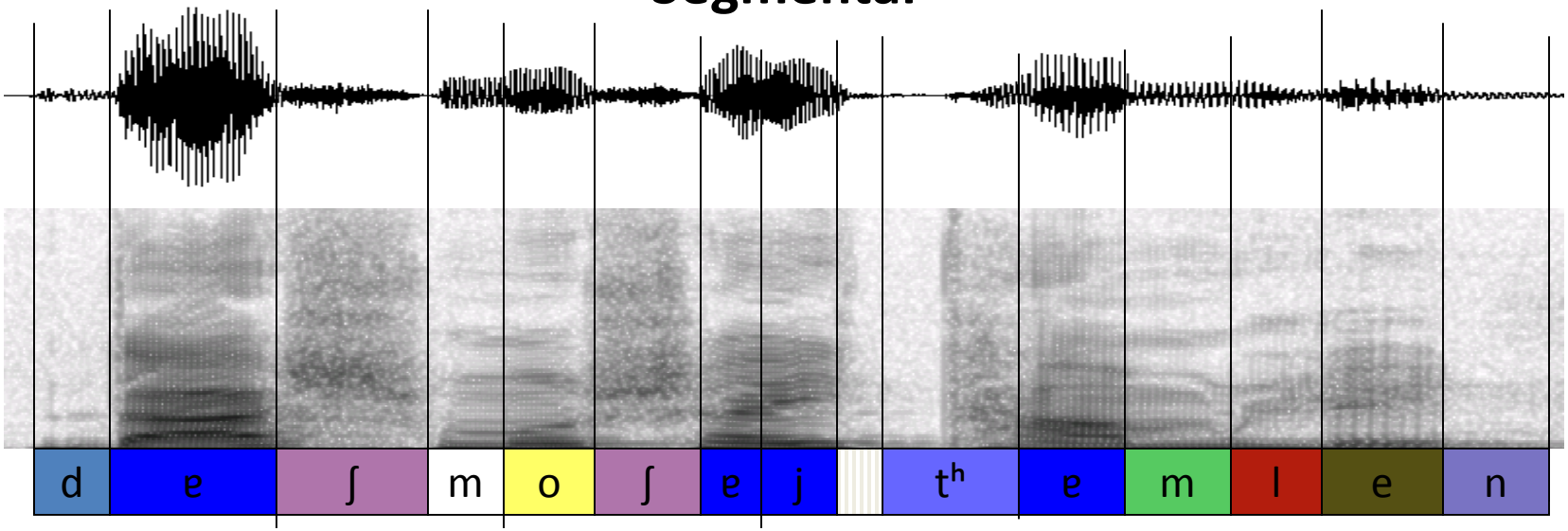
example: two instances of /a:/ vowel spoken in isolation, with time interval between repetitions < 1 second:

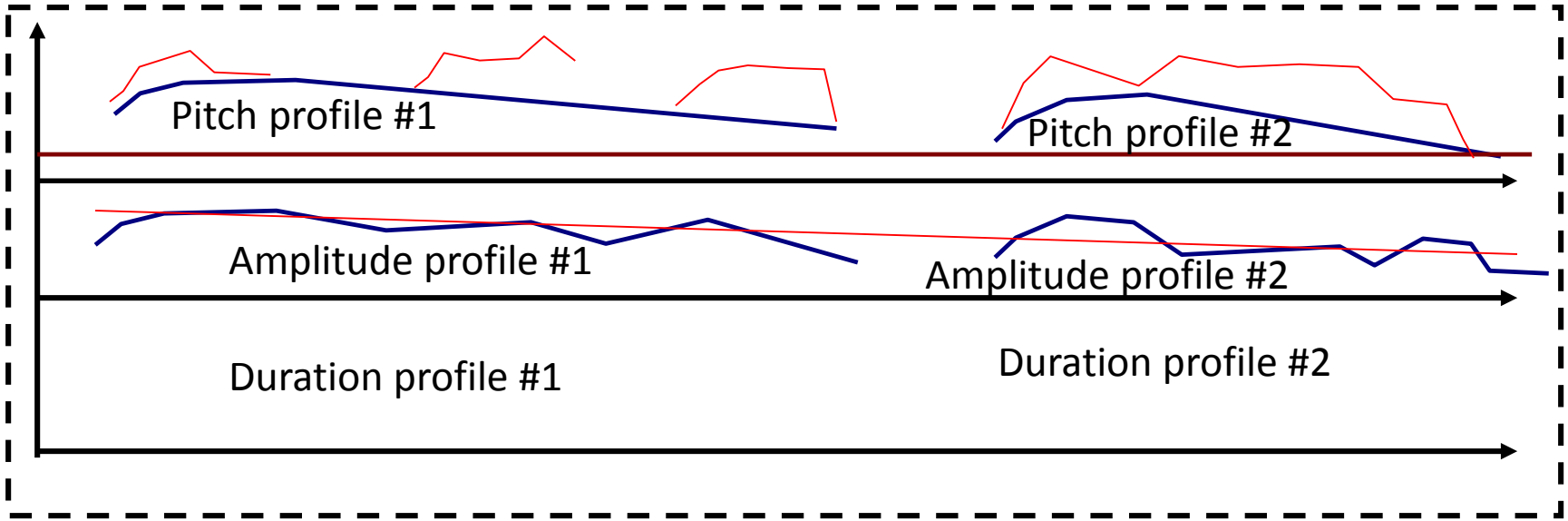


What is Features?

- Feature = a measure of a property of the speech waveform
- Reasons for feature extraction:
  - Redundancy and harmful information is removed
  - Reduced computation time
  - Easier modeling of the feature distribution
- Speech has many “natural” (Acoustic-phonetic) features:
  - Fundamental frequency (F0), formant frequencies, formant bandwidths, spectral tilt, intensity, phone durations, articulation, etc
- Not-so-natural features:
  - Cepstrum, linear predictive coefficients, line spectral frequencies, vocal tract area function, delta and double-delta coefficients, etc

Segmental





Supra-Segmental

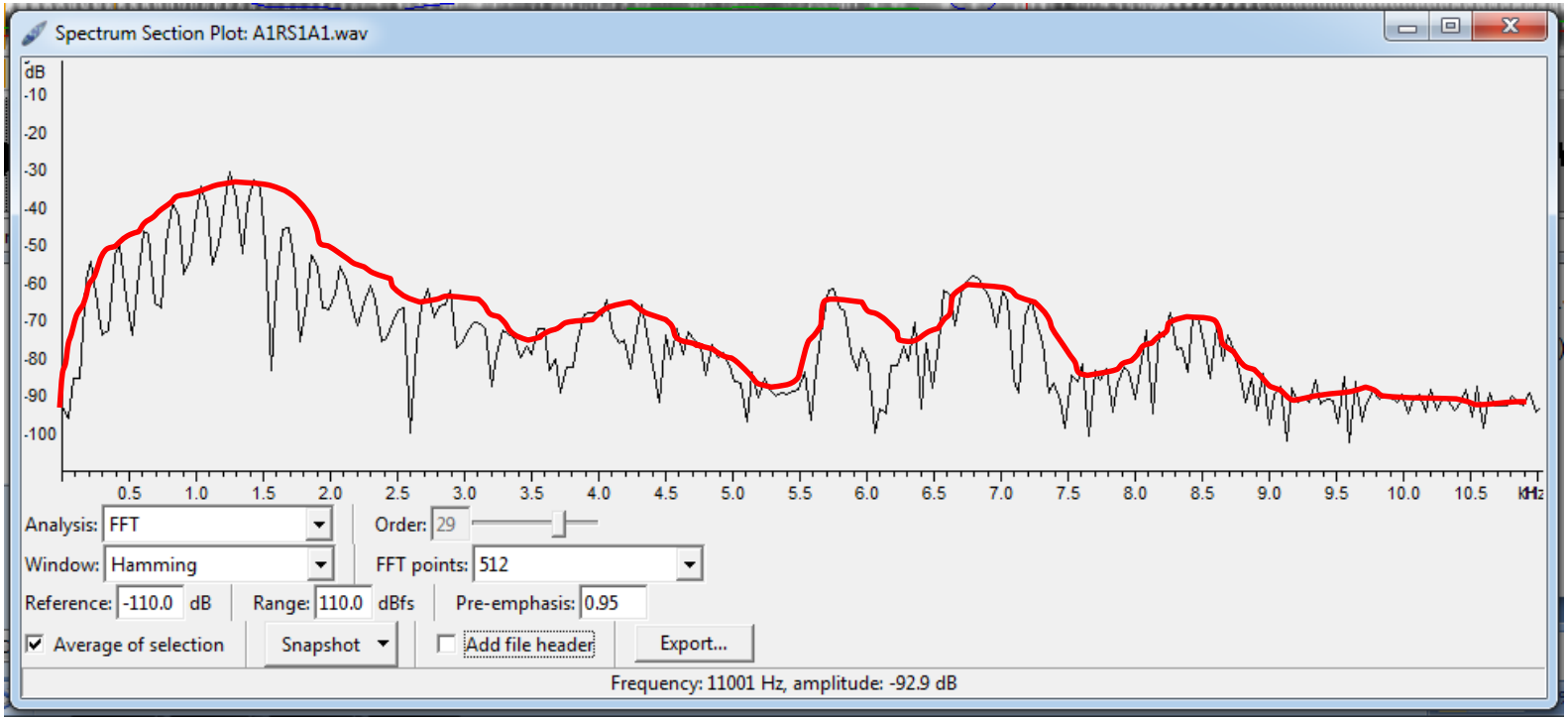
Speech Events

Segmental

Supra-segmental

Supra-segmental features and Prosody

- Intonation, pause, duration, stress together are called prosodic or supra-segmental features and may be considered as the melody, rhythm, and emphasis of the speech at the perceptual level.
- The prosody of a sentence is important for naturalness and for conveying the correct meaning of a sentence.



- ❑ Peaks denote dominant frequency components in the speech signal
- ❑ Peaks are referred to as formants
- ❑ Formants carry the identity of the sound

Parameter / Feature Classification

Frequency Domain Parameters

- Filter Bank Analysis
- Short-term spectral analysis
- Cepstral Transfer Coefficient (CC)
- Formant Parameters
- MFCC, Delta MFCC, Delta-Delta MFCC

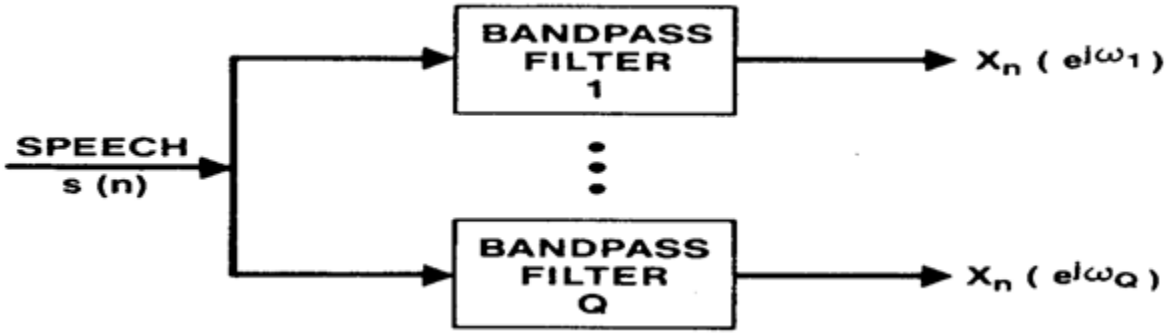
Time Domain Parameters

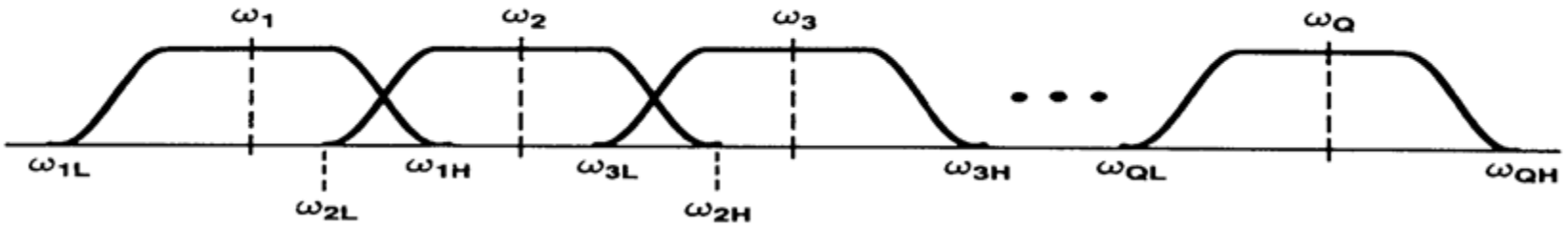
- LPC
- Shape Parameters

Time- Frequency Domain Parameters

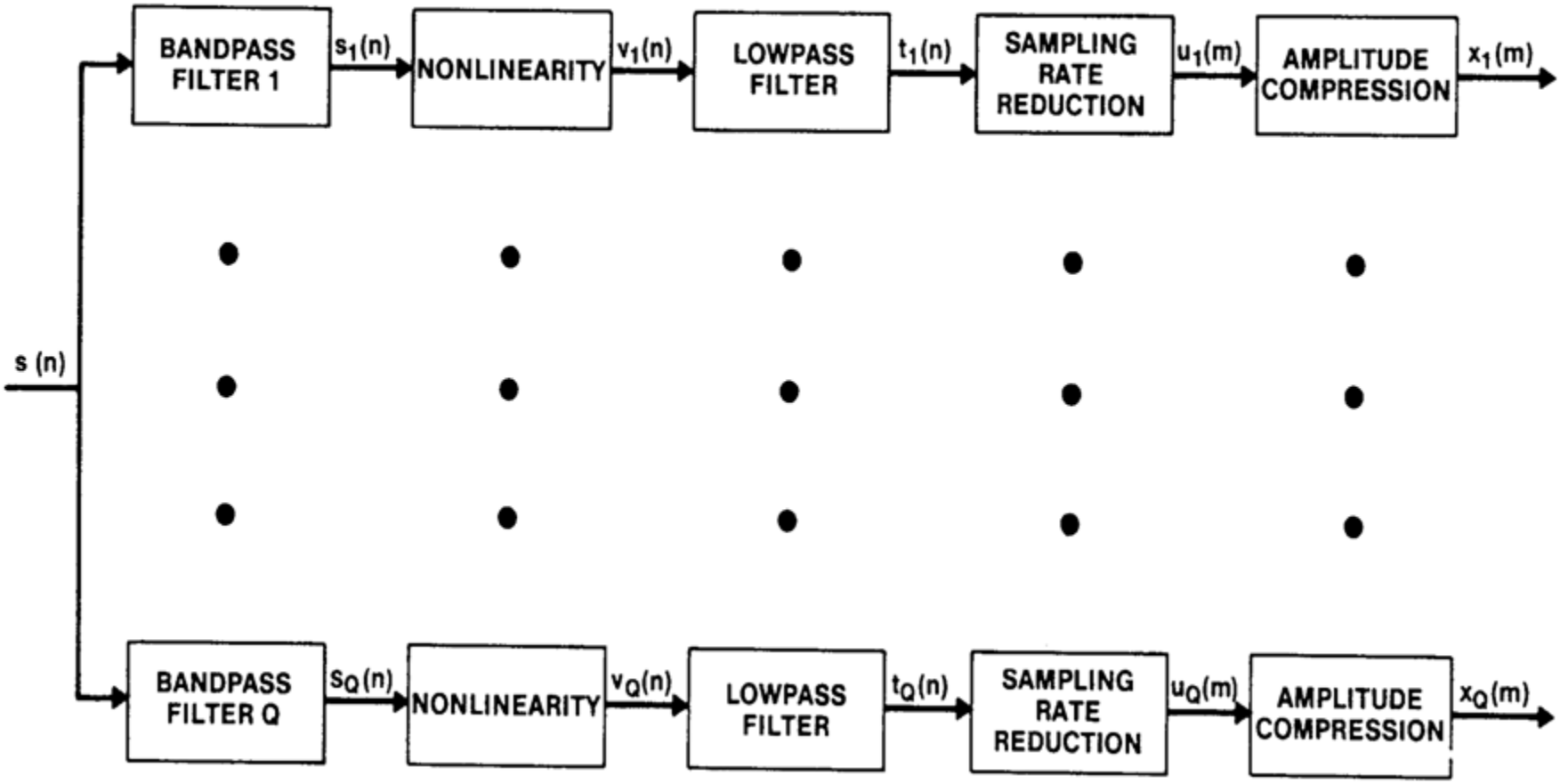
- Perceptual Linear Prediction (PLP):
- Wavelet Analysis

Filter Bank Analysis





Complete Filter Bank Analysis Model

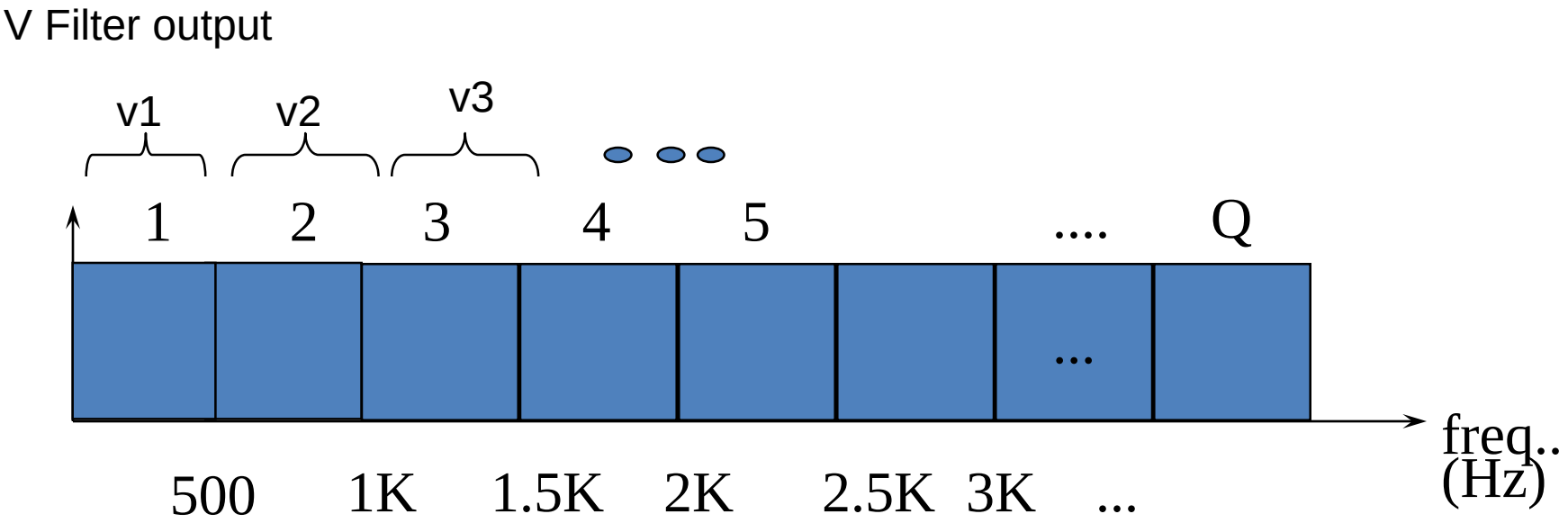


How to determine filter band ranges

- Uniform filter banks
- Log frequency banks
- Mel filter bands

Uniform Filter Banks

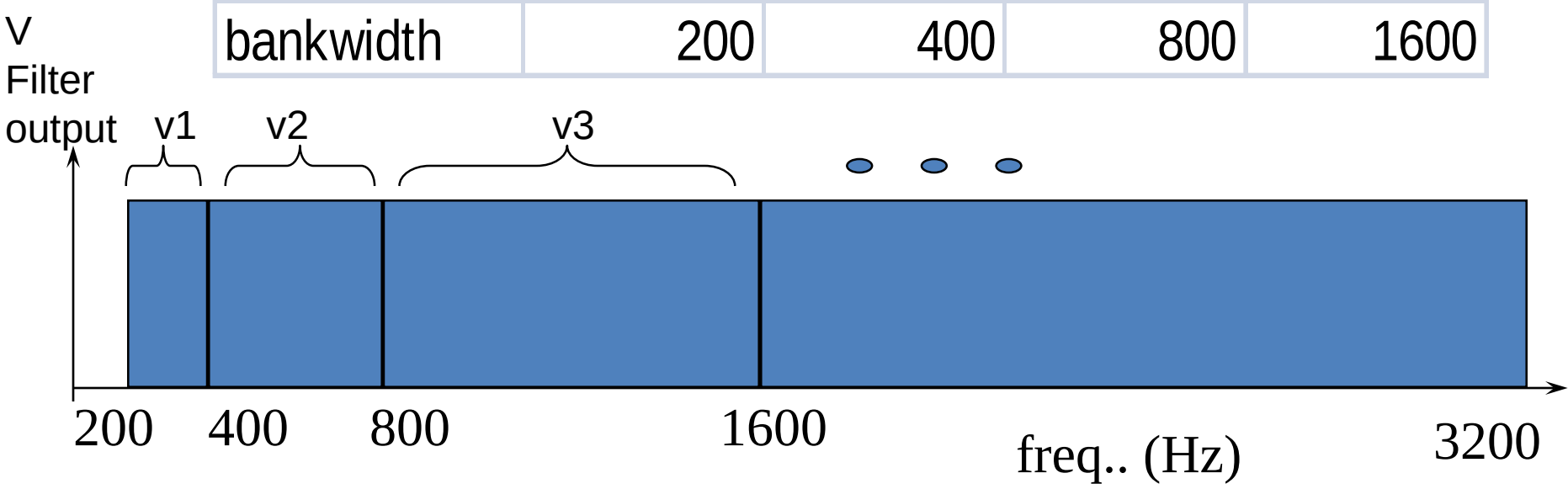
- Uniform filter banks
  - bandwidth B= Sampling Freq... (Fs)/no. of banks (N)
  - For example Fs=10Kz, N=20 then B=500Hz
  - Simple to implement but not too useful



Non-uniform filter banks: Log frequency

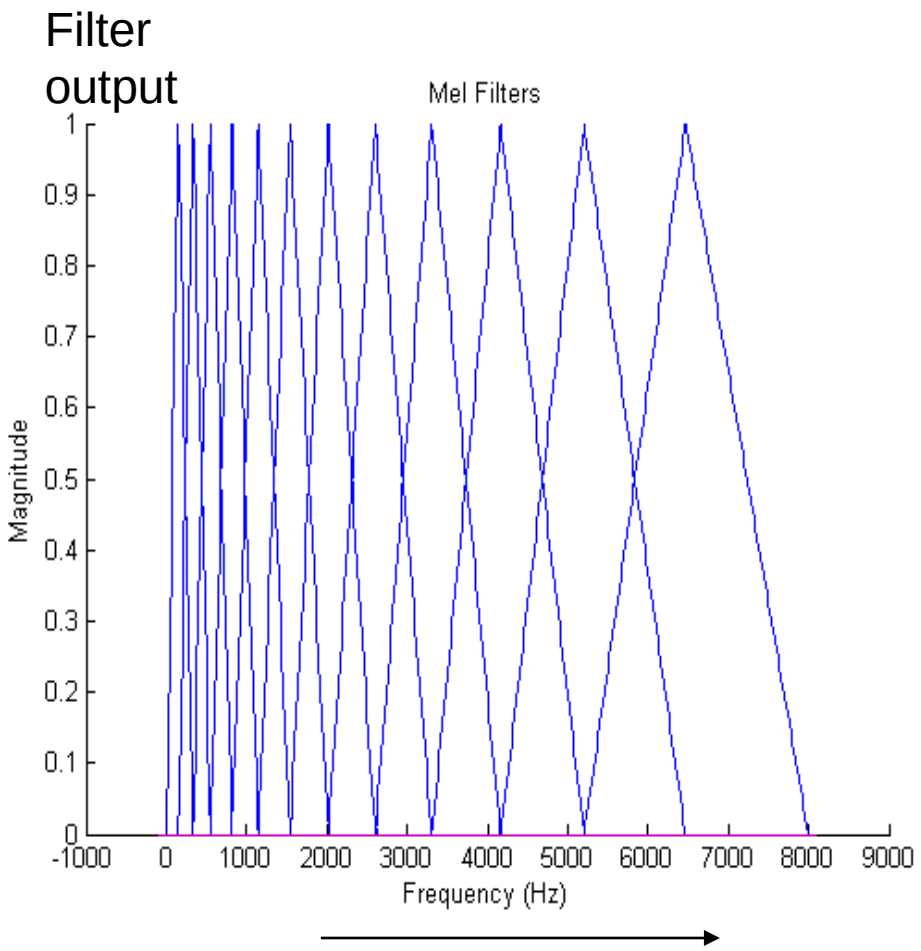
- Log. Freq... scale : close to human ear

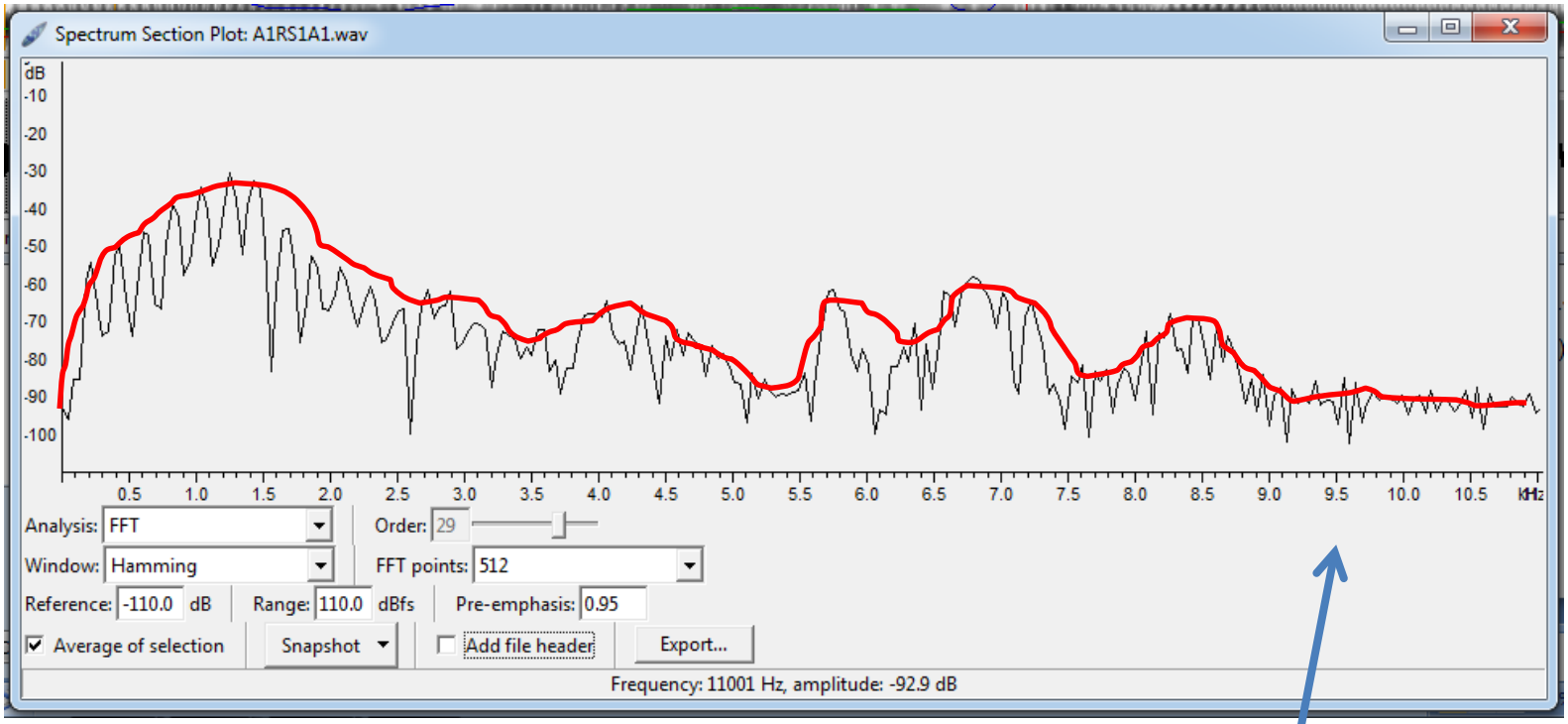
	filter 1	filter 2	filter 3	filter 4
Center freq.	300	600	1200	2400
bankwidth	200	400	800	1600



Mel filter bands

- Freq. lower than 1 KHz has narrower bands (and in linear scale)
- Higher frequencies have larger bands (and in log scale)
- More filter below 1KHz
- Less filters above 1KHz

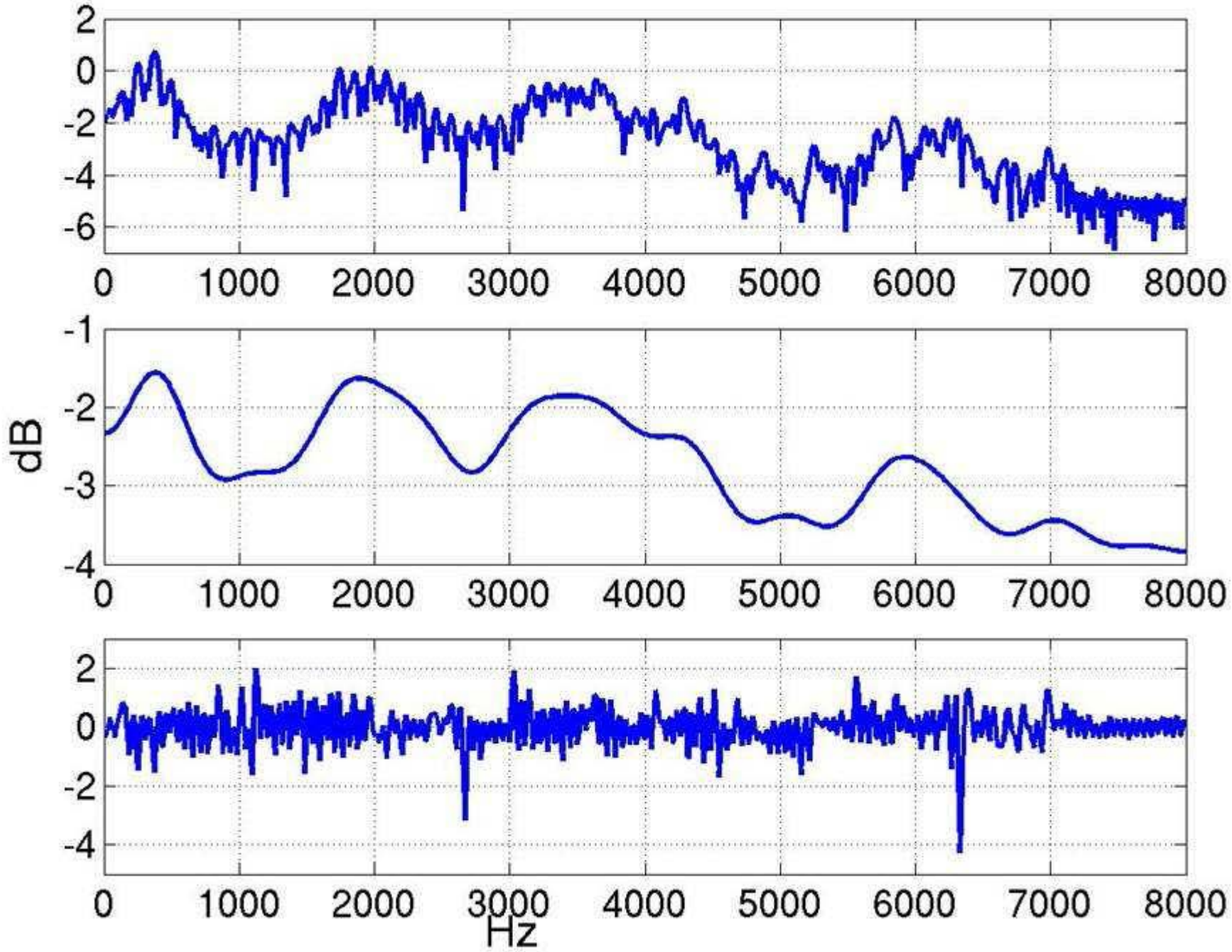




$$s[n] = h[n] * e[n]$$

DFT of s[n] $$S[k] = H[k]E[k]$$

$$\log(|S[k]|) = \log(|H[k]|) + \log(|E[k]|)$$


2
0
-2
-4
-6
0
1000
2000
3000
4000
5000
6000
7000
8000
-1
-2
-3
-4
dB
0
1000
2000
3000
4000
5000
6000
7000
8000
2
0
-2
-4
0
1000
2000
3000
4000
5000
6000
7000
8000
Hz

Cepstral analysis

- **Homomorphic speech processing**
  - Speech is modelled as the output of a linear, time varying system (linear time-invariant (LTI) in short seg.) excited by either quasi-periodic pulses or random noise.
  - The problem of speech analysis is to estimate the parameters of the speech model and to measure their variations with time.
  - Since the excitation and impulse response of a LTI system are combined in a convolutional manner, the problem of speech analysis can also been viewed as a problem in separating the components of a convolution, called ”deconvolution”.

$$y[n] = x[n] * h[n]$$

The principle of superposition for conventional linear systems:

$$\begin{cases} L[x(n)] = L[x_1(n) + x_2(n)] = L[x_1(n)] + L[x_2(n)] \\ \qquad\qquad = y_1(n) + y_2(n) = y(n) \\ L[ax(n)] = aL[x(n)] = ay(n) \end{cases}$$

If signals fall in non-overlapping frequency bands then they are separable

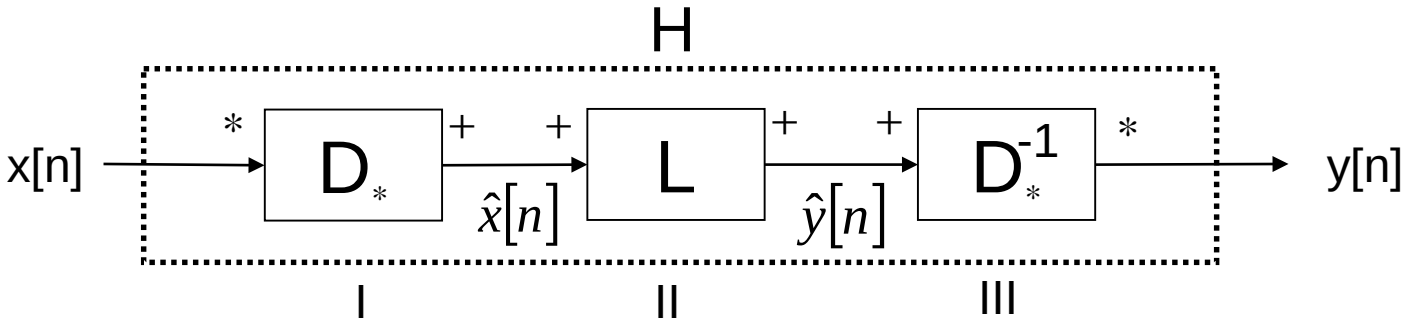
$x[n]=x_1[n]+x_2[n]$

$X_1(\omega)=\mathcal{F}\{x_1[n]\}$ & $X_1(\omega)$ $[0, \pi/2]$,

$X_2(\omega)=\mathcal{F}\{x_2[n]\}$ & $X_2(\omega)$ $[\pi/2, \pi]$,

Principles of Homomorphic Processing

- Importance of homomorphic systems for speech processing lies in their capability of transforming nonlinearly combined signals to additively combined signals so that linear filtering can be performed on them.
- Homomorphic systems can be expressed as a cascade of three homomorphic sub-systems → referred to as the ***canonic representation***:



Homomorphic Systems for Convolution

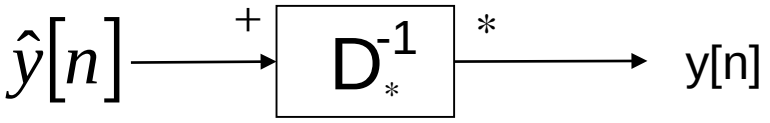
Canonic Representation of a Homomorphic System

I. System takes inputs combined by convolution and transforms them into additive outputs
II. System is a conventional linear system
III. Inverse of first system--takes additive inputs and transforms them into convolution outputs

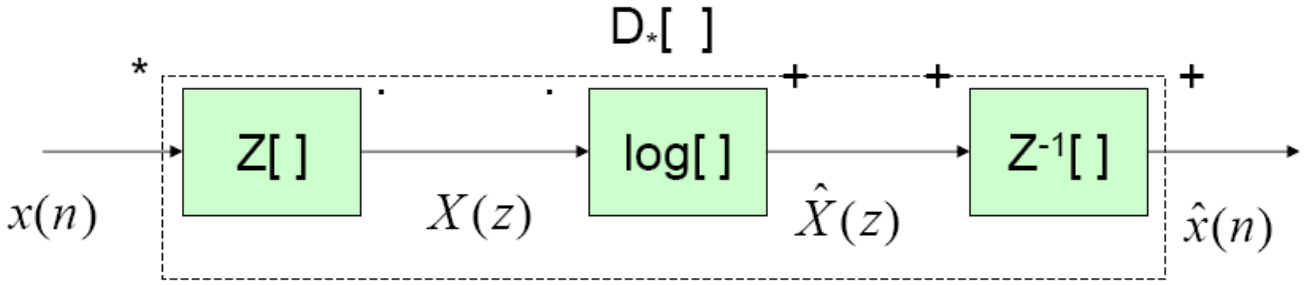
I

II

III



- The characteristic system for homomorphic deconvolution



Cepstral analysis

Observation:

$$x[n] = x_1[n] * x_2[n] \Leftrightarrow X(z) = X_1(z)X_2(z)$$

taking logarithm of X(z), then

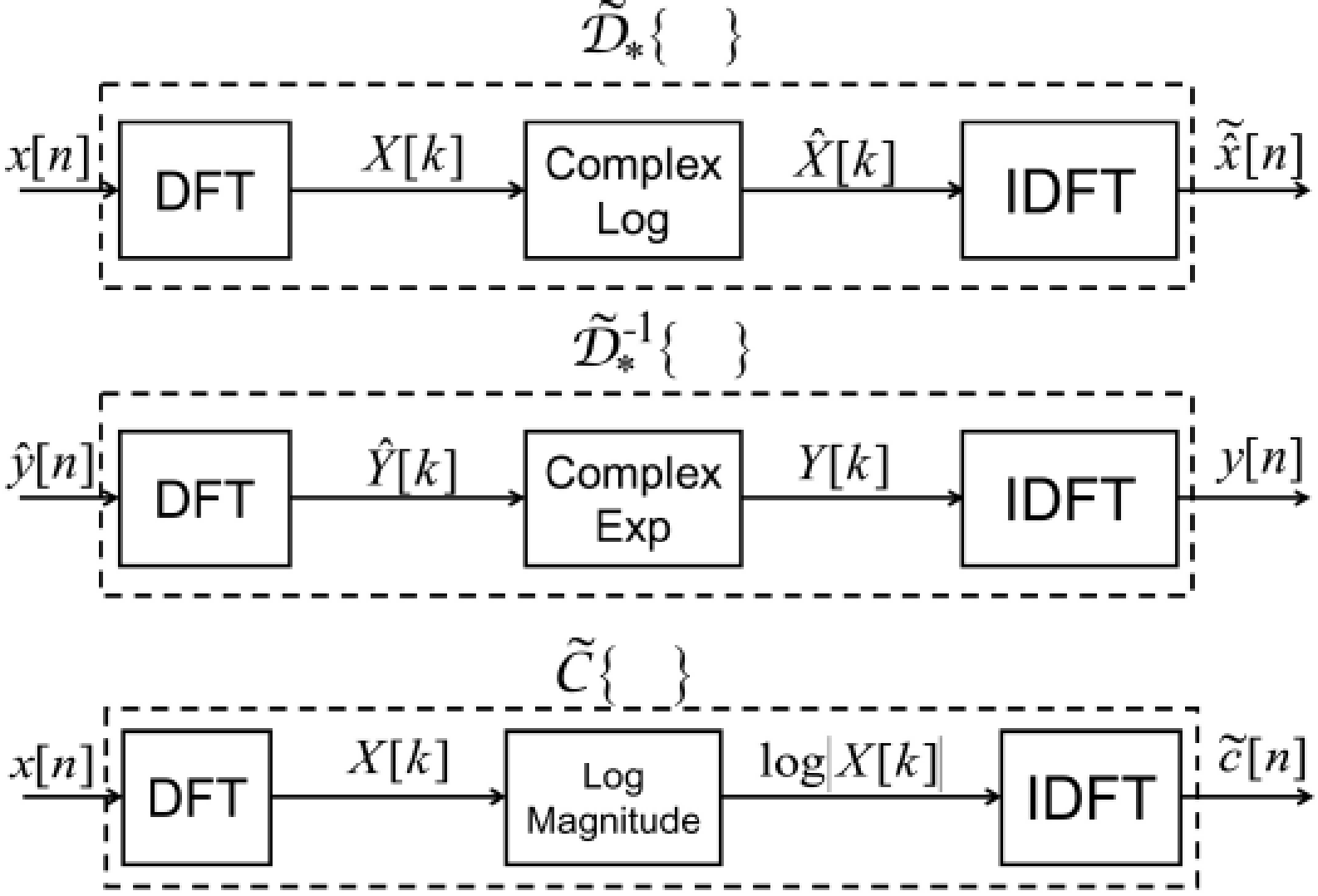
$$\log\{X(z)\} = \log\{X_1(z)\} + \log\{X_2(z)\}$$

$$\text{i.e.,}\ \hat{X}(z) = \hat{X}_1(z) + \hat{X}_2(z)$$

$$\hat{x}[n] = \hat{x}_1[n] + \hat{x}_2[n]$$ in the cepstral domain

- So, the two convolved signals are additive in the cepstral domain

Computational Considerations



Cepstral analysis

Real cepstrum *c[n]* is the even part of $\hat{x}[n]$

$$\begin{cases} \hat{x}[n] = \dfrac{1}{2\pi}\int_{-\pi}^{\pi} \hat{X}(e^{jw})e^{jwn}dw \\ \qquad = \dfrac{1}{2\pi}\int_{-\pi}^{\pi} \log\{X(e^{jw})\}e^{jwn}dw & \text{complex cepstrum} \\ c[n] = \dfrac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(e^{jw})|e^{jwn}dw & \text{cepstrum} \end{cases}$$

- Relationship of complex cepstrum $\hat{x}[n]$ to real cepstrum c[n]:
  - If *x[n]* real then:
    - |X(ω)| is real and even and thus log[|X(ω)|] is real and even
    - ∠X(ω) is odd, and hence

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

$\hat{x}[n]$ is referred to as the ***complex cepstrum***.

- Even component of the complex cepstrum, c[n] is referred to as the ***real cepstrum***.

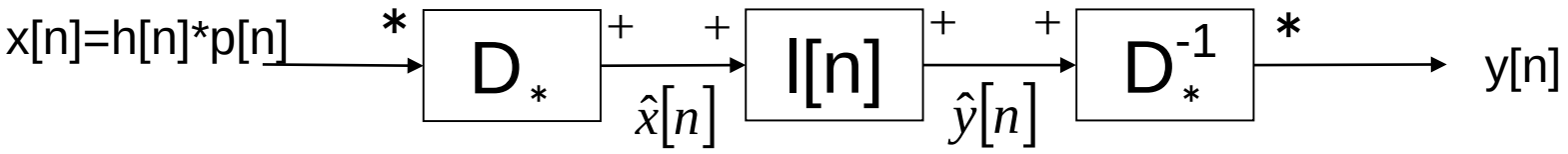
$$c[n]=\frac{\hat{x}[n]+\hat{x}[-n]}{2}$$

Homomorphic Filtering

- In the cepstral domain:
  - Pseudo-time ⇔ ***Quefrency***
  - Low Quefrency ⇔ Slowly varying components.
  - High Quefrency ⇔ Fast varying components.
- Removal of unwanted components (i.e., filtering) can be attempted in the cepstral domain (on the signal $\hat{x}[n]$, in which case filtering is referred to as ***liftering***):
- When the complex cestrum of h[n] resides in a quefrency interval less than a pitch period, then the two components can be separated form each other.

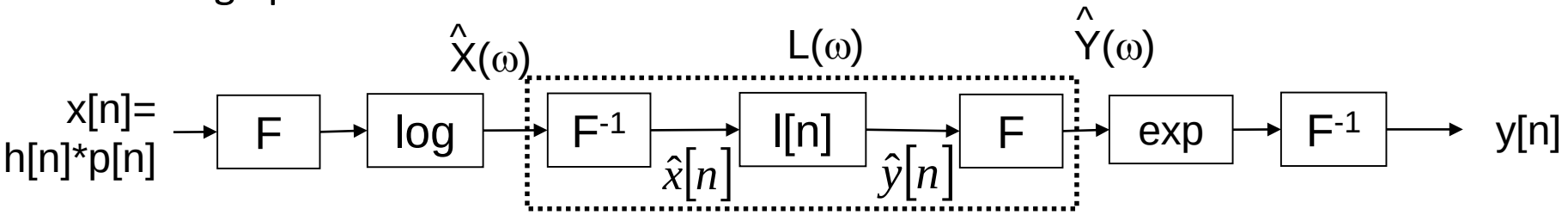
Homomorphic Filtering

- If log[X(ω)]
  - Is viewed as a “time signal”
  - Consisting of low-frequency and high-frequency contributions.
  - Separation of this signal with a high-pass/low-pass filter.
- One implementation of low pass filter:



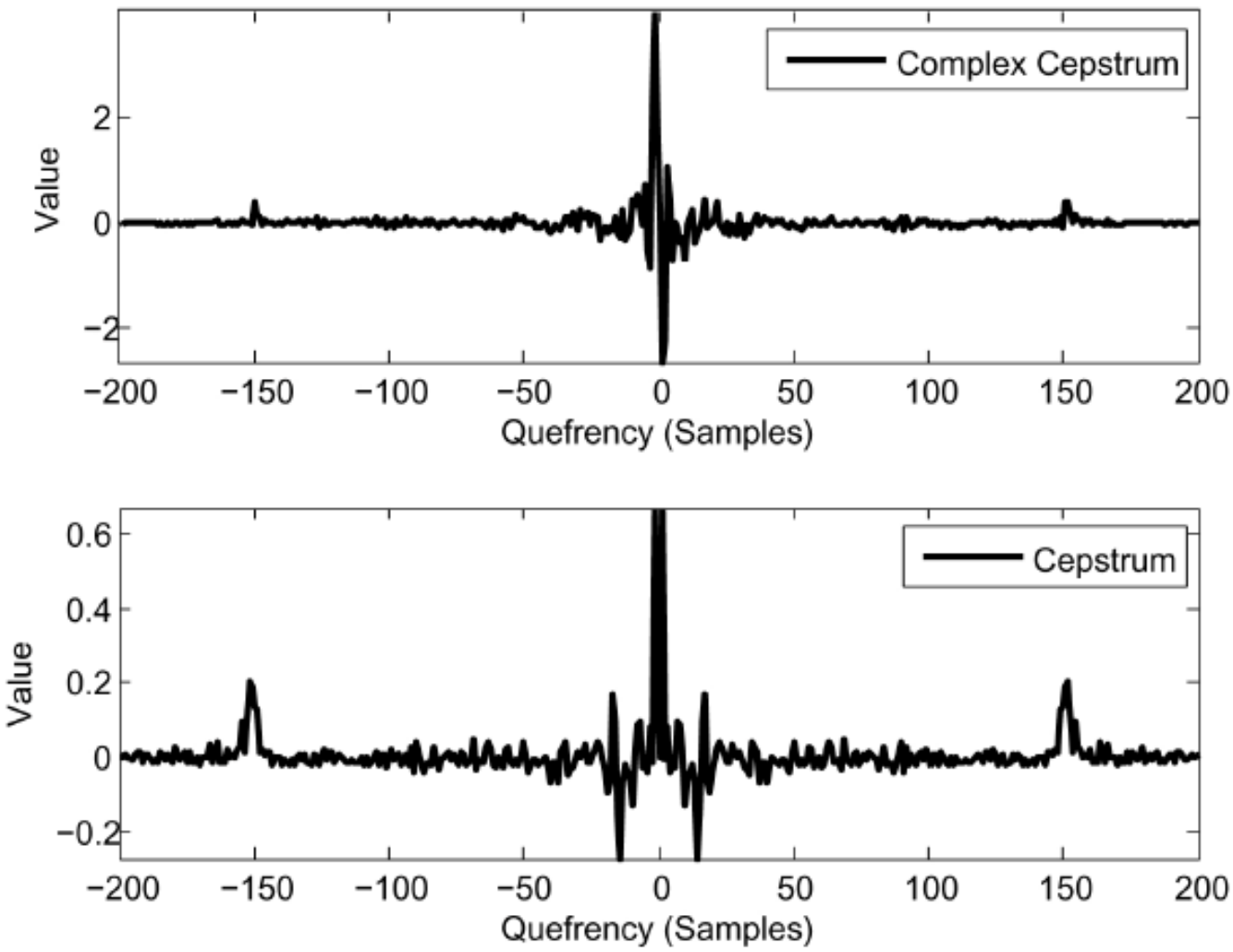
Homomorphic Filtering

- Alternate view of "liftering" operation: Filtering operation L(ω) applied in the log-spectral domain



- Interchange of time and frequency domain by viewing the frequency-domain signal log[X(ω)] as a time signal to be filtered. ⇒
  - "Cepstrum" can be thought of as spectrum of log[X (ω)]
  - Time axes of $\hat{x}[n]$ is referred to as "quefrency"
  - Filter l[n] as the "lifter".

Voiced Speech Example



Basic Speech processing steps for Frequency Parameter

Signal: **S(t) = x[n]**

Basic Signal Processing Block

Pre-emphasis

Framing

Windowing

DFT

Power Spectrum

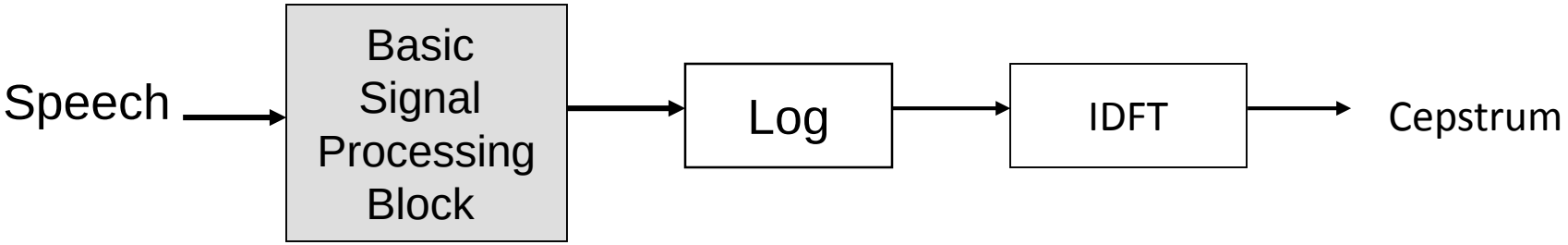
y[n] = x[n] – α * x[n-1] *(where α = 0.95)*

x(n) = x(n)*W(n)

Hamming	:	0.54 – 0.46 cos(2*pi*n/(N-1))
Hanning	:	0.5(1- cos(2*pi*n/(N-1)))
Cosine	:	sin(pi*n/(N-1))

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

Cepstral Transform Coefficients (CC)



$$\text{Cepstrum} = \text{IDFT}(\log(\text{DFT}(\text{S}(\text{n}))))$$

- Relationship of complex cepstrum $\hat{x}[n]$ to real cepstrum c[n]:
  - If *x[n]* real then:
    - |X(ω)| is real and even and thus log[|X(ω)|] is real and even
    - ∠X(ω) is odd, and hence

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

$\hat{x}[n]$ is referred to as the ***complex cepstrum***.

- Even component of the complex cepstrum, c[n] is referred to as the ***real cepstrum***.

$$c[n]=\frac{\hat{x}[n]+\hat{x}[-n]}{2}$$

$$\hat{x}[n] \leftrightarrow \hat{X}(e^{j\omega}) = \log|X(e^{j\omega})| + j\arg[X(e^{j\omega})]$$
$$c[n] \leftrightarrow \log|X(e^{j\omega})|$$

By definition we have

$$\begin{aligned} c[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(e^{j\omega})|e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(e^{j\omega})|[\cos(\omega n) + j\sin(\omega n)]d\omega \end{aligned}$$

Recall that for $x[n]$ real, $|X(e^{j\omega})|$ is an even function; therefore

$$\int_{-\pi}^{\pi} \log|X(e^{j\omega})|[j\sin(\omega n)]d\omega = 0$$

Leading to the result

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

By inverse transforming $\hat{X}(e^{j\omega})$ we obtain

$$\hat{x}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}[\log|X(e^{j\omega})| + j\arg\{X(e^{j\omega})\}]\cdot[\cos(\omega n) + j\sin(\omega n)]d\omega$$

For $x[n]$ real, $\arg\{X(e^{j\omega})\}$ is an odd function, therefore

$$\int_{-\pi}^{\pi} j\arg\{X(e^{j\omega})\}\cos(\omega n)d\omega = 0$$

Similarly, since $\log|X(e^{j\omega})|$ is an even function of ω and $\sin(\omega n)$ is an odd function of ω, therefore we have

$$\int_{-\pi}^{\pi}\log|X(e^{j\omega})|(j\sin(\omega n))d\omega = 0$$

Thus we get:

$$\begin{aligned}
\hat{x}[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi}[\log|X(e^{j\omega})|][\cos(\omega n)]d\omega - \frac{1}{2\pi}\int_{-\pi}^{\pi}\arg[X(e^{j\omega})][\sin(\omega n)]d\omega \\
\hat{x}[-n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi}[\log|X(e^{j\omega})|][\cos(\omega n)]d\omega + \frac{1}{2\pi}\int_{-\pi}^{\pi}\arg[X(e^{j\omega})][\sin(\omega n)]d\omega \\
\frac{\hat{x}[n]+\hat{x}[-n]}{2} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}[\log|X(e^{j\omega})|][\cos(\omega n)]d\omega = c[n]
\end{aligned}$$

LPC Cepstrum

The LPC vector is defined by $[a_0, a_1, a_2, \ldots a_p]$ and the CC vector is defined by $[c_0 c_1 c_2 \ldots c_p \ldots c_{n-1}$

LPC Cepstrum (c_m)

$$c_0 = \log G^2$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \le m \le p$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad m > p$$

$$G = e^{c_0/2}$$

$$a_m = c_m - \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \le m \le p$$

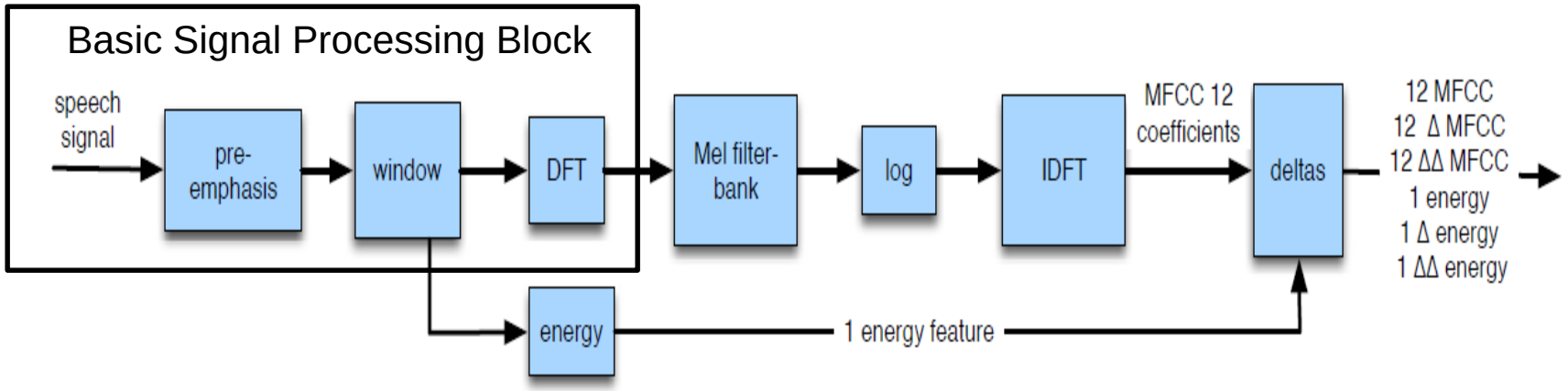
Mel Frequency Cepstral Coefficients (MFCC)

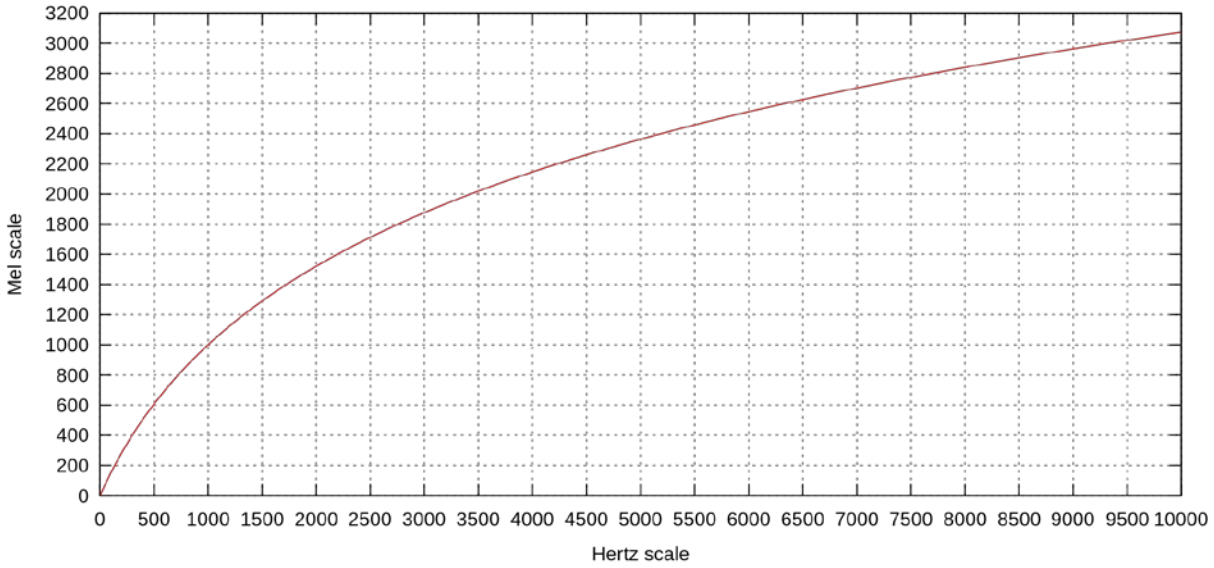
MFCC is the most used parameters in Speech Technology development.

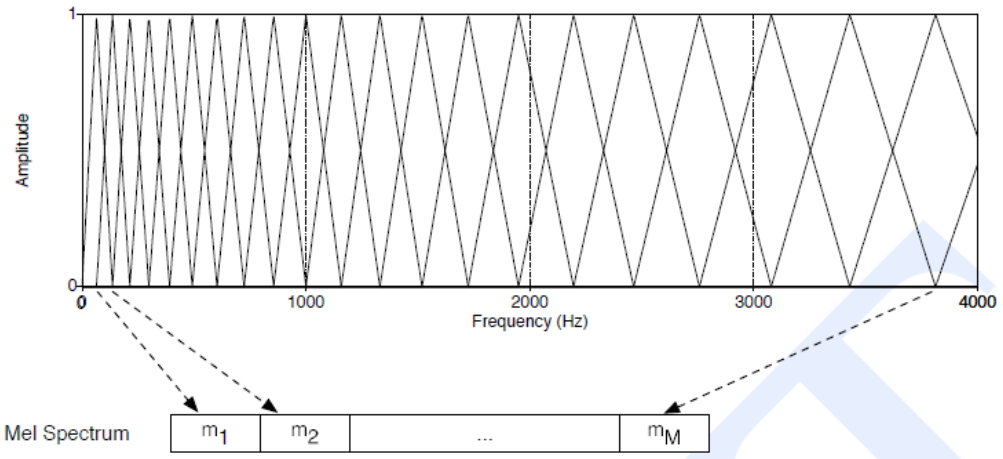
MFCC computed from the speech signal using the following three steps:

1. Compute the FFT power spectrum of the speech signal
2. Apply a Mel-space filter-bank to the power spectrum to get energies
3. Compute discrete cosine transform (DCT) of log filter-bank energies to get uncorrelated MFCC’s

Block diagram of Extracting a sequence of 39-dimensional MFCC feature vectors





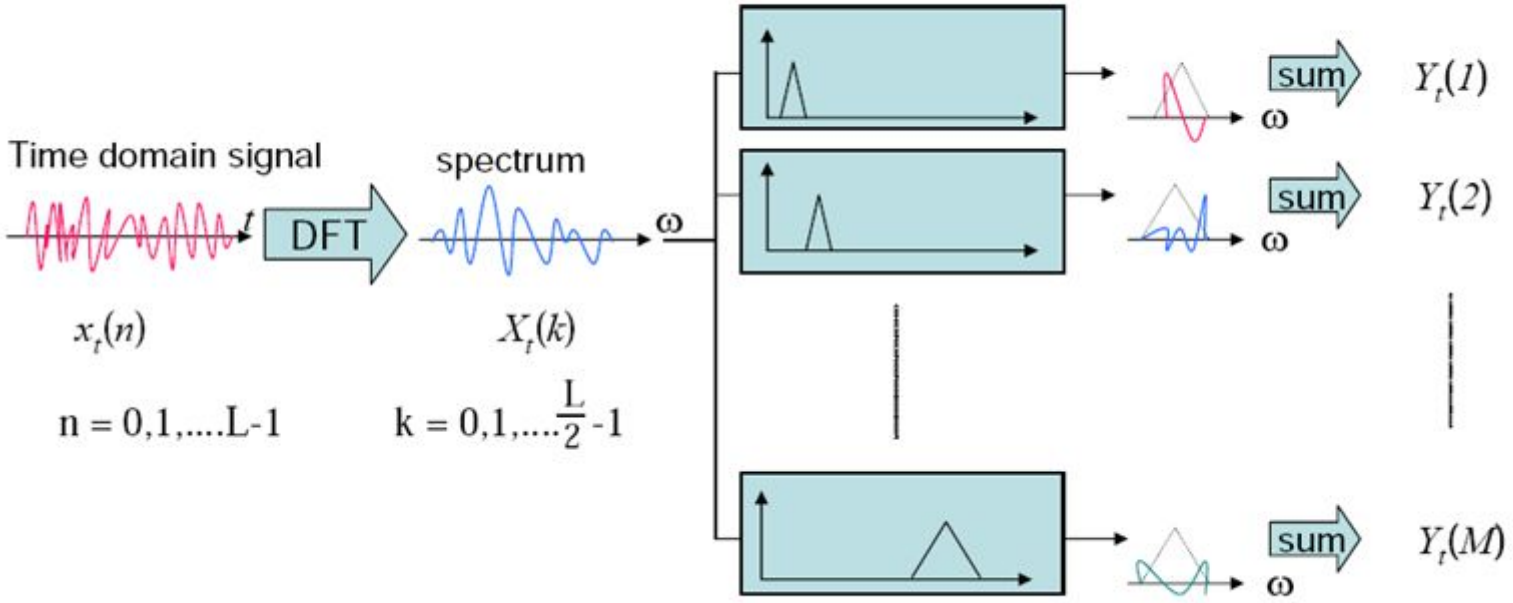


Pitch $(mels) = 3322 \log_{10}(1 + f / 1000)$

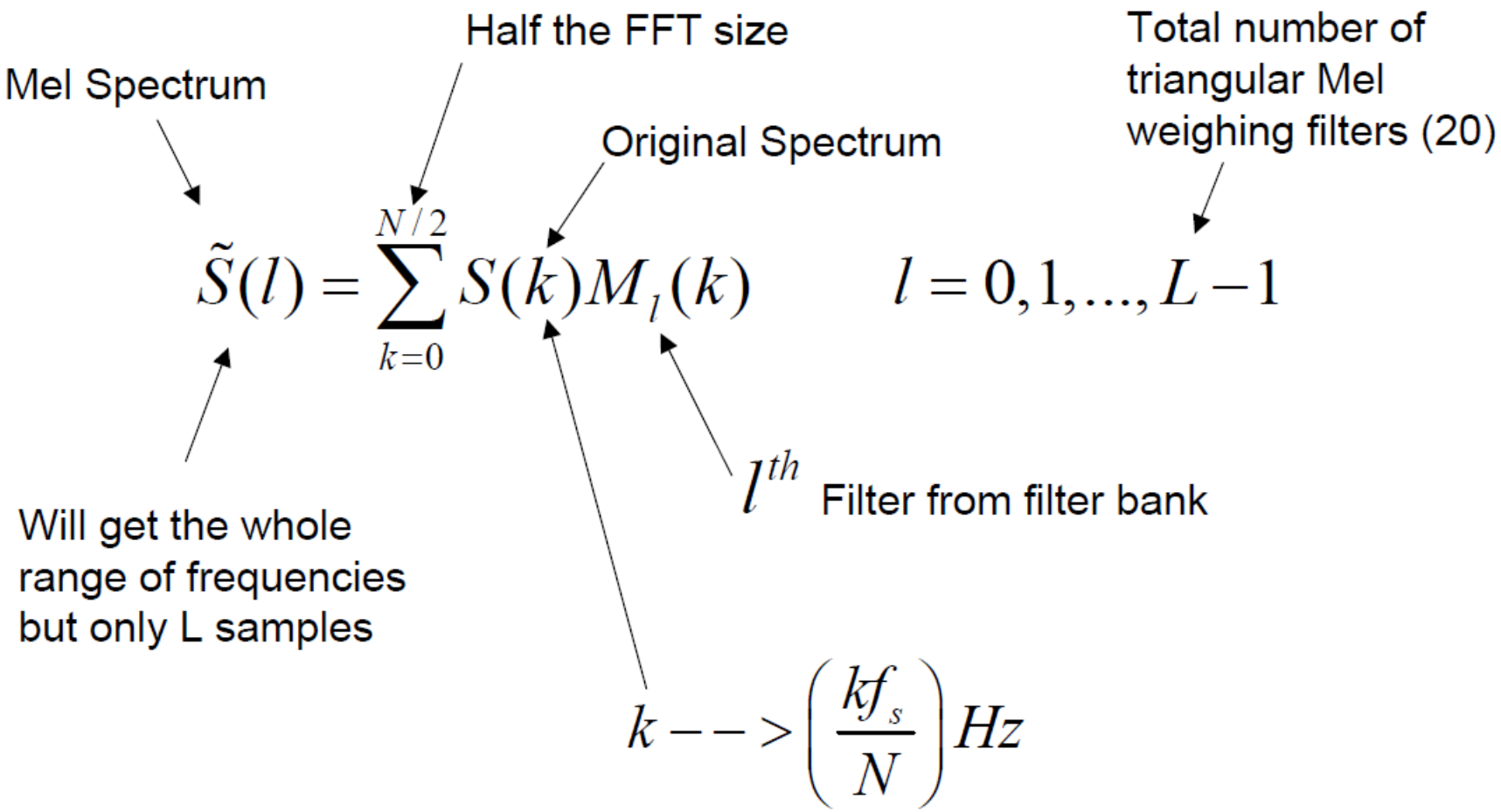
Alternatively, we can approximate curve as:

Pitch $(mels) = 1127 \log_{e}(1 + f / 700)$

Mel Filter bank



Mel Filter bank

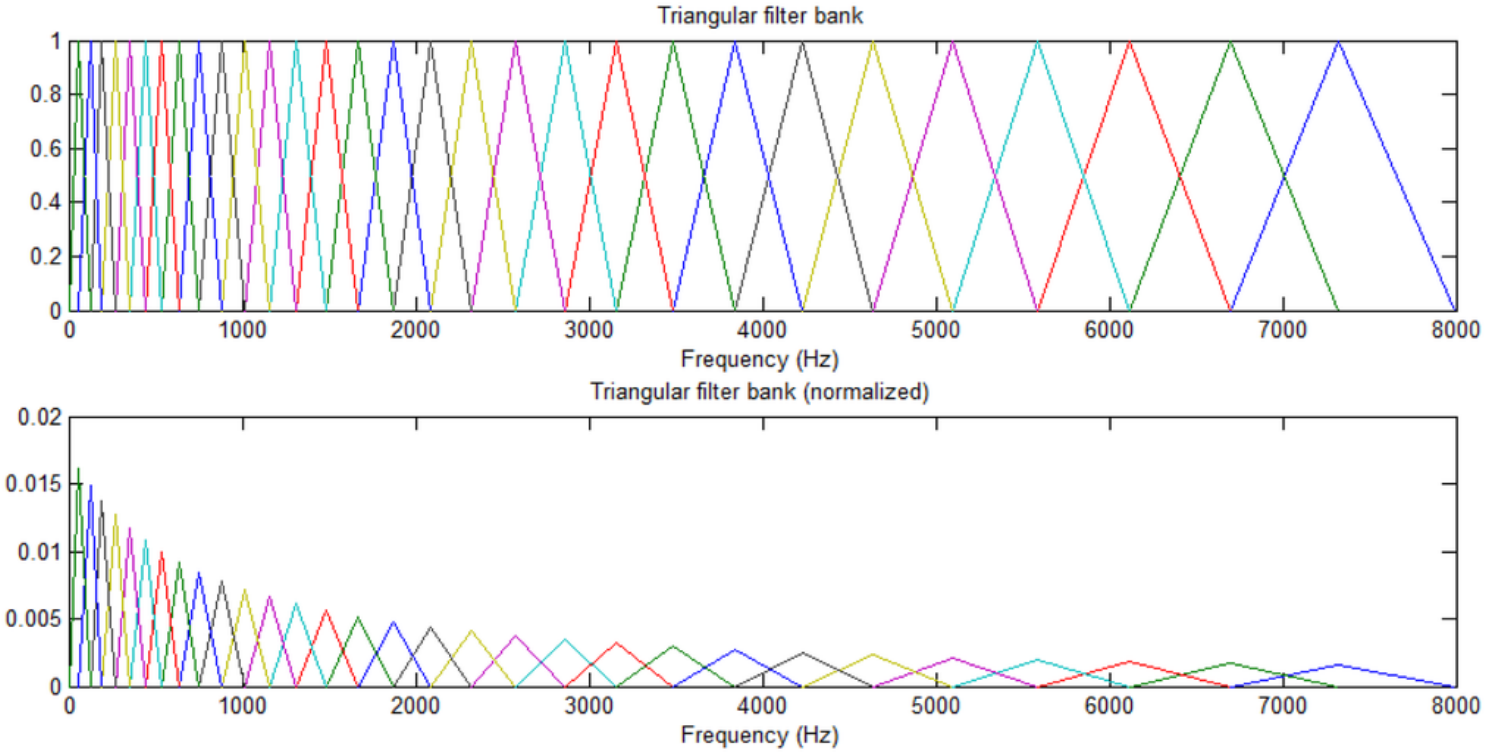


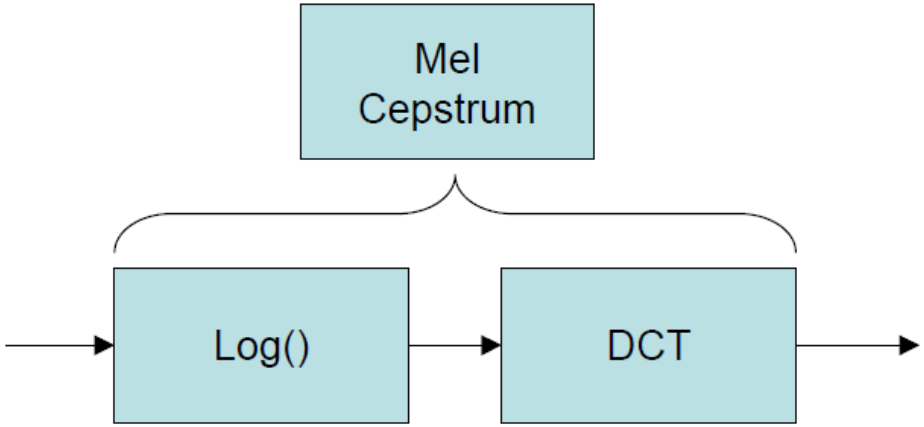
$$\hat{S}(l) = \sum_{k=L_l}^{U_l} S(k) M_l(k)$$

$M_l(k)$ the filter weighting function can be normalized

$$\hat{S}(l) = \frac{1}{M_l} \sum_{k=L_l}^{U_l} S(k) M_l(k)$$

$$M_l = \sum_{k=L_l}^{U_l} M_l(k)$$





Total number of triangular Mel weighing filters (20)

DCT Equation: $$c(i) = \sqrt{\frac{2}{L}} \sum_{m=1}^{L} \log(\tilde{S}(m)) \cos\left(\frac{\pi i}{L}(m - 0.5)\right) \qquad i = 0, 1, \ldots C - 1$$

Mel Spectrum

cepstral coefficients desired

Why the DCT?

- The signal is real with mirror symmetry
- The IFFT requires complex arithmetic
- The DCT does NOT
- The DCT implements the same function as the FFT more efficiently by taking advantage of the redundancy in a real signal.
- The DCT is more efficient computationally

Delta Cepstrum

- The set of mel frequency cepstral coefficients provide perceptually meaningful and smooth estimates of speech spectra, over time
- Since speech is inherently a dynamic signal, it is reasonable to seek a representation that includes some aspect of the dynamic nature of the time derivatives (both first and second order derivatives) of the short-term cepstrum
- The resulting parameter sets are called the delta cepstrum (first derivative) and the delta-delta cepstrum (second derivative).
- The simplest method of computing delta cepstrum parameters is a first difference of cepstral vectors, of the form:

$$\Delta\text{mfcc}_m[n] = \text{mfcc}_m[n] - \text{mfcc}_{m-1}[n]$$

- The simple difference is a poor approximation to the first derivative and is not generally used. Instead a least-squares approximation to the local slope (over a region around the current sample) is used, and is of the form:

$$d_i = \frac{\sum_{n=1}^{N} n\,(c_{n+i} - c_{n-i})}{2\sum_{n=1}^{N} n^2}$$

Perceptual Linear Prediction

- PLP parameters are the coefficients that result from standard all-pole modeling or linear predictive analysis, of a specially modified, short-term speech spectrum.
- In PLP the speech spectrum is modified by a set of transformations that are based on models of the human auditory system
- The spectral resolution of human hearing is roughly linear up to 800 or 1000Hz, but it decreases with increasing frequency above this linear range

Perceptually motivated analyses

- **Critical-band spectral resolution**: PLP incorporates critical-band spectral-resolution into its spectrum estimate by remapping the frequency axis to the Bark scale and integrating the energy in the critical bands to produce a critical-band spectrum approximation.
- **Equal-loudness pre-emphasis**: At conversational speech levels, human hearing is more sensitive to the middle frequency range of the audible spectrum. PLP incorporates the effect of this phenomenon by multiplying the critical-band spectrum by an equal loudness curve that suppresses both the low- and high-frequency regions relative to the midrange from 400 to 1200 Hz.
- **Intensity-loudness power law:** There is a nonlinear relationship between the intensity of sound and the perceived loudness. PLP approximates the power-law of hearing by using a cube-root amplitude compression of the loudness-equalized critical band spectrum estimate.

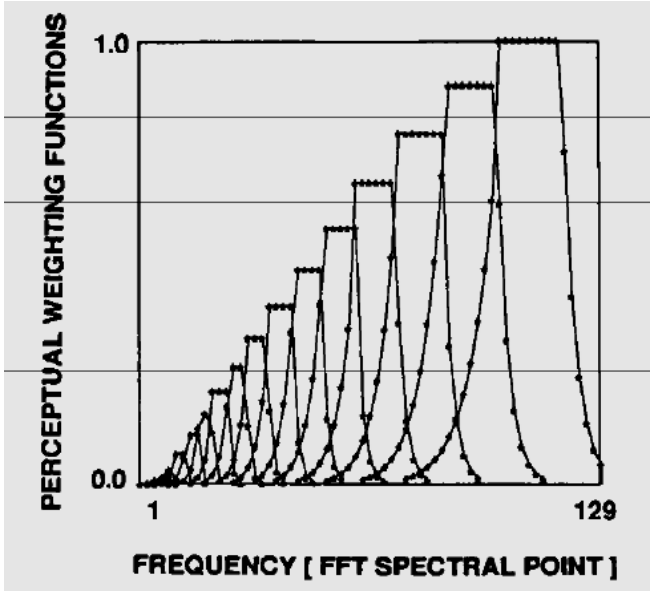
Perceptual LPC

(Hermansky, *J. Acoust. Soc. Am.*, 1990)

- First, warp the spectrum to a Bark scale:

$$\tilde{S}(b) = \sum_{k=0}^{N-1} |H_b(k)|^2 |X(k)|^2, \quad b = 1, \ldots, K$$

- The filters, $H_b(k)$, are uniformly spaced in Bark frequency. Their amplitudes are scaled by the equal-loudness contour (an estimate of how loud each frequency sounds):



Perceptual LPC

- Second, compute the cube-root of the power spectrum
  - Cube root replaces the logarithm that would be used in MFCC
  - Loudness of a tone is proportional to cube root of its power

$$Y(b) = S(b)^{0.33}$$

- Third, inverse Fourier transform to find the "Perceptual Autocorrelation:"

$$\begin{aligned}\tilde{R}(m) &= \frac{1}{2K}\sum_{b=0}^{2K} Y(b) e^{\frac{j2\pi bm}{2K}} \\ &= \frac{1}{K}\sum_{b=1}^{K} Y(b)\cos\left(\frac{\pi bm}{K}\right) + \frac{(-1)^m}{2K} Y(K)\end{aligned}$$

Perceptual LPC

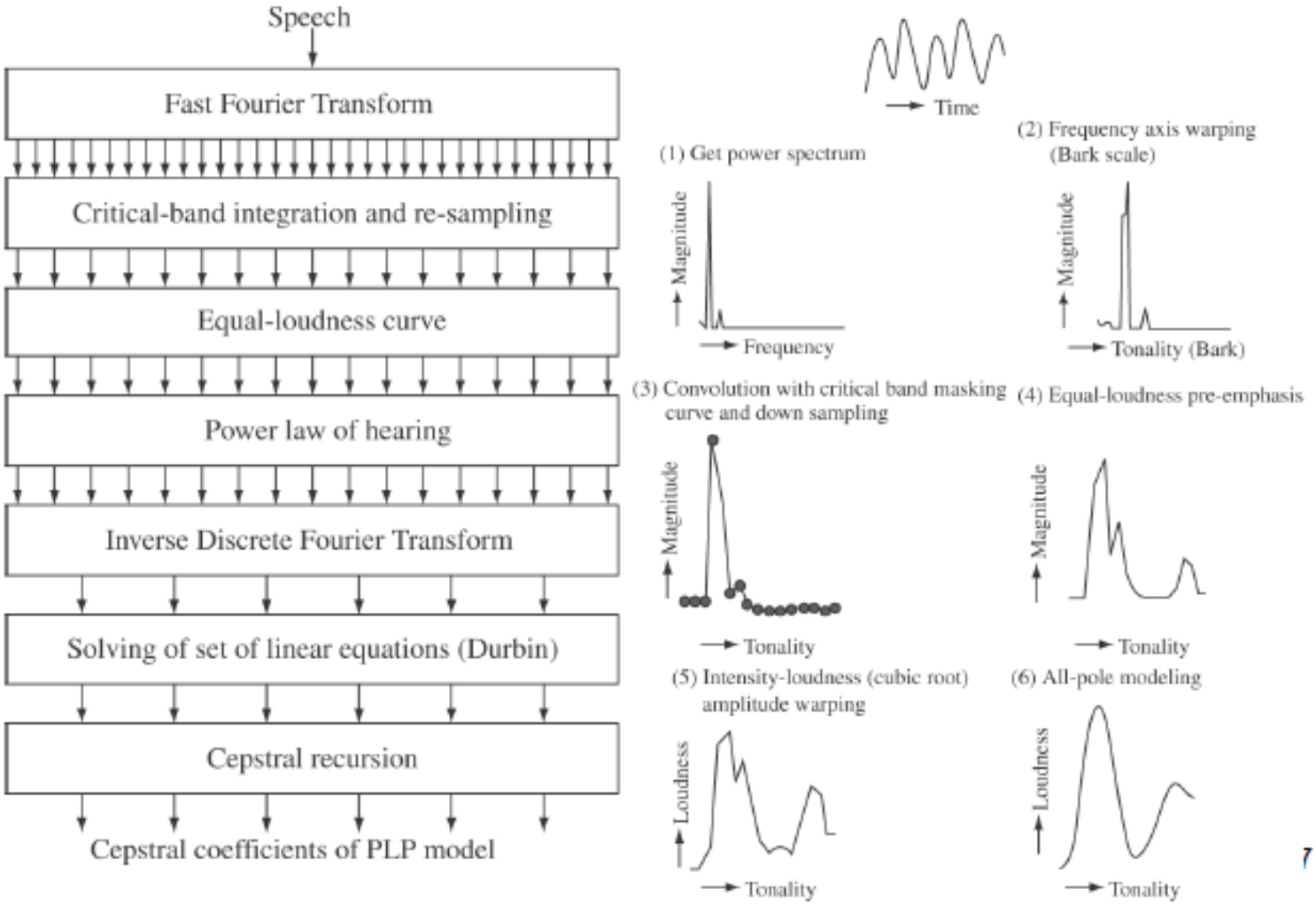
- Fourth, use Normal Equations to find the Perceptual LPC (PLP) coefficients:

$$\tilde{R}(m) = \sum_{k=1}^{p} \tilde{a}_k \tilde{R}(|m-k|)$$

- Fifth, use the LPC Cepstral recursion to find Perceptual LPC Cepstrum (PLPCC):

$$\tilde{c}(m) = \tilde{a}_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) \tilde{c}(k)\tilde{a}_{m-k}, \quad 1 \leq m \leq p$$

Perceptual Linear Prediction



RASTA(**R**el**A**tive **S**pec**T**r**A**)

- The rate of change of nonlinguistic components of speech and background noise environments often lies outside the typical rate-of-change of vocal-tract shapes in conversational speech
- Hearing is relatively insensitive to slowly varying stimuli
- The basic idea of RASTA filtering is to exploit these phenomena by suppressing constant and slowly varying elements in each spectral component of the short term auditory-like spectrum prior to computation of the linear prediction coefficients

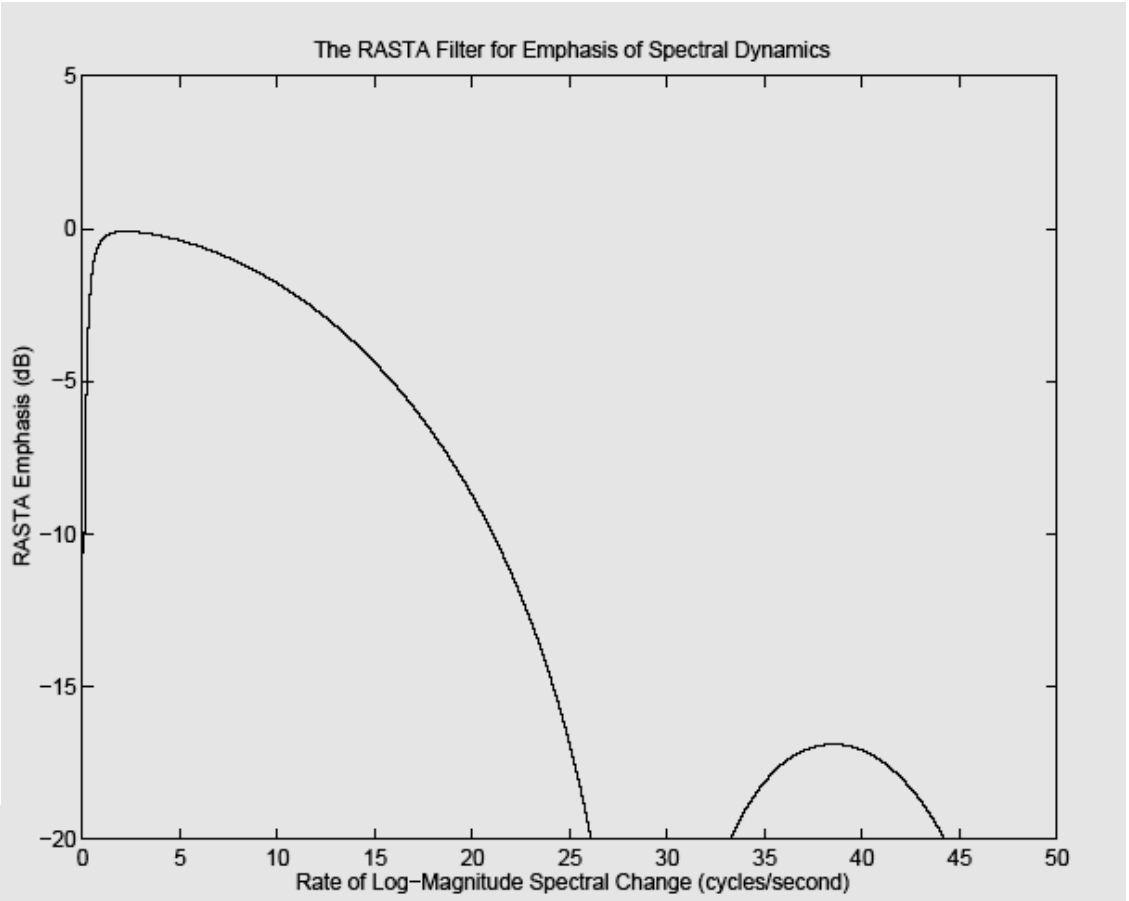
RASTA (RelAtive SpecTral Amplitude)

(Hermansky, *IEEE Trans. Speech and Audio Proc.*, 1994)

- Modulation-filtering of the cepstrum is equivalent to modulation-filtering of the log spectrum:

  $c_t^*[m] = \Sigma_k\, h_k\, c_{t-k}[m]$

- RASTA is a particular kind of modulation filter:



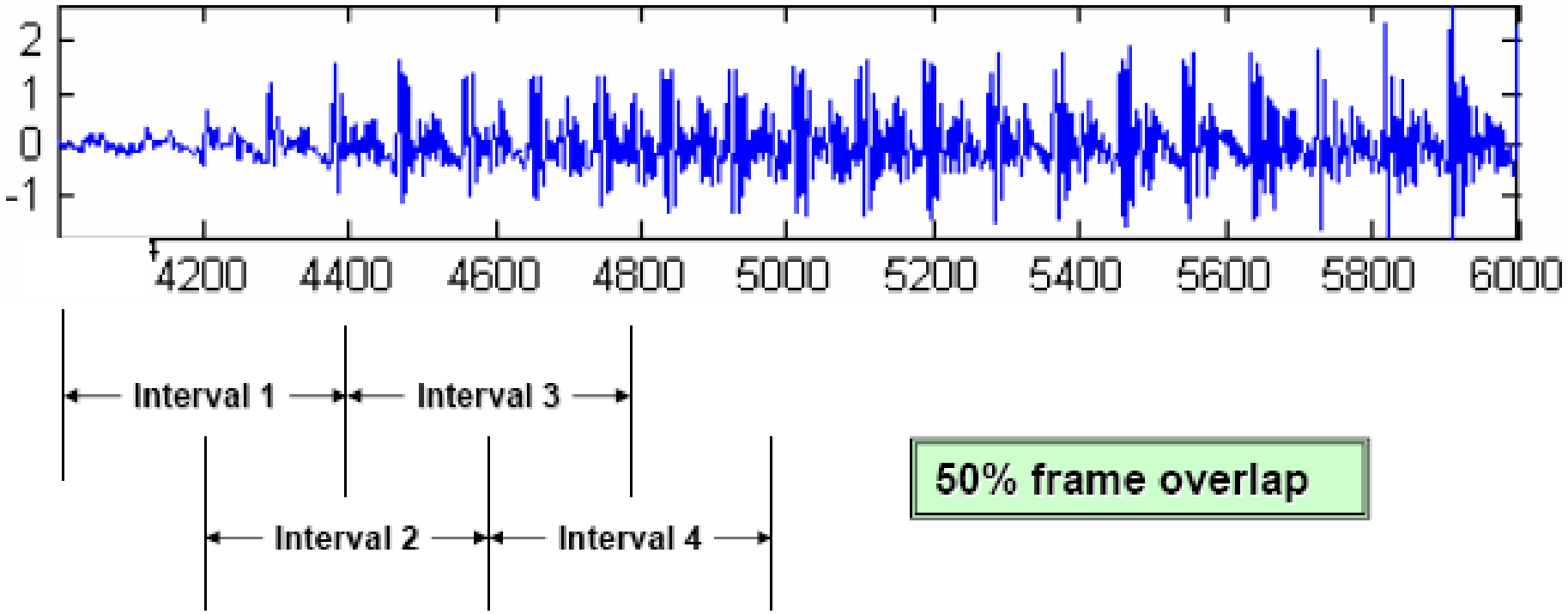
$$H(z) = \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{10z^{-2}(1 - 0.98z^{-1})}$$

Time Domain Methods in Speech Processing

Fundemental Assumptions

- Properties of Speech Signal change relatively slowly with time (5-10 sounds per second)
- Uncertainty in short/Long time measurements and estimates
  - Over very short (5-20ms) intervals
    - Uncertainty due to small amount of data, varying pitch and amplitude
  - Over medium Length intervals (20-100ms)
    - Uncertainty due to changes in sound quality, transition between sounds, rapid transients in speech
  - Overlong Intervals (100-500ms)
    - Uncertainty due to large amount of sound changes

Frame-by-Frame Processing



50% frame overlap

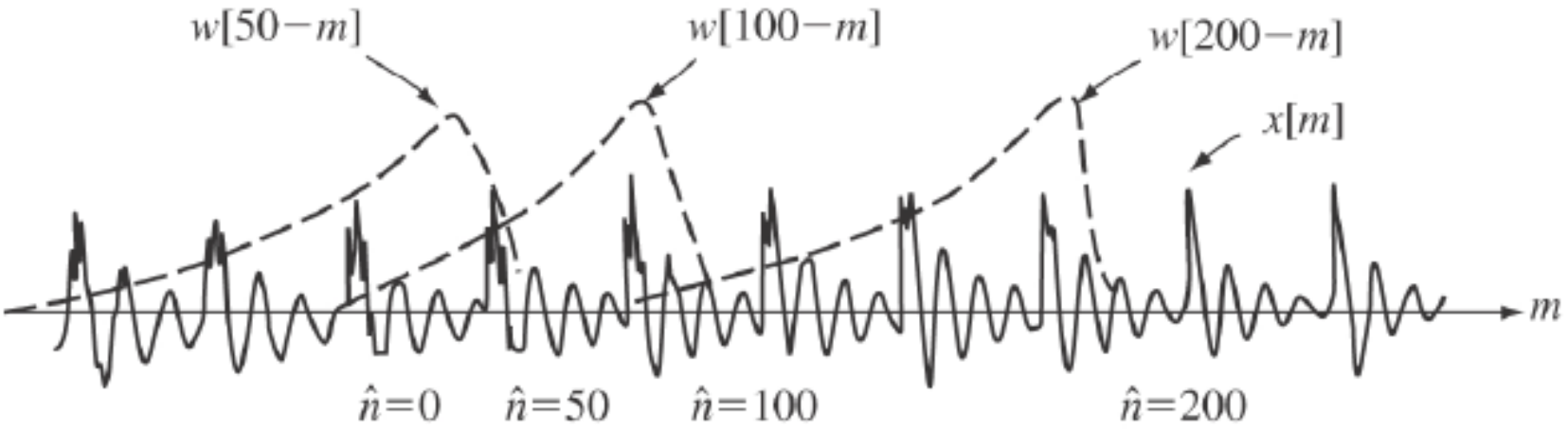
- speech is processed frame-by-frame in overlapping intervals until entire region of speech is covered by at least one such frame
- results of analysis of individual frames used to drive model parameters in some manner

Definition of STFT

$$X_{\hat{n}}(e^{j\hat{\omega}}) = \sum_{m=-\infty}^{\infty} x(m)w(\hat{n}-m)e^{-j\hat{\omega}m}$$

both $\hat{n}$ and $\hat{\omega}$ are variables

- $w(\hat{n}-m)$ is a real window which determines the portion of $x(\hat{n})$ that is used in the computation of $X_{\hat{n}}(e^{j\hat{\omega}})$



Time-domain processing

- **Time-domain parameters**
  - Short-time energy
  - Short-time average magnitude
  - Short-time zero crossing rate
  - Short-time autocorrelation
  - Short-time average magnitude difference

Short-Time Energy

$$E = \sum_{m=-\infty}^{\infty} x^2[m]$$

-- this is the long term definition of signal energy

-- there is little or no utility of this definition for time-varying signals

$$E_{\hat{n}} = \sum_{m=\hat{n}-N+1}^{\hat{n}} x^2[m] = x^2[\hat{n}-N+1] + \ldots + x^2[\hat{n}]$$

-- short-time energy in vicinity of time $\hat{n}$

$$T(x) = x^2$$

$$\tilde{w}[n] = 1 \qquad 0 \le n \le N-1$$

$$= 0 \qquad \text{otherwise}$$

Computation of Short-Time Energy

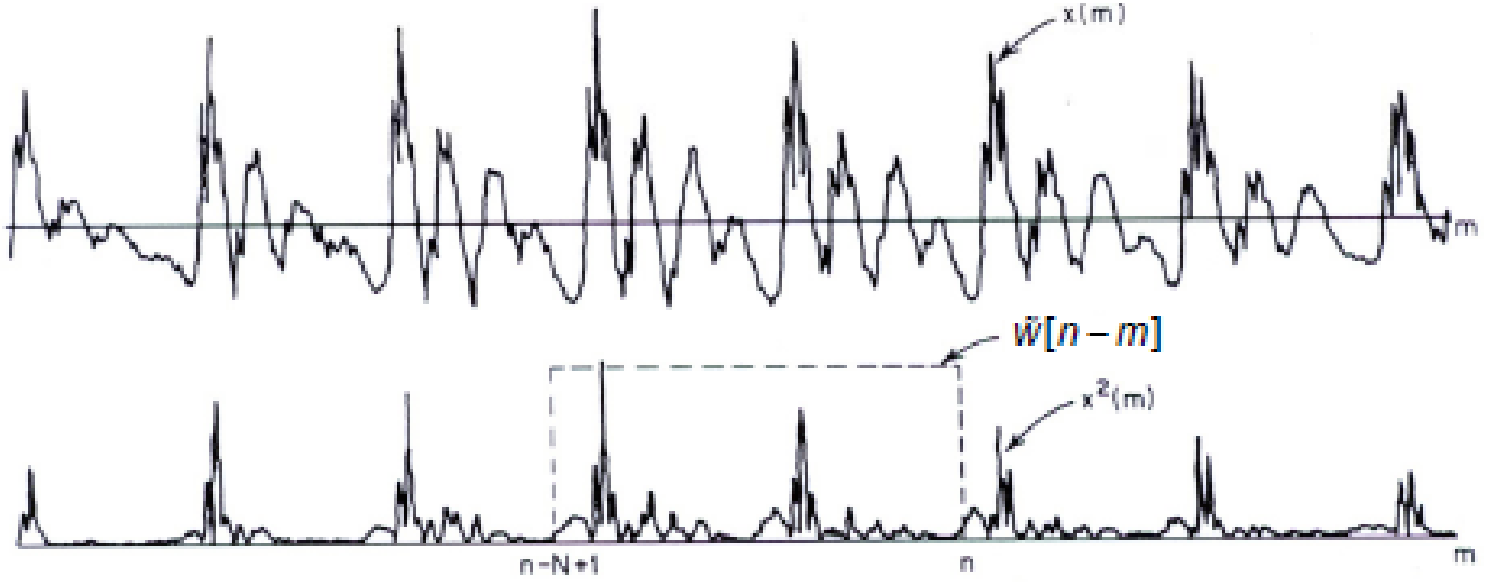


Fig. 4.2 Illustration of the computation of short-time energy.

- window jumps/slides across sequence of squared values, selecting interval for processing
- what happens to E_n as sequence jumps by 2,4,8,...,L samples ($E_{\hat{n}}$ is a lowpass function—so it can be decimated without lost of information; why is E_n lowpass?)
- effects of decimation depend on L; if L is small, then $E_{\hat{n}}$ is a lot more variable than if L is large (window bandwidth changes with L!)

Short-Time Energy Properties

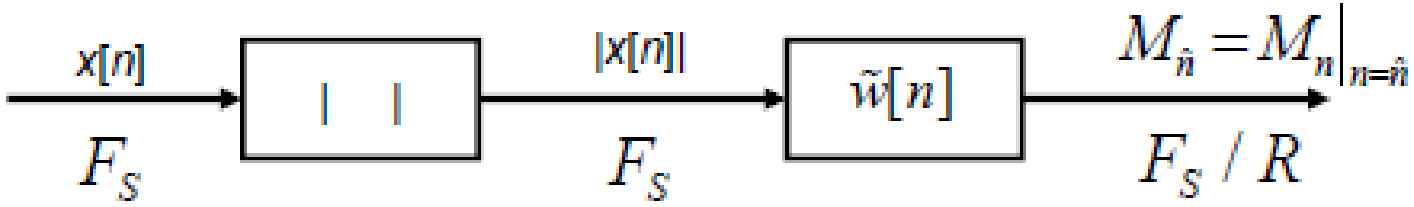
- depends on choice of $h[n]$, or equivalently, window $\tilde{w}[n]$
  - if $w[n]$ duration very long and constant amplitude ($\tilde{w}[n]=1,\ n=0,1,\ldots,L-1$), E_n would not change much over time, and would not reflect the short-time amplitudes of the sounds of the speech
  - very long duration windows correspond to **narrowband lowpass filters**
  - want E_n to change at a rate comparable to the changing sounds of the speech => this is the essential ***conflict*** in all speech processing, namely we need short duration window to be responsive to rapid sound changes, but short windows will not provide sufficient averaging to give smooth and reliable energy function

Short-Time Magnitude

- short-time energy is very sensitive to large signal levels due to $x^2[n]$ terms
  - consider a new definition of 'pseudo-energy' based on average signal magnitude (rather than energy)

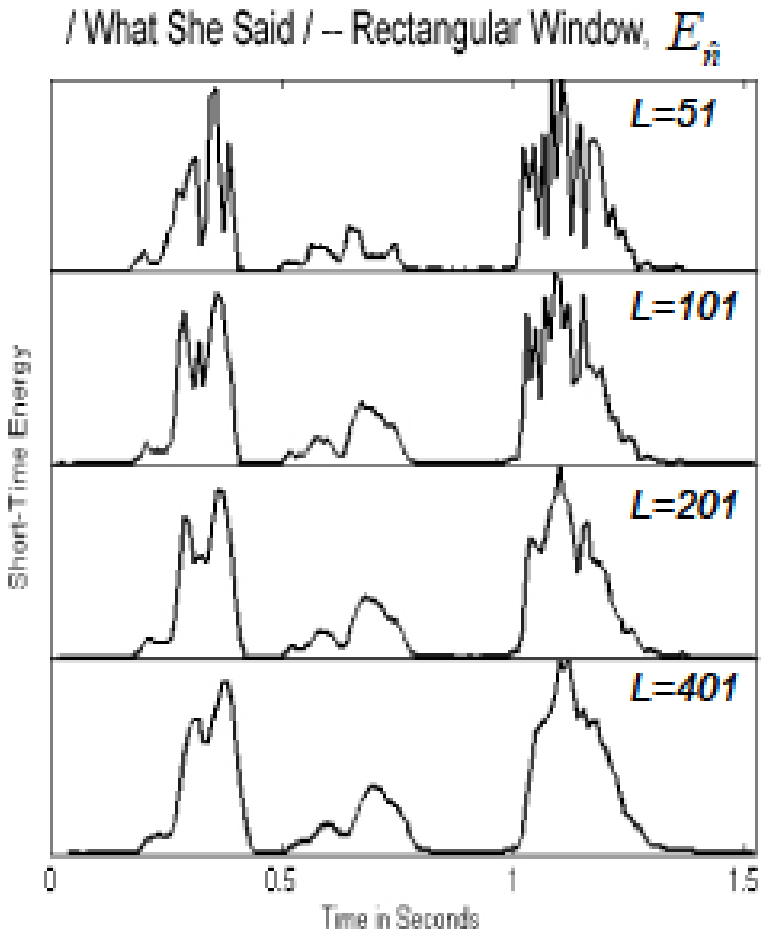
$$M_{\hat{n}} = \sum_{m=-\infty}^{\infty} |x[m]|\, \tilde{w}[\hat{n}-m]$$

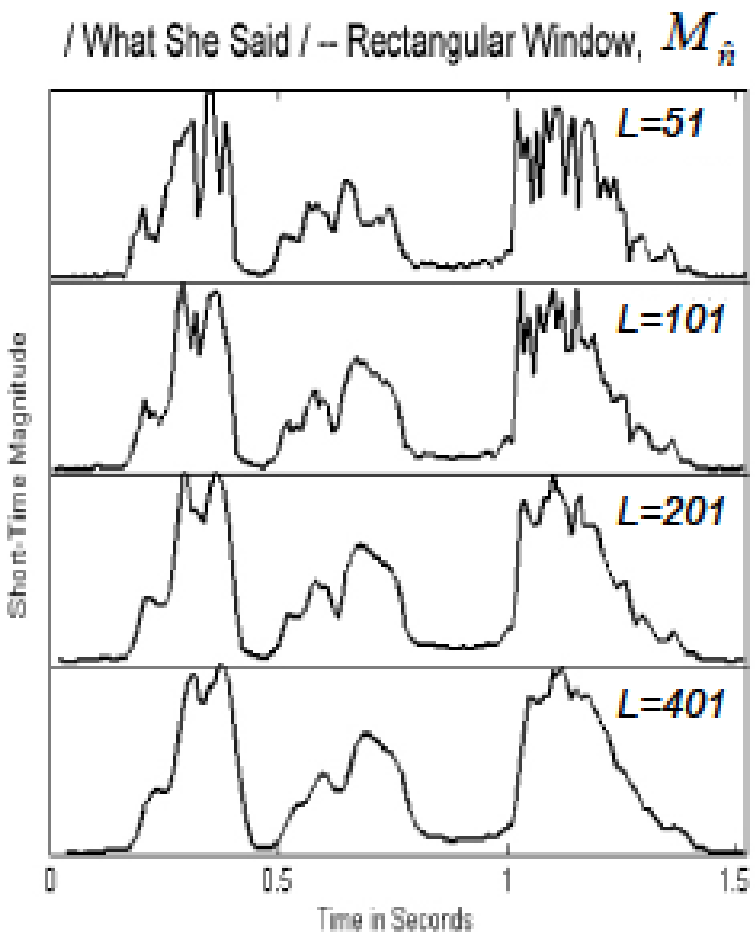
  - weighted sum of magnitudes, rather than weighted sum of squares



- computation avoids multiplications of signal with itself (the squared term)

Short Time Energy and Magnitude—Rectangular Window

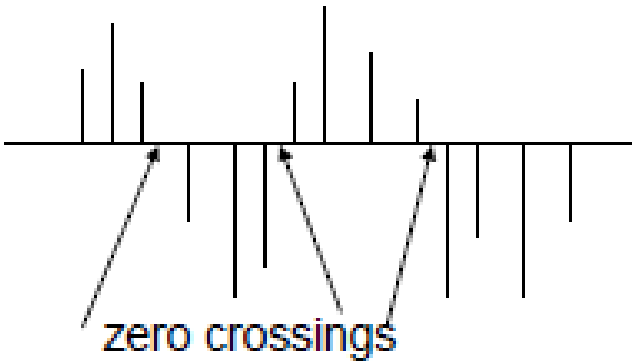




Zero Crossing

- Number of times unvoiced speech crosses the zero line is significantly higher than that of voiced speech.
- Gender of speaker can also have an effect on zero crossing.
- Small pitch weighting can be used to weight the decision threshold.

Short-Time Average ZC Rate



zero crossing => successive samples have different algebraic signs

- zero crossing rate is a simple measure of the 'frequency content' of a signal—especially true for narrowband signals (e.g., sinusoids)
- sinusoid at frequency F_0 with sampling rate F_S has F_S/F_0 samples per cycle with two zero crossings per cycle, giving an average zero crossing rate of

z_1=(2) crossings/cycle x (F_0 / F_S) cycles/sample

$z_1 = 2F_0 / F_S$ crossings/sample (i.e., z_1 proportional to F_0)

$z_M = M\ (2F_0 / F_S)$ crossings/(M samples)

Sinusoid Zero Crossing Rates

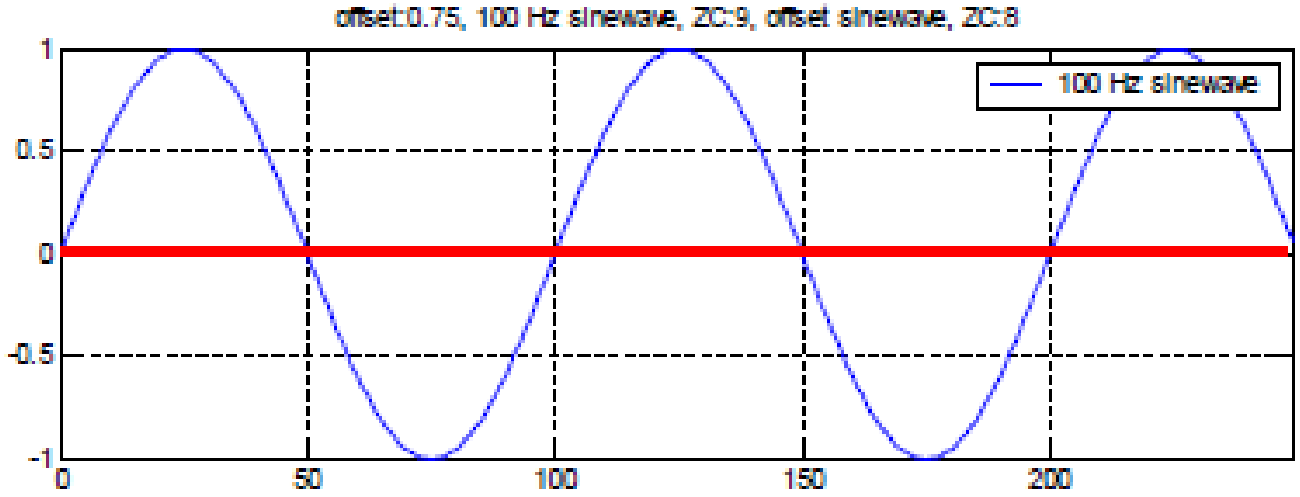
Assume the sampling rate is $F_S = 10{,}000$ Hz

1. $F_0 = 100$ Hz sinusoid has $F_S / F_0 = 10{,}000 / 100 = 100$ samples/cycle;
   or $z_1 = 2/100$ crossings/sample, or $z_{100} = 2/100 * 100 =$
   2 crossings/10 msec interval

2. $F_0 = 1000$ Hz sinusoid has $F_S / F_0 = 10{,}000 / 1000 = 10$ samples/cycle;
   or $z_1 = 2/10$ crossings/sample, or $z_{100} = 2/10 * 100 =$
   20 crossings/10 msec interval

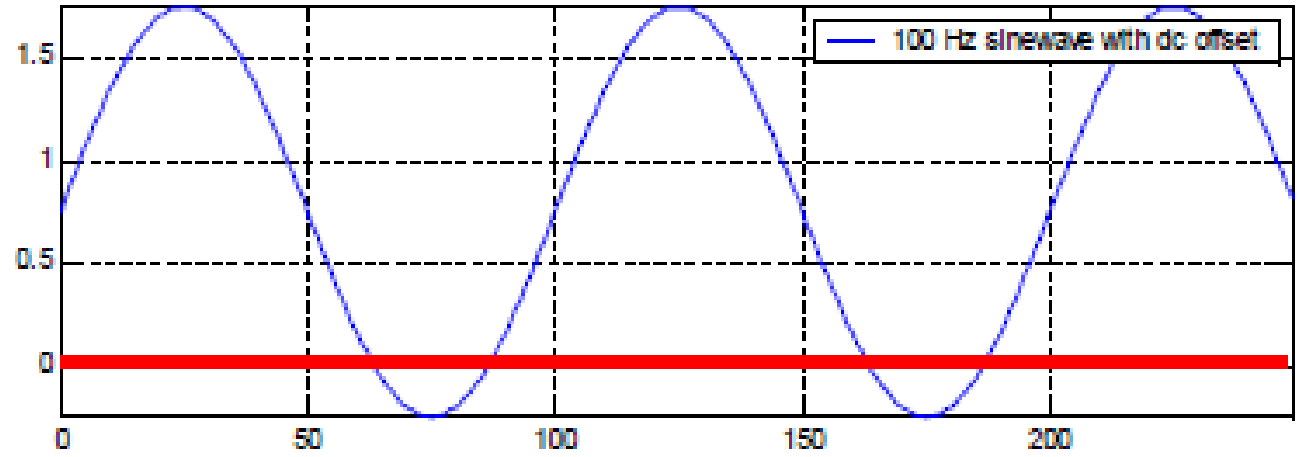
3. $F_0 = 5000$ Hz sinusoid has $F_S / F_0 = 10{,}000 / 5000 = 2$ samples/cycle;
   or $z_1 = 2/2$ crossings/sample, or $z_{100} = 2/2 * 100 =$
   100 crossings/10 msec interval

Zero Crossing for Sinusoids



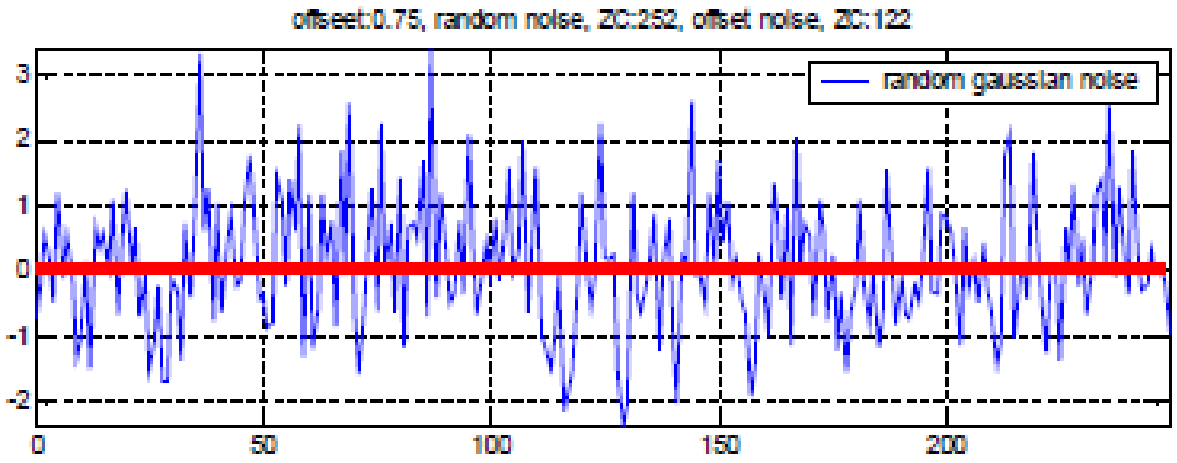
ZC=9

Offset=0.75



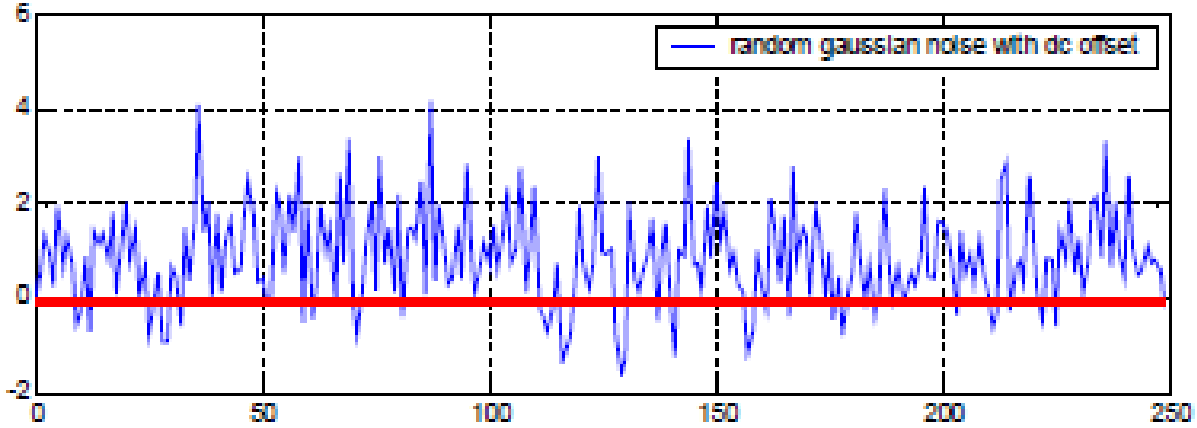
ZC=8

Zero Crossings for Noise



ZC=252

Offset=0.75



ZC=122

ZC Rate Definitions

$$Z_{\hat{n}} = \frac{1}{2L_{\text{eff}}} \sum_{m=\hat{n}-L+1}^{\hat{n}} |\,\text{sgn}(x[m]) - \text{sgn}(x[m-1])\,|\,\tilde{w}[\hat{n}-m]$$

$$\text{sgn}(x[n]) = 1 \qquad x[n] \geq 0$$

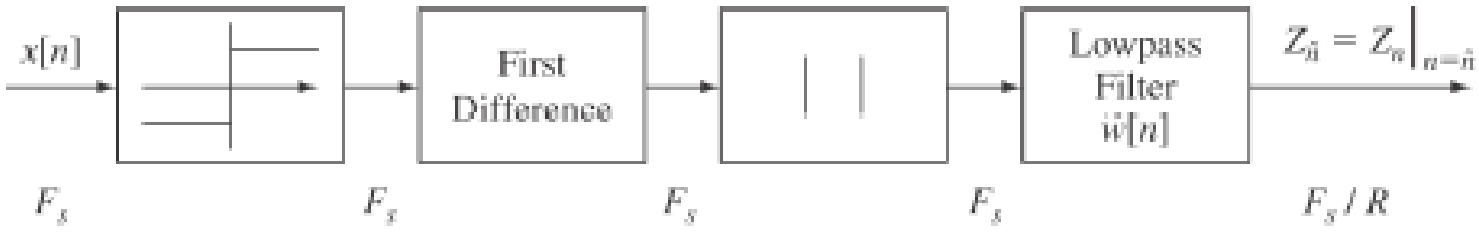
$$= -1 \qquad x[n] < 0$$

- simple rectangular window:

$$\tilde{w}[n] = 1 \qquad 0 \leq n \leq L-1$$

$$= 0 \qquad \text{otherwise}$$

$$L_{\text{eff}} = L$$



Same form for $Z_{\hat{n}}$ as for $E_{\hat{n}}$ or $M_{\hat{n}}$

ZC Normalization

- The formal definition of z_n is:

$$Z_{\hat{n}} = z_1 = \frac{1}{2L} \sum_{m=\hat{n}-L+1}^{\hat{n}} |\operatorname{sgn}(x[m]) - \operatorname{sgn}(x[m-1])|$$

  is interpreted as the number of zero crossings per sample.
- For most practical applications, we need the rate of zero crossings per fixed interval of M samples, which is

$$z_M = z_1 \cdot M = \text{rate of zero crossings per } M \text{ sample interval}$$

  Thus, for an interval of τ sec., corresponding to M samples we get

$$z_M = z_1 \cdot M; \quad M = \tau F_S = \tau / T$$

ZC Normalization

- For a 1000 Hz sinewave as input, using a 40 msec window length (L), with various values of sampling rate (F_S), we get the following:

F_S	L	z_1	M	z_M
8000	320	1 / 4	80	20
10000	400	1 / 5	100	20
16000	640	1 / 8	160	20

- Thus we see that the normalized (per interval) zero crossing rate, z_M, is independent of the sampling rate and can be used as a measure of the dominant energy in a band.

Autocorrelation Technique

- Autocorrelation is a cross-correlation of a signal with itself.

$$\phi(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n+\tau)$$

  - The maximum of similarity occurs for time shifting of zero.
  - An other maximum should occur in theory when the time-shifting of the signal corresponds to the fundamental period.

Autocorrelation function

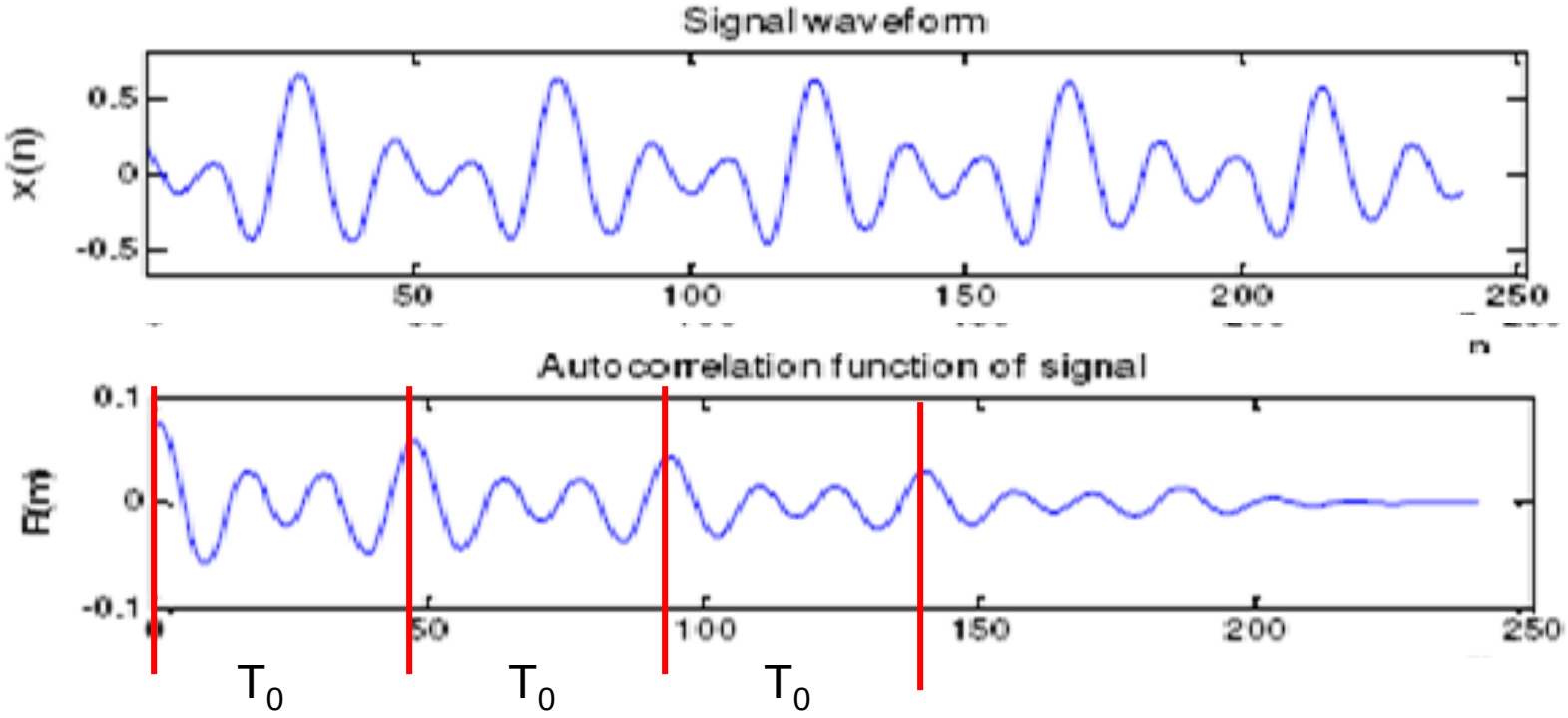
By definition, auto - correlation is

$$R[k] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n] \cdot x[n+k], \;\; 0 \le k \le K_0$$

Pr*oparties of* Autocorrelations is

1. R[k] = R[-k]
2. R[k]is maximum at k = 0

$$R[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n] \cdot x[n+k], \;\; 0 \le k \le K_0$$



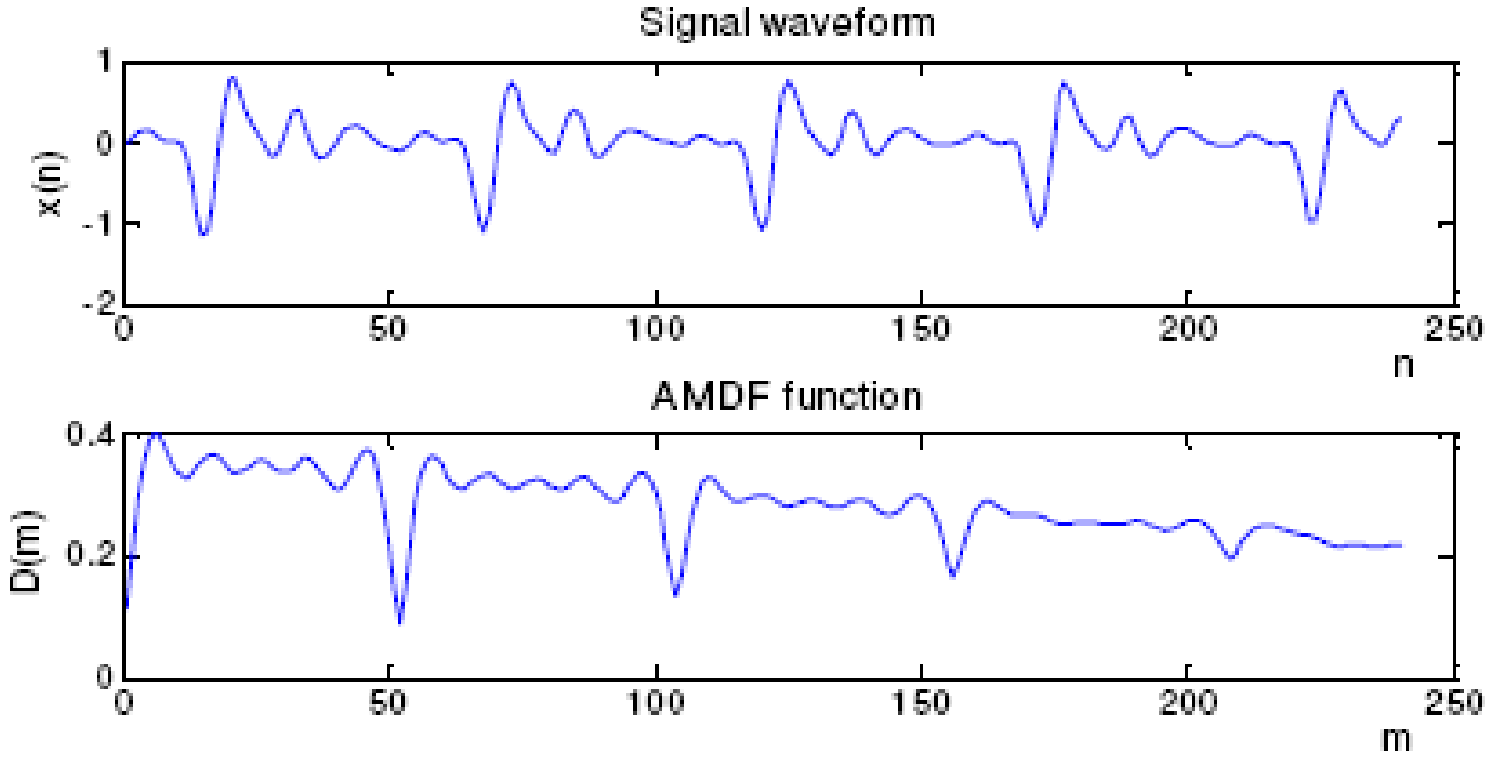
When a segment of a signal is correlated with itself, the distance (*Lag_time_in_samples*) between the positions of the maximum and the second maximum is defined as the *fundamental period* (pitch) of the signal.

Average Magnitude Difference Function(AMDF)

- It is an alternate to Autocorrelation function.
- It compute the difference between the signal and a time-shifted version of itself.

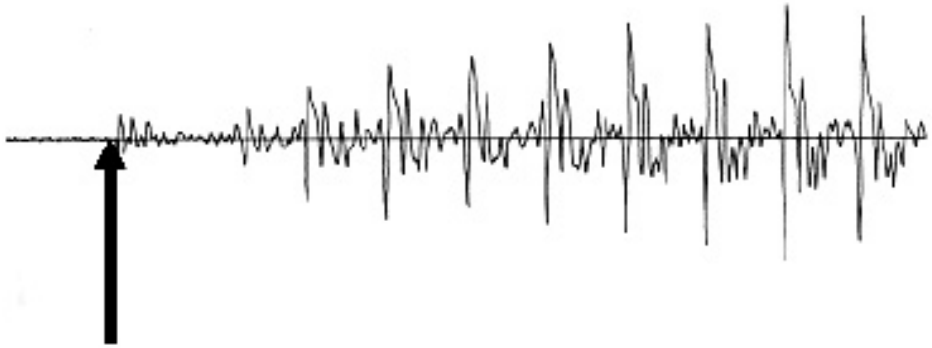
$$D_x[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} \left| x(k) - x(n+k) \right|, \qquad 0 \le k \le K_0$$

- While autocorrelation have peaks at maximum similarity, there will be valleys in the average magnitude difference function.


Signal waveform
x(n)
1
0
-1
-2
0
50
100
150
200
250
n
AMDF function
D(m)
0.4
0.2
0
0
50
100
150
200
250
m

Speech/Non-speech Detection

Ideal Speech/Non-Speech Detection

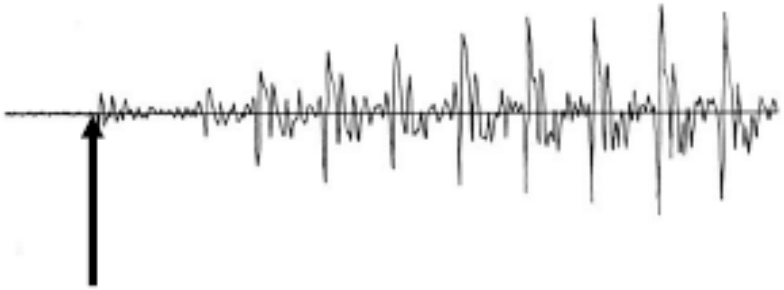


Beginning of
speech interval

Ending of speech
interval

Speech Detection Issues

- key problem in speech processing is locating accurately the beginning and end of a speech utterance in noise/background signal



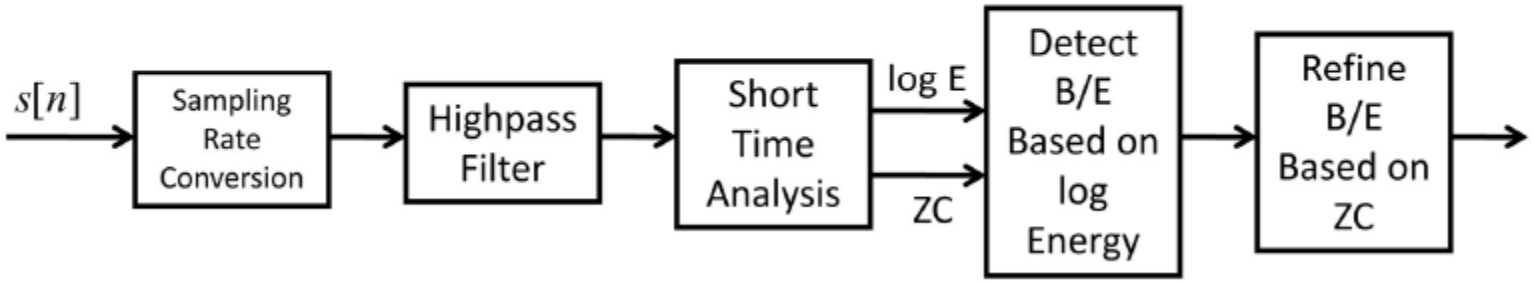
- need endpoint detection to enable:
  - computation reduction (don't have to process background signal)
  - better recognition performance (can't mistake background for speech)
- non-trivial problem except for high SNR recordings

Problems for Reliable Speech Detection

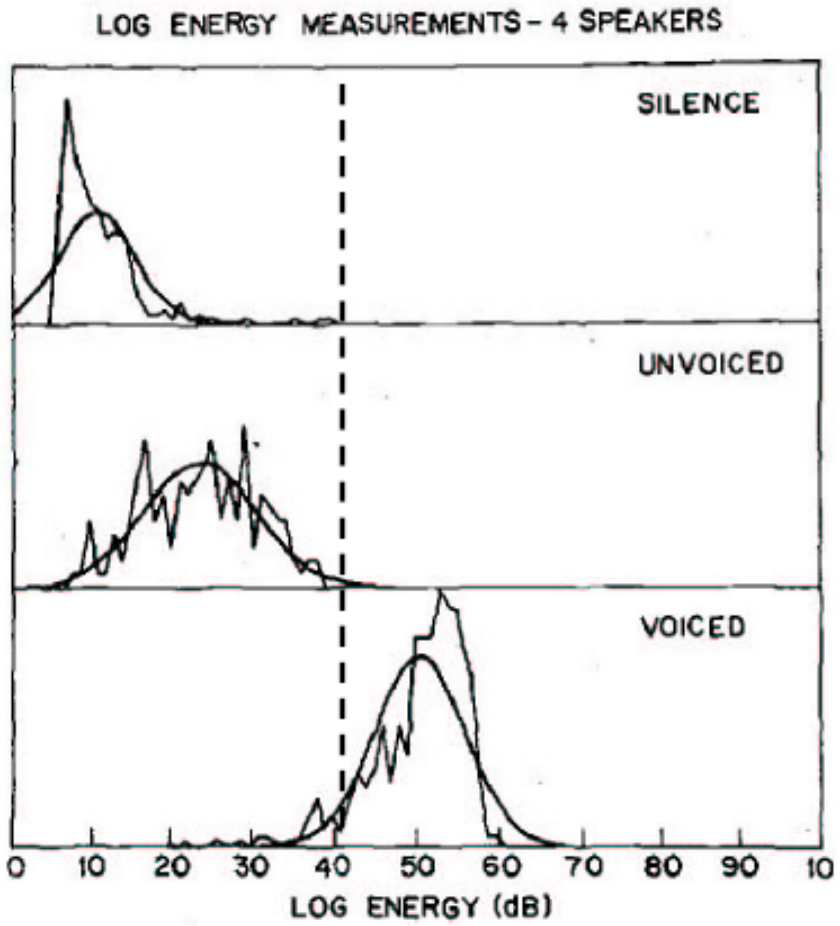
- weak fricatives (/f/, /th/, /h/) at beginning or end of utterance
- weak plosive bursts for /p/, /t/, or /k/
- nasals at end of utterance (often devoiced and reduced levels)
- voiced fricatives which become devoiced at end of utterance
- trailing off of vowel sounds at end of utterance

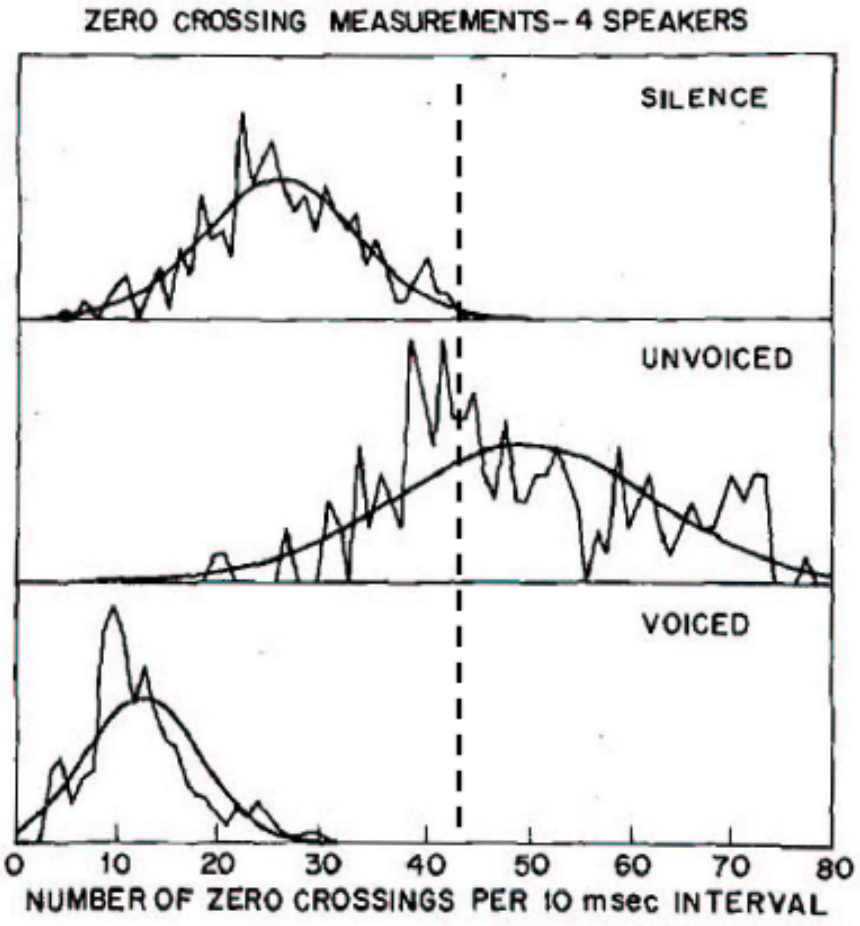
the good news is that highly reliable endpoint detection is not required for most practical applications; also we will see how some applications can process background signal/silence in the same way that speech is processed, so endpoint detection becomes a moot issue

Speech/Non-Speech Detection



Speech/Non-Speech Detection





Log energy separates Voiced from Unvoiced and Silence

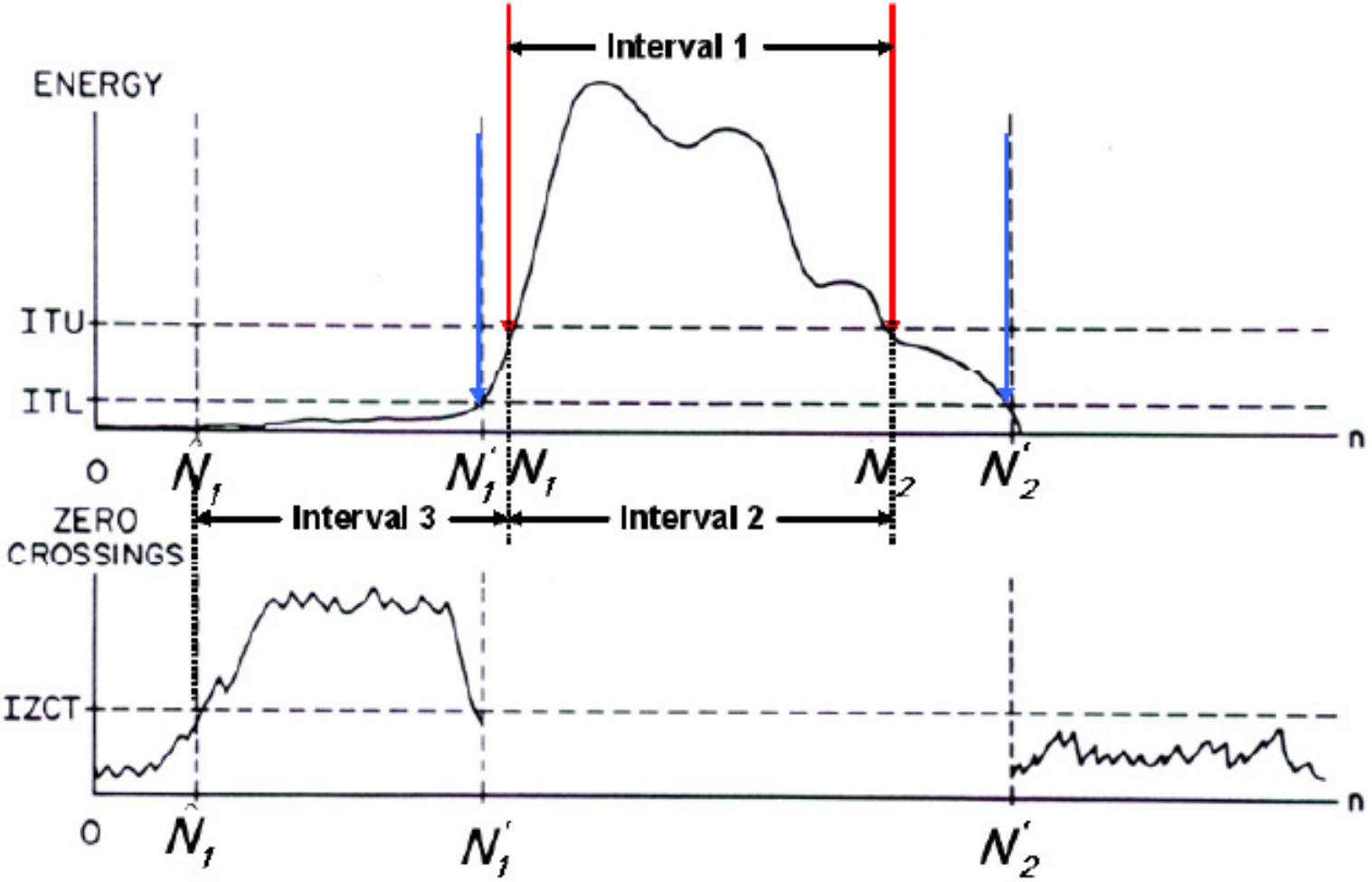
Zero crossings separate Unvoiced from Silence and Voiced

Rule-Based Short-Time Measurements of Speech

Algorithm for endpoint detection:

1. compute mean and σ of log E_n and Z_{100} for first 100 msec of signal (assuming no speech in this interval and assuming F_S=*10,000 Hz*).
2. determine maximum value of log E_n for entire recording => normalization.
3. compute log E_n thresholds based on results of steps 1 and 2—e.g., take some percentage of the peaks over the entire interval. Use threshold for zero crossings based on ZC distribution for unvoiced speech.
4. find an interval of log E_n that exceeds a high threshold ITU.
5. find a putative starting point (N_1) where log E_n crosses ITL from above; find a putative ending point (N_2) where log E_n crosses ITL from above.
6. move backwards from N_1 by comparing Z_{100} to IZCT, and find the first point where Z_{100} exceeds IZCT; similarly move forward from N_2 by comparing Z_{100} to IZCT and finding last point where Z_{100} exceeds IZCT.

Endpoint Detection Algorithm



Speech Parameters

$X = [x_1, x_2, x_3, x_4, x_5]$

$x_1 = \log E_S$ -- short-time log energy of the signal

$x_2 = Z_{100}$ -- short-time zero crossing rate of the signal for a 100-sample frame

$x_3 = C_1$ -- short-time autocorrelation coefficient at unit sample delay

$x_4 = \alpha_1$ -- first predictor coefficient of a p^{th} order linear predictor

$x_5 = E_p$ -- normalized energy of the prediction error of a p^{th} order linear predictor

Manual Training

- Using a designated training set of sentences, each 10 msec interval is classified manually (based on waveform displays and plots of parameter values) as either:
  - Voiced speech – clear periodicity seen in waveform
  - Unvoiced speech – clear indication of frication or whisper
  - Background signal – lack of voicing or unvoicing traits
  - Unclassified – unclear as to whether low level voiced, low level unvoiced, or background signal (usually at speech beginnings and endings); not used as part of the training set
- Each classified frame is used to train a single Gaussian model, for each speech parameter and for each pattern class; i.e., the mean and variance of each speech parameter is measured for each of the 3 classes

Frequency-domain Processing

- **Spectrogram – short-time Fourier analysis**
  - two-dimensional waveform (amplitude/time) is converted into a three-dimensional pattern (amplitude/frequency/time)
- **Wideband spectrogram:**
  - analyzed on 15ms sections of waveform with a step of 1ms
  - voiced regions with vertical striations due to the periodicity of the time waveform (each vertical line represents a pulse of vocal folds) while unvoiced regions are solid/random, or ‘snowy’
- **Narrowband spectrogram:**
  - analyzed on 50ms sections of waveform with a step of 1ms
  - pitch for voiced intervals in horizontal lines

Frequency-domain Processing