Signal Processing Reminder and Feature Extraction

DT2119 Speech and Speaker Recognition

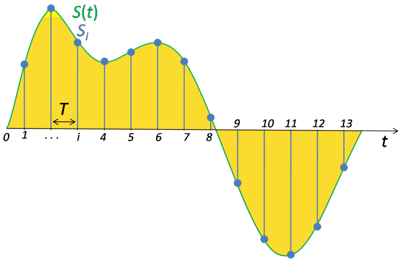
Giampiero Salvi

KTH/CSC/TMH giampi@kth.se

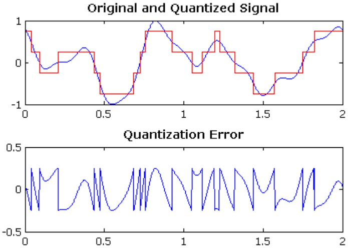
VT 2017

Continuous vs Digital Signals

sampling: discretisation in time



quantisation: discretisation in amplitude



(Figures from Wikipedia)

Linear Time-Invariant (LTI) Systems

$$x[n] \longrightarrow \boxed{T(.)} \longrightarrow y[n]$$

In general:

$$y[n] = T(x[n])$$

Time invariance:

$$y[n - n_0] = T(x[n - n_0])$$

Linearity:

$$T(a_1 x_1[n] + a_2 x_2[n]) = a_1 T(x_1[n]) + a_2 T(x_2[n])$$

LTI: Impulse Response

In general we can always write:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

For the linearity:

$$y[n] = T(x[n]) = \sum_{k=-\infty}^{\infty} x[k]T(\delta[n-k])$$

$h[n] \equiv T(\delta[n])$ is the system's response to an impulse $\delta[n]$
For the time invariance:

$$T(\delta[n-k]) = h[n-k]$$

$h[n]$ is a complete description of the system!

Convolution

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

$$y[n] = T(x[n]) = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n]$$

Properties:

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

Kind of complicated to interpret.

Sinusoidal Signals

Sinusoidal signals are eigensignals for LTI systems: if $x[n] = e^{j\omega_0 n}$ then

$$
\begin{aligned}
y[n] &= x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] = (k \equiv n-m) \\
&= \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} h[k]e^{j\omega_0(n-k)} \\
&= \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega_0 k}e^{j\omega_0 n} = e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega_0 k} \\
&= H(\omega_0)e^{j\omega_0 n}
\end{aligned}
$$

Transfer Function

$$x[n] \longrightarrow \boxed{H(\omega)} \longrightarrow y[n]$$

$$H(\omega) = \sum_{k=-\infty}^{\infty} h[k]e^{j\omega k}$$

Sinusoidal signals:

$$x[n] = e^{j\omega_0 n} \rightarrow y[n] = H(\omega_0)e^{j\omega_0 n}$$

$\omega = 2\pi f$, where f is the frequency

Fourier Transforms

Fourier transform of continuous signals

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{j\omega t} dt$$

Fourier transform of discrete signals

$$X(\omega) = \sum_{k=-\infty}^{\infty} x[k] e^{j\omega k}$$

Discrete Fourier Transform

$$X[n] = \sum_{k=-\infty}^{\infty} x[k] e^{j 2\pi \frac{n}{N} k}$$

Transfer Function for Generic Signals

$$x[n] \longrightarrow \boxed{H(\omega)} \longrightarrow y[n]$$

Sinusoidal signals:

$$x[n] = e^{j\omega_0 n} \rightarrow y[n] = H(\omega_0)e^{j\omega_0 n}$$

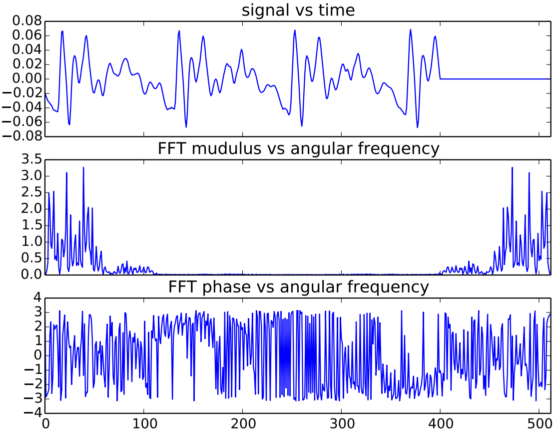
Generic signals (can be decomposed in sinusoids):

$$Y(\omega) = H(\omega)X(\omega)$$

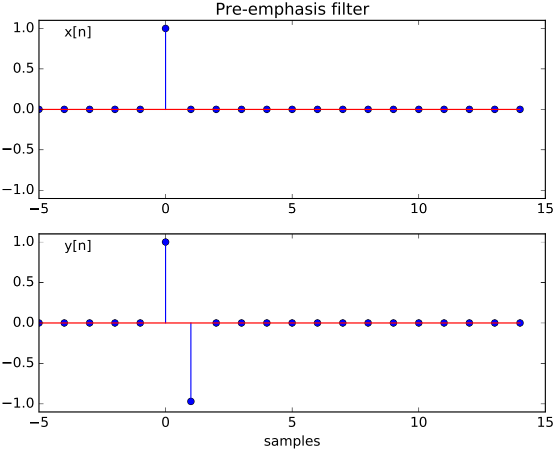
$\omega = 2\pi f$, where f is the frequency

More on DFT ($N = 512$)

$$X[n] = \sum_{k=-\infty}^{\infty} x[k] e^{j2\pi \frac{n}{N} k}$$

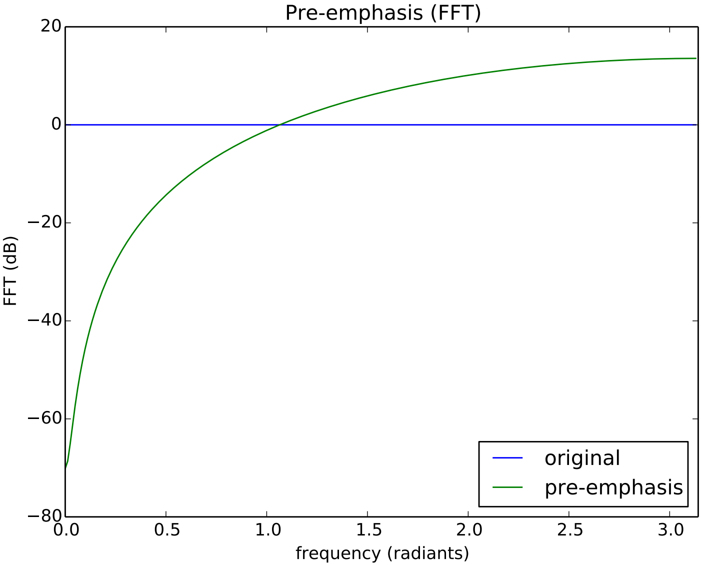


Examples of LTIs: Pre-emphasis

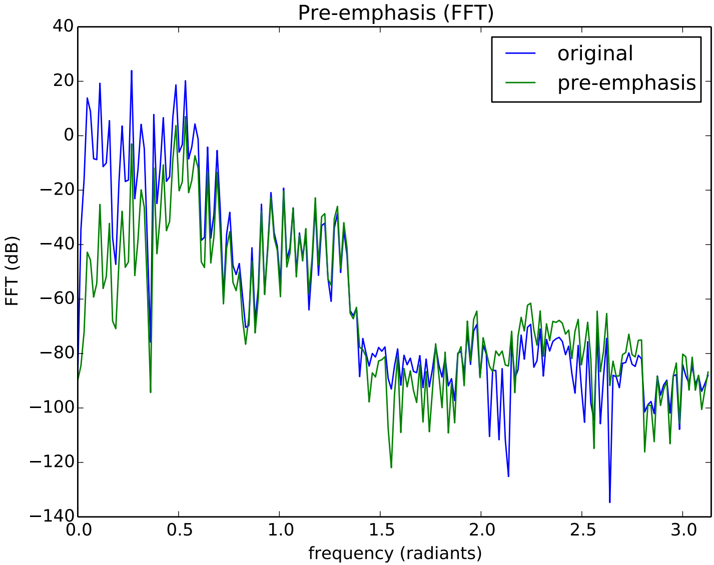


$$y[n] = x[n] - \alpha x[n-1], \quad \text{with } \alpha = 0.97$$

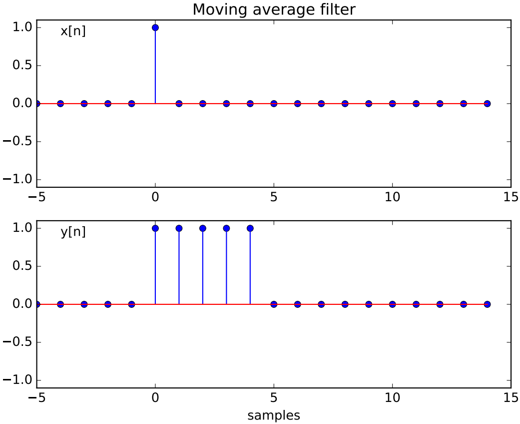
Pre-emphasis in frequency domain



Pre-emphasis applied to vowel



Examples of LTIs: Moving Average



$$y[n] = x[n] + x[n-1] + \cdots + x[n-P]$$

Finite Impulse Response (FIR) Systems

y only depends on (delayed) samples of the input (no feedback)

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \cdots + b_P x[n-P] \\ &= \sum_{i=0}^{P} b_i x[n-i] \end{aligned}$$

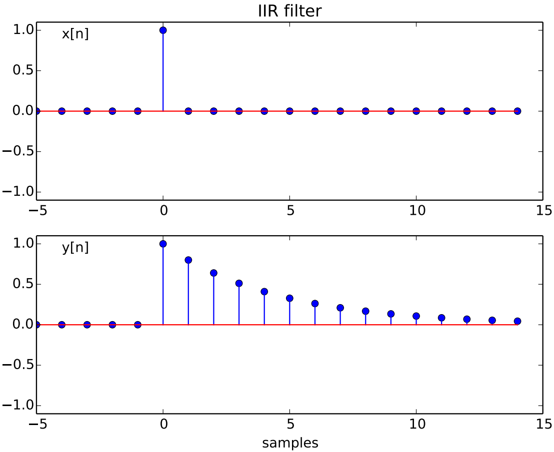
Infinite Impulse Response (IIR) Systems

Auto regressive (AR): y depends on (delayed) samples of the input, as well as the output at previous times (feedback)

$$
\begin{aligned}
y[n] &= \frac{1}{a_0}\left(b_0 x[n] + b_1 x[n-1] + \cdots + b_P x[n-P] + \right. \\
&\quad \left. -a_1 y[n-1] - a_2 y[n-2] - \cdots + a_Q y[n-Q]\right) \\
&= \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j]\right)
\end{aligned}
$$

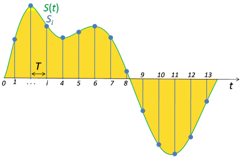
IIR Example

$$y[n] = x[n] - ay[n-1]$$



stable only if $|a| < 1$, here $a = -0.8$

Sampling Theorem (Nyquist-Shannon)



$$x(t), t \in \mathbb{R} \longrightarrow \boxed{\text{sampling}} \longrightarrow x[n], n \in \mathbb{N}$$

If $x(t)$ contains energy up to B_x, in order to reconstruct the signal we need to sample with

$$f_s > 2B_x$$

Aliasing

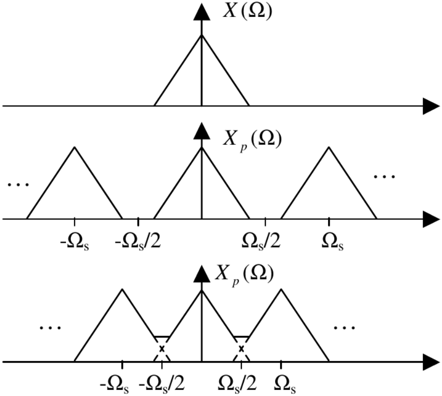
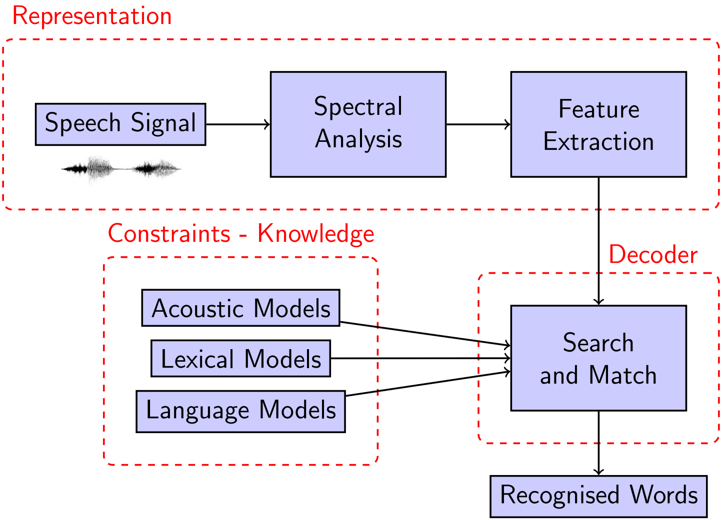


Figure from Huang, Acero and Hon (2001)

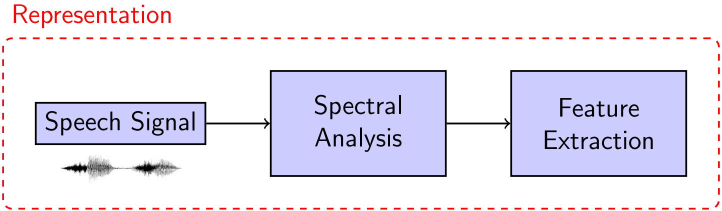
Aliasing: Illustration

Video from `https://youtu.be/usN47Jvy9PY`

Components of ASR System



Speech Signal Representations



Goals:

- disregard irrelevant information
- optimise relevant information for modelling

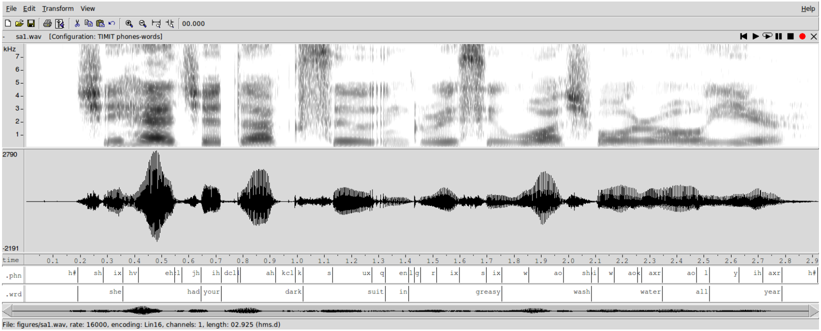
Means:

- try to model essential aspects of speech production
- imitate auditory processes
- consider properties of statistical modelling

First step: represent speech signal

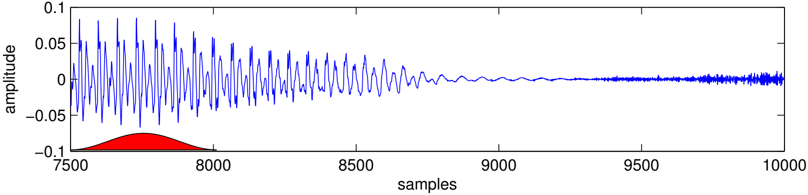
- Pressure wave converted into electric current (microphone)
- Sampling
  - Nyquist-Shannon Theorem: sample at twice the band
  - 8kHz (4kHz band, telephone), 16kHz (8 kHz band, high quality)
  - TIDIGITS sampled at 20kHz
  - TIMIT sampled at 16kHz
- Quantisation
  - Type of quantisation: linear, a-law, μ-law
  - 8, 16 bits (more rare 32, floating point)
  - TIDIGITS and TIMIT are quantised with 16 bits linear

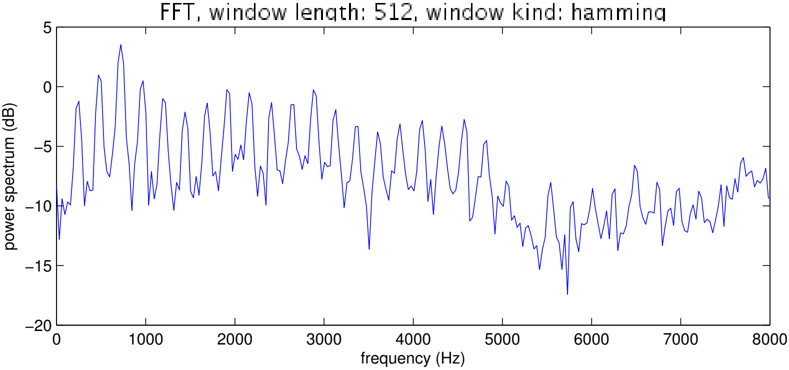
A time varying signal



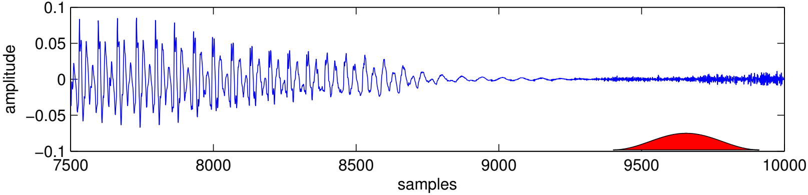
- speech is time varying
- short segment are quasi-stationary
- use short time analysis

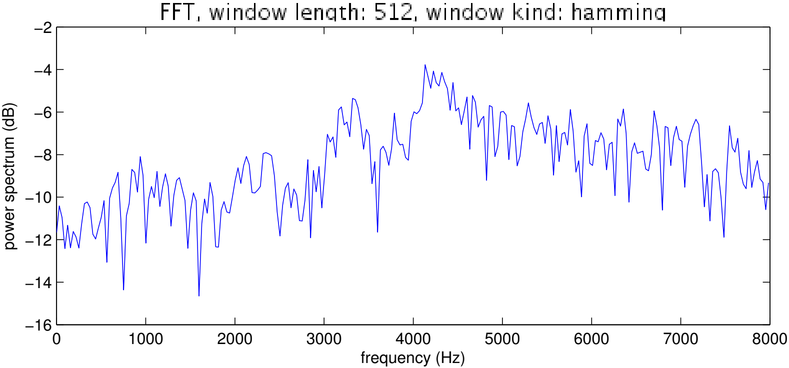
Short-Time Fourier Analysis





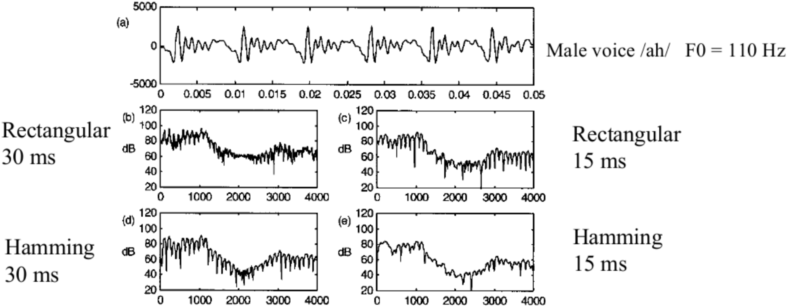
Short-Time Fourier Analysis





Short-Time Fourier Analysis

Effect of different window functions



Window should be long enough to cover 2 pitch pulses
Short enough to capture short events and transitions

Windowing, typical values

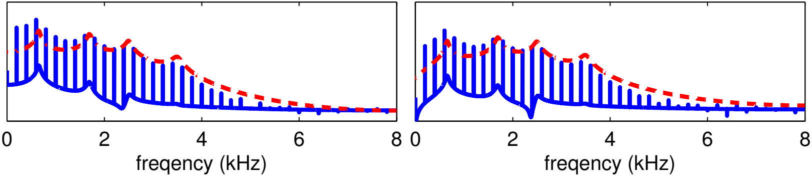
- signal sampling frequency: 8–20kHz
- analysis window: 10–50ms
- frame step: 10–25ms (100–40Hz)

Pre-emphasis

Compensate for the 6db/octave drop (radiation at the lips)

$$y[n] = x[n] - \alpha x[n-1]$$

Corresponds to a linear filter with $A = 1$ and $B = [1 \quad -\alpha]$

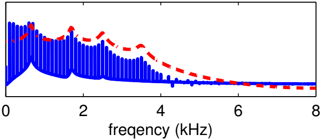


α is usually 0.95–0.97

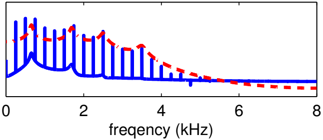
F_0 and Formants

- Varying F_0 (vocal fold oscillation rate)

spectrum (log) f0 = 100Hz

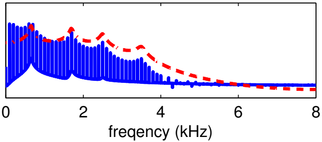


spectrum (log) f0 = 250Hz

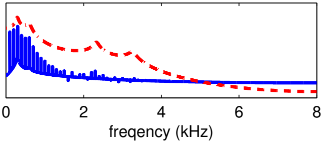


- Varying Formants (vocal tract shape)

spectrum (log) vowel [ε]



spectrum (log) vowel [u]



Linear Prediction Coefficients (LPC)

- assume all-pole model:

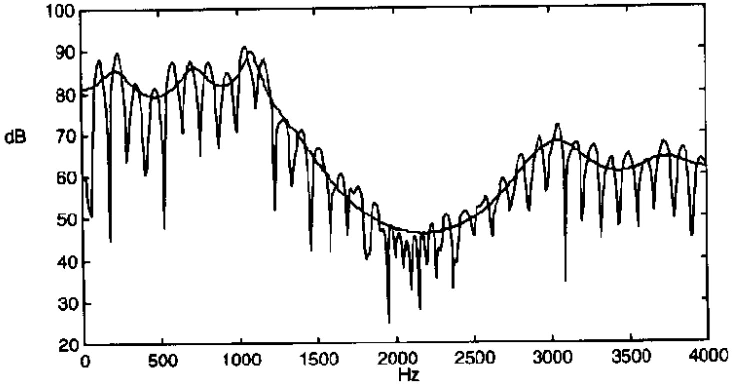
$$H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

- the output signal $s[n]$ can be expressed as the sum of the input $u_g[n]$ and a number of previous samples $a_k s[n-k]$:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + A u_g[n]$$

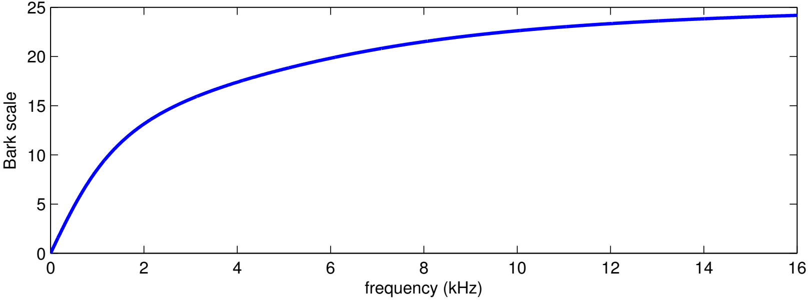
LPC Example

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Au_g[n]$$



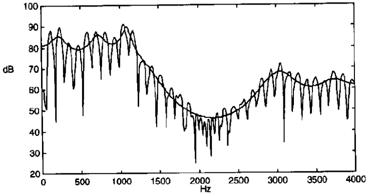
Perceptual Linear Prediction

- Transform to the Bark frequency scale before computing the LPC coefficients
- Cubic root of energy instead of logarithm



LPC Limitations

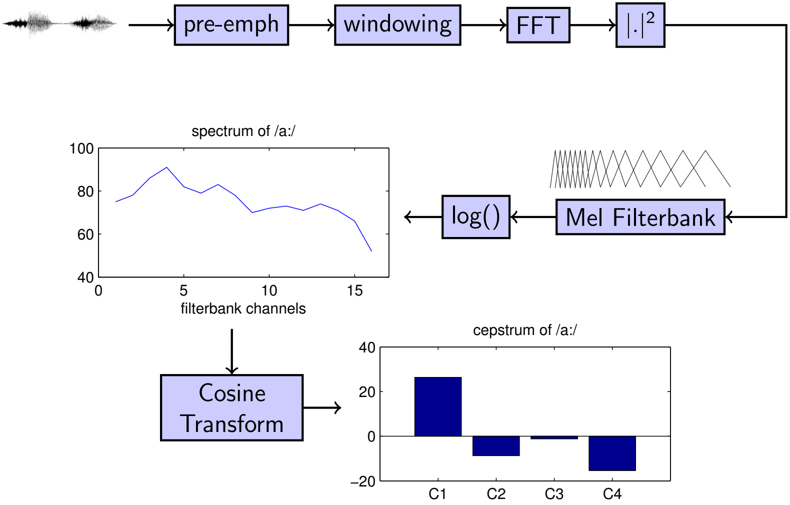
- better match at spectral peaks than at valleys
- not accurate if transfer function contain zeros (nasals, fricatives. . .)



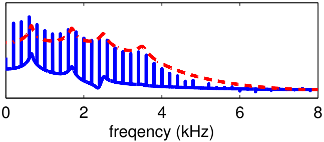
Mel Frequency Cepstrum Coefficients

- *de facto* standard in ASR (before Deep Learning)
- imitate aspects of auditory processing
- does not assume all-pole model of the spectrum
- uncorrelated: easier to model statistically

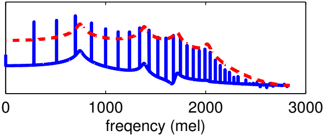
MFCCs Calculation



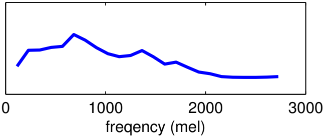
Mel Frequency Cepstral Coefficients



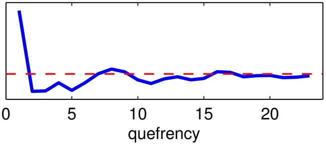
Linear to Mel frequency



Filterbank ($\sim$ 20-25 filters) + log()

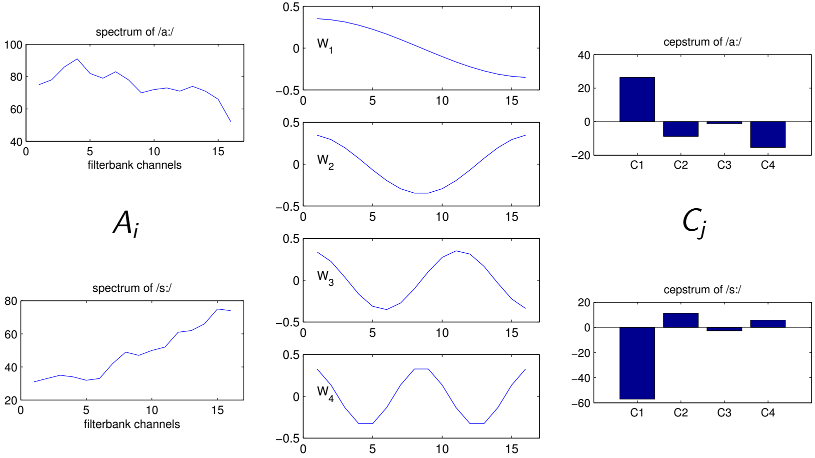


Discrete Cosine Transform



MFCC: Cosine Transform

$$C_j = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} A_i \cos\left(\frac{j\pi(i-0.5)}{N}\right)$$



MFCC Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$
- to analise the different contribution perform Fourier transform (DCT if not interested in phase information).
- Terminology:
  - frequency vs quefrency
  - spectrum vs cepstrum
  - filter vs lifter
  - ...

MFCC Advantages [1]

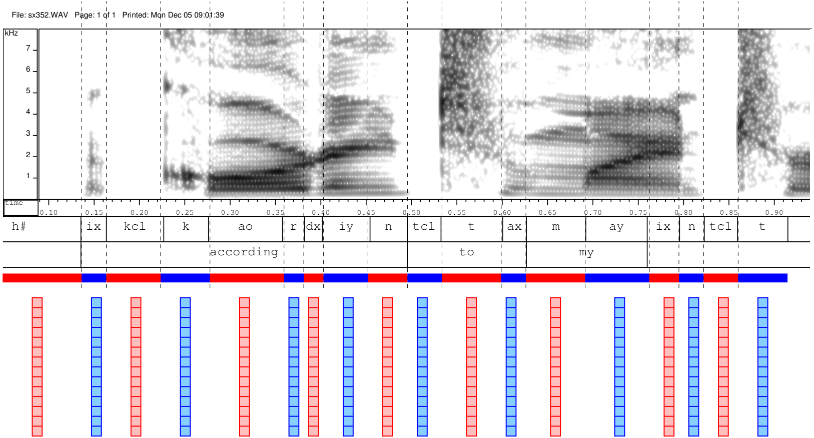
- fairly uncorrelated coefficients (simpler statistical models)
- high phonetic discrimination (empirically shown)
- do not assume all-pole model
- low number of coeff. enough to capture coarse structure of spectrum
- Cepstral Mean Subtraction corresponds to channel removal

[1] B. Bogert, M. Healy, and J. Tukey. "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-Cepstrum and Saphe Cracking". In: *Proc. Symp. Time Series Analysis*. Ed. by M. Rosemblatt. John Wiley & Sons, 1963, pp. 209–243

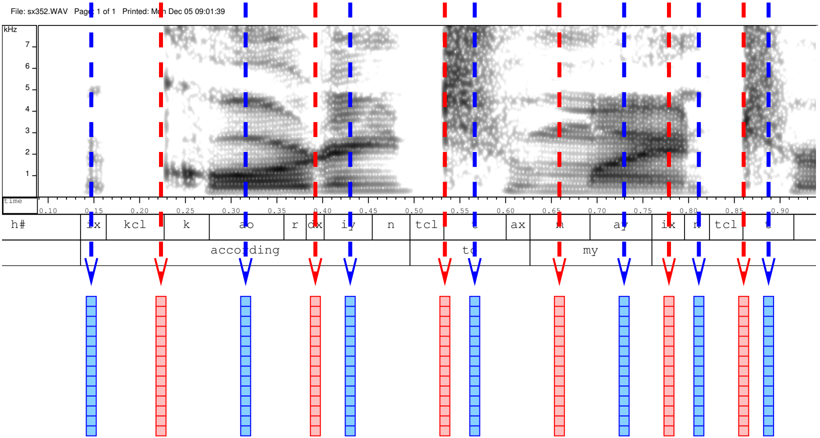
MFCCs: typical values

- 12 Coefficients C1–C12
- Energy (could be C0)
- Delta coefficients (derivatives in time)
- Delta-delta (second order derivatives)
- total: 39 coefficients per frame (analysis window)

Segment-Based Processing



Landmark-Based Processing



Frame-Based Processing