Signal Processing

10.B Speech features

Włodzimierz Kasprzak

2021

1. Speech features

Frame-based speech signal features

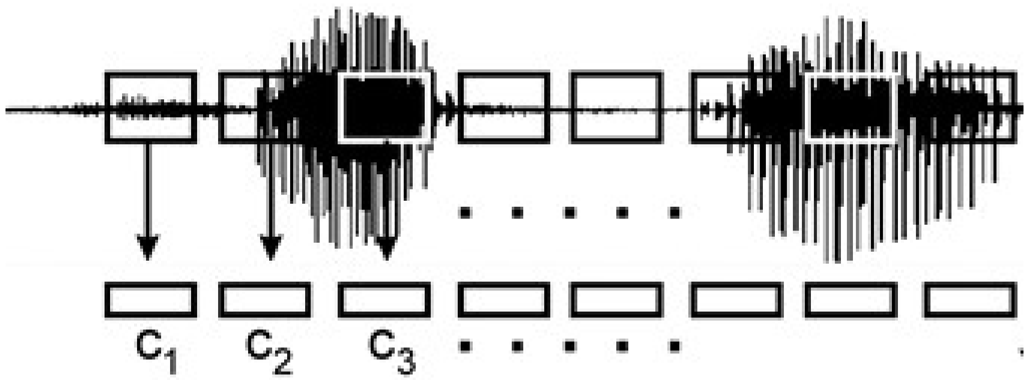
1. Mel-frequency cepstral coefficients (MFCC), extended by their first derivatives in time.

or

2. Speech features based on *Linear Predictive Coding* (LPC), e.g., LPCC – **linear predictive cepstral coefficients**

Speech signal (frames)

Features



Cepstrum (1)

The **cepstrum** of a signal $x[n]$ is the result of a homomorphic transformation:

$$cepstrum(x) = F^{-1}(\log |F(x)|),$$

where F is the discrete-time Fourier Transform (DFT) for MFCC or the Z transform for LPCC..

Note: spectrum → spec | trum → ceps | trum → cepstrum

MFCC:

Signal frame → $X=\mathrm{DFT}(x)$ → $mfc=\log mel|X|$ → $c=\mathrm{DFT}^{-1}(mfc)$ → cepstrum

LPCC:

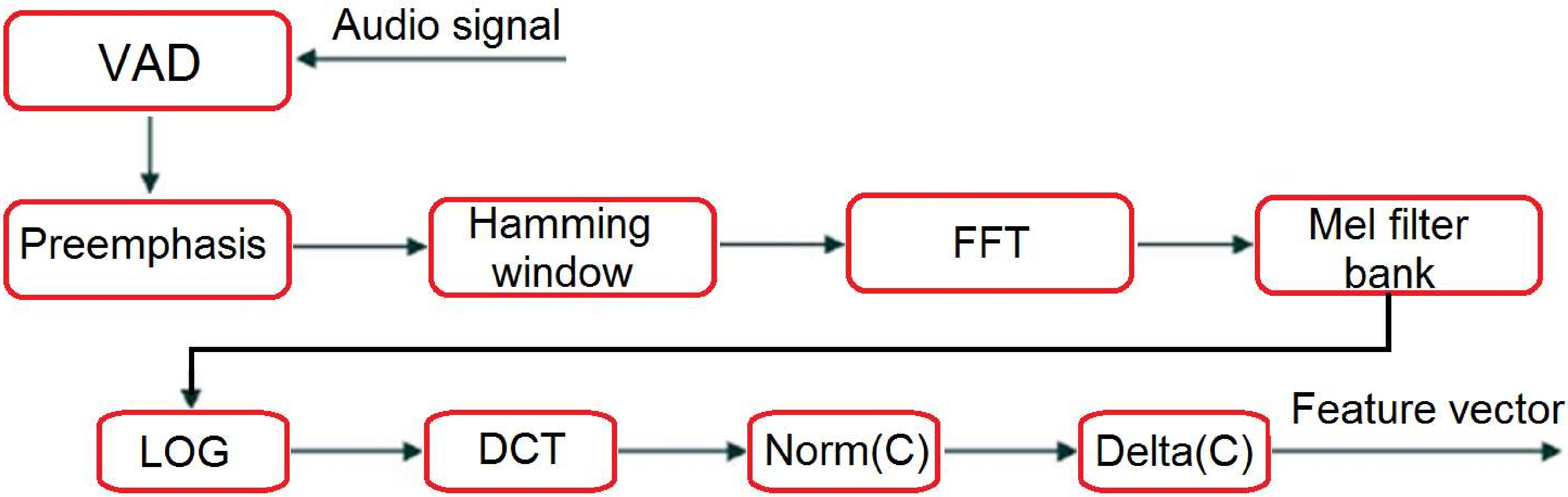
Signal frame → $\mathbf{H}=Z(\mathrm{LPC}(x))$ → $\boldsymbol{h}=\log(\mathbf{H})$ → $\boldsymbol{c}=Z^{-1}(\boldsymbol{h})$ → cepstrum

Cepstrum (2)

Why are **cepstrum features** useful for speech recognition?

- The cepstrum features characterizing the (impulse response of the) **vocal tract** are located near the “zero” feature $k=0$; whereas the input impulse components, corresponding to the **larynx-modulated oscillations** (that are not useful for speech recognition) are located at higher values of k (“longer” cepstrum time), where the cepstrum features achieve a maximum value;
- The useful features can be separated from the others by selecting some first-indexed features only, starting from $k=0$, and by additional decorrelation, called **liftering**.
- The speech part can also be separated from the acquisition channel’s (microphone) response by using **centered cepstrum** features.

2. MFCC



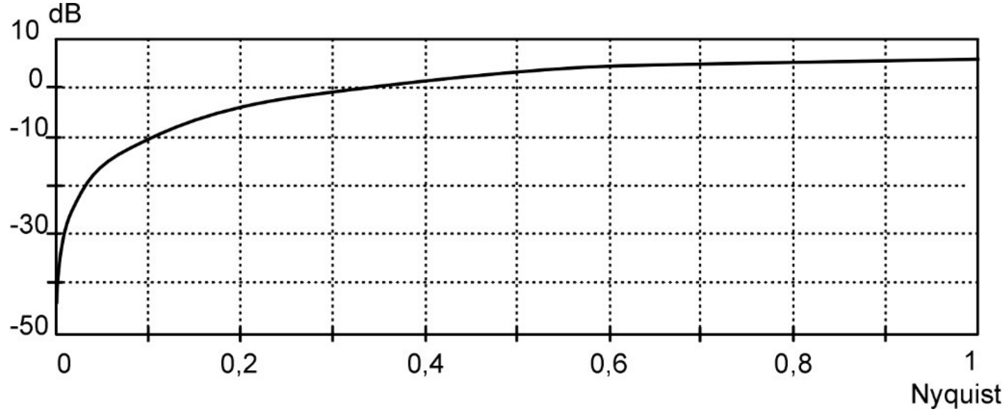
Audio signal
VAD
Preemphasis
Hamming window
FFT
Mel filter bank
LOG
DCT
Norm(C)
Delta(C)
Feature vector

Pre-emphasis filter

The goal of "pre-emphasis" is to strengthen the higher frequencies (is performed in the time domain):

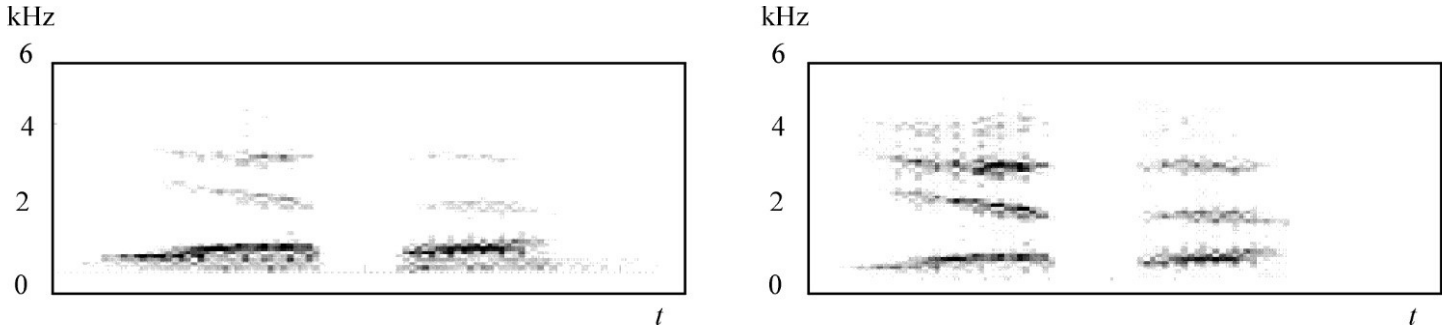
$$f_t' = f_t - \varphi \cdot f_{t-1}\ , \quad \text{where} \quad \varphi \in \langle 0.9, 1.0 \rangle\ .$$

The magnitude part of the frequency characteristics:



Example:

A spectrogram before and after pre-emphasis:

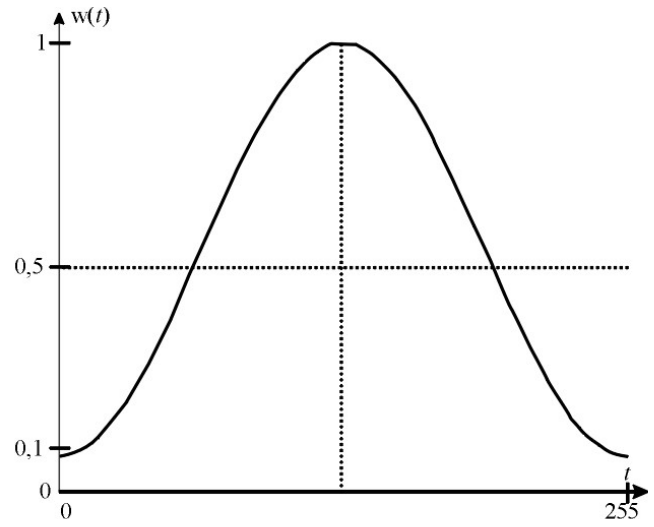


STFT

1. Short-time Fourier Transform (STFT)

A windowed DFT for every frame τ of the input signal:

$$F(k,\tau) = \sum_{t=0}^{M-1} \left(x[\tau+t] \cdot e^{-i2\pi kt/M} \cdot w_\tau[t] \right) \quad , \quad k=0,\ 1,\ \ldots,\ M\text{-}1$$



Window functions w[t]

1. Rectangular window
2. Triangle window
3. **Hamming** window

etc.

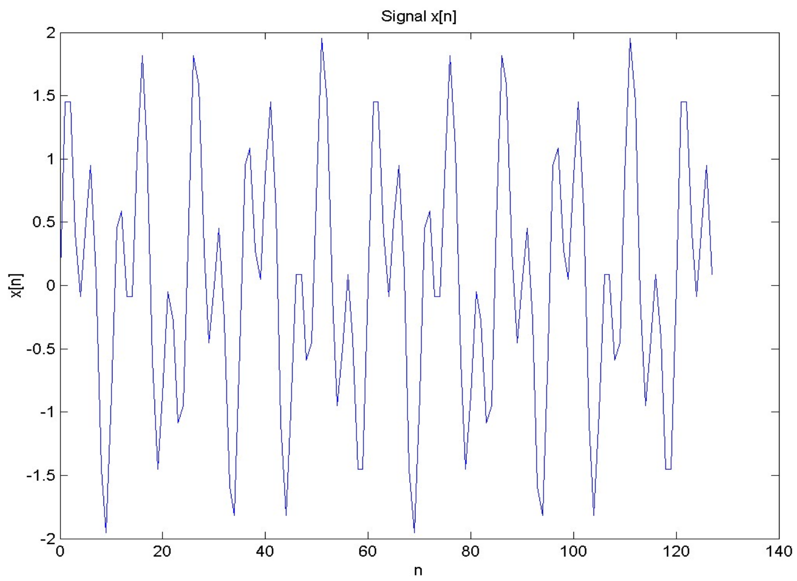
$$w_\tau[t] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi t}{M}\right), & for \quad t = \{0,1,\ldots,M-1\} \\ 0\,, & otherwise \end{cases}$$

Windowing example

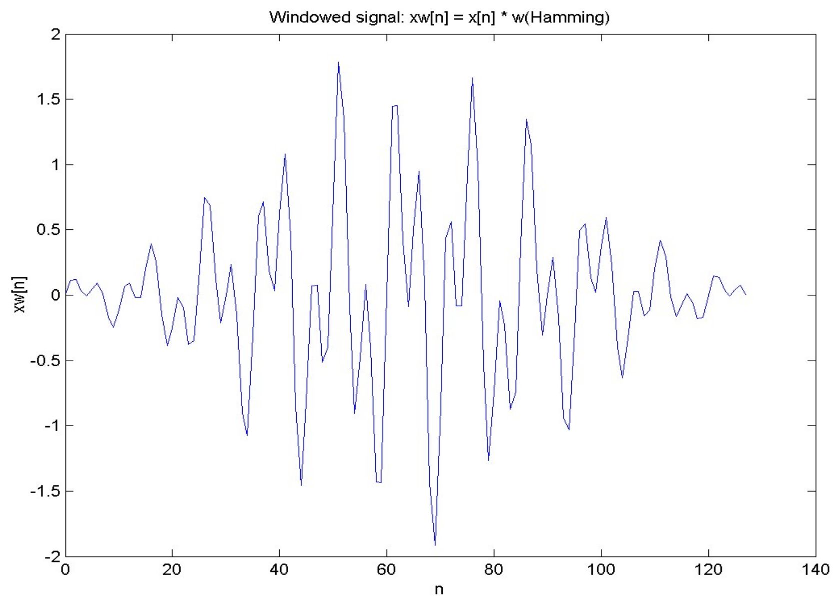
Example. $\mathrm{x}[n]$ is the sum of two sinus functions uniformly sampled from 0 to 2π by 128 samples:

$$\mathrm{x}[n] = \sin(2\pi\, n/5) + \sin(2\pi\, n\,/12), \qquad n=0,1,2,\ldots,127.$$

Single frame (rectangular window applied)



Single frame (Hamming window applied)

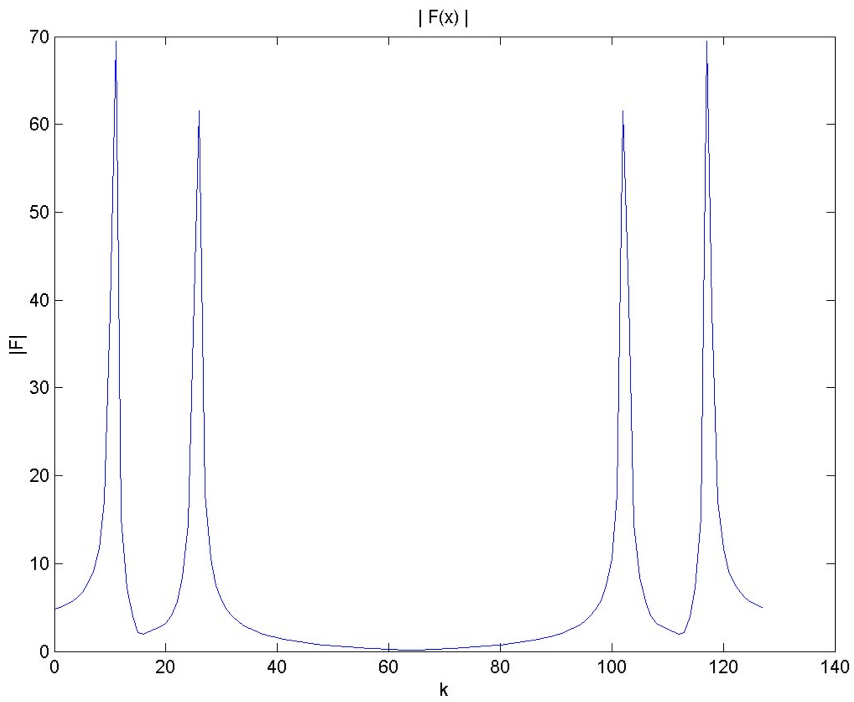


Windowing example (2)

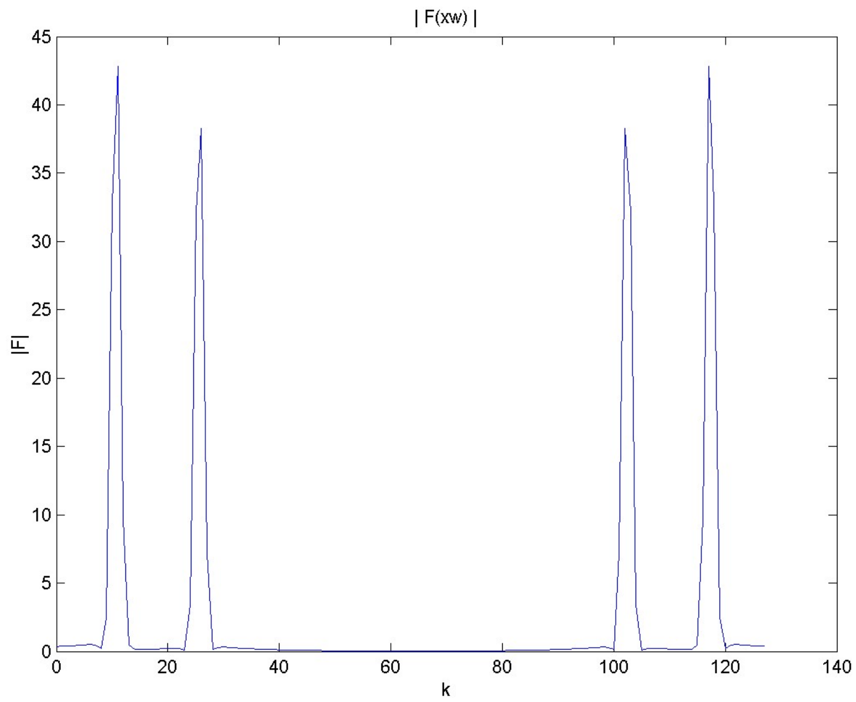
Example (cont.)

Magnitude of Fourier coefficients:

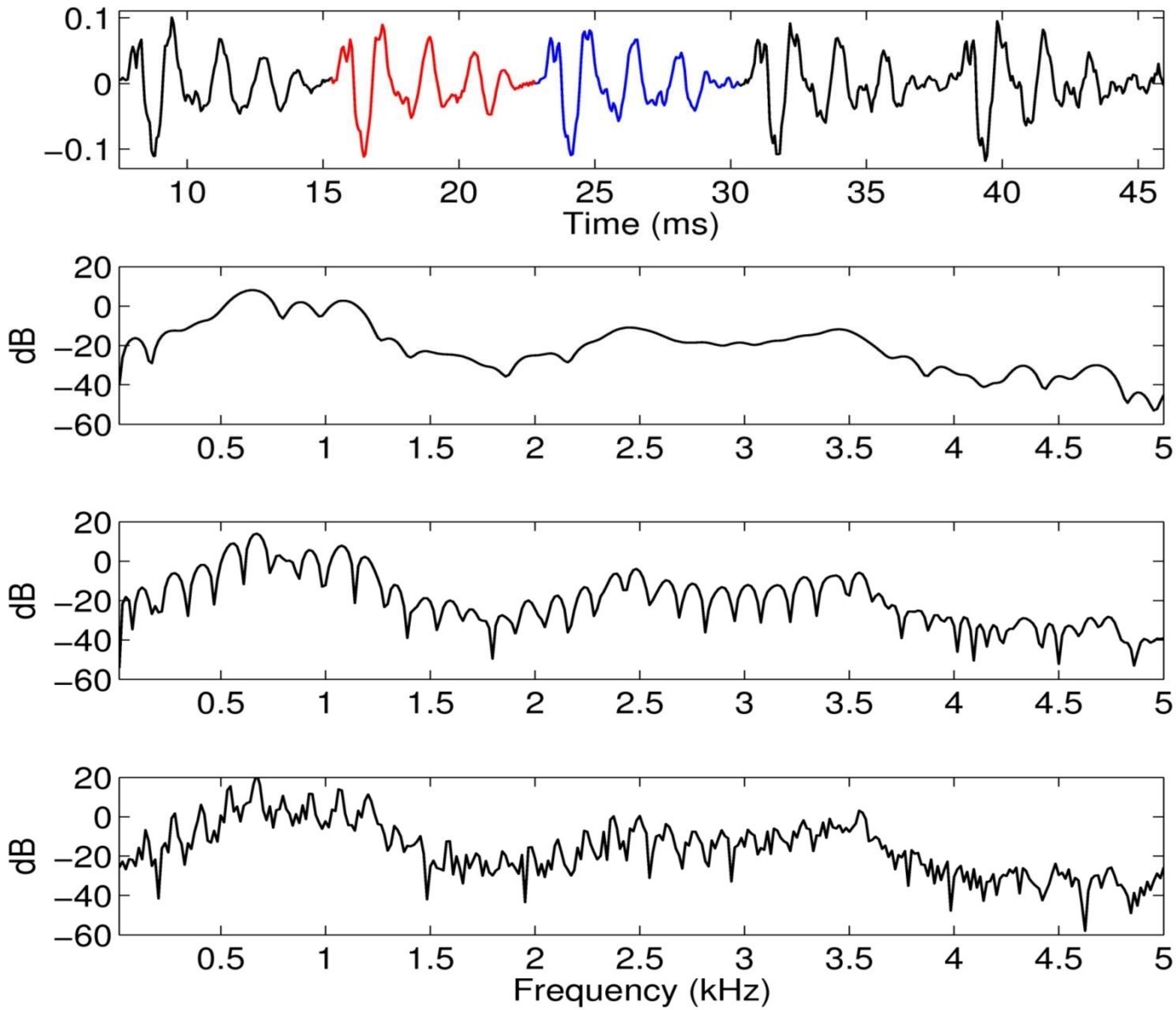
With rectangular window.



With Hamming window



Effect of frame size on the STFT



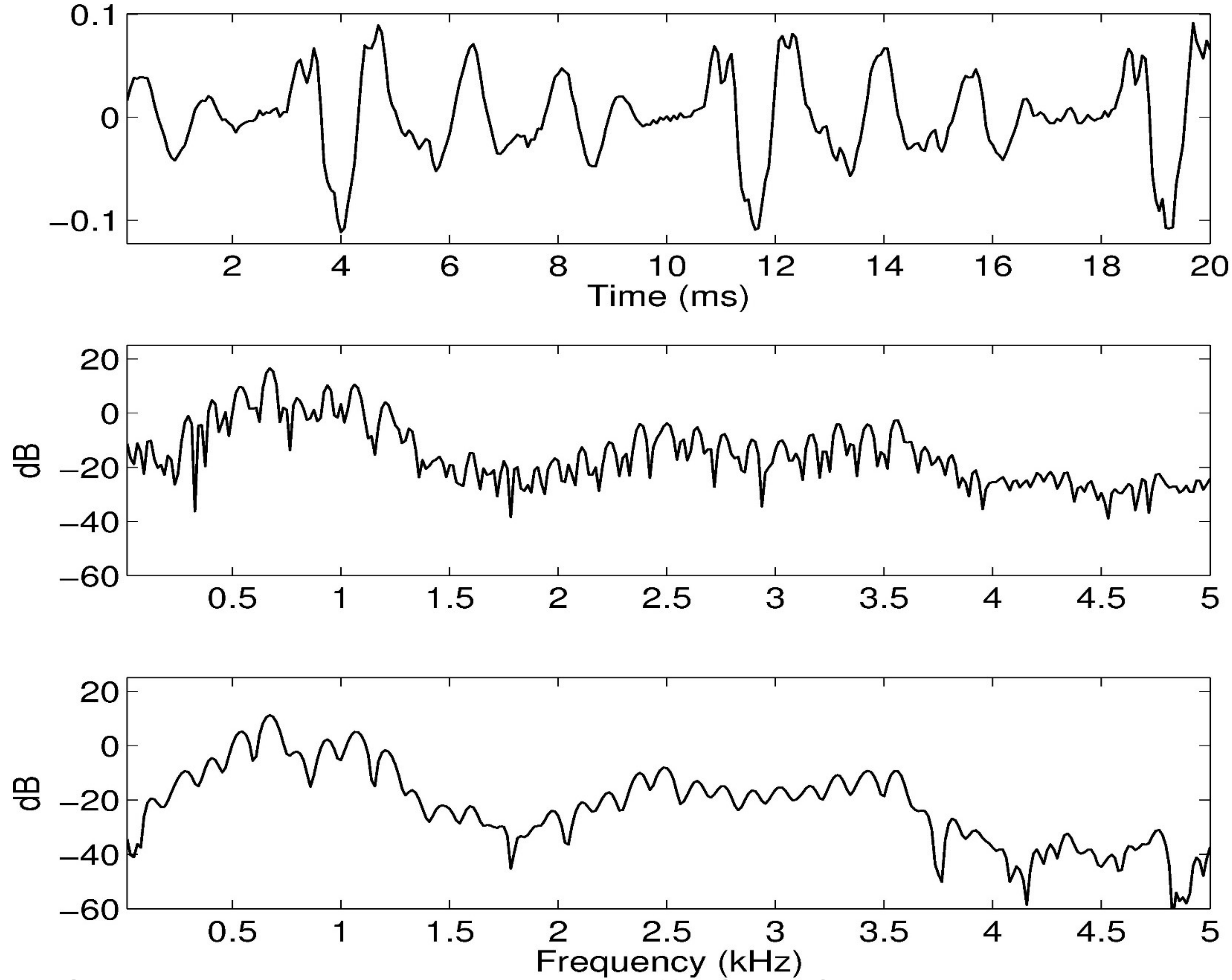
Speech signal – 5 pitch periods

STFT with frame = 1 pitch period : 10ms

STFT with frame = 2 pitch periods: 20ms

STFT with frame = 5 pitch periods: 50ms

Effect of window shape on STFT



Audio signal

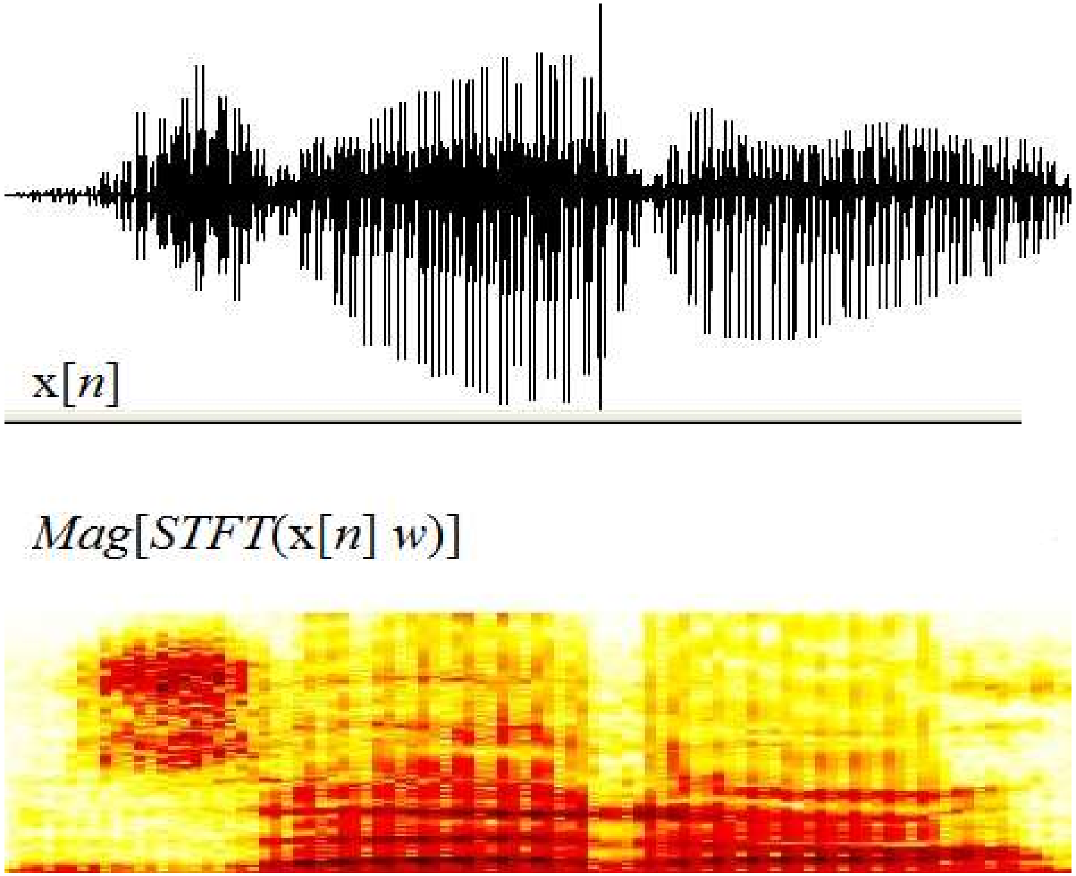
STFT with rectangular window

STFT with Hamming window

Spectrogram

Power of Fourier coefficients (squared magnitude)

$$FC(k,\tau) = \left| F(k,\tau) \right|^2 = \left| \sum_{t=0}^{M-1} (x[\tau+t] e^{-i2\pi kt/M} \cdot w_\tau[t]) \right|^2 \quad , \quad k = 0 ,..., \text{M-1}$$



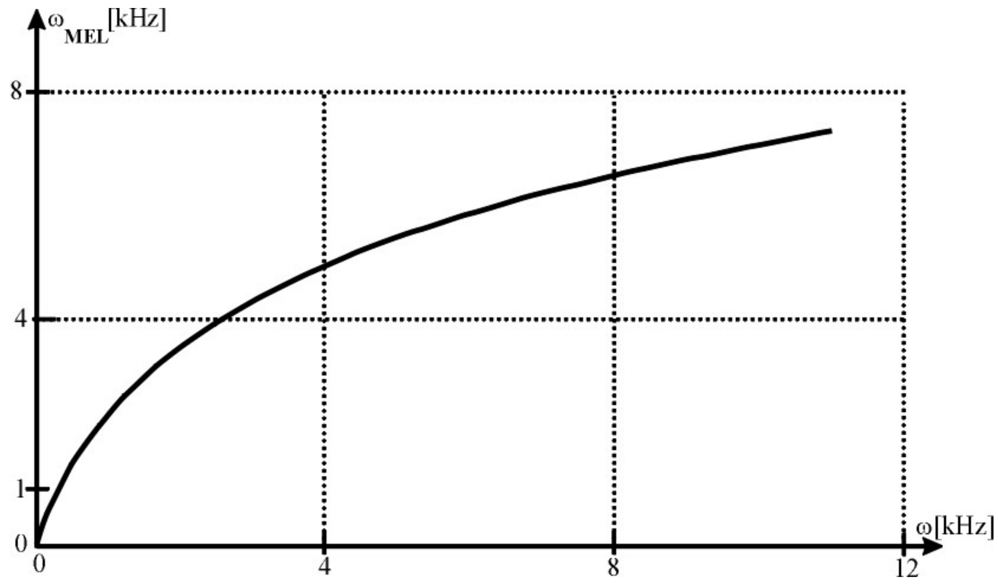
Mel frequency scale

Non-linear response of the human ear to the frequency components in the audio spectrum: differences in frequencies at the low end (< 1 kHz) are easier detectable than differences of the same magnitude in the high end of the audible spectrum.

Approach: a non-linear frequency analysis performed by the human ear - the higher the frequency the lower its resolution

MEL scale (empirical result):

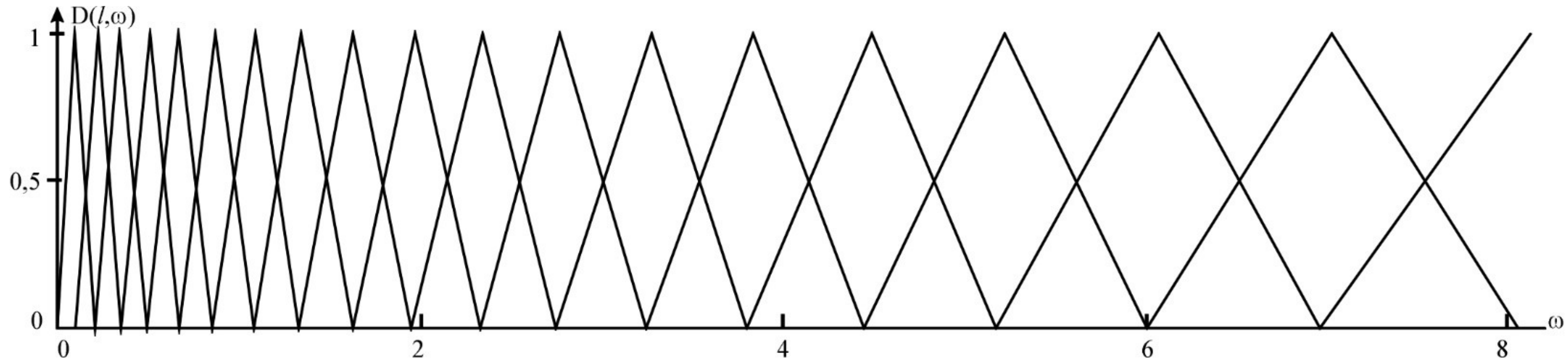
$$f_{mel} = 2595 \log\left(1 + \frac{f}{700[Hz]}\right)$$



Mel frequency filter

<u>Mel frequency coefficients (MFC)</u>

Triangular filters are located uniformly in the Mel frequency scale:



$$MFC(l,\tau) = \sum_{k=0}^{M-1} [D(l,k) \cdot FC(k,\tau)] \qquad l = 1,\dots, L$$

The MFC value associated with each bin corresponds to a weighted average of the power spectral values in the particular frequency range specified by the shape of the filter.

MFCC

The Mel-frequency cepstrum coeffcients are computed by the **homomorphic** transformation

$$MFCC(h) = FT^{-1}\{\log MFC\{FT\{h\}\}\},\ \text{for} \quad h = x \otimes w$$

The last step is the inverse Fourier Transform of logarithmic Mel frequency coefficients:

$$MFCC(k,\tau) = \sum_{l=0}^{L-1} [\log MFC(l,\tau) \cdot \cos(\frac{k \cdot (2l+1)\pi}{2L})] \qquad k = 1,\ldots,K$$

Centered MFCC

$$MFCC_{centered}(k,\tau) = MFCC(k,\tau) - mean\{MFCC(k,\tau)) \mid \tau = 1,2,\ldots]$$

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

Delta features

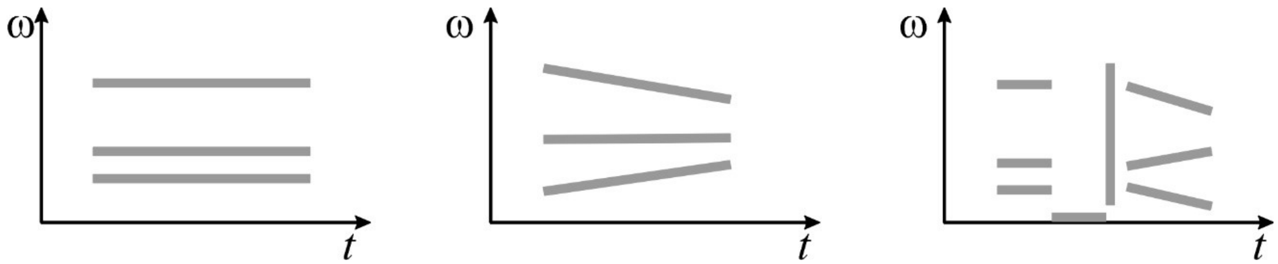
Energy feature

Additional feature - the total energy of signal in a single frame:

$$\mathrm{E}(\tau) = \log\left(\sum_{i=1}^{M} x_i^2\right)$$

Gradients of features in time („delta" features)

A schematic view of spectrograms for different phoneme types: single vowels (left), diphthongs (middle), plosives (right).



A linear regression in 5 consecutive frames is applied to find delta coefficients „d" (of MFCCs and energy feature „c"):

$$d(\tau) = \frac{2c(\tau+2) + c(\tau+1) - c(\tau-1) - 2c(\tau-2)}{10}$$

An extended feature vector

Energy	c0	E_mel
MFCC	c1	mfcc_1
	c2	mfcc_2
	...	...
	c18	mfcc_18
Delta energy	c19	ΔE_mel
Delta MFCC	c20	Δmfcc_1
	c21	Δmfcc_2
	...	...
	c37	Δmfcc_18
General features per frame (Total energy, mean and variance, norm. max. auto- correlation, low-band ratio)	c38	E
	c39	M1
	c40	MC2
	c41	r_max
	c42	L_p

3. LPC

The ***Z* transform** is a discrete-time signal transform, which is dual to the Laplace transform of continuous-time signals, that means a probing of signal by sinusoids and (decaying) exponentials:

$$\boxed{X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n}}$$

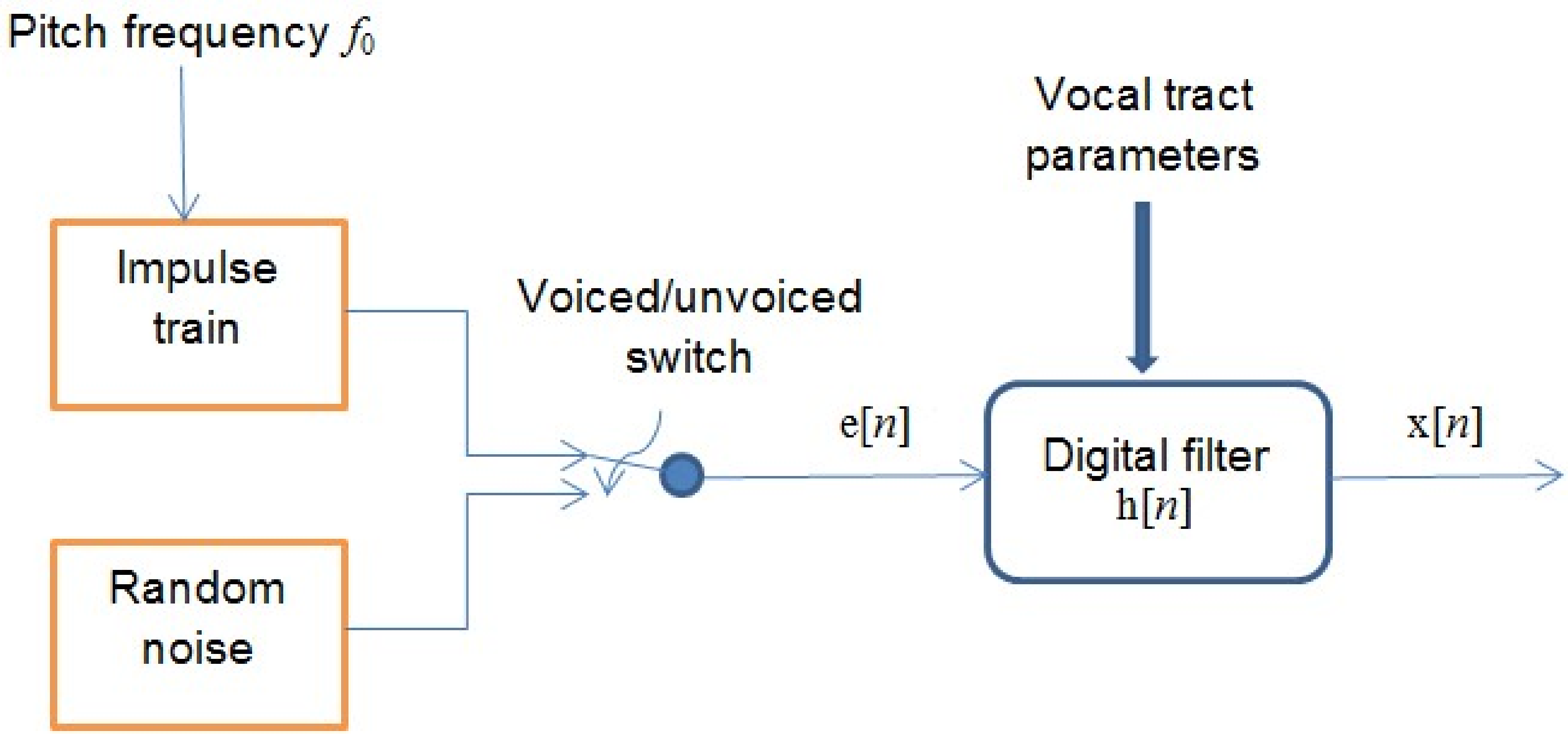
and z is a complex number: $z = r\, e^{-j\omega}$, $r = e^{-\sigma}$.

The **synthesis model** of human speech (in z-domain) consists of:

- an excitation source $E(z)$ on the input,
- a linear filter with transmittance $H(z)$,
- the speech signal $X(z)$ on its output;

(the signals and the filter are represented by their transforms in the complex-valued domain z).

Speech synthesis model



Let us denote by $\mathbf{H}(z)$ the transmittance of the filter (the z transform of its frequency response $\mathrm{h}[n]$). In the z domain:

$$\mathbf{X}(z) = \mathbf{H}(z)\mathbf{E}(z),$$

$$\mathbf{E}(z) = \mathbf{A}(z)\mathbf{X}(z)$$

IIR filter

A **digital IIR filter** is characterized by a recursive equation:

$$x[n] = b_0 e[n] + b_1 e[n-1] + b_2 e[n-2] + \cdots + b_p e[n-p]$$
$$+ a_1 x[n-1] + a_2 x[n-2] + \cdots + a_m x[n-m]$$

The n-th output sample, $x[n]$, is computed from the current and previous input samples and previous output samples. In short:

$$x[n] - \sum_{k=1}^{m} a_k \, x[n-k] = \sum_{k=0}^{p} b_k \, e[n-k]$$

A corresponding description in the z-domain is:

$$\mathbf{X}(z) = \mathbf{E}(z) \frac{\sum_{k=0}^{p} b_k \, z^{-k}}{1 - \sum_{k=1}^{m} a_k \, z^{-k}} \qquad \mathbf{H}(z) = \frac{\sum_{k=0}^{p} b_k \, z^{-k}}{1 - \sum_{k=1}^{m} a_k \, z^{-k}}$$

LPC

The **Auto-Regressive** (AR) model assumes that the numerator is 1:

$$\mathbf{H}(z) = \frac{1}{1 - \sum_{k=1}^{m} a_k z^{-k}}$$

Thus, in the AR model the n-th output sample, x_n, is estimated only on m previous output samples and current input sample as:

$$x[n] = e[n] + a_1 x[n-1] + a_2 x[n-2] + \cdots + a_m x[n-m]$$

In short:

$$x_n = e_n + \sum_{k=1}^{m} a_k x_{n-k}$$

Ideally, for voiced parts the vocal tract is cyclically fed by a Dirac delta impulse. Then: $e_0=1, \ e_n=0$, for short-time frames.

Thus, the n-th speech sample (in a frame) is estimated as a linear combination of the previous m samples:

$$\hat{x}_n = \sum_{k=1}^{m} a_k x_{n-k}$$

Auto-correlation method for LPC

The task is to compute the parameters, $\{ a_k \mid k=1, \ldots , m \}$, for every signal frame. By the LSE approach, for given frame, we have:

$$\varepsilon = \sum_{n=n0}^{n1} (x_n - \hat{x}_n)^2 \qquad \frac{\partial \varepsilon}{\partial a_i} = \sum_n \left(x_n - \sum_k a_k x_{n-k} \right) 2x_{n-i} = 0$$

where n_0, n_1 are training sample indices in given frame.

We get m equations with m unknowns:

$$\sum_k a_k \sum_n x_{n-k} x_{n-i} = \sum_n x_n x_{n-i} \cdots, \quad i = 1, \ldots, m$$

By introducing the first $m+1$ auto-correlation coefficients:

$$r_{|i-k|} = \sum_{n=0}^{M-1-|i-k|} x_n x_{n+|i-k|} = \sum_n x_{n-k} x_{n-i}$$

the equation system takes the form:

$$\sum_{k=0}^{m} a_k r_{|i-k|} = r_i \,, \quad i = 1, \ldots, m$$

LPC computation

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-1} \\ r_1 & r_0 & r_1 & \cdots & r_{m-2} \\ \vdots & & & & \vdots \\ r_{m-1} & r_{m-2} & r_{m-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{pmatrix}$$

$$\boldsymbol{Ra} = \boldsymbol{r}$$

The matrix $\boldsymbol{R}$ is a Toeplitz matrix (it is symmetric with equal diagonal elements). Due this Toeplitz property an efficient algorithm is available for computing $\boldsymbol{a}$ without computing the inverse matrix $\boldsymbol{R}^{-1}$.

Alternative method

The *Levinson-Durbin algorithm* is an iterative method for the solution of an equation system given by a Toeplitz matrix. It can be applied to solve the above system and give LPC parameters.

4. LPCC (1)

LPCC - the cepstral LPC

Recall, the speech synthesis filter function is transformed to the z-domain **transmittance** function:

$$\mathbf{H}(z) = \frac{1}{1 - \sum_{k=1}^{m} a_k z^{-k}}$$

The polynomial in the denominator part can be reorganized giving an all-pole transmittance function:

$$\mathbf{H}(z) = \frac{1}{1 - \sum_{k=1}^{m} a_k z^{-k}} = \frac{1}{\prod_{k=1}^{m} (1 - p_k z^{-1})}$$

Next, use the $\ln$ - function and apply the inverse Z transform:

$$\mathbf{c}[1:m] = Z^{-1}(\ln[\mathbf{H}(z)]) = Z^{-1}(\sum_{k=1}^{m} \ln[p_k z^{-1}])$$

LPCC (2)

A direct iterative method for computing the LPCC features

Instead of performing the particular steps of the cepstrum transformation of LPC coefficients, there exists an iterative method for a direct computation of LPCC features from the LPC coefficients.

For $1 \le n \le m$ (where m is the order of LPC transform) :

$$c[n] = -a_n - \sum_{k=1}^{n-1} (1 - \frac{k}{n}) c[n-k] a_k \ ; \quad n = 1,2,\ldots,m$$

For $n > m$:

$$c[n] = -\sum_{k=1}^{n-1} (1 - \frac{k}{n}) c[n-k] a_k \ ; \quad n > m$$

Exercises 10

Task 10.1

Compute MFC features for the following signal frame:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$x[n]$	20	10	5	5	5	0	-10	-10	0	0	0	0	0	0	0	0

Use a rectangular window. Assume, the magnitudes of Fourier coefficients to be as follows:

k	0	1	2	3	4	5	6	7	8
$\|F_k\|$	25	51.4	36.6	16.2	33.4	9.2	18.7	10.7	15

The sampling rate is 8 kHz. Use 3 triangle filters uniformly located according to the Mel-scale.

Exercises 10

Task 10.2

Compute the set of 4 LPC features for the following signal frame:

n	0	1	2	3	4	5	6	7
$x[n]$	20	10	5	5	5	0	-10	-10

Define and solve the linear system given by a Toeplitz matrix:

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-1} \\ r_1 & r_0 & r_1 & \cdots & r_{m-2} \\ \vdots & & & & \vdots \\ r_{m-1} & r_{m-2} & r_{m-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{pmatrix}$$