LPC

Jan Černocký, Valentina Hubeika

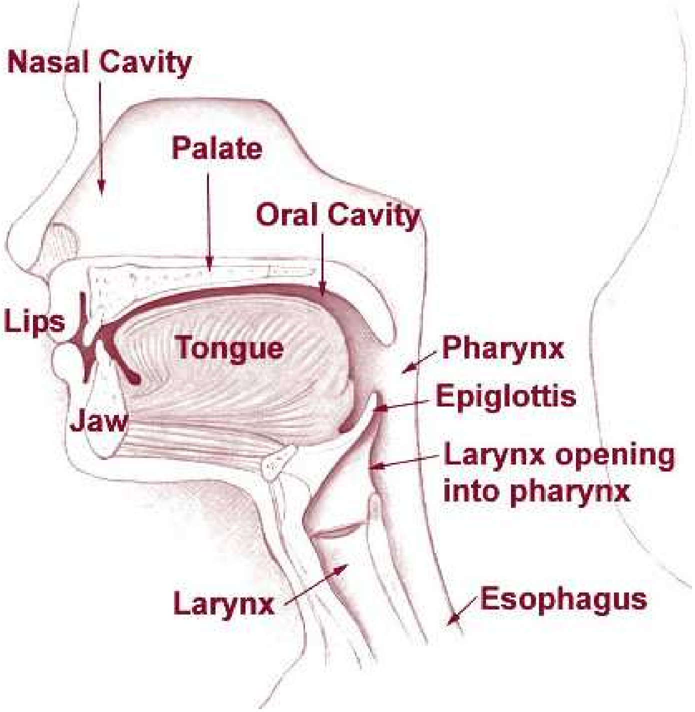
{cernocky,ihubeika}@fit.vutbr.cz

DCGM FIT BUT Brno

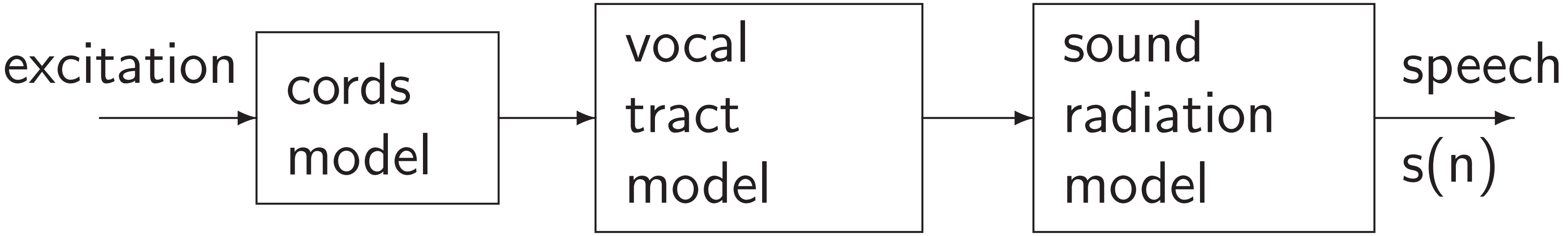
Agenda

- Signal model of articulatory tract.
- Motivation for linear prediction.
- Filter coefficients estimation.
- Levinson-Durbin algorithm.
- Power Spectral Density (PSD) using LPC.
- Parameters derived from LPC.

Recap – speech processing and its model



Articulatory Tract Model



Objective: estimate parameters of the speech production model.

This lecture is focused on filters.

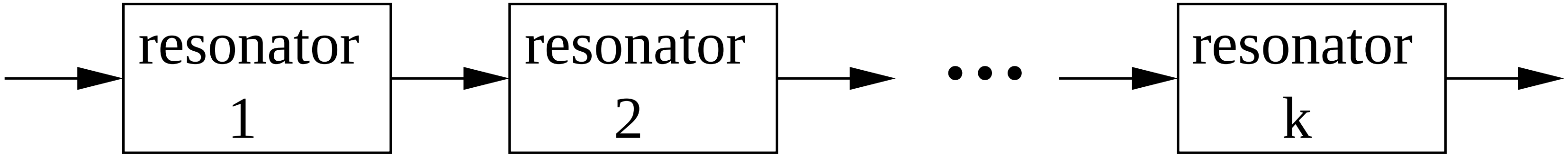
Vocal Cords

Low band filter 2d order with the cutoff frequency at about 100 Hz:

$$G(z) = \frac{1}{[1 - e^{-cT_s} z^{-1}]^2} \tag{1}$$

Vocal Tract

a cascade of two-pole *resonators* corresponding to *formants*.



For the k formants F_i with the band pass B_i:

$$V(z) = \frac{1}{\prod_{i=1}^{K}[1 - 2e^{-\alpha_i T_s}\cos\beta_i T_s z^{-1} + e^{-2\alpha_i T_s} z^{-2}]} \quad (2)$$

where parameters α_i and β_i are given by the location and bandwidth of the formants.

Model of the Sound Radiation

$$L(z) = 1 - z^{-1} \tag{3}$$

which is a high-pass filter.

It gives . . .

$$\begin{aligned} H(z) &= G(z)V(z)L(z) = \\ &= \frac{1-z^{-1}}{(1-e^{-cT_s}z^{-1})^2 \prod_{i=1}^{K}[1-2e^{-\alpha_i T_s}\cos\beta_i T_s z^{-1} + e^{-2\alpha_i T_s}z^{-2}]} \end{aligned} \quad (4)$$

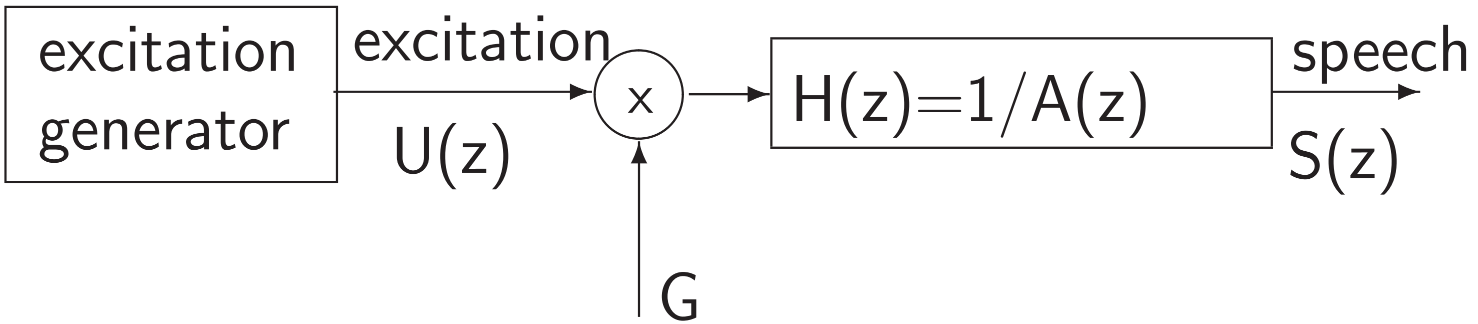
Component $cT_s \rightarrow 0$, hence we can cancel out the $1 - z^{-1}$ component from the nominator and denominator. The model is thus the **all-pole** filter. (consists of denominator only – purely recursive IIR filter). Usually denoted as:

$$H(z) = \frac{1}{1 + \sum_{i=1}^{P} a_i z^{-i}} = \frac{1}{A(z)}, \quad (5)$$

where the polynomu $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_P z^{-P}$ is of order $P = 2k + 1$ (k is number of formants). The most informative are first 4 or 5 formants, thus P is set to 10 (for F_s=8 kHz). Higher sampling frequencies require higher P (for instance 16), to cover higher part of the spectrum.

Estimation of the Model Parameters using Linear Prediction (LP)

Speech is produced using the following filter:



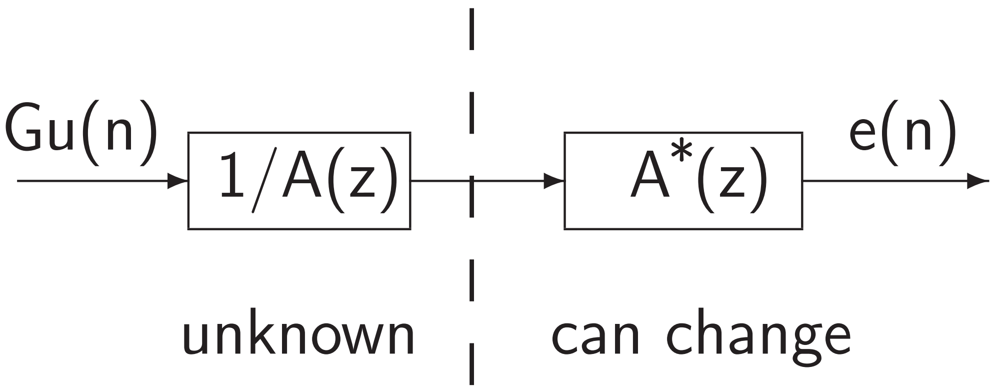
n-th speech sample is thus given by:

$$s(n) = Gu(n) - \sum_{i=1}^{P} a_i s(n-i) \qquad (6)$$

Parameters (coefficients) a_i of the filter are **unknown** and have to be **estimated**, or **identified** (system identification).

Filter Parameters Estimation

We can construct so called *inverse filter* $A^{\star}(z)$ with coefficients α_i:



For stationary signals $s(n)$, the coefficients a_i are *identified* using the coefficients α_i, when the $e(n)$' *output energy is minimized* : $\mathcal{E}\{e^2(n)\}$. "tune the filter parameters as long untill the output signal energy is minimal. . . "

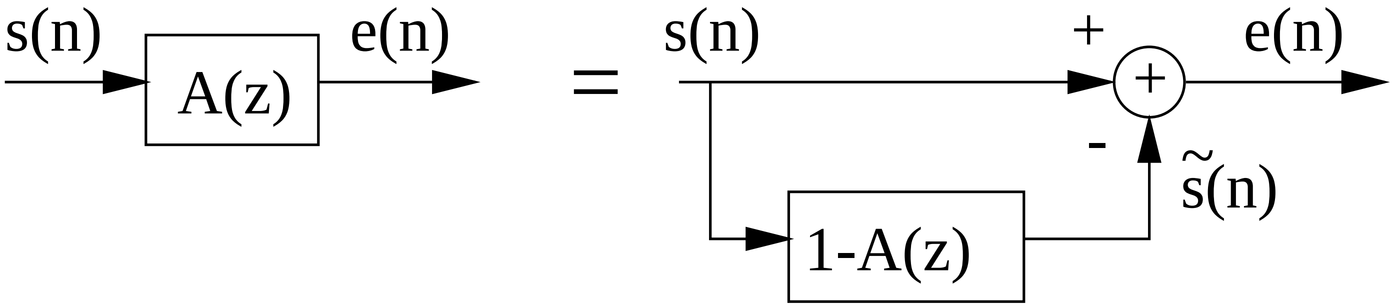
Why "Linear Prediction" ?

Assume, $\mathcal{E}\{e^2(n)\}$ is minimized. Thus $A^\star(z) = A(z)$.

$A(z)$ can be re-written as:

$$A(z) = 1 - [1 - A(z)] \tag{7}$$

Thus:



The signal sample $\tilde{s}(n)$ is given by a linear combination of a number of the preceding samples,
$\tilde{s}(n)$ is interpreted as *prediction* of the true sample $s(n)$:

$$\tilde{s}(n) = -\sum_{i=1}^{P} a_i s(n-i) \tag{8}$$

Prediction Error is given as the difference between the true and the estimated sample:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - [-\sum_{i=1}^{P} a_i s(n-i)] = s(n) + \sum_{i=1}^{P} a_i s(n-i). \tag{9}$$

the better prediction the smaller error.

In the plane z:

$$E(z) = S(z)A(z) \tag{10}$$

Advantages of the method:

- if $\alpha_i = a_i$, the prediction error is equal to *excitation*. (we can get to the root of vocal tract without a scalpel ;)).
- coefficient prediction using LP leads to a system of easy-to-solve linear equations.

Solution

Unnormalized energy of the prediction error is given by:

$$E = \sum_n e^2(n) \tag{11}$$

The value of E has to be minimized. Lets rewrite it using the signal $s(n)$ (known value) and unknown coefficients a_i. To get to the minimum, the expression must be partially derivated with respect to each a_i (gradient) and the derivations set to zero:

$$\begin{aligned}
\frac{\delta}{\delta a_j}\left\{\sum_n [s(n) + \sum_{i=1}^{P} a_i s(n-i)]^2\right\} &= 0 && (12)\\
\sum_n 2[s(n) + \sum_{i=1}^{P} a_i s(n-i)]s(n-j) &= 0 && (13)\\
\sum_n s(n)s(n-j) + \sum_{i=1}^{P} a_i \sum_n s(n-i)s(n-j) &= 0. && (14)\\
& && (15)
\end{aligned}$$

Denote:

$$\sum_n s(n-i)s(n-j) = \phi(i,j), \tag{16}$$

then

$$\sum_{i=1}^{P} a_i \phi(i,j) = -\phi(0,j) \quad \text{for} \quad 1 \leq j \leq P \tag{17}$$

Which is a system of linear equations:

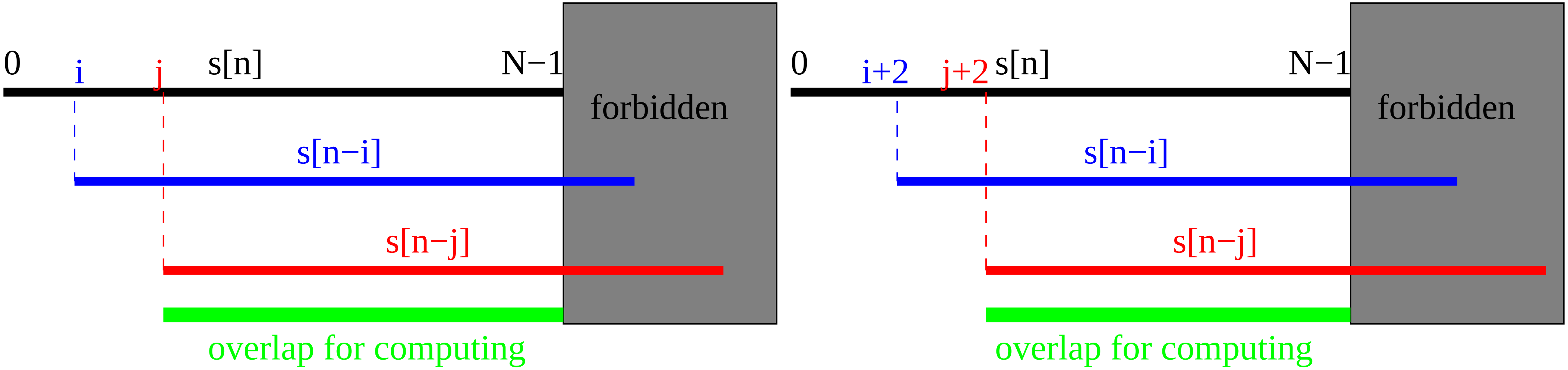
$$\begin{array}{ccc}
\phi(1,1)a_1 + \phi(2,1)a_2 + & \cdots & +\phi(P,1)a_P = -\phi(0,1) \\
\phi(1,2)a_1 + \phi(2,2)a_2 + & \cdots & +\phi(P,2)a_P = -\phi(0,2) \\
 & \vdots & \\
\phi(1,P)a_1 + \phi(2,P)a_2 + & \cdots & +\phi(P,P)a_P = -\phi(0,P),
\end{array} \tag{18}$$

Estimation of $\phi(\cdot,\cdot)$

The coefficients are estimated from frames of N samples. There are two methods differing in treating of the signal outside the frame (thus the samples for $n<0$ and $n>N-1$):

Covariance method

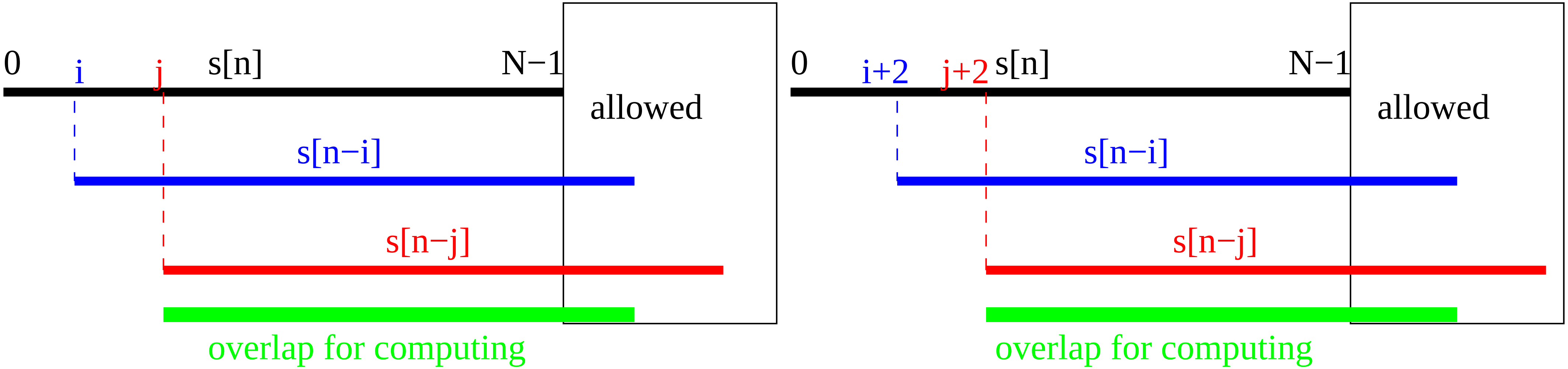
The signal outside the frame is **unknown**: samples beyond $[0, N-1]$ are simply not considered.



$\Rightarrow \phi(i,j)$ and $\phi(i+const, j+const)$ are not equal (we have a different number of samples - we have to compute the whole system of linear equations). Difficult, moreover the covariance method leads to an **unstable** filter $1/A(z)$.

Correlation method

The signal outside the frame is considered as **known** but having **null values**.



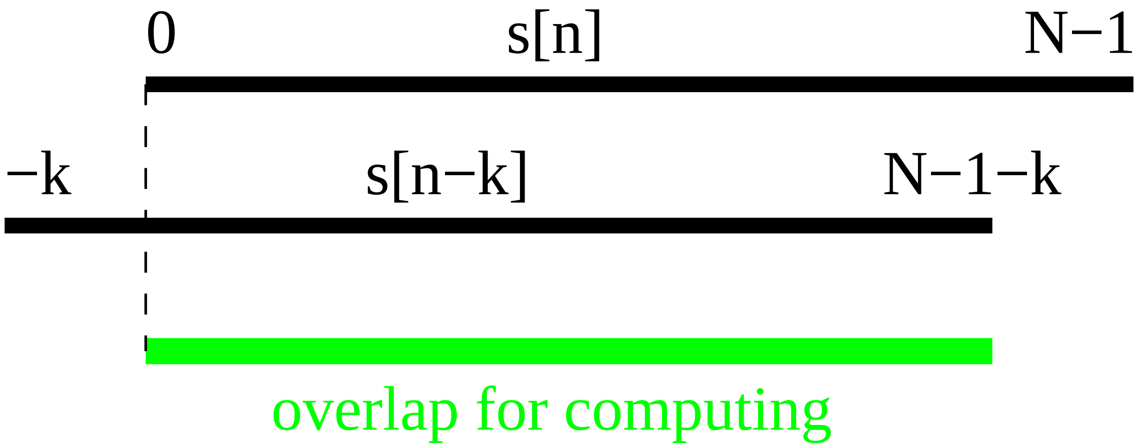
$\Rightarrow \phi(i,j)$ and $\phi(i+const, j+const)$ are equal (we have consistent number of samples) - easier to solve the linear equation system - the values on the diagonal are mutually equal (for instance, $\phi(2,1) = \phi(3,2) = \phi(4,3) = \cdots$).

Why ϕ are the autocorrelation coefficients

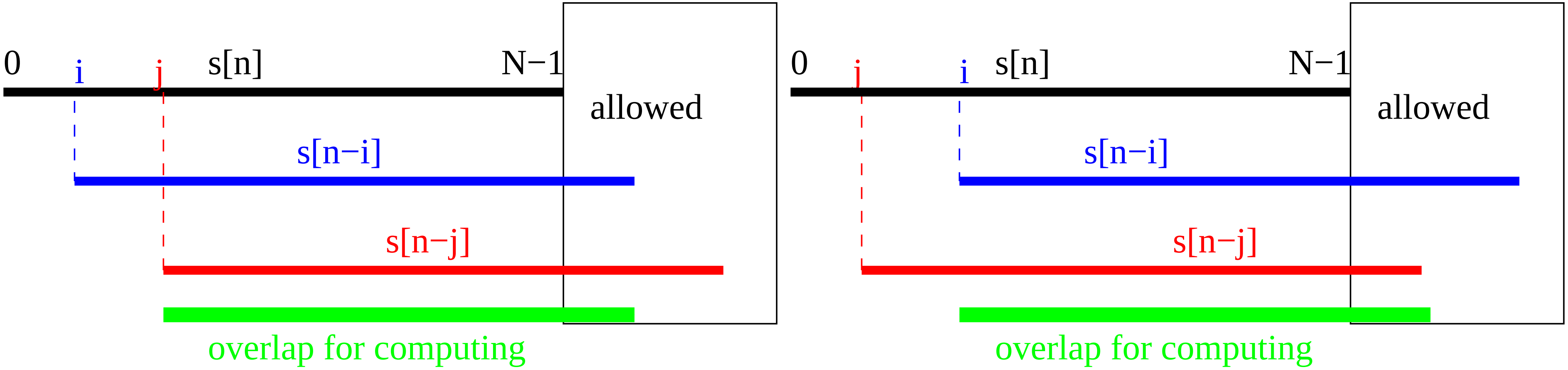
Autocorrelation coefficients' estimation (with no normalization) for a signal of the length N with positive k, see the Signals and Systems course, Random Processes II.: `http://www.fit.vutbr.cz/~cernocky/sig`

$$R(k) = \sum_{n=0}^{N-1-k} s(n)s(n+k)$$

Correlation coefficients "indicate signal similarity to its copy shifted by n samples"



For $\phi(i,j)$ and $\phi(j,i)$:



⇒ in both examples the computation is done using the same samples ⇒ both are equal to the autocorrelation coefficient $R(|i-j|)$. That is, the matrix is symmetric. Symmetric matrix containing identical diagonal components is called **Töplitz**.

Resulting equation system for the coefficients $a_1 \dots a_P$

$$
\begin{array}{lcr}
R(0)a_1 + R(1)a_2 + & \cdots & +R(P-1)a_P = -R(1) \\
R(1)a_1 + R(0)a_2 + & \cdots & +R(P-2)a_P = -R(2) \\
 & \vdots & \\
R(P-1)a_1 + R(P-2)a_2 + & \cdots & +R(0)a_P = -R(P),
\end{array}
\tag{19}
$$

Prediction Error Energy

Without derivation, using LPC, **unnormalized** prediction error is:

$$E = \sum_{n=0}^{N+P-1} e^2(n) = R(0) + \sum_{i=1}^{P} a_i R(i) \tag{20}$$

When the exciting signal has *normalized energy* (equal to 1) — for instance white noise with variability 1 or a series of impulses with $\frac{1}{N}\sum_{n=0}^{N-1} u^2(n) = 1$, then, to achieve the same energy as for the original signal $s(n)$, the filter gain (boost) has to be set to:

$$G^2 = \frac{E}{N} = \frac{1}{N}\left[R(0) + \sum_{i=1}^{P} a_i R(i)\right]. \tag{21}$$

... will be useful in encoding.

Levinson–Durbin

As the matrix $\mathbf{R}$ is symmetric and Töplitz (the diagonal items are equal), the Levinson and Durbin algorithm can be used to get fast solution for the equation system 19:

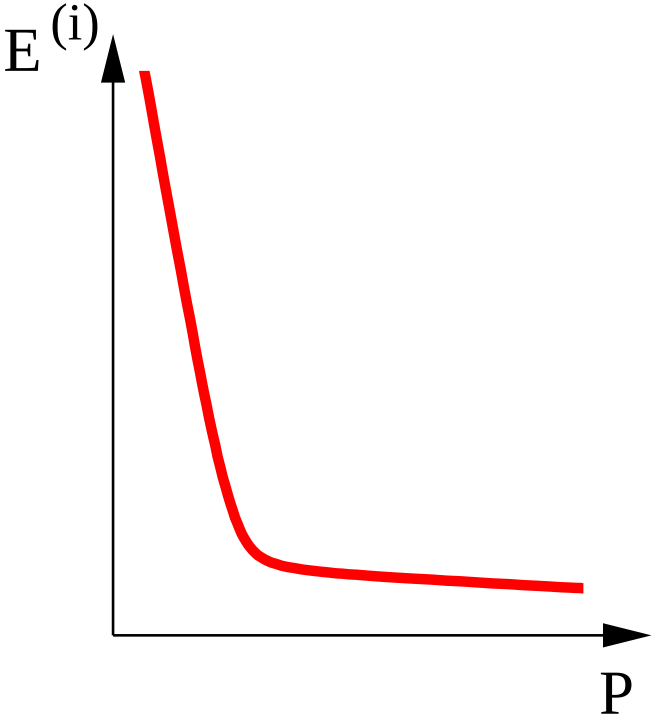
$$
\begin{aligned}
E^{(0)} &= R(0) && (22)\\
k_i &= -\left[R(i)+\sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)\right] / E^{(i-1)} && (23)\\
a_i^{(i)} &= k_i && (24)\\
a_j^{(i)} &= a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad \text{pro} \;\; 1 \leq j \leq i-1 && (25)\\
E^{(i)} &= (1-k_i^2)E^{(i-1)} && (26)
\end{aligned}
$$

Levinson-Durbin II.

- Subsequently, increase the predictor order (columns of the following table). $a_j^{(i)}$ is the j-th coefficient of the i-th order predictor:

$$\begin{array}{ccccc}
a_1^{(1)} & a_1^{(2)} & a_1^{(3)} & \cdots & a_1^{(P)} \\
 & a_2^{(2)} & a_2^{(3)} & \cdots & a_2^{(P)} \\
 & & a_3^{(3)} & \cdots & a_3^{(P)} \\
 & & & \ddots & \vdots \\
 & & & & a_P^{(P)}
\end{array} \tag{27}$$

- According to the plot of prediction error $E(i)$ as a function of the predictor order, an optimal order can be chosen:



Increasing the predictor order P after reaching the function's inflection point brings no improvement in the error energy.

Estimation of Power Spectral Density (PSD) using LPC Model

So far, PSD was estimated using DFT (contained the "fine" component containing folds of the fundamental frequency). PSD can as well be estimated using the filter $1/A(z)$ frequency characteristics:

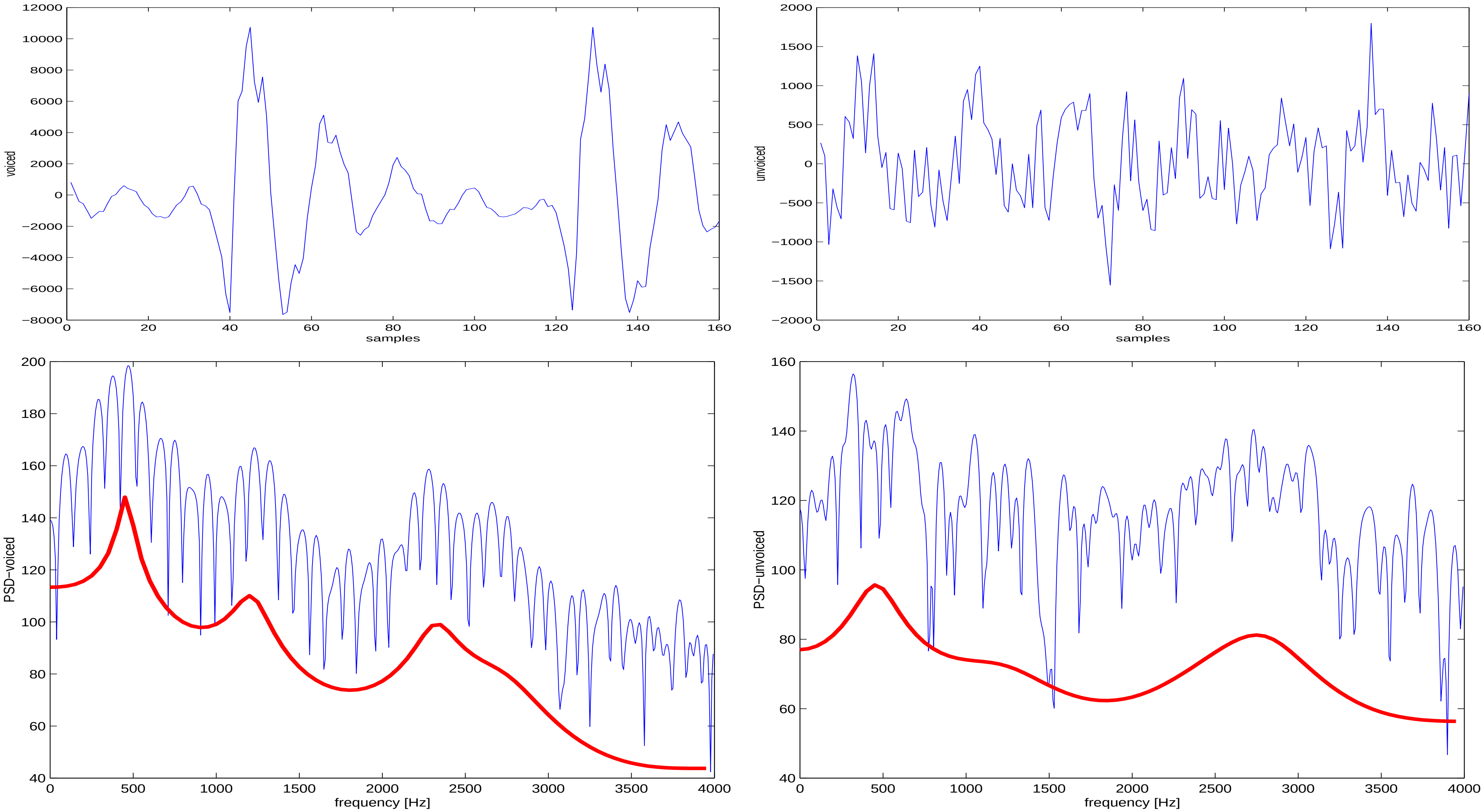
$$\hat{G}_{LPC} = \left| \frac{G}{A(z)} \right|^2_{z=e^{j2\pi f}}, \qquad (28)$$

where f is the normalized frequency $f = \frac{F}{F_s}$. After the substitution:

$$\hat{G}_{LPC} = \frac{G^2}{\left| 1 + \sum_{i=1}^{P} a_i e^{-j2\pi f i} \right|^2} \qquad (29)$$

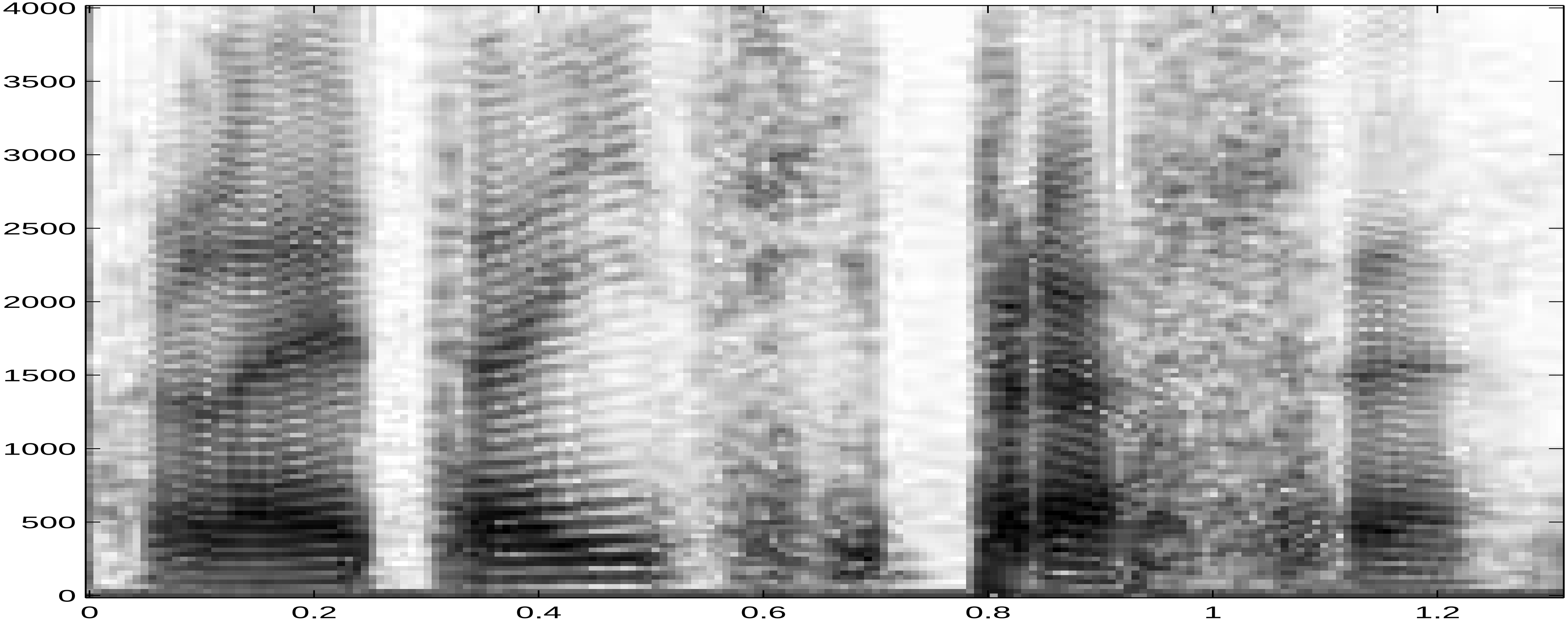
This PSD provides better formant resolution since the influence of the fundamental frequency is suppressed.

Example: PSD estimation using DFT and LPC on a voiced and an unvoiced frame.

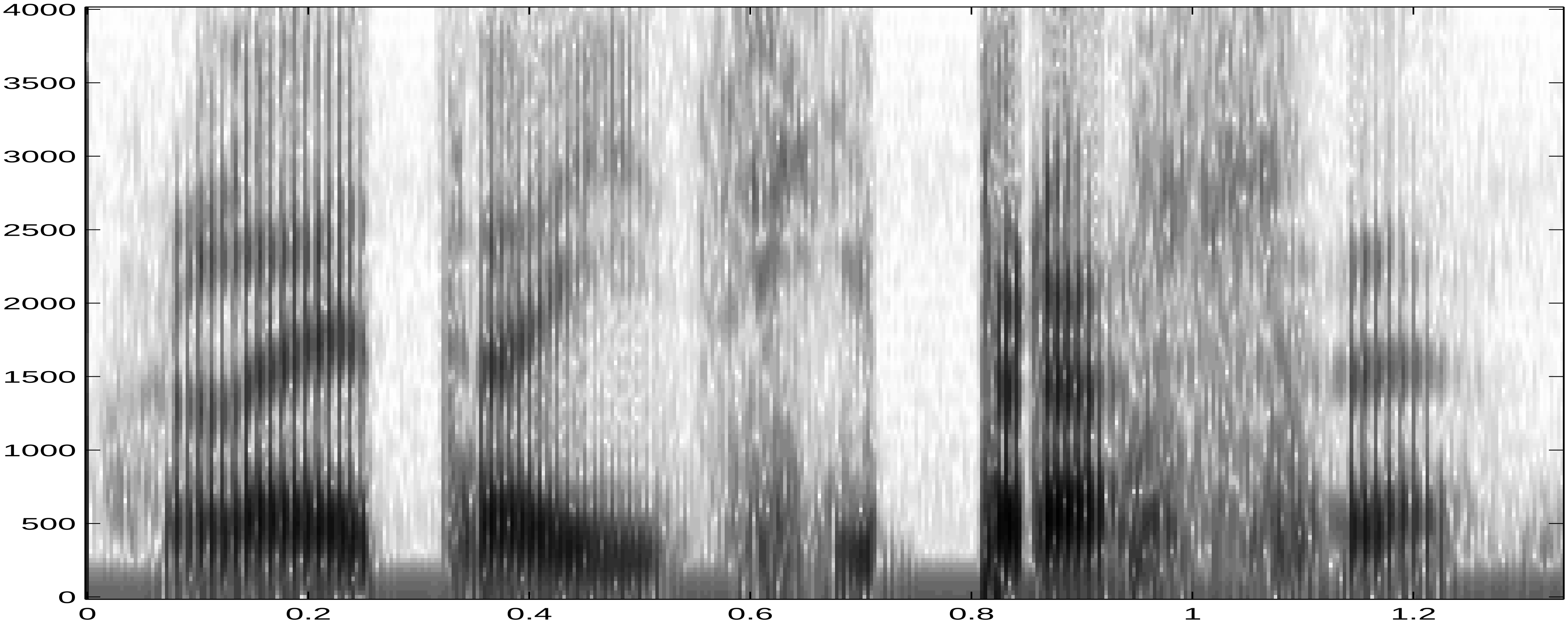


Spectrogram comparison

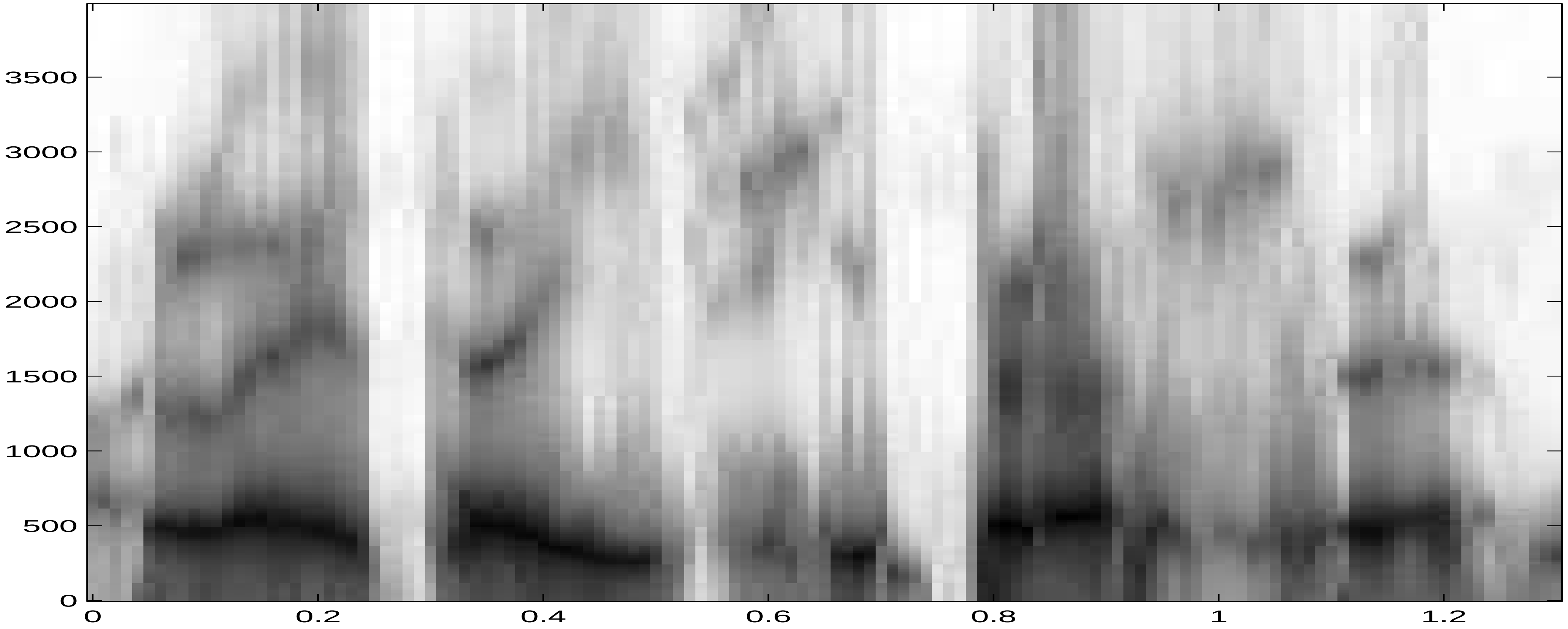
Long-term spectrogram: `specgram(s,256,8000,hamming(256),200);`



Short-term spectrogram: `specgram(s,256,8000,hamming(50));`



LPC spectrogram



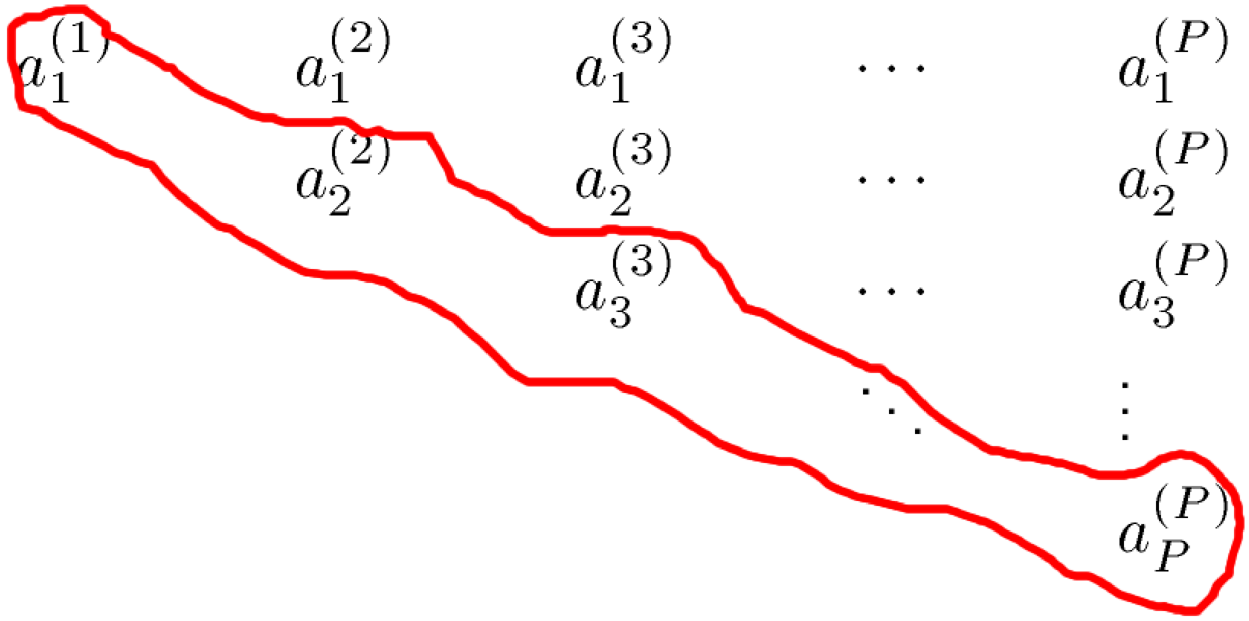
Parameters Derived from LPC Coefficients

Why ? Simple coefficients a_i are suitable for filtering, and thats about it:

- Difficult quantization (filter sensibility to the quantization error: $a_i \in < -\infty, +\infty >$).
- The coefficients are strongly correlated – not suitable for HMM base recognition.
- Distance between the coefficients a_i does not correspond to the similarity of the speech frames – cannot be used even in recognition based on direct parameter comparison (DTW).

⇒ Is there anything better?

PARCOR

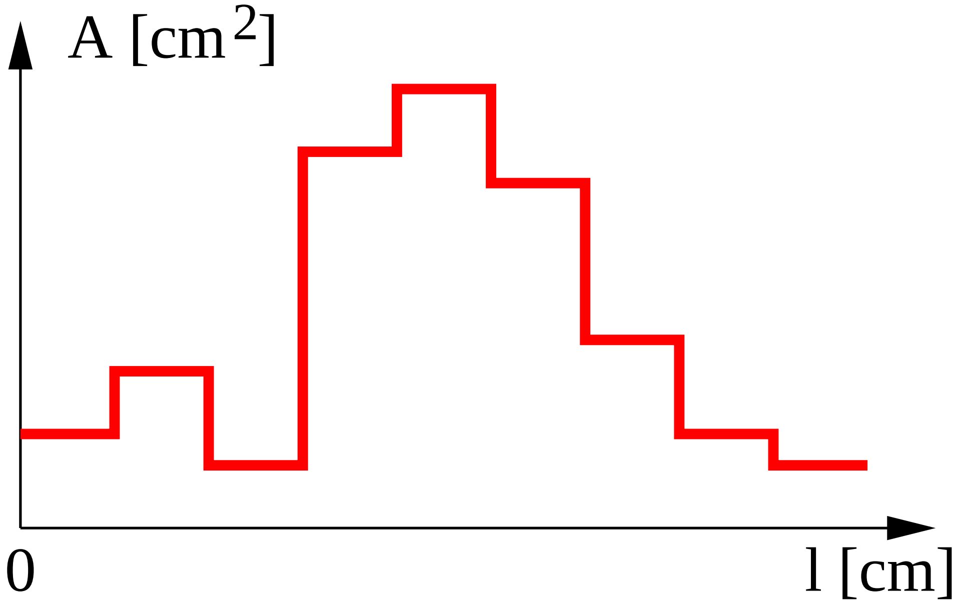


$$\begin{array}{ccccc}
a_1^{(1)} & a_1^{(2)} & a_1^{(3)} & \cdots & a_1^{(P)} \\
 & a_2^{(2)} & a_2^{(3)} & \cdots & a_2^{(P)} \\
 & & a_3^{(3)} & \cdots & a_3^{(P)} \\
 & & & \ddots & \vdots \\
 & & & & a_P^{(P)}
\end{array}$$

- Byproducts of the Levinson-Durbin algorithm: coefficients $k_i = a_i^{(i)}$ are denoted as *reflection coefficients* or *PARCOR coefficients* (partial correlation).
- It holds: $k_i \in < -1, 1 >$, are thus suitable for using in encoding in contrast to a_i.
- Coefficients a_i and k_i are mutually convertible.

Cylinder Model of the Vocal Tract

A vocal tract can be modeled by cylindric sections of the same length and variable diameter (thus different cuts):



Relationship of Cylinder Model to LPC

the ratio of the neighboring sections:

$$\frac{A_{m-1}}{A_m} = \frac{1+k_m}{1-k_m} \tag{30}$$

for $m = P, P-1, \ldots, 1$. Area A_P is fictive – we don't know the true value, thus put $A_P = 1$. Values $\frac{A_{m-1}}{A_m}$ are then area ratios (AR). Usually, logarithmic ratios are used – log area ratios (LAR):

$$g_m = \log \frac{A_{m-1}}{A_m} = \log \frac{1+k_m}{1-k_m} \tag{31}$$

Advantage of g_m over k_i is in linear sensibility of the spectrum. Linear quantifier g_m can be used. At the values k_i the spectrum is very sensitive to values $k_i \rightarrow 0$.

LSF or LSP

Line Spectrum Frequencies (LSF) or *Line Spectrum Pairs (LSP)*, are derived from the roots of the two polynomes:

$$\begin{aligned} M(z) &= A(z) - z^{-(P+1)}A(z^{-1}) \\ Q(z) &= A(z) + z^{-(P+1)}A(z^{-1}). \end{aligned} \tag{32}$$

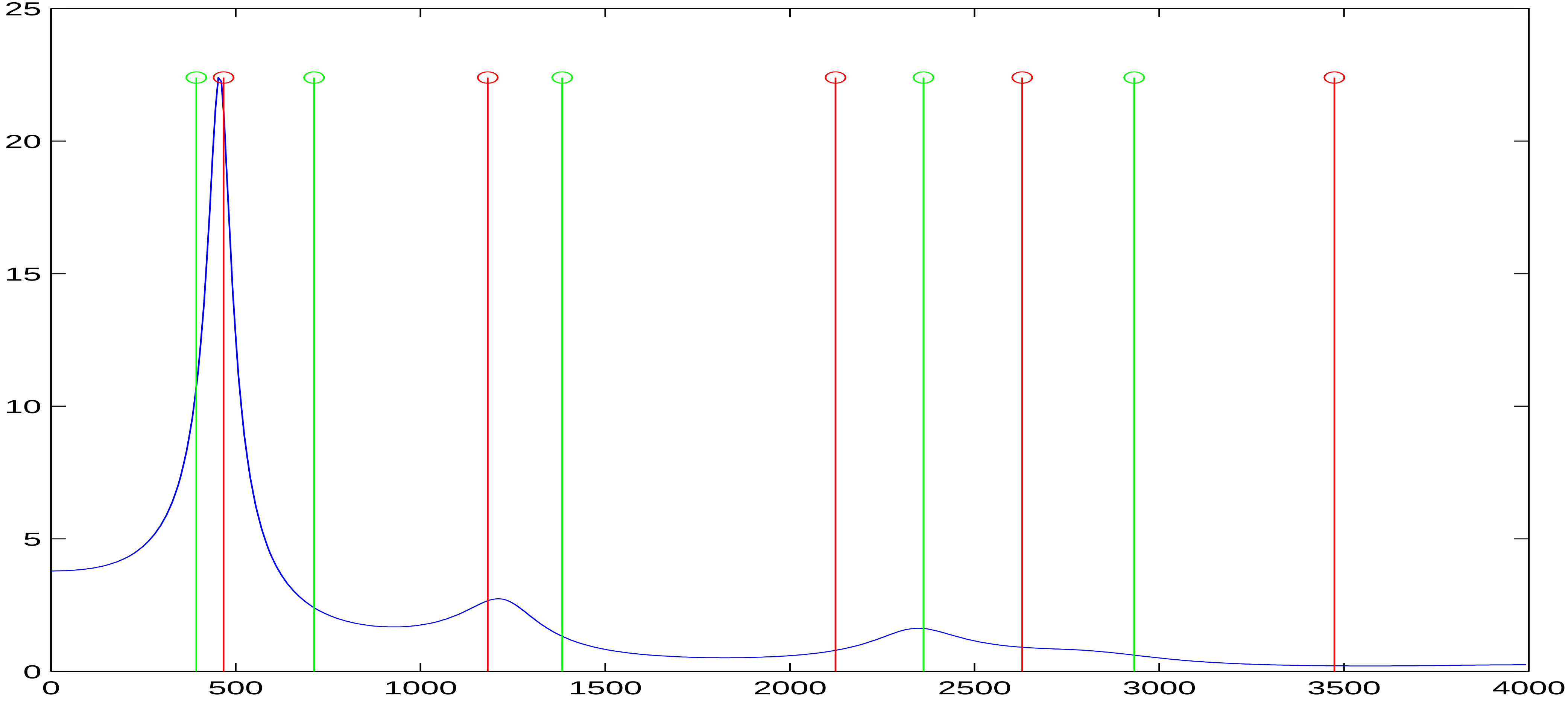
Can be rewritten using the roots:

$$\begin{aligned} M(z) &= (1 - z^{-1}) \prod_{i=2,4,\dots,P} (1 - 2z^{-1}\cos\omega_i + z^{-2}) \\ Q(z) &= (1 + z^{-1}) \prod_{i=1,3,\dots,P-1} (1 - 2z^{-1}\cos\omega_i + z^{-2}). \end{aligned} \tag{33}$$

where ω is the normalized angular frequency $\omega = 2\pi f$ (f is the "usual" frequency). Line spectral frequencies f_i lie within interval (0,0.5) and are sorted upward:

$$0 < f_1 < f_2 < \dots < f_{P-1} < f_P < \frac{1}{2}. \tag{34}$$

If we use LSFs (quantized) in a transfer function definition, we can test accuracy of the decoder by checking correct order of the frequencies.



LPC-cepstrum

Cepstral coefficients were so far calculated using DFT. Cepstrum can as well be estimated from PSD calculated using LPC, thus:

$$\hat{G}_{LPC}(f) = \left|\frac{G}{A(z)}\right|^2_{z=e^{j2\pi f}}, \tag{35}$$

where G is the gain of the filter and $A(z)$ is a polynome of the order P. Thus we talk about LPC-cepstrum (LPCC):

$$c(n) = \mathcal{F}^{-1}[\ln \hat{G}_{LPC}(f)] \tag{36}$$

We can derive the following properties of the LPC-cepstral coefficients:

$$c(0) = \ln G^2. \tag{37}$$

Zero cepstral coefficient carries information on the *energy* of the given speech frame. The following coefficients can be calculated from the LPC coefficients using recurrent relations:

$$\begin{aligned} c(n) &= -a_n - \frac{1}{n}\sum_{k=1}^{n-1} k c_k a_{n-k} && \text{for} \quad 1 \leq n \leq P \\ c(n) &= -\frac{1}{n}\sum_{k=1}^{n-1} k c_k a_{n-k} && \text{for} \quad n > P \end{aligned} \tag{38}$$

⇒ very simple estimation.

Use of LPCC coefficients

- LPCC coefficients are used in *speech recognition*. The advantage lies in weaker correlation than for instance in case of LPC coefficients, which means we can use diagonal covariance matrices $\mathbf{\Sigma}$ (vectors of standard deviations) in recognizers based on Hidden Markov models (HMMs).
- Given two sets of LPCC coefficients we can simply compute logarithmic spectral distance between two speech frames.