Lecture

Adaptive Filters

Lecture 4: Applications of Linear Prediction

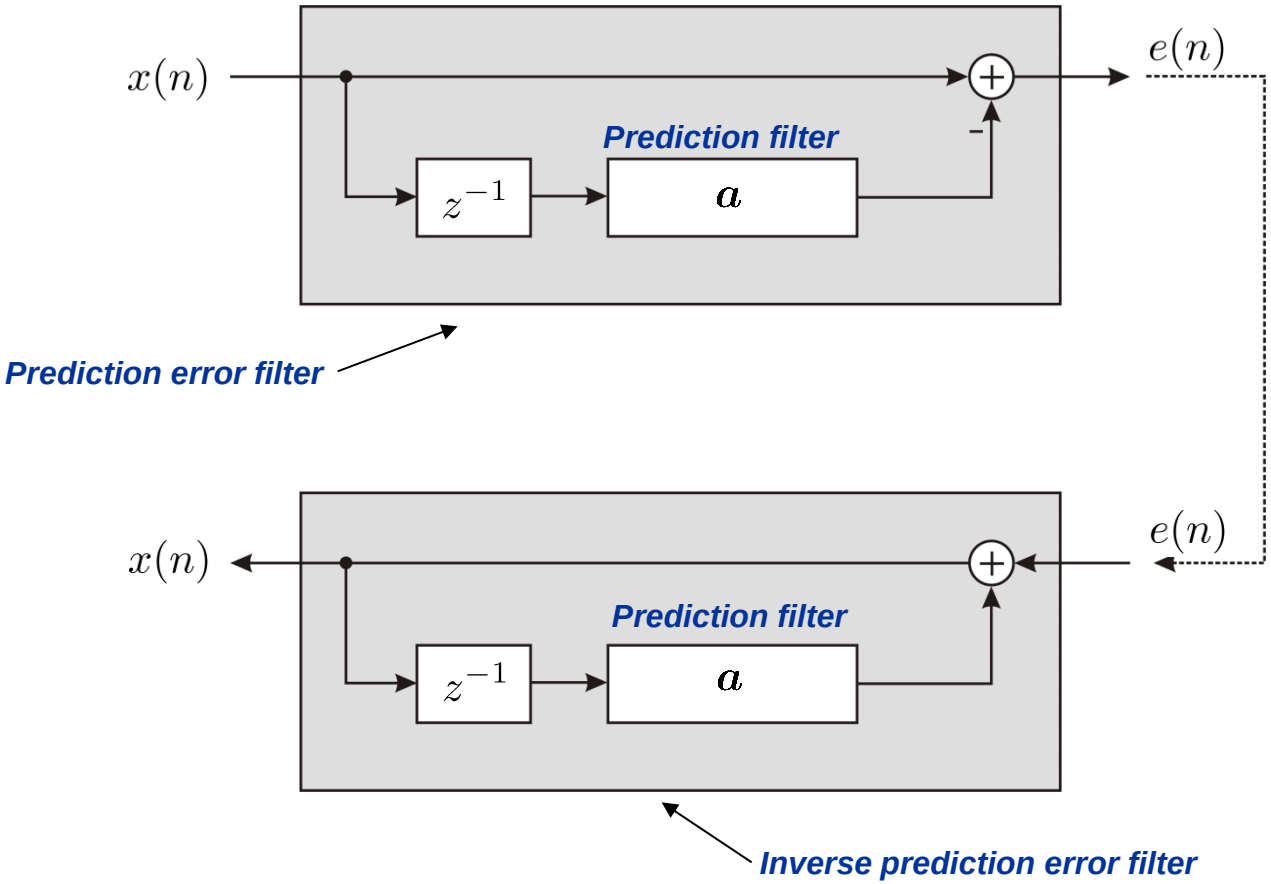
TECHNISCHE UNIVERSITÄT DARMSTADT

Linear Prediction – Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

- Repetition of Linear Prediction
- Properties of prediction filters
- Application examples
  - Codebook based audio coding
  - Speaker recognition
  - Filter design
  - Video coding

Linear prediction: Prediction error filtering & inverse filtering



Linear prediction: Filter design

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Cost function:

Minimization of the mean square error

$$\mathrm{E}\{e^2(n)\} \;\rightarrow\; \min$$

2. Solution:

Yule-Walker equation system:

$$\underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(N-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(N-1) & r_{xx}(N-2) & \cdots & r_{xx}(0) \end{bmatrix}^{-1}}_{\boldsymbol{R}_{xx}^{-1}} \underbrace{\begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(N) \end{bmatrix}}_{\boldsymbol{r}_{xx}(1)}$$

3. Robust and computationally efficient solution:

Levinson-Durbin recursion

Levinson-Durbin recursion - Overview

Initialization:

- Predictor: $a_1^{(1)} = \tilde{a}_1^{(1)} = r_{xx}(1)/r_{xx}(0)$
- Error signal power (minimum): $E_{\min}^{(0)} = r_{xx}(0)$

Recursion:

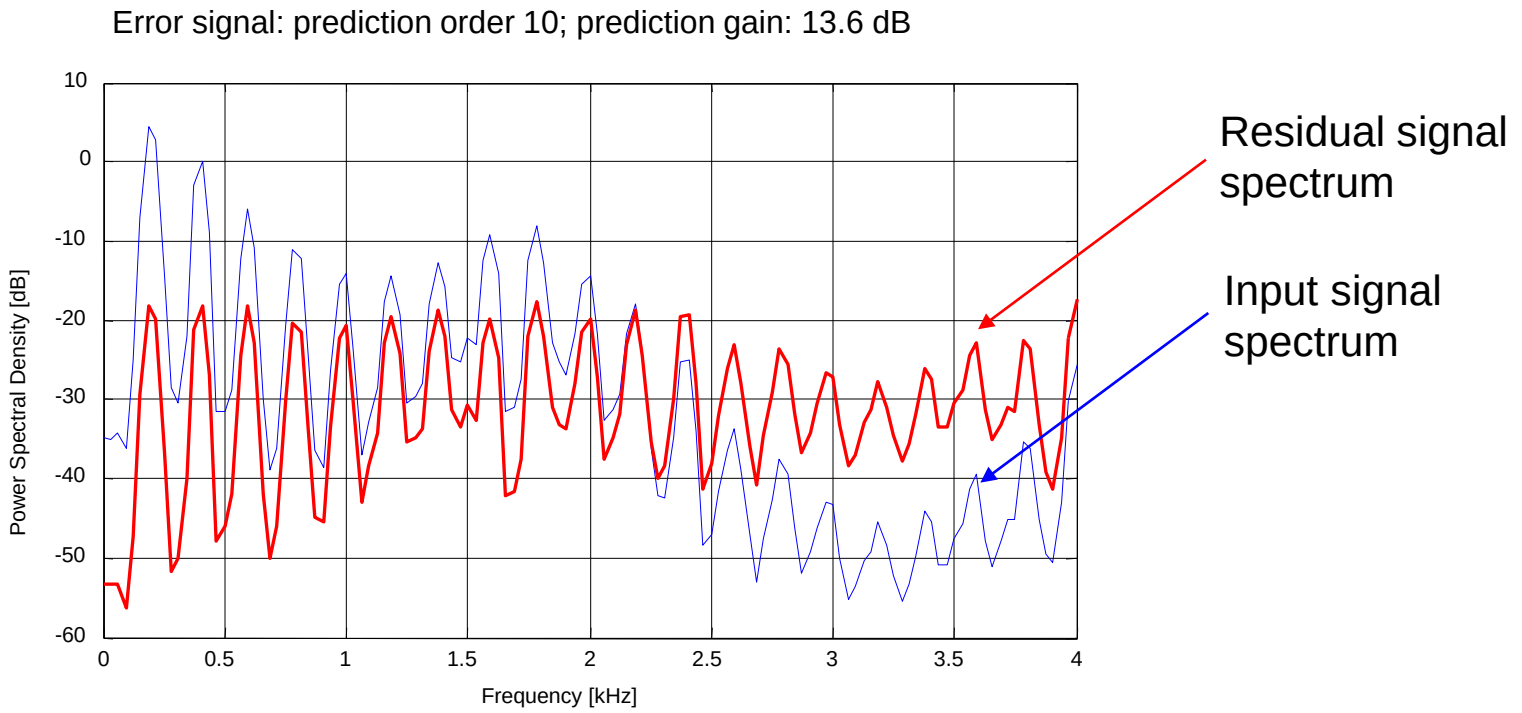
- Reflection coefficient: $a_{N+1}^{(N+1)} = \dfrac{r_{xx}(N+1) - \boldsymbol{r}_{xx}^{\mathrm{T}}(1)\,\tilde{\boldsymbol{a}}^{(N)}}{r_{xx}(0) - \boldsymbol{r}_{xx}^{\mathrm{T}}(1)\,\boldsymbol{a}^{(N)}}$
- Forward prediction: $[a_1^{(N+1)}\, a_2^{(N+1)} \cdots a_N^{(N+1)}]^T = \boldsymbol{a}^{(N)} - a_{N+1}^{(N+1)}\tilde{\boldsymbol{a}}^{(N)}$
- Backward prediction: $\tilde{a}_i^{(N)} = a_{N-i}^{(N)}$
- Error power (minimum): $\mathrm{E}_{\min}^{(N+1)} = \mathrm{E}_{\min}^{(N)}(1 - |a_{N+1}^{(N+1)}|^2)$

Stop criteria:

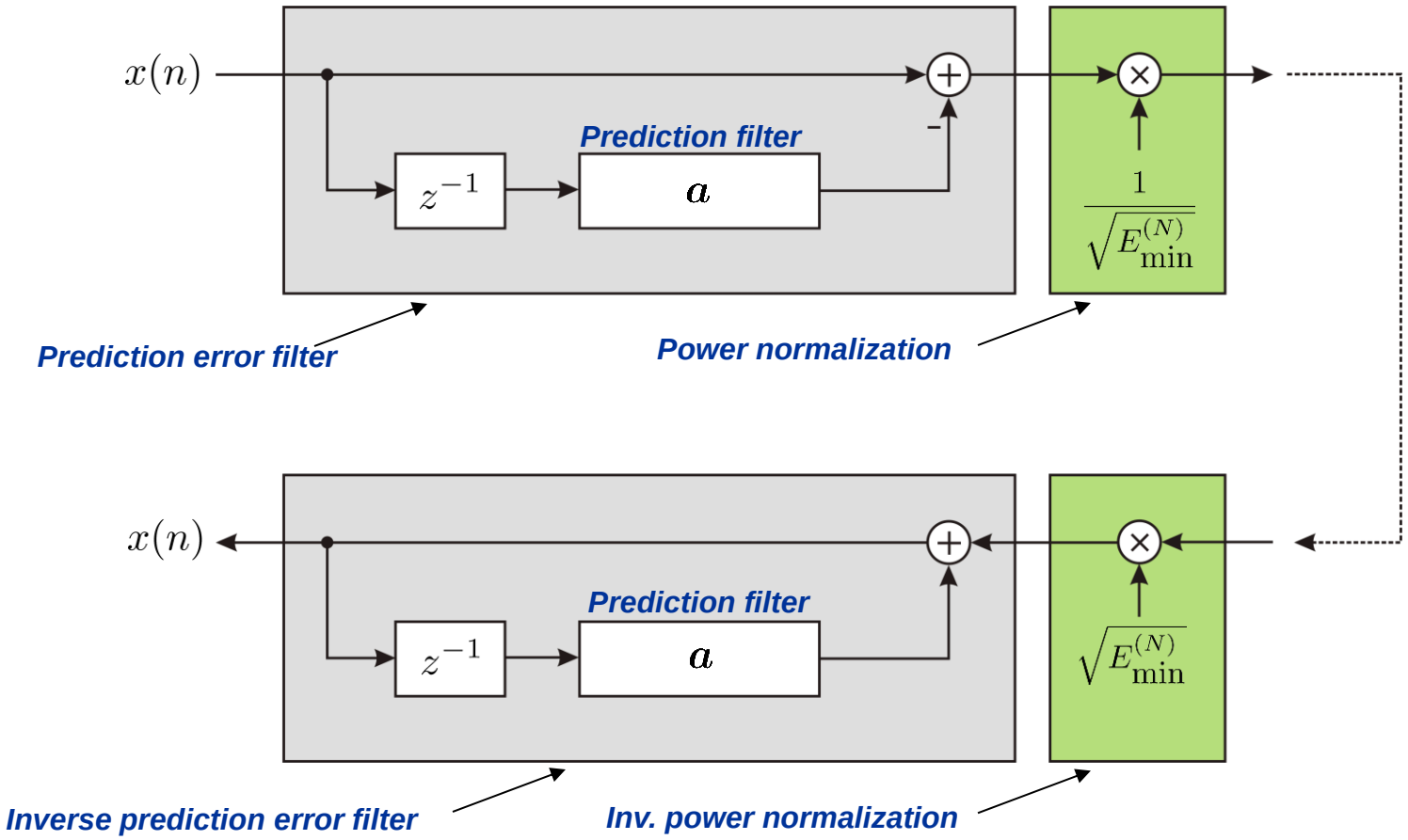
- Numeric criterion: When $|a_{N+1}^{(N+1)}|^2 < \epsilon$, use the previous recursion step and stop the recursion.
- Order criterion: In case N has reached the desired order => recursion stop.

Spectral properties of the prediction error filter (I)

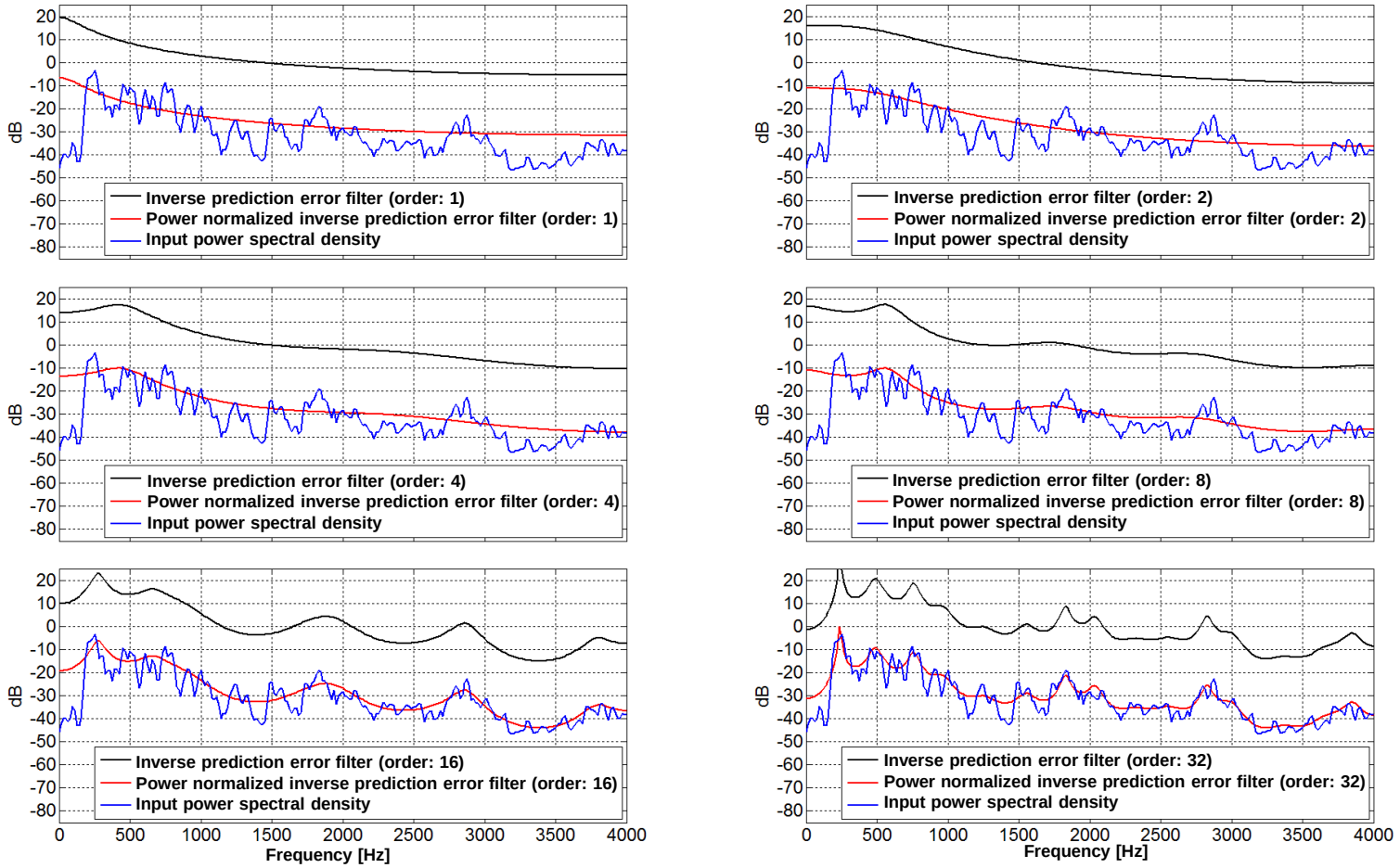
Signal whitening:



Spectral properties of the prediction error filter (II)



Spectral properties of the prediction error filter (III)



Properties of the prediction error filter (I)

TECHNISCHE UNIVERSITÄT DARMSTADT

Minimization without constraints:

- Cost function:

$$\mathrm{E}\{e^2(n)\} \;=\; \mathrm{E}\left\{\left(x(n) - \sum_{i=1}^{N} a_i\, x(n-i)\right)^2\right\} \;\rightarrow\; \min$$

The resulting prediction error filter is phase minimal (magn. of all reflection coefficients < 1):

- All nulls of the filter are located within the unit circle.
- All signals pass the filter with minimum latency.
- The inverse prediction error filter (IIR filter) is therefore stable, since all nulls of the prediction filter are now poles. These poles are thus located in the unit circle as well => stable filter.

Thus, normalized filters are generated:

- Frequency response of the prediction error filter:

$$H_{\mathrm{PEF}}\left(e^{j\Omega}\right) = 1 - \sum_{i=1}^{N} a_i\, e^{-j\Omega i}$$

- Frequency response of the inverse prediction error filter:

$$H_{\mathrm{inv.\,PEF}}\left(e^{j\Omega}\right) = \frac{1}{1 - \sum_{i=1}^{N} a_i\, e^{-j\Omega i}}$$

Inverse prediction error filter: estimation of the spectral envelope

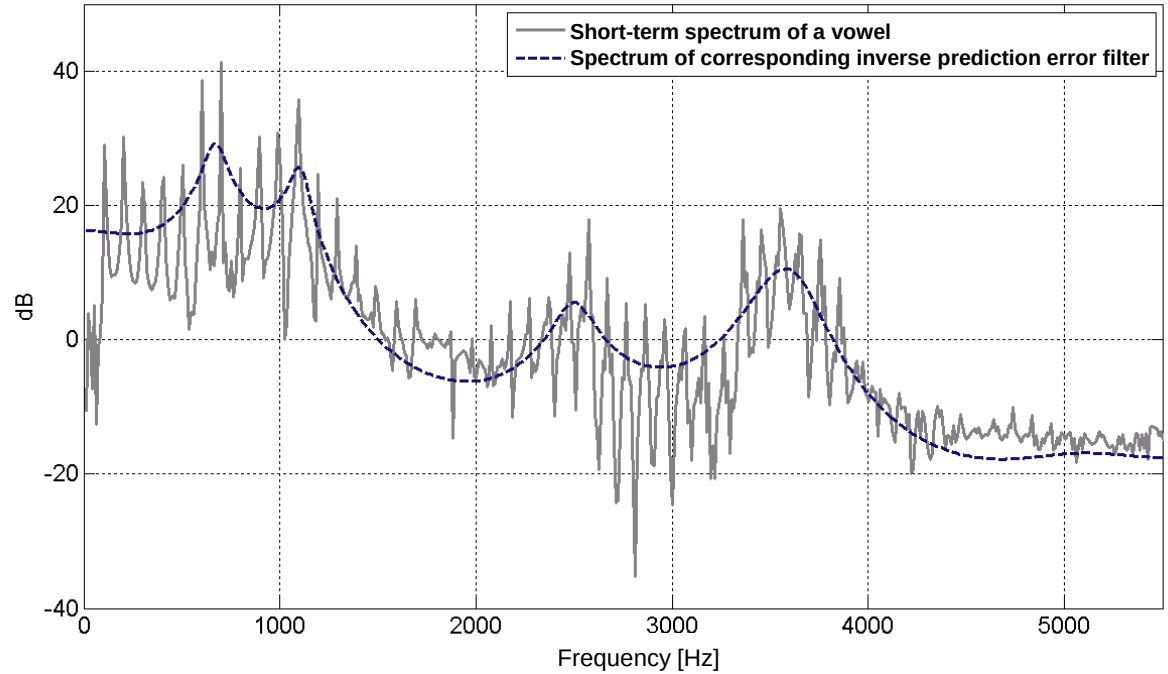
TECHNISCHE UNIVERSITÄT DARMSTADT

Parametric spectral estimation:

- Description of the spectrum by the inverse prediction error filter.

$$H_{\text{inv. PEF}}\left(e^{j\Omega}\right) = \frac{1}{1 - \sum_{i=1}^{N} a_i \, e^{-j\Omega i}}$$

- In case of limited order, only the envelope is characterized => determine formants spectral description without considering the pitch frequency.

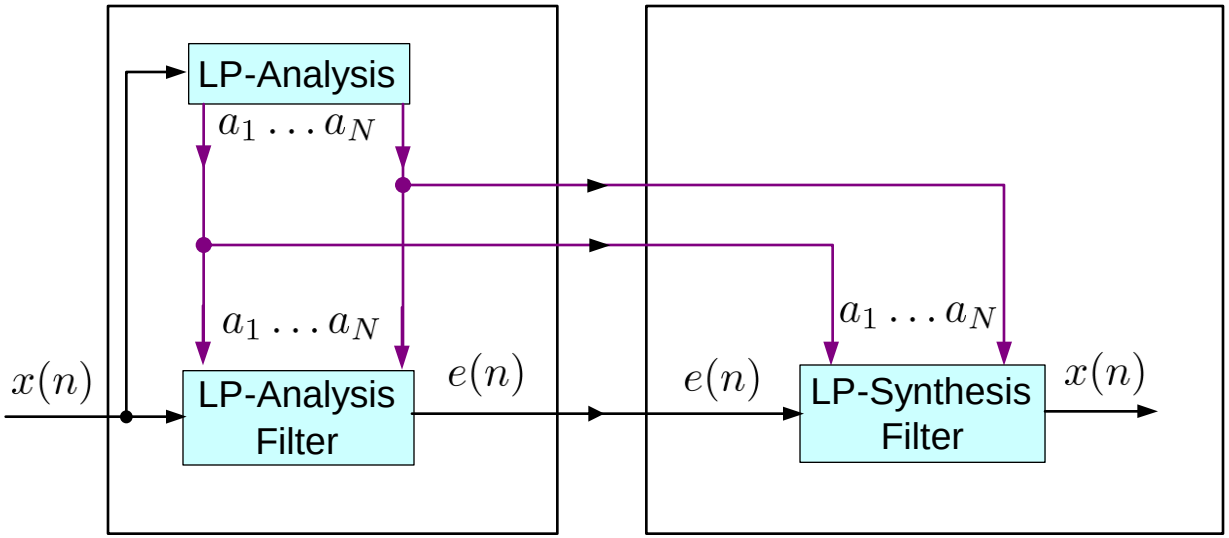


Application of prediction filtering – Part 1

Codebook based audio coding and decoding

Audio coding with linear prediction

- Determine prediction filter for consecutive blocks of the signal to code / transmit
- Determine residual signal => prediction error signal
- Transmit the residual signal and the prediction coefficients

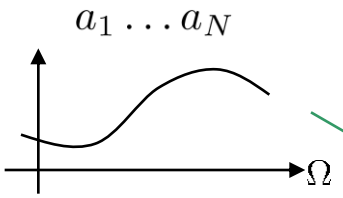


LP audio coding with codebooks for spectral envelopes

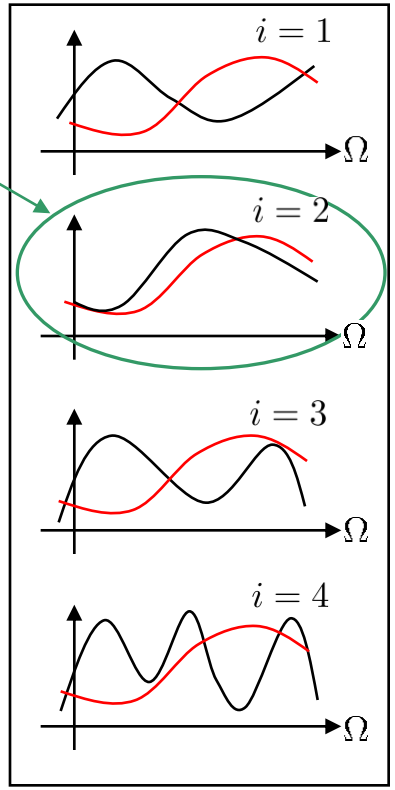
- For LP filter $a_1 \ldots a_N$ search for the best "fitting" codebook entry with index i and coefficients: $\tilde{a}_1 \ldots \tilde{a}_N$

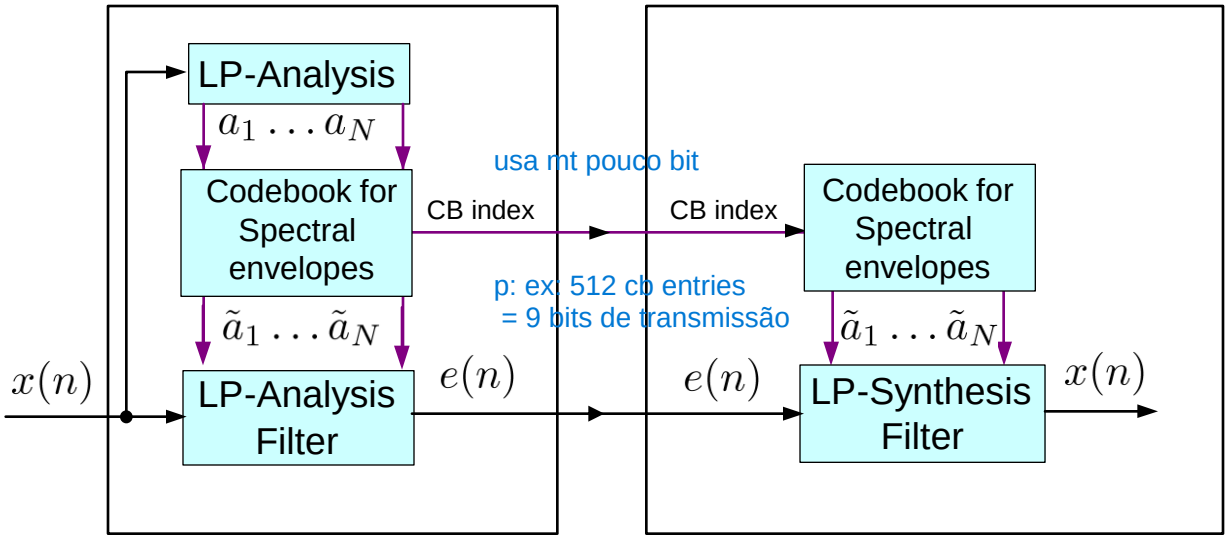
ideal em que senso? mudar para essa entrada no codebook?

Calculated opt. LP filter:

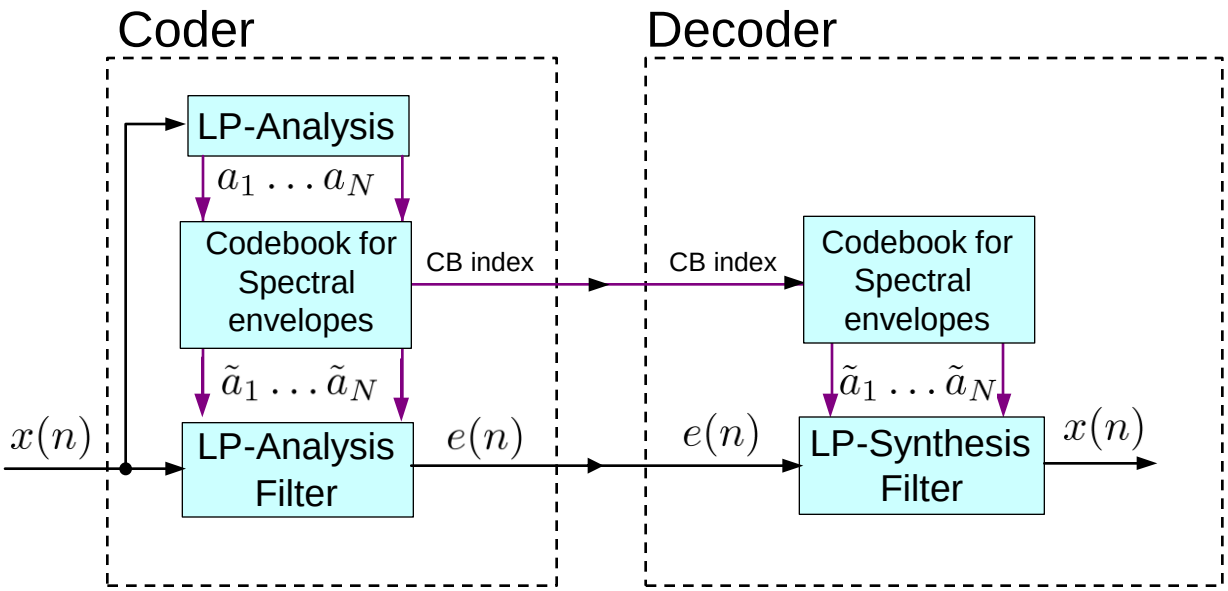


Codebook for spectral envelopes:





LP audio coding with codebooks for spectral envelopes



- Known linear prediction with codebook coding of prediction vector
- Next step: also codebook coding of the residual signal

Application of prediction filtering – Part 2

Speaker recognition

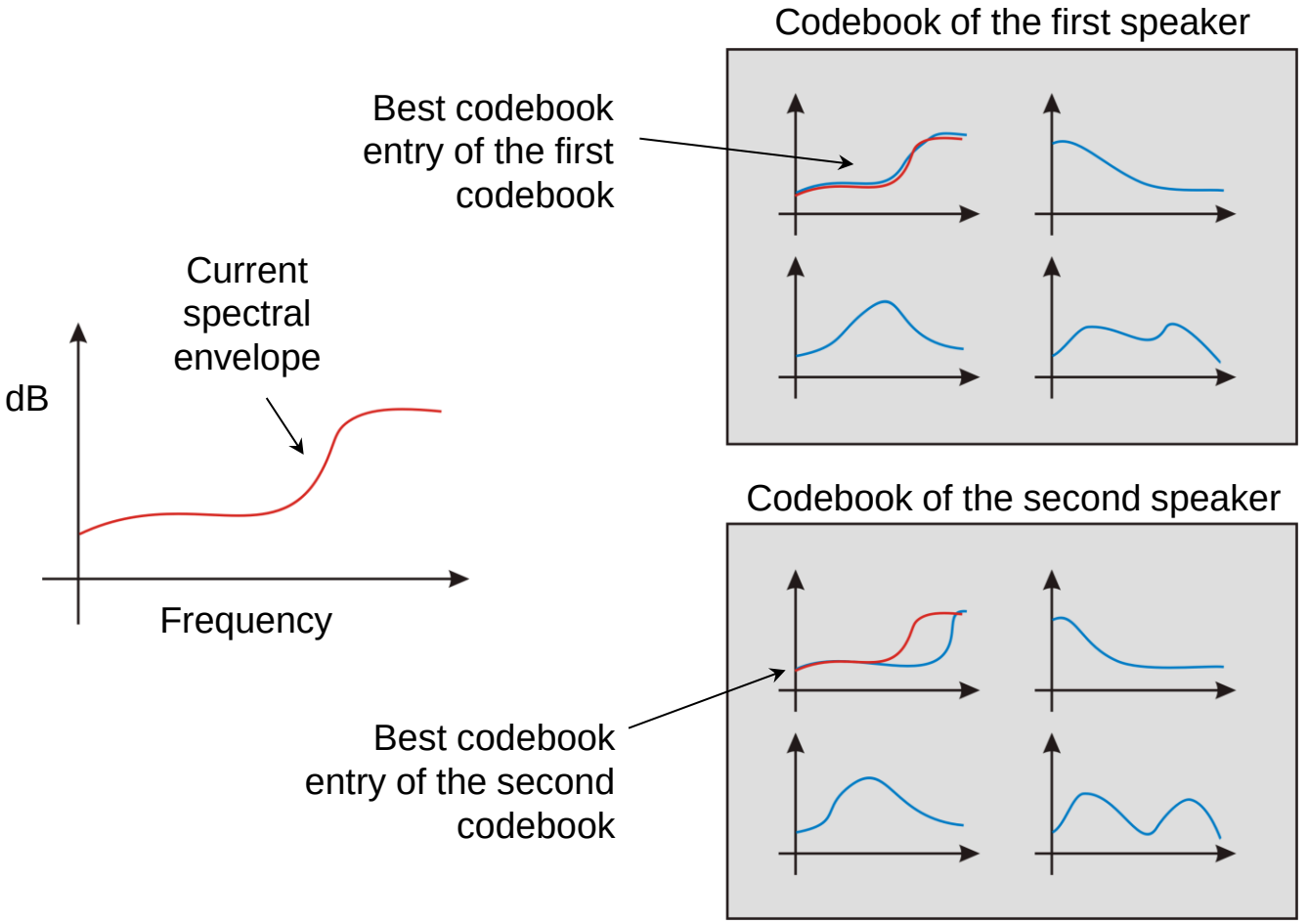
A simple speaker recognition

TECHNISCHE UNIVERSITÄT DARMSTADT

Principle:

- For speaker recognition, first relevant signal characteristics are extracted, e.g. the spectral envelope. Performed for block lengths of 5 to 30 ms.
- Then, comparison of the calculated spectral envelope with codebook entries.
- The comparison is performed for several codebooks, each corresponding to a specific speaker.
- The minimum distances for each codebook are determined.
- The minimum distance (summed over several blocks) determines the speaker which is detected.
- The model of the known speaker competes against universal models.
- Generally an adjustment of the “winner“ codebook is performed.

A simple speaker recognition



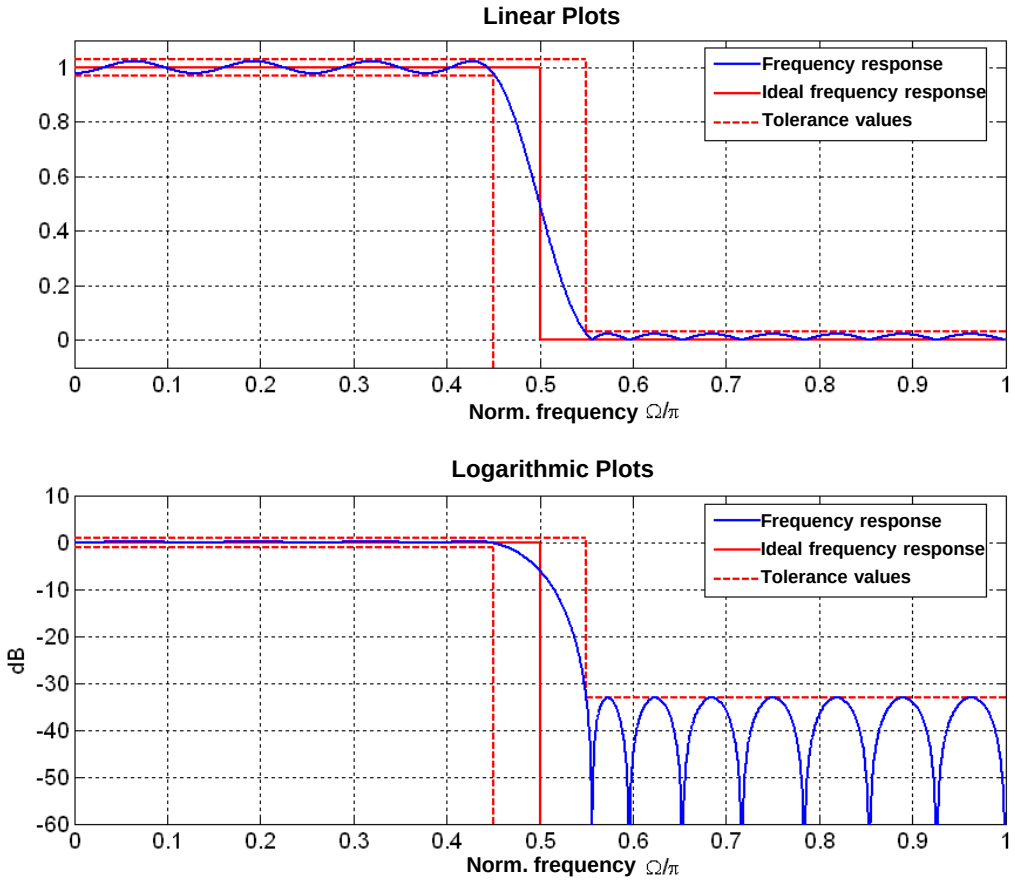
Application of prediction filtering – Part 3

Filter design

Filter design (I)

Definition of an ideal filters with tolerances values:

- Mostly high-pass, low-pass filters and band-pass, band-stop filters are designed.
- The solution is typically found with iterative procedures (also available in Matlab).
- FIR and IIR filters are designed.

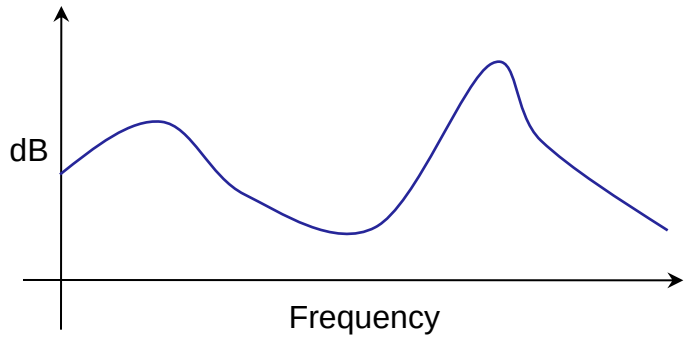


Filter design (II)

TECHNISCHE UNIVERSITÄT DARMSTADT

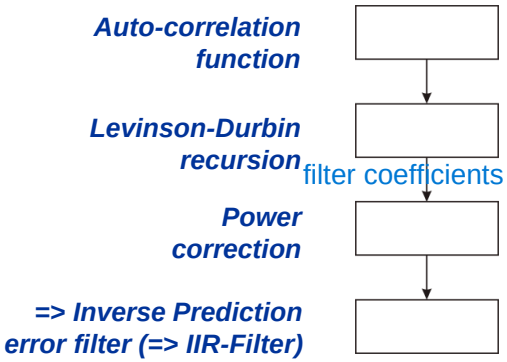
… but what to do when:

- A filter with an arbitrary frequency response should be designed (possibly online during real-time processing).

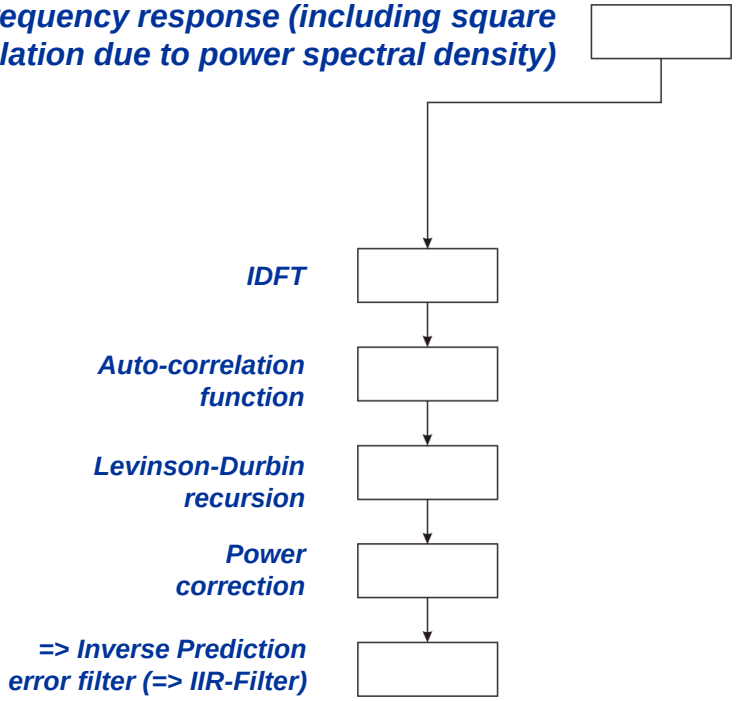


- The filter should have either IIR or FIR structure.
- The IIR filter should be stable and the FIR filter phase minimal
- The design should be done computationally efficient.

Filter design with prediction methods (I)

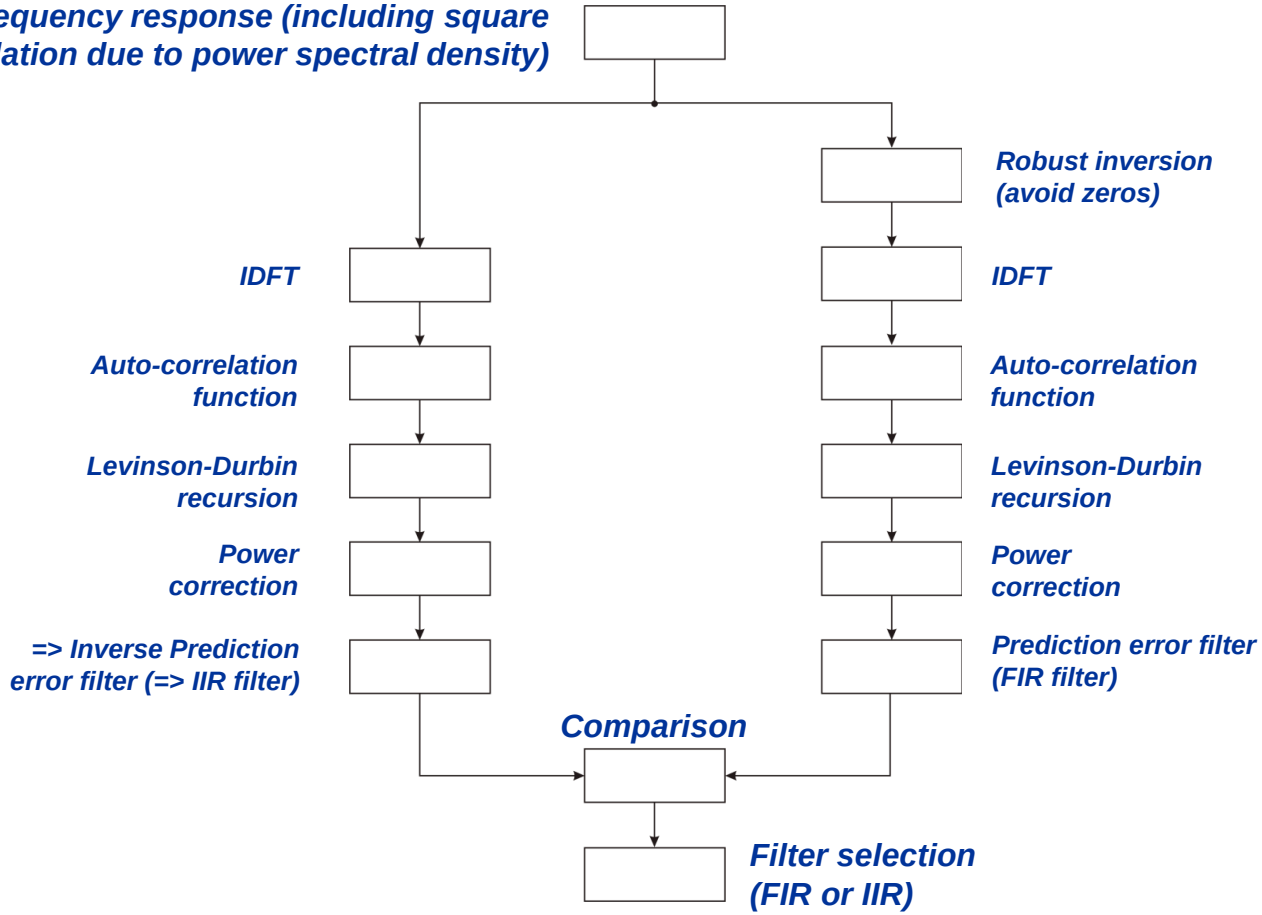


Filter design with prediction methods (II)



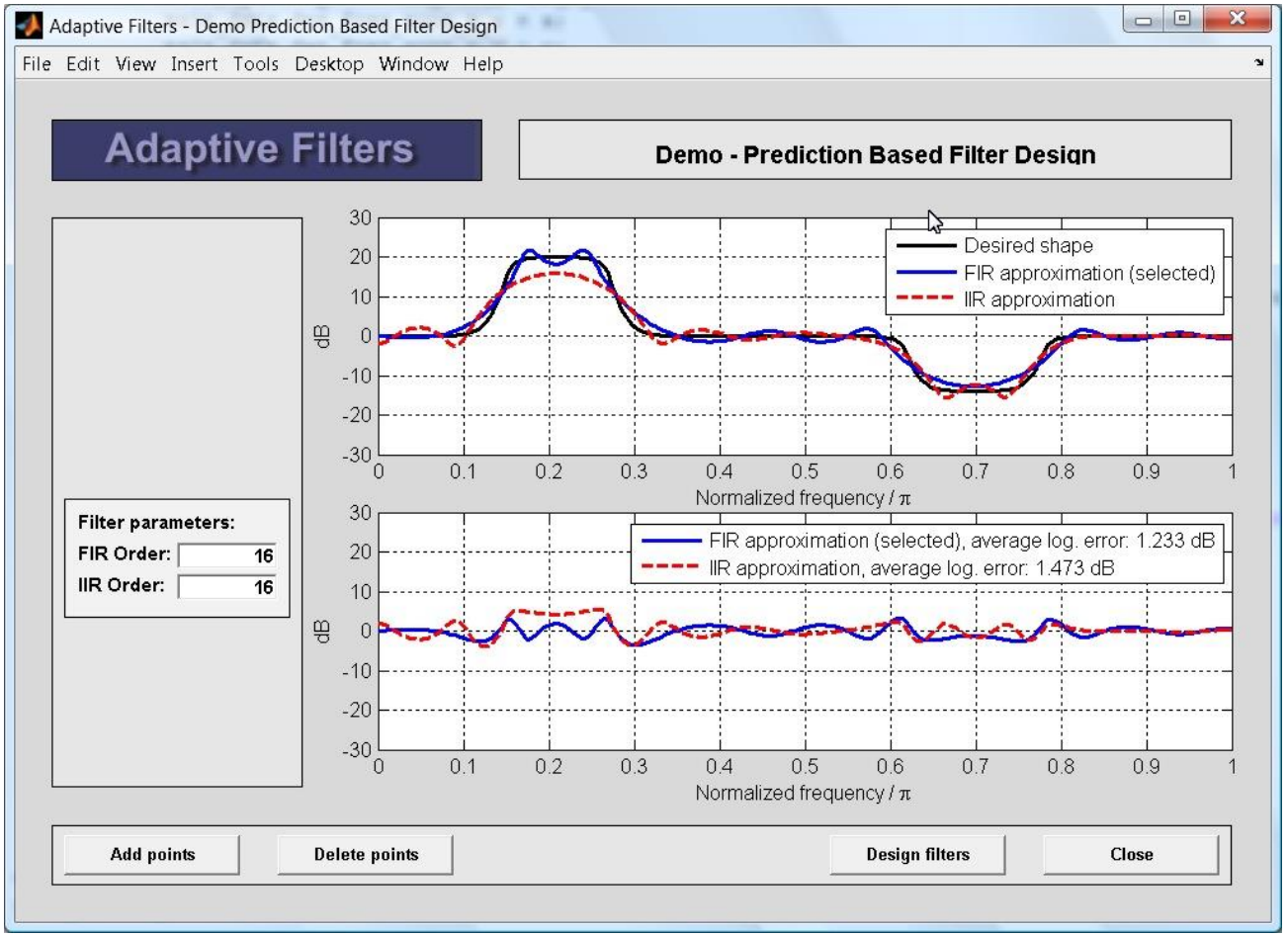
vai conseguir entender o que na frequencia gerou a sequencia no tempo dps de IDFT

Filter design with prediction methods (III)



Design examples

esse programa em matlab é mt massa
mostra bem que o iir é bom pra calcular um vale, e o fir é bom pra modelar um pico. - mas pra uma ordem grande, os 2 funcionam decentemente



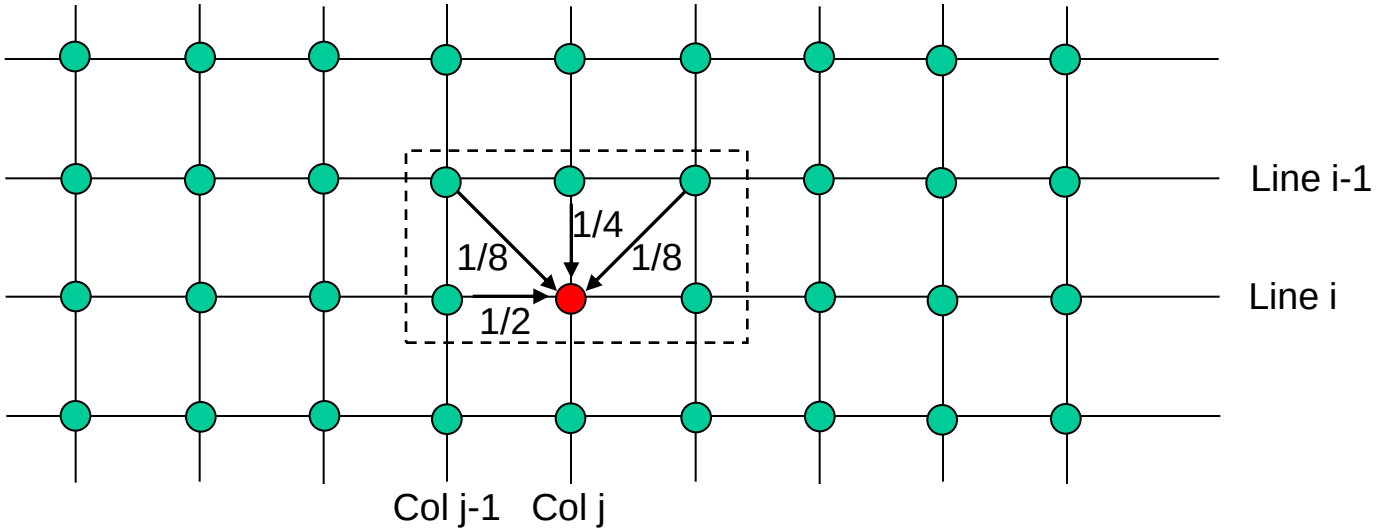
Application of prediction filtering – Part 4

Video coding

aplicação bem interessante na vdd

Video coding – Intra-frame prediction

- **Video signal:**
  - A two dimensional signal with lines and columns:



  - Prediction on previous line and previous value (intra-frame prediction):

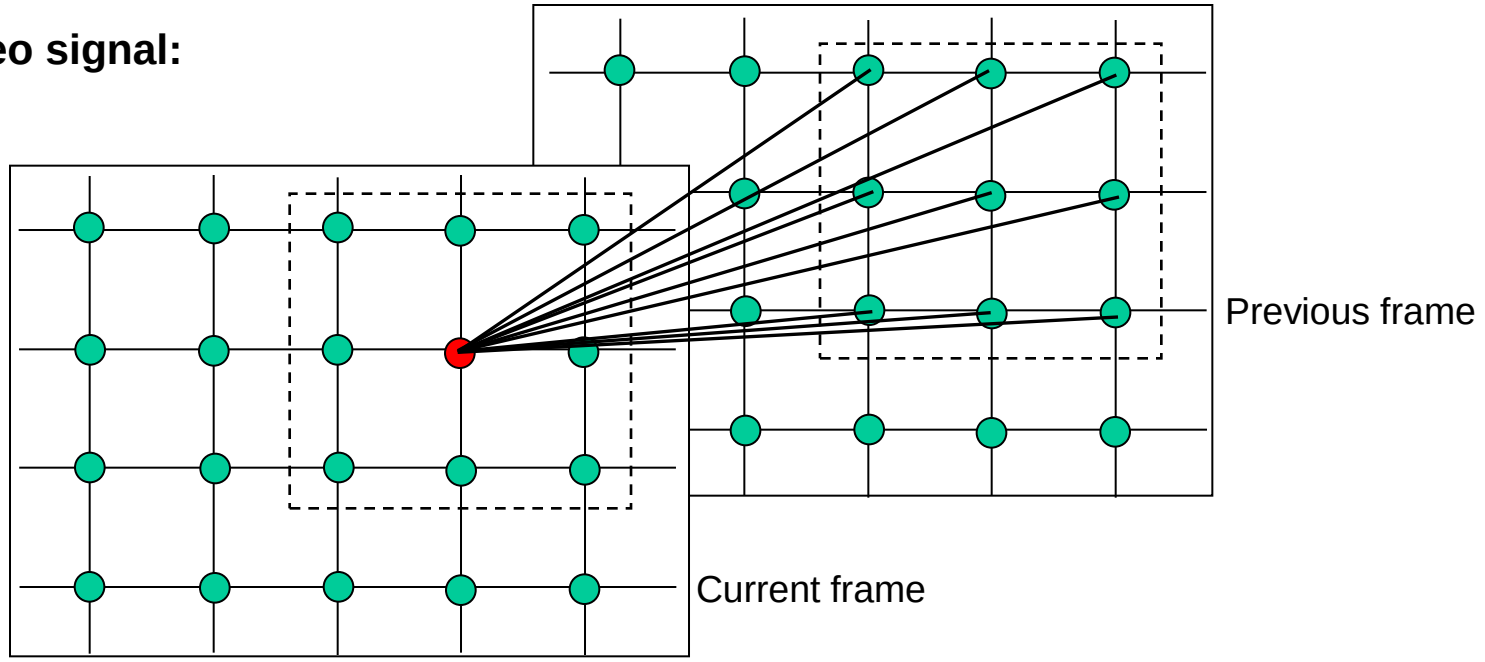
$$\boldsymbol{X}_{i,j}(n) = \begin{bmatrix} x_{i-1,j-1}(n) & x_{i-1,j}(n) & x_{i-1,j+1}(n) \\ x_{i,j-1}(n) & x_{i,j}(n) & x_{i,j+1}(n) \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1/8 & 1/4 & 1/8 \\ 1/2 & 0 & 0 \end{bmatrix}$$

$$\hat{x}_{i,j}(n) = \boldsymbol{A} \circ \boldsymbol{X}_{i,j}(n) \quad \text{Element-wise multiplication and summation}$$

$$= 1/8\, x_{i-1,j-1}(n) + 1/4\, x_{i-1,j}(n) + 1/8\, x_{i-1,j+1}(n) + 1/2\, x_{i,j-1}(n)$$

Video coding – Inter-frame prediction

- **Video signal:**

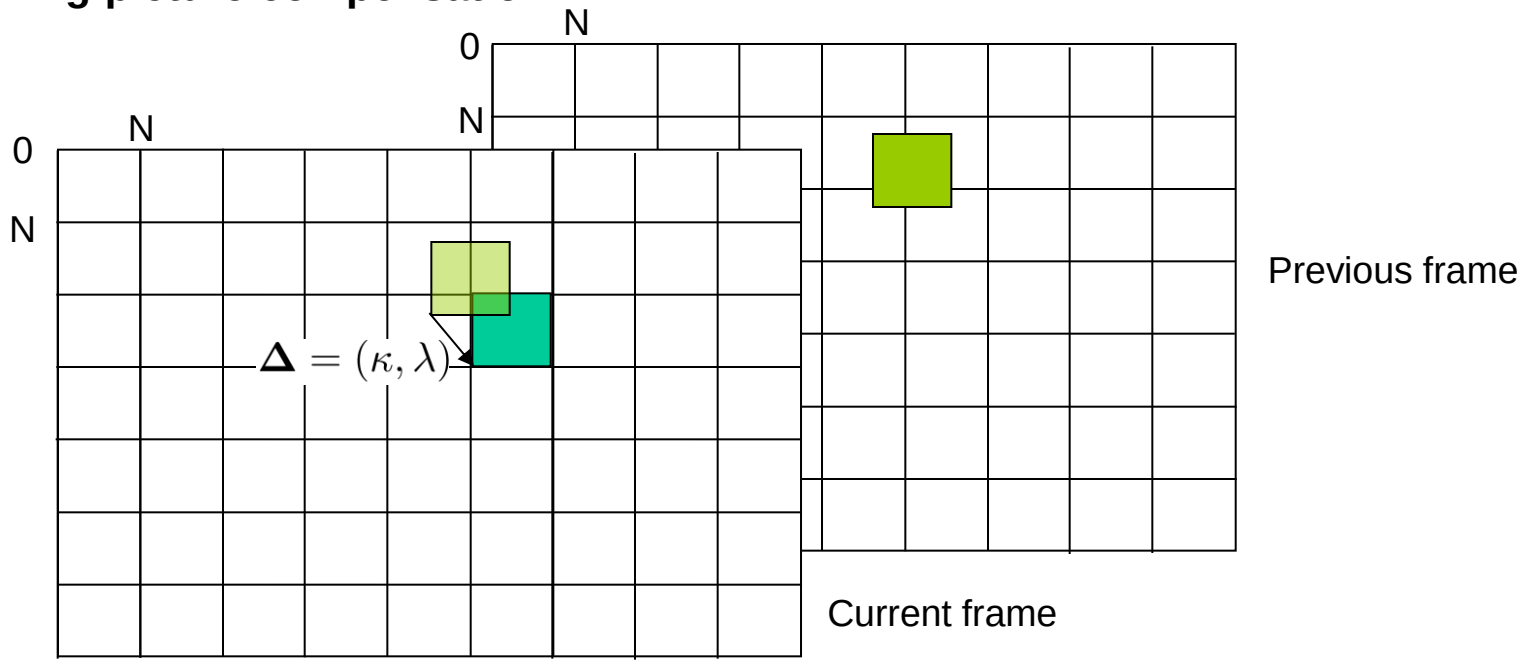


- Prediction on previous frame (inter-frame prediction):

$$\hat{x}_{i,j}(n) = \sum_{\mu=-1}^{+1} \sum_{\nu=-1}^{+1} a_{\mu+2,\nu+2}\; x_{i+\mu,j+\nu}(n-1) \qquad \boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Video coding – Inter-frame prediction

- **Moving picture compensation:**



In an NxN block coding:

Find corresponding block (translation by $\boldsymbol{\Delta} = (\kappa, \lambda)$) which shows the smallest distance δ.

$$\delta(\kappa, \lambda) = \sum_{\mu=1}^{N} \sum_{\nu=1}^{N} (x_{i,j}(n) - x_{i+\kappa, j+\lambda}(n-1))^2$$

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Moving picture compensation:**
  - Prediction on previous frame (inter-frame prediction) with translation by Δ:

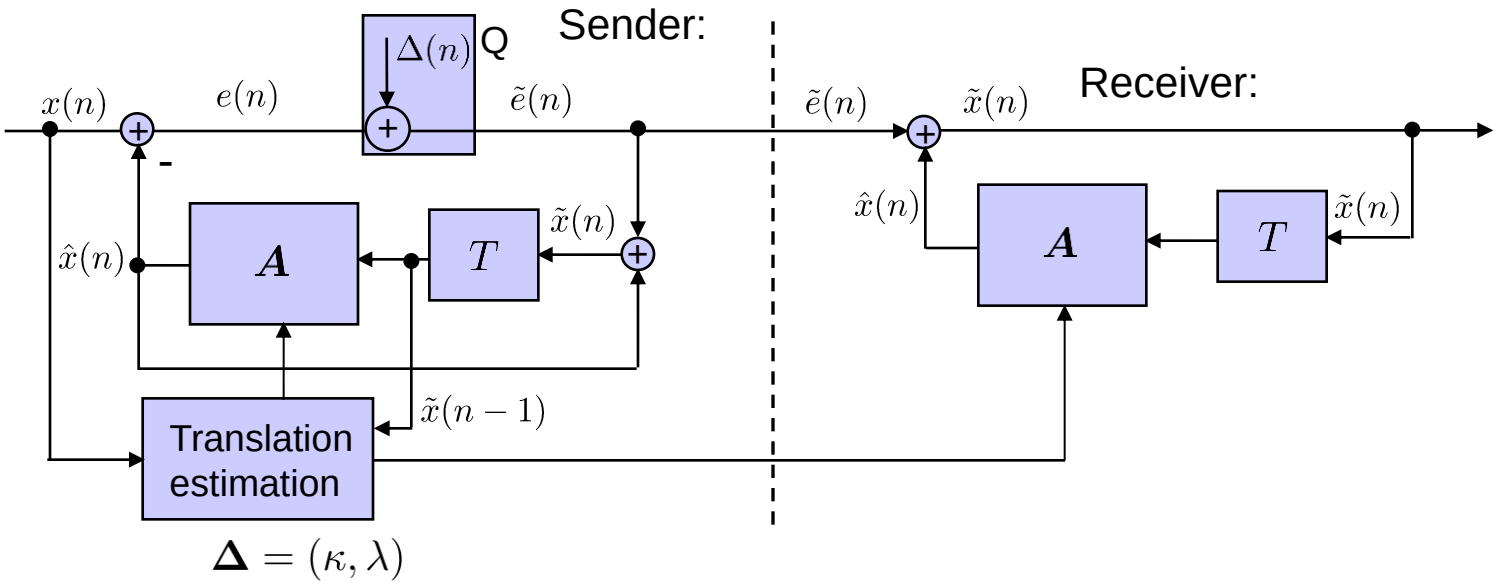
$$\hat{x}_{i,j}(n) = \sum_{\mu=-1}^{+1} \sum_{\nu=-1}^{+1} a_{\mu+2,\nu+2}\; x_{i+\mu+\kappa,j+\nu+\lambda}(n-1) \qquad \boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\boldsymbol{\Delta} = (\kappa, \lambda)$$

  - Application of the same translation within the block NxN.

Coding and decoding structure

- **Closed-loop structure**
  (prediction and translation estimation on the quantized signal):



Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

This week:

- Several examples of linear prediction, exploring the two main properties of linear prediction error filters:
  - Signal power reduction
  - Spectral envelope modeling
- Applications:
  - Codebook based audio coding
  - Speaker recognition
  - Filter design
  - Video coding

Next week:

- Adaptive filters: Our first procedure: Recursive Least Squares (RLS)

tutoriall - 1b - plot de 3 pontos

ree(l) = sum(1-N)sum(1-N) hpef(i) hpef(j) rxx(i-j + l)

da 1 aula: ree(0) = E{|e(n)|²} = hpefT * Rxx * hpef