Adaptive Filters – Application of Linear Prediction

Gerhard Schmidt

Christian-Albrechts-Universität zu Kiel
Faculty of Engineering
Electrical Engineering and Information Technology
Digital Signal Processing and System Theory

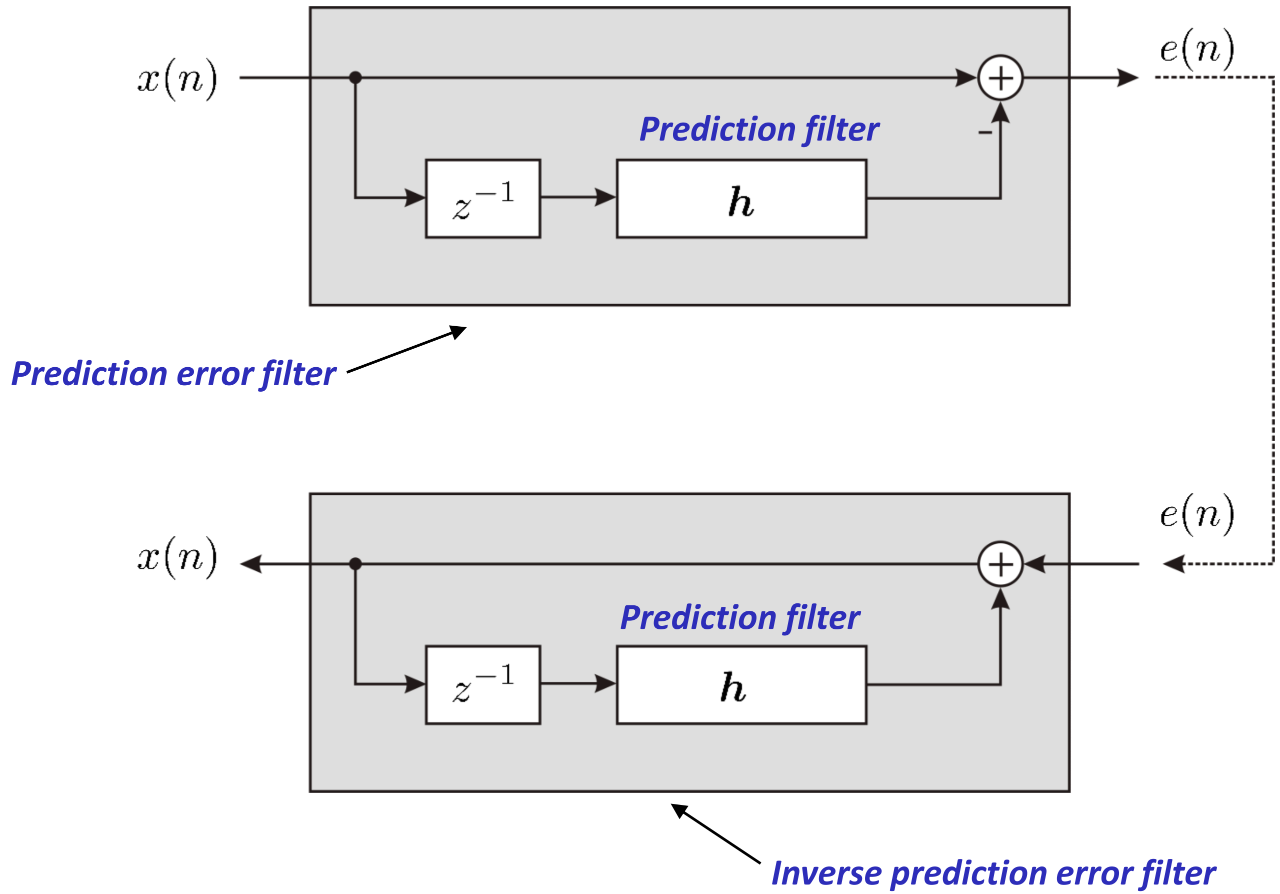
Contents of the Lecture

Today:

- Repetition of linear prediction
- Properties of prediction filters
- Application examples
  - Improving the convergence speed of adaptive filters
  - Speech and speaker recognition
  - Filter design

C | A | U
Christian-Albrechts-Universität zu Kiel

Structure Consisting of an Prediction Filter and of an Inverse Prediction Filter



Design of a Prediction Filter

Cost function:

Minimizing the mean squared error

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

Solution:

Yule-Walker equation system

$$\underbrace{\begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(N) \end{bmatrix}}_{\boldsymbol{r}_{ss}(1)} = \underbrace{\begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r(1) & r(0) & \dots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(N-1) & r(N-2) & \dots & r(0) \end{bmatrix}}_{\boldsymbol{R}_{ss}} \underbrace{\begin{bmatrix} h_{\mathrm{opt},0} \\ h_{\mathrm{opt},1} \\ \vdots \\ h_{\mathrm{opt},N-1} \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{opt}}}$$

Robust and efficient implementation:

Levinson-Durbin recursion

Levinson-Durbin Recursion

Initialization:

- Predictor: $$h_0^{(1)} = \tilde{h}_0^{(1)} = r(1)/r(0)$$
- Error power (optional): $$E_{\min}^{(0)} = r(0)$$

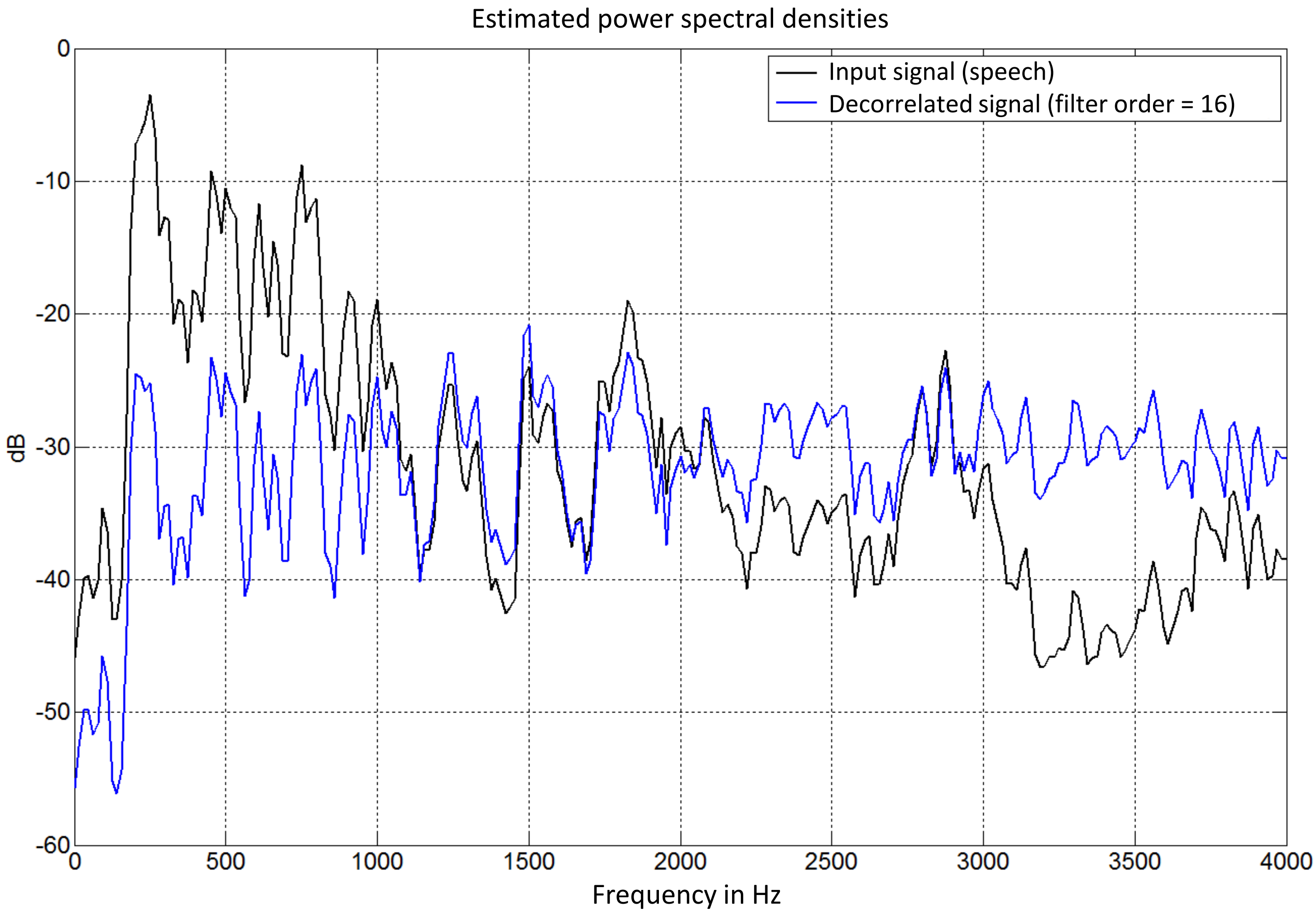
Recursion:

- PARCOR coefficient: $$h_{N-1}^{(N)} = \frac{r(N) - \left(\tilde{\boldsymbol{r}}_{ss}^{(N-1)}(1)\right)^{\mathrm{T}} \boldsymbol{h}_{\mathrm{opt}}^{(N-1)}}{r(0) - \left(\tilde{\boldsymbol{r}}_{ss}^{(N-1)}(1)\right)^{\mathrm{T}} \tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}}$$
- Forward predictor: $$\left[h_0^{(N)},\, h_1^{(N)},\, \ldots,\, h_{N-2}^{(N)}\right]^{\mathrm{T}} = \boldsymbol{h}_{\mathrm{opt}}^{(N-1)} - h_{N-1}^{(N)}\, \tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}$$
- Backward predictor: $$\tilde{h}_i^{(N)} = h_{N-i-1}^{(N)}$$
- Error power (optional): $$E_{\min}^{(N)} = E_{\min}^{(N-1)} \left(1 - \left(h_{N-1}^{(N)}\right)^2\right)$$

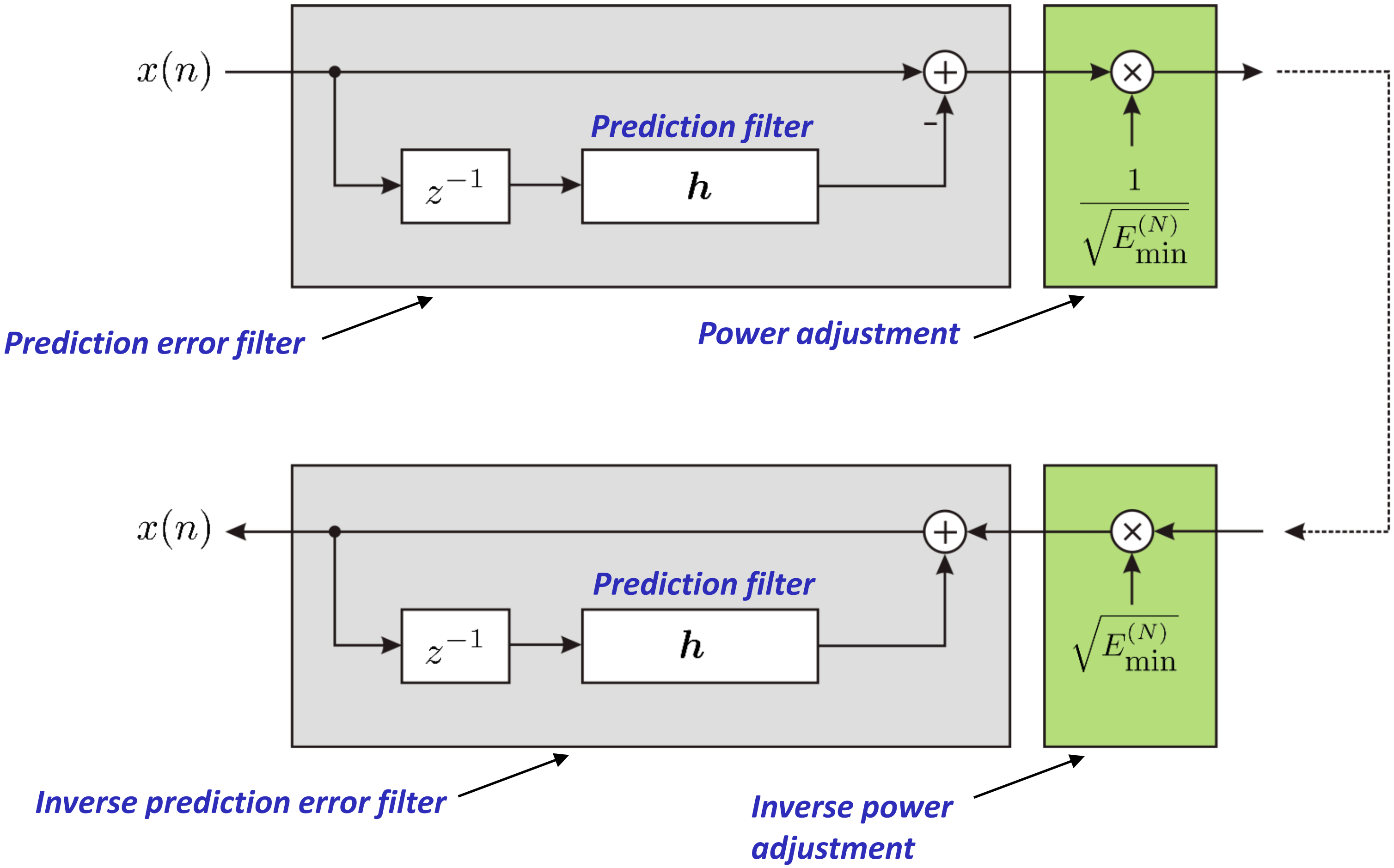
Termination:

- Numerical problems: If $\left(h_{N-1}^{(N)}\right)^2 > 1 - \varepsilon$, use the coefficients of the previous step and stop the recursion.
- Final order reached: If N has reached the desired order stop the recursion.

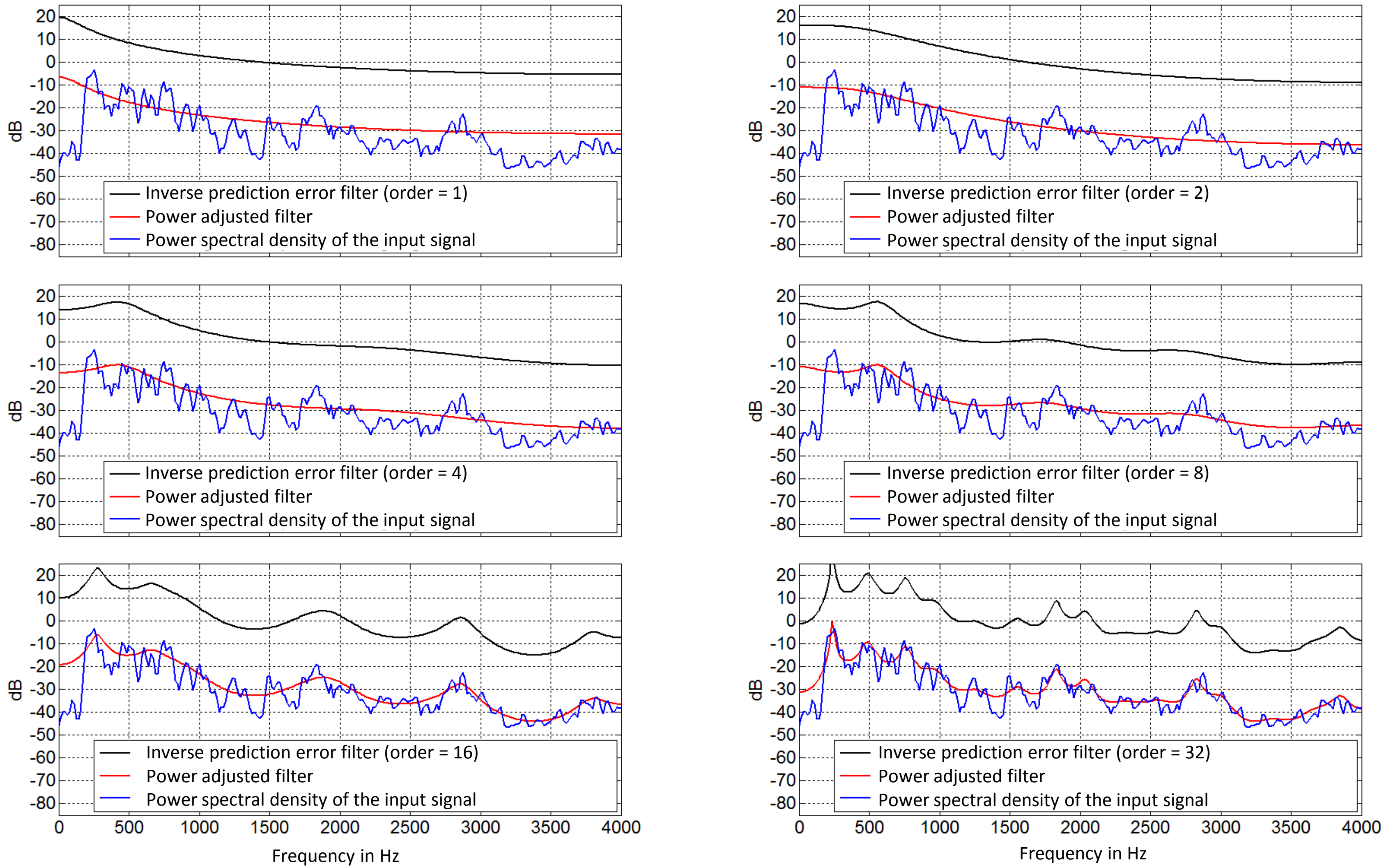
Impact of a Prediction Error Filter in the Frequency Domain – Part 1



Impact of a Prediction Error Filter in the Frequency Domain – Part 2



Impact of a Prediction Error Filter in the Frequency Domain – Part 3



Properties – Part 1

Minimization without restrictions (included in the filter structure)

- Cost function:

$$\mathrm{E}\{e^2(n)\} = \mathrm{E}\left\{\left(x(n) - \sum_{i=0}^{N-1} h_i\, x(n-i-1)\right)^2\right\} \to \min$$

The resulting filter has minimum phase:

- An FIR filter is computed with all its zeros within the unit circle.
- Signals can pass the filter with minimum delay.
- The inverse prediction filter is stable, since all zeros become poles and the zeros are located in the unit circle.

Normalized filters are generated – Part 1:

- Frequency response of the filter: $H_{\mathrm{PF}}\big(e^{j\Omega}\big) = 1 - \sum_{i=1}^{N} h_{i-1}\, e^{-j\Omega i}$
- Frequency response of the inverse: $H_{\mathrm{inv.\,PF}}\big(e^{j\Omega}\big) = \dfrac{1}{1 - \sum_{i=1}^{N} h_{i-1}\, e^{-j\Omega i}}$

Properties – Part 2

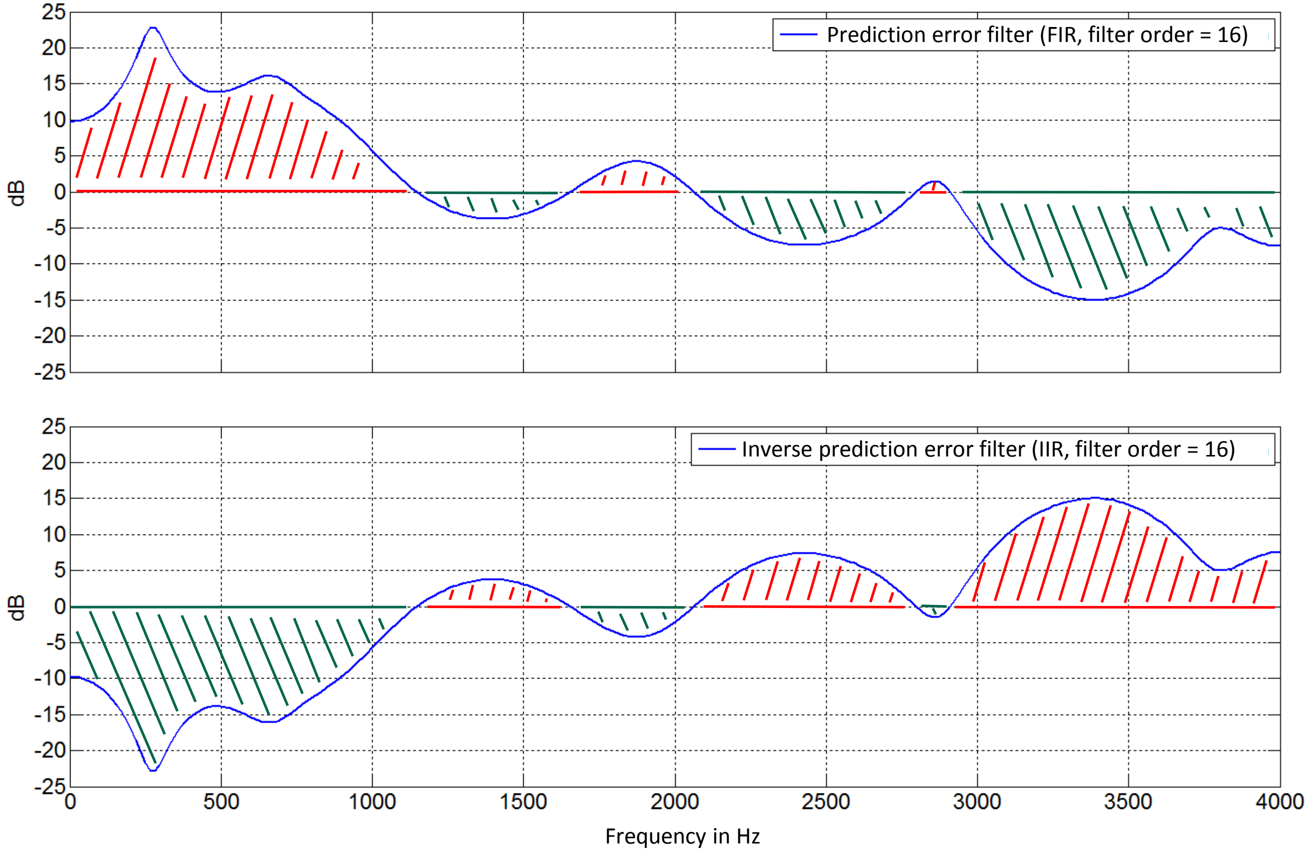
Normalized filters are generated (true for the prediction filter as well as for the inverse filter) – Part 2:

- Frequency response of the prediction filter: $$H_{\mathrm{PF}}\big(e^{j\Omega}\big) = 1 - \sum_{i=1}^{N} h_{i-1}\, e^{-j\Omega i}$$
- Frequency response of the inverse filter: $$H_{\mathrm{inv.\,PF}}\big(e^{j\Omega}\big) = \frac{1}{1 - \sum\limits_{i=1}^{N} h_{i-1}\, e^{-j\Omega i}}$$
- Type of normalization:

$$\boxed{\int\limits_{\Omega=0}^{2\pi} 20\,\log_{10}\Big\{\Big|H_{\mathrm{PF}}\big(e^{j\Omega}\big)\Big|\Big\}\,d\Omega \;=\; \int\limits_{\Omega=0}^{2\pi} 20\,\log_{10}\Big\{\Big|H_{\mathrm{inv.\,PF}}\big(e^{j\Omega}\big)\Big|\Big\}\,d\Omega \;=\; 0}$$

Properties – Part 3

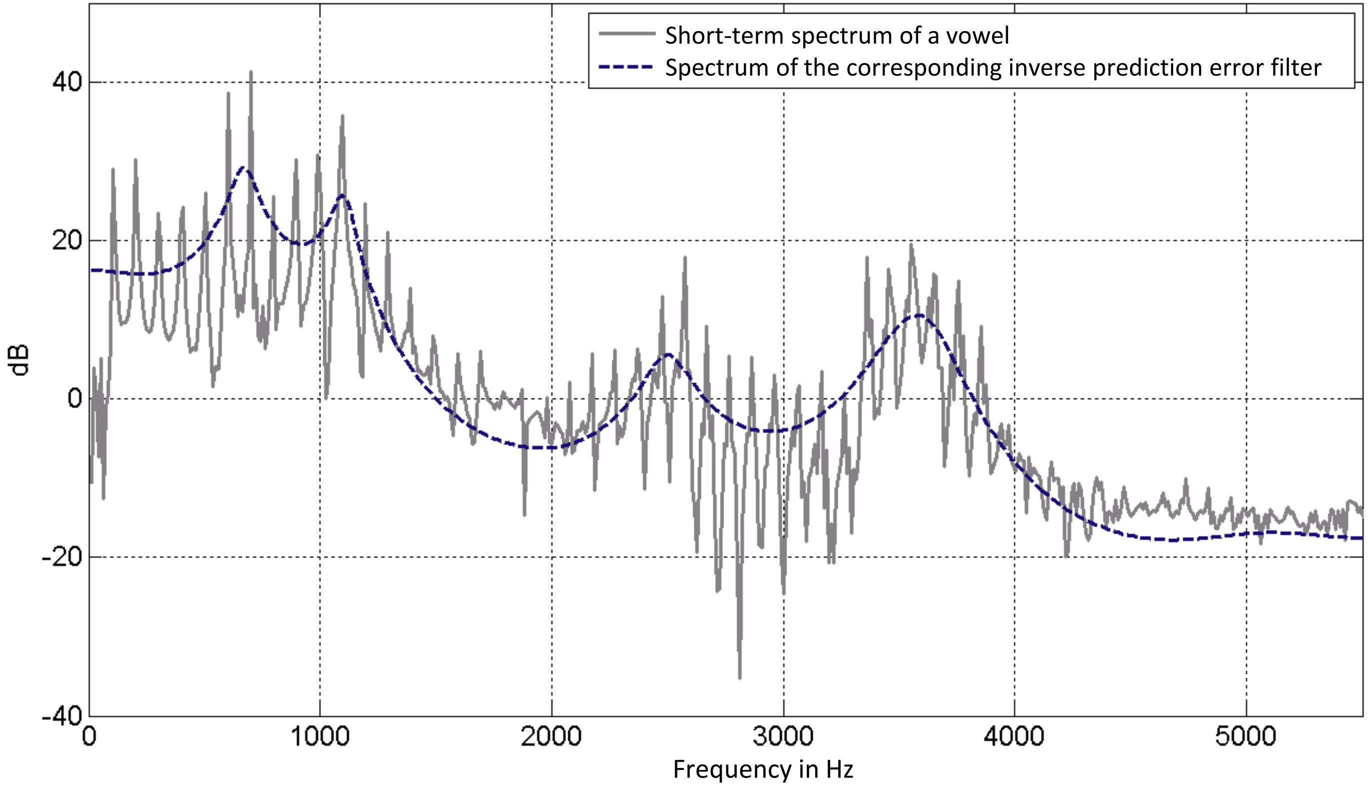
Normalized filters are generated (true for the prediction filter as well as for the inverse filter) – Part 3:



Estimation of the Spectral Envelope

Parametric estimation of the spectral envelope:

- Reducing the amount of parameters required to describe the specral envelope (compared to short-term spectrum)
- Independence of other signal properties (such as the pitch frequency)



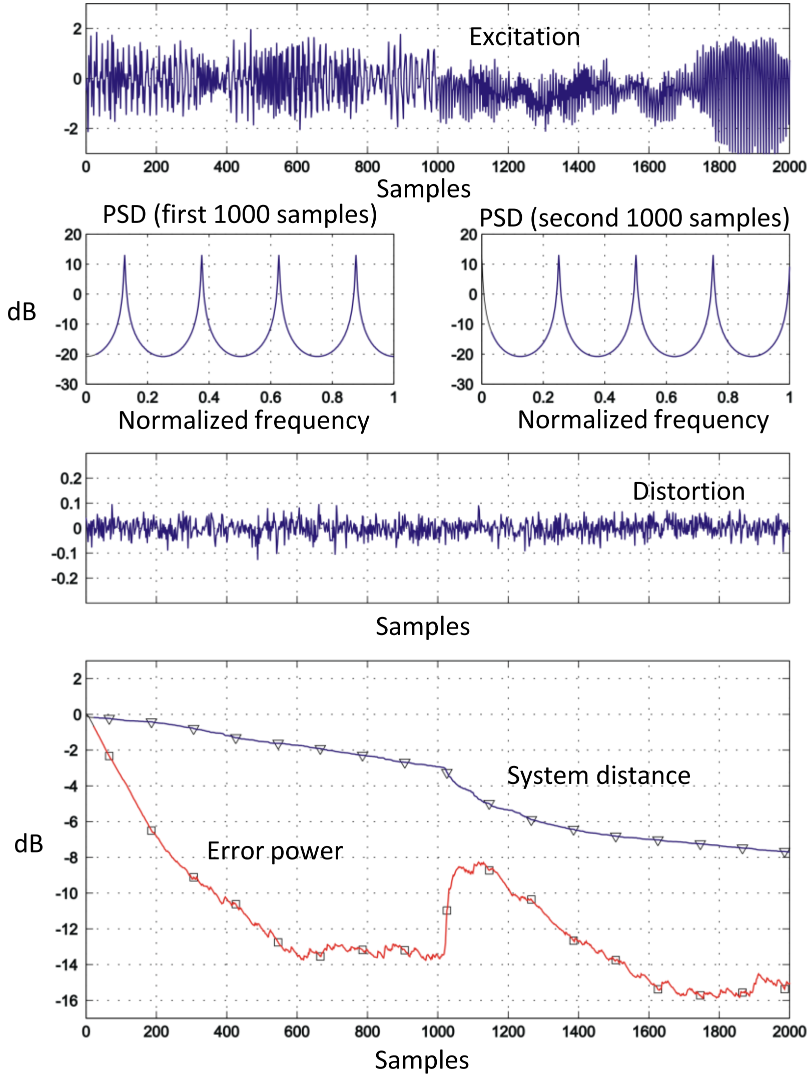
Applications of Linear Prediction – Part 1

Improving the Speed of Convergence of Adaptive Filters

Improving the Speed of Convergence of Adaptive Filters – Part 1

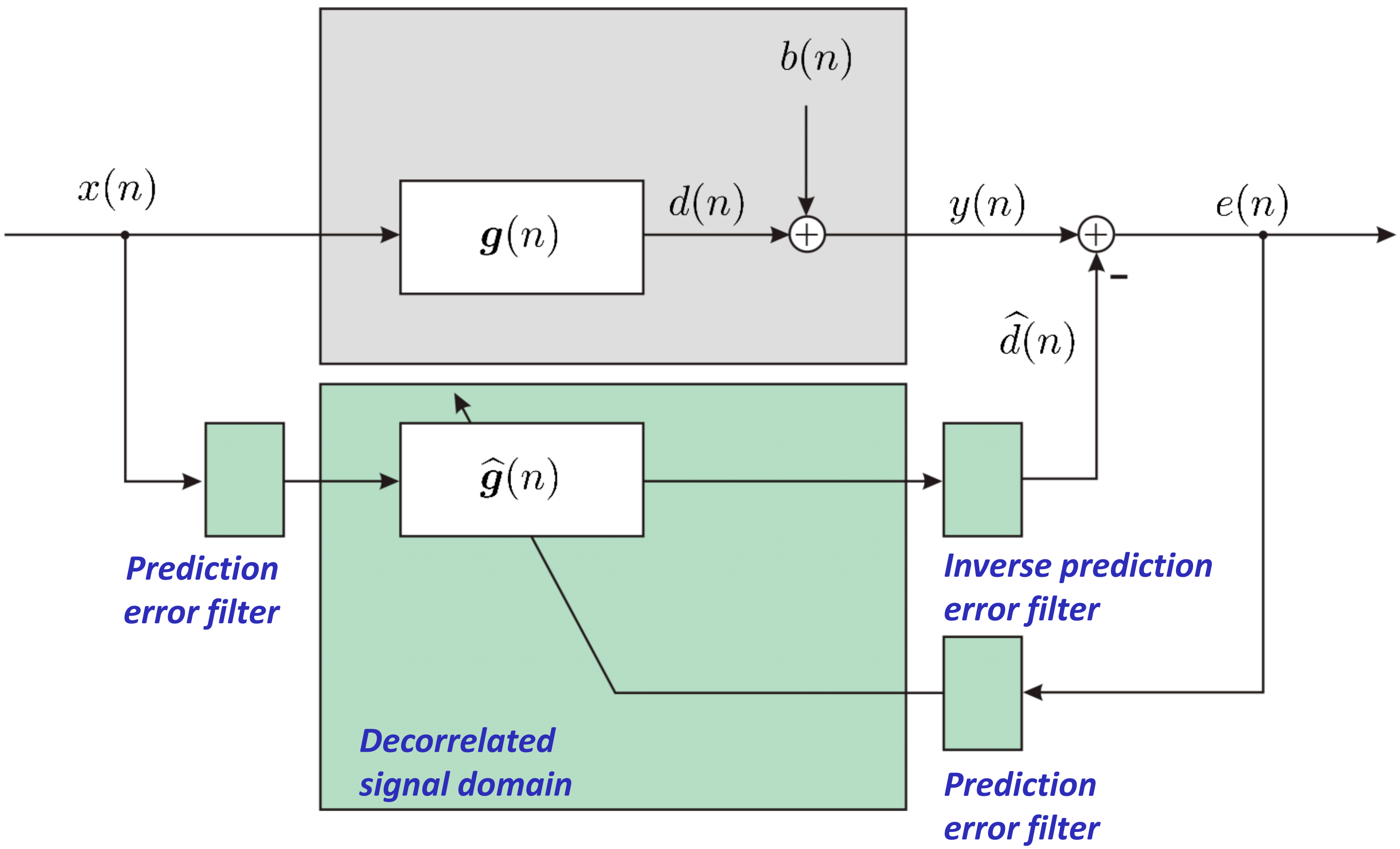
Simulation example:

- Excitation: colored noise (power spectral density [PSD] of the excitation is changed after 1000 samples)
- Distortion: white noise
- Monitoring the error power and the system distance



Improving the Speed of Convergence of Adaptive Filters – Part 2

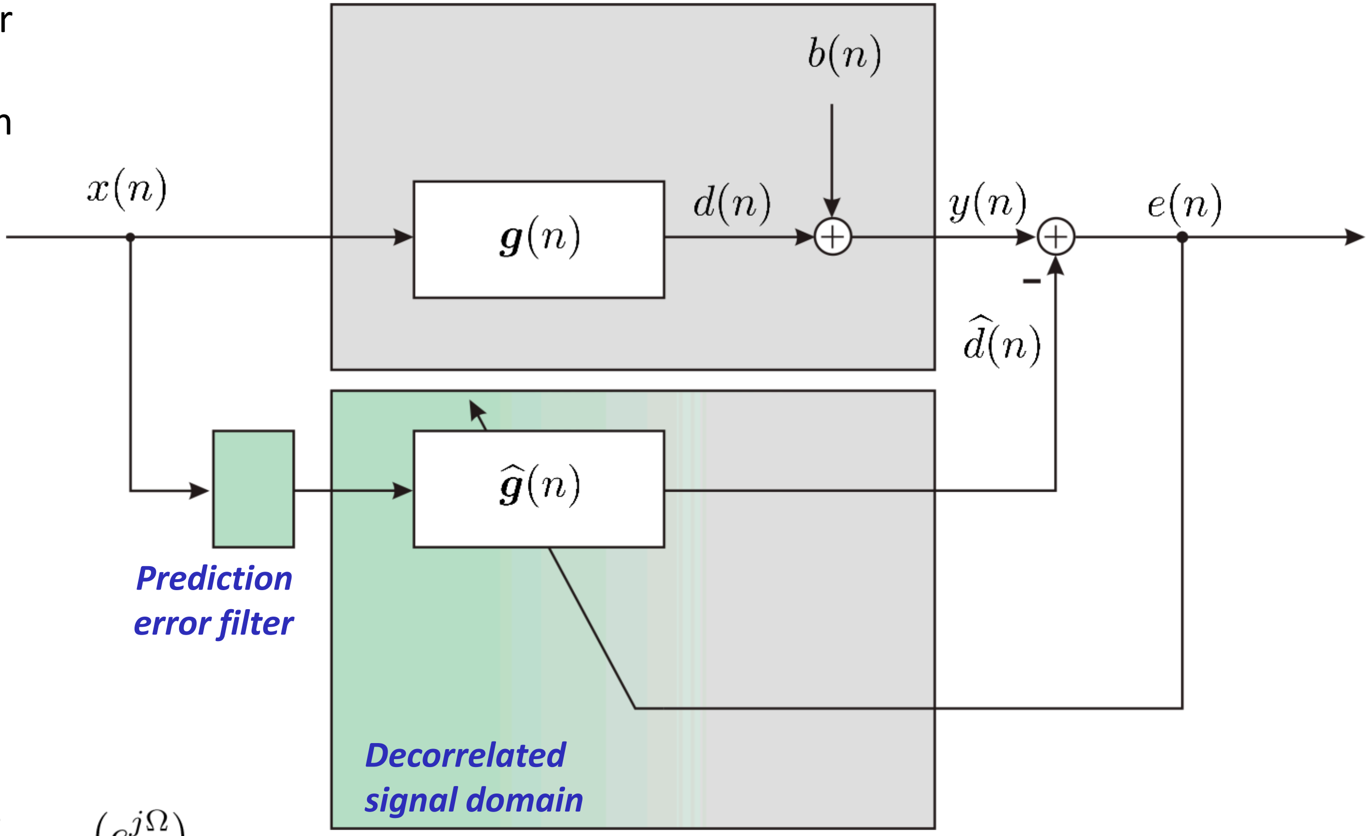
Time-invariant decorrelation:



Improving the Speed of Convergence of Adaptive Filters – Part 3

Simplified time-invariant decorrelation:

- The adaptive filter has to model the (unknown) system in series with the inverse prediction error filter (the convolution of both impulse responses)



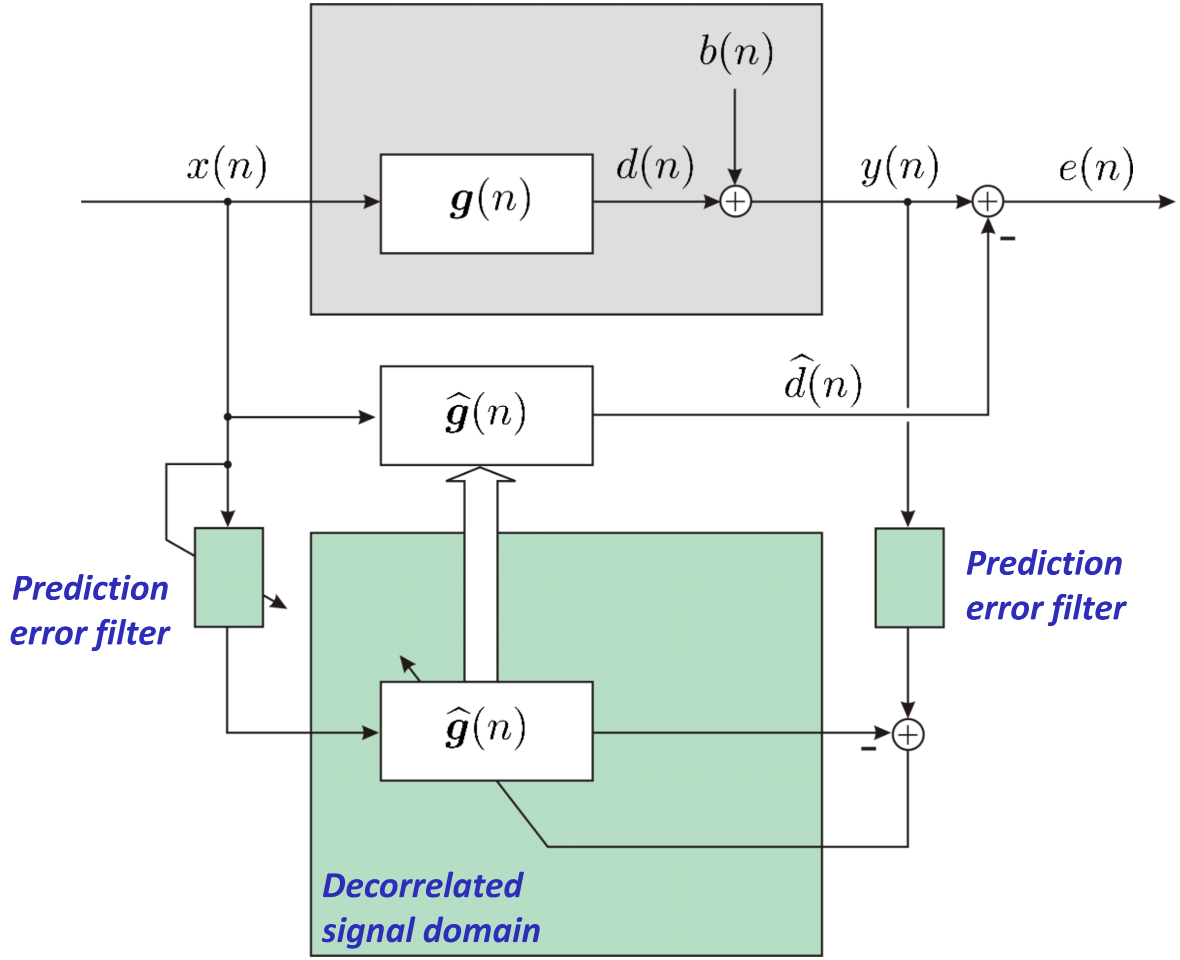
- Wiener solution:

$$\widehat{G}_{\text{opt}}\big(e^{j\Omega}, n\big) = G\big(e^{j\Omega}, n\big)\, H_{\text{inv. PF}}\big(e^{j\Omega}\big)$$

Improving the Speed of Convergence of Adaptive Filters – Part 4

Time-variant decorrelation

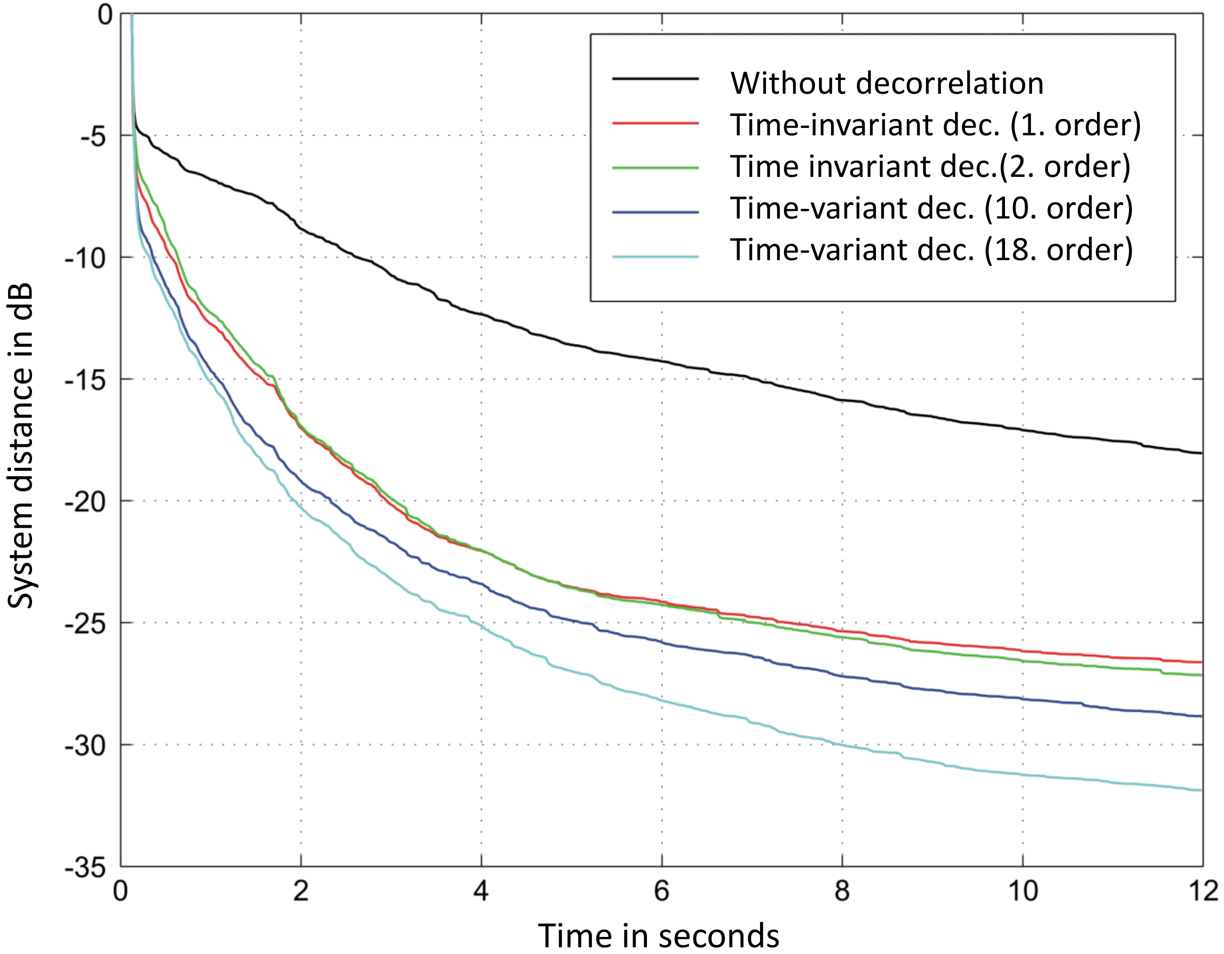
- Every 10 to 50 ms the prediction filters are updated.
- With the update also the signal memory of the adaptive filters needs to be corrected.
- This can be realized in an efficient manner by using a so-called double-filter structure.



Improving the Speed of Convergence of Adaptive Filters – Part 5

Convergence runs

(averaged over several simulations, speech was used as excitation):



Application of Linear Prediction – Part 2

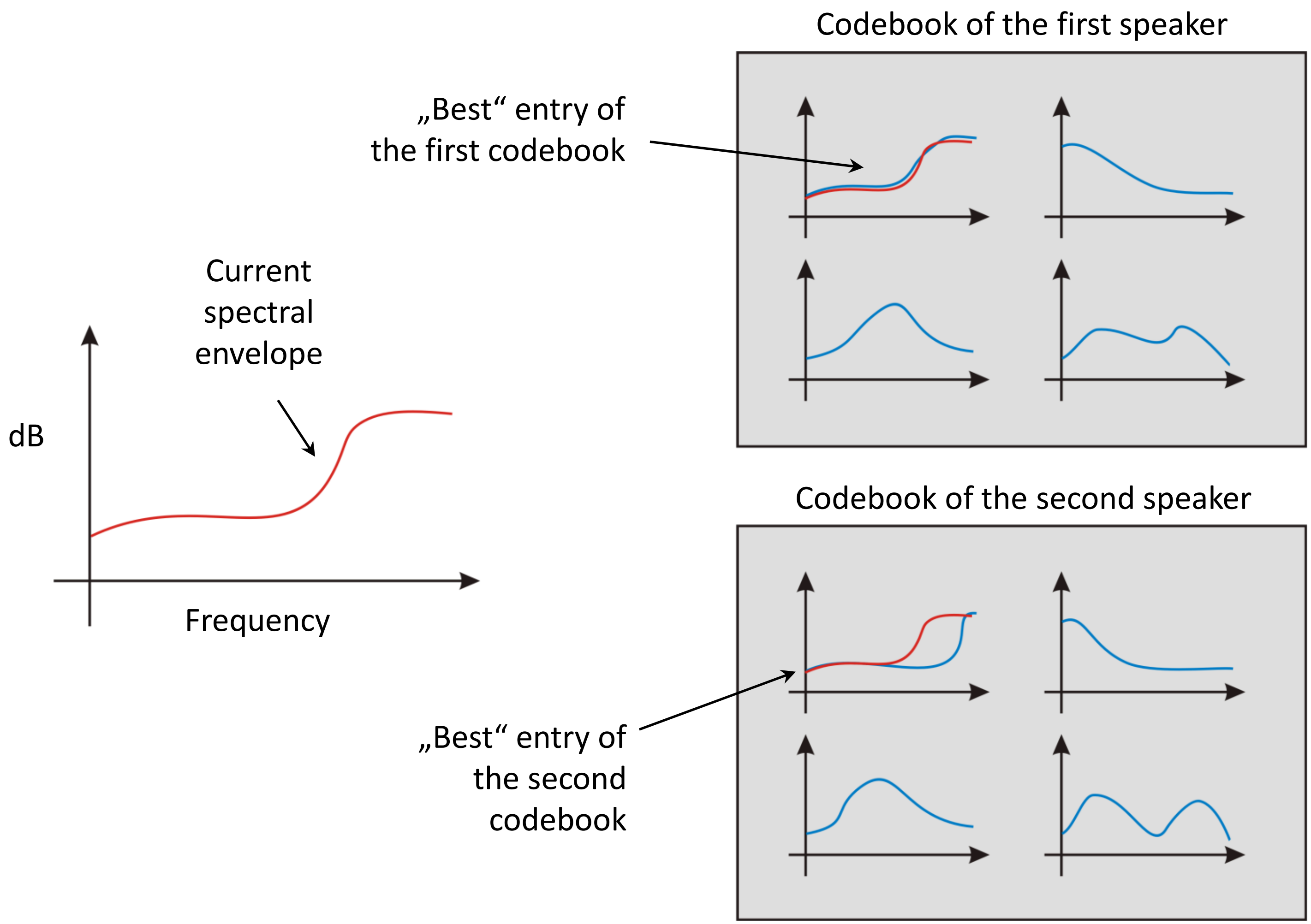
Speech and Speaker Recognition

Basics of Speaker Recognition – Part 1

Basic Principle:

- To recognize a speaker, first features are extracted out of the signal, e.g. the spectral envelope. This is performed every 5 to 30 ms.
- After extracting the feature vector it is compared with all entries of a codebook and the entry with minimum distance is detected.
- This has to be done for several codebooks, each belonging to an individual speaker.
- For each codebook the minimum distances are accumulated.
- The accumulated minimum distances determine which speaker is the one with the largest likelihood.
- Models for known speakers are competing with “universal” models.
- Often the winning codebook is adapted according to the new features.

Basics of Speaker Recognition – Part 2



Appropriate Cost Functions for Speech and Speaker Recognition – Part 1

Requirements:

- An appropriate cost function should measure the „perceived" distance between spectral envelopes. Similar envelopes should result in a small distance, very different envelopes in a large one, and the distance of equal envelopes should be zero.
- The cost function should be invariant to different amplitude settings when recording the speech signal.
- The cost function should have low computational complexity.
- The cost function should mimic the human perception (e.g. having a logarithmic loudness scale).

Ansatz:

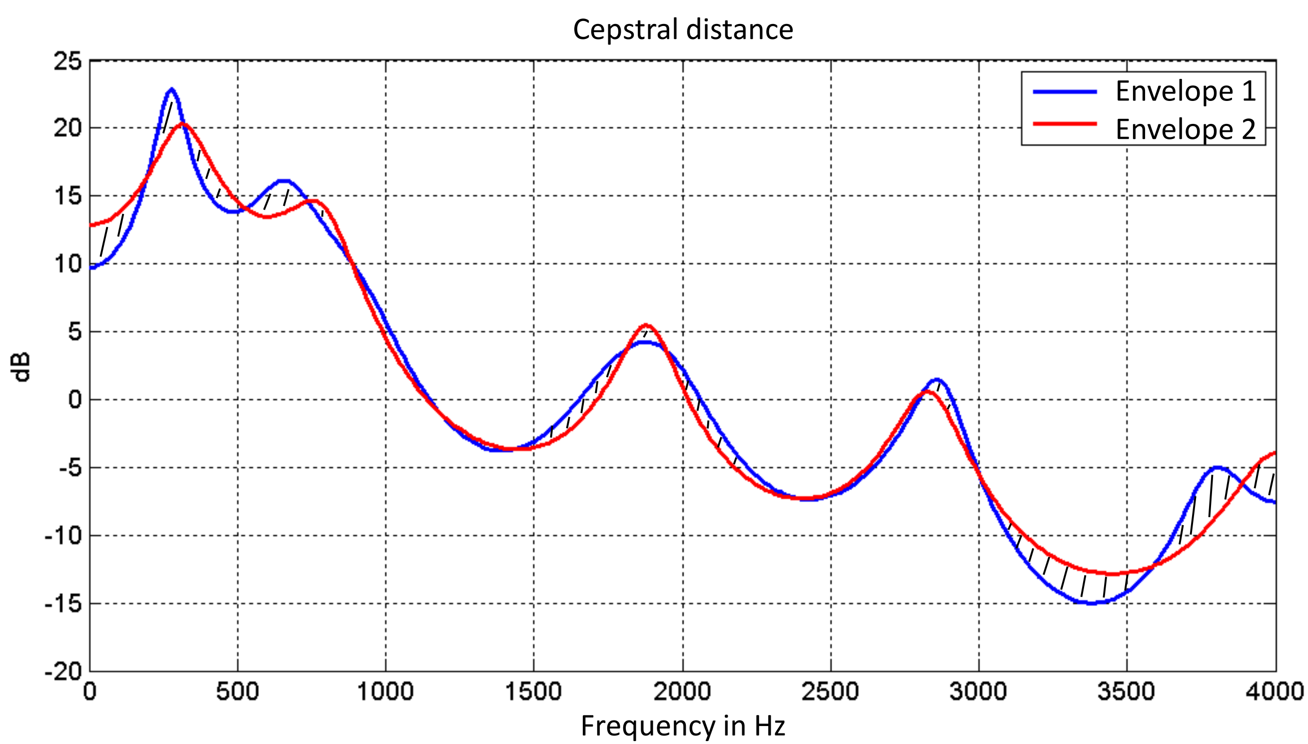
$$d_{\text{ceps}}(\ldots,\ldots) \;=\; \int\limits_{\Omega=0}^{2\pi} \Big| \ln\Big\{H_{\text{inv. PF},\,1}\big(e^{j\Omega}\big)\Big\} - \ln\Big\{H_{\text{inv. PF},\,2}\big(e^{j\Omega}\big)\Big\}\Big| \, d\Omega$$

Cepstral distance

Appropriate Cost Functions for Speech and Speaker Recognition – Part 2

Ansatz:

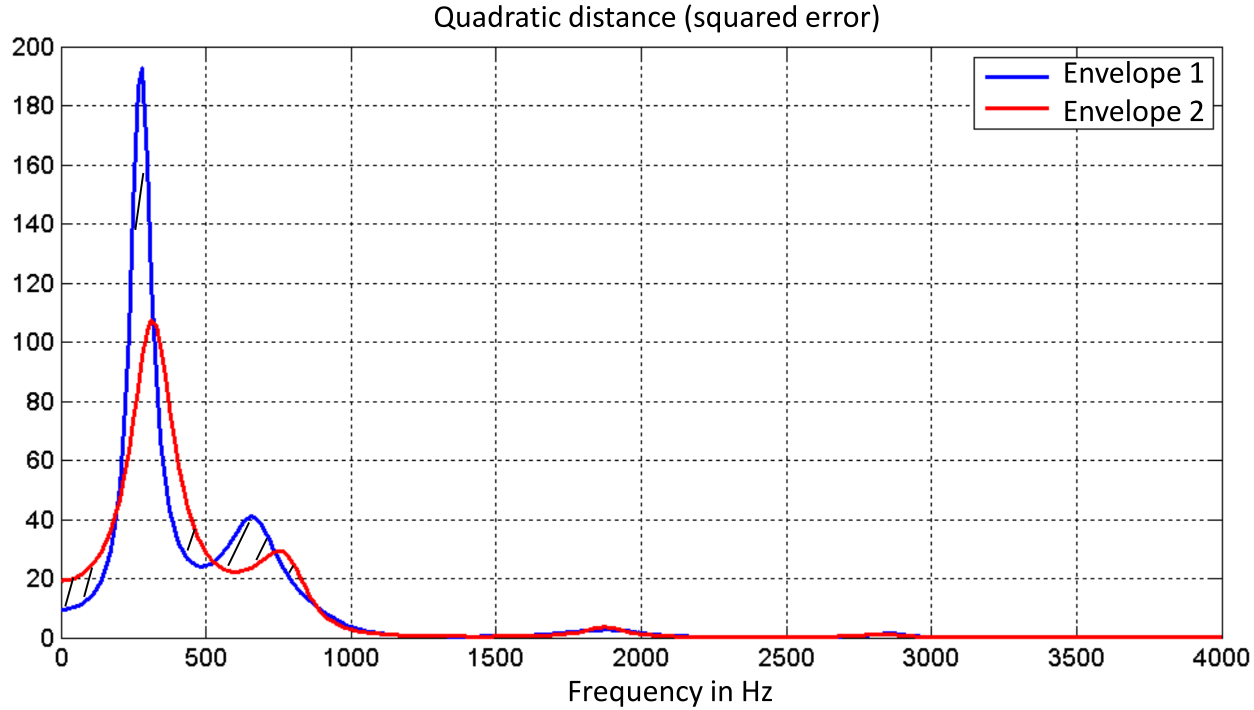
$$d_{\mathrm{ceps}}(\ldots,\ldots) \;=\; \int\limits_{\Omega=0}^{2\pi} \left| \ln\left\{H_{\mathrm{inv.\,PF},\,1}\big(e^{j\Omega}\big)\right\} - \ln\left\{H_{\mathrm{inv.\,PF},\,2}\big(e^{j\Omega}\big)\right\} \right| d\Omega$$



Appropriate Cost Functions for Speech and Speaker Recognition – Part 3

A „well known“ alternative – The (mean) squared error:

$$d_{\mathrm{mse}}(\ldots,\ldots) = \int\limits_{\Omega=0}^{2\pi} \left| H_{\mathrm{inv.\,PF},\,1}\big(e^{j\Omega}\big) - H_{\mathrm{inv.\,PF},\,2}\big(e^{j\Omega}\big) \right|^2 d\Omega$$



Appropriate Cost Functions for Speech and Speaker Recognition – Part 4

Cepstral distance:

$$d_{\text{ceps}}(...,...) = \int\limits_{\Omega=0}^{2\pi} \left| \ln\left\{H_{\text{inv. PF},1}\left(e^{j\Omega}\right)\right\} - \ln\left\{H_{\text{inv. PF},2}\left(e^{j\Omega}\right)\right\} \right| d\Omega$$

$$\Big\downarrow \text{Parseval}$$

$$\begin{aligned} d_{\text{ceps}}(...,...) &= \sum_{i=-\infty}^{\infty} \left(c_{i,1} - c_{i,2}\right)^2 \\ &\approx \sum_{i=1}^{3/2\,N} \left(c_{i,1} - c_{i,2}\right)^2 \end{aligned}$$

$$c_i = \frac{1}{2\pi} \int\limits_{\Omega=0}^{2\pi} \ln\left\{H_{\text{inv. PF}}\left(e^{j\Omega}\right)\right\} e^{j\Omega_i}\, d\Omega$$

$$\ln\{z\} = \ln|z| + j\arg\{z\}$$

Appropriate Cost Functions for Speech and Speaker Recognition – Part 5

Efficient transformation of prediction into cepstral coefficients:

- Definition

$$c_i = \frac{1}{2\pi} \int_{\Omega=0}^{2\pi} \ln\left\{H_{\text{inv. PF}}\left(e^{j\Omega}\right)\right\} e^{j\Omega i}\, d\Omega$$

- Fourier transform for discrete signals and systems

$$\sum_{i=-\infty}^{\infty} c_i\, e^{-j\Omega i} = \ln\left\{H_{\text{inv. PF}}\left(e^{j\Omega}\right)\right\}$$

- Replacing $e^{j\Omega}$ with z (z-transform)

$$\left.\sum_{i=-\infty}^{\infty} c_i\, z^{-i}\right|_{z=e^{j\Omega}} = \left.\ln\left\{H_{\text{inv. PF}}(z)\right\}\right|_{z=e^{j\Omega}}$$

C | A | U
Christian-Albrechts-Universität zu Kiel

Appropriate Cost Functions for Speech and Speaker Recognition – Part 6

Efficient transformation of prediction into cepstral coefficients:

- Previous result

$$\sum_{i=-\infty}^{\infty} c_i \, z^{-i} \;=\; \ln\left\{H_{\text{inv. PF}}(z)\right\}$$

- Inserting the structure of an inverse prediction error filter

$$\begin{aligned}\sum_{i=-\infty}^{\infty} c_i \, z^{-i} \;&=\; \ln\left\{\frac{1}{1-\sum\limits_{i=1}^{N} h_{i-1}\, z^{-i}}\right\}\\ &=\; -\ln\left\{1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\}\end{aligned}$$

Appropriate Cost Functions for Speech and Speaker Recognition – Part 7

Efficient transformation of prediction into cepstral coefficients:

- Previous result

$$\sum_{i=-\infty}^{\infty} c_i\, z^{-i} = -\ln\left\{1 - \sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\}$$

- Computing the coefficients with non-positive index:

$$\begin{aligned}\ln\left\{1 - \sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\} &= \ln\left\{\prod_{i=0}^{N}\left(1 - b_i\, z^{-1}\right)\right\} \\ &= \sum_{i=0}^{N} \ln\left\{1 - b_i\, z^{-1}\right\}\end{aligned}$$

Inserting

- Using the following series:

$$\ln\left\{1 - b\, z^{-1}\right\} = -\sum_{k=1}^{\infty} \frac{b^k}{k}\, z^{-k}, \qquad \text{for } |z| > |b|$$

Appropriate Cost Functions for Speech and Speaker Recognition – Part 8

Efficient transformation of prediction into cepstral coefficients:

- Computing the coefficients with non-positive index

  - After inserting the result of the last slide we get:

$$\ln\left\{1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\} \;=\; -\sum_{i=0}^{N}\sum_{k=1}^{\infty}\frac{b_i^k}{k}\, z^{-k}$$

  - Thus, we obtain

$$\sum_{i=1}^{\infty} c_i\, z^{-i} \;=\; -\ln\left\{1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\}$$

All coefficients with non-positive index are zero!

Appropriate Cost Functions for Speech and Speaker Recognition – Part 9

Efficient transformation of prediction into cepstral coefficients:

- Previous result

$$\sum_{i=1}^{\infty} c_i\, z^{-i} \;=\; -\ln\left\{1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\}$$

- Differentiation

$$\frac{d}{dz}\left[\sum_{i=1}^{\infty} c_i\, z^{-i}\right] \;=\; -\frac{d}{dz}\left[\ln\left\{1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right\}\right]$$

$$-\sum_{i=1}^{\infty} i\, c_i\, z^{-i-1} \;=\; -\sum_{i=1}^{N} i\, h_{i-1}\, z^{-i-1}\left[1-\sum_{i=1}^{N} h_{i-1}\, z^{-i}\right]^{-1}$$

- Multiplication of both sides with […]

$$\sum_{i=1}^{\infty} i\, c_i\, z^{-i-1} \;-\; \sum_{k=1}^{\infty}\sum_{i=1}^{N} k\, c_k\, h_{i-1}\, z^{-k-i-1} \;=\; \sum_{i=1}^{N} i\, h_{i-1}\, z^{-i-1}$$

Efficient transformation of prediction into cepstral coefficients:

- Previous result

$$\sum_{i=1}^{\infty} i\, c_i\, z^{-i-1} \;-\; \sum_{k=1}^{\infty} \sum_{i=1}^{N} k\, c_k\, h_{i-1}\, z^{-k-i-1} \;=\; \sum_{i=1}^{N} i\, h_{i-1}\, z^{-i-1}$$

- Comparing the coefficients for $i \in \{1, ..., N\}$

$$i\, c_i \;-\; \sum_{k=1}^{i-1} k\, c_k\, h_{i-k-1} \;=\; i\, h_{i-1}$$

- Comparing the coefficients for $i > N$

$$i\, c_i \;-\; \sum_{k=1}^{i-1} k\, c_k\, h_{i-k-1} \;=\; 0$$

Appropriate Cost Functions for Speech and Speaker Recognition – Part 11

Efficient transformation of prediction into cepstral coefficients:

$$c_i = \begin{cases} 0, & \text{if } i < 1, \\ h_{i-1} + \frac{1}{i} \sum\limits_{k=1}^{i-1} k\, c_k\, h_{i-k-1}, & \text{if } 1 \leq i \leq N, \\ \frac{1}{i} \sum\limits_{k=1}^{i-1} k\, c_k\, h_{i-k-1}, & \text{else.} \end{cases}$$

Recursive method with low complexity. The sum can be truncated after 3/2 N, since cepstral coefficients with a larger index usually do not contribute significantly to the result.

Applications of Linear Prediction – Part 3

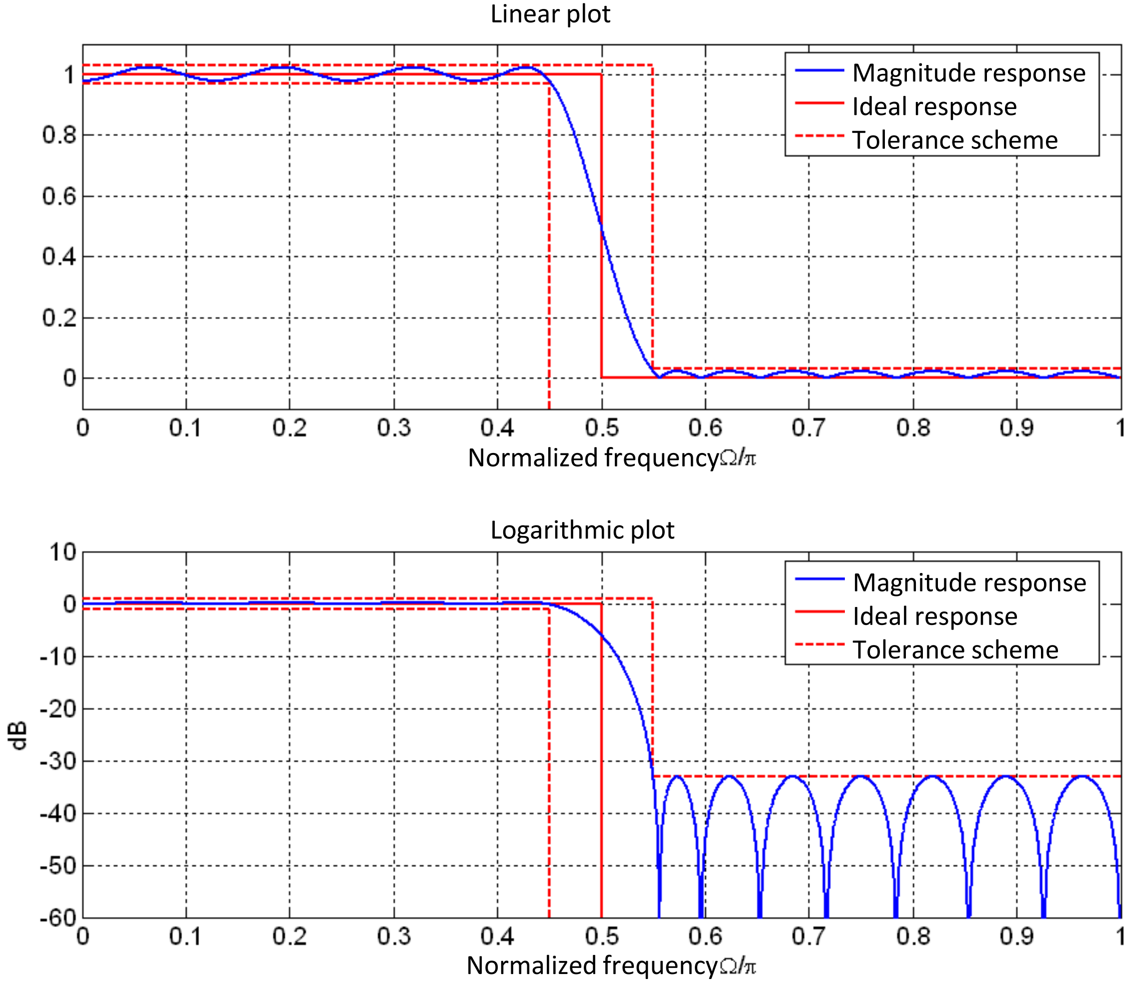
Filter Design

C | A | U
Christian-Albrechts-Universität zu Kiel

Filter Design – Part 1

Specification of a tolerance scheme:

- Often a lowpass, bandpass, bandstop, or highpass filter is specified.
- The solution is computed iteratively (e.g. by means of programs such as Matlab).
- FIR or IIR filters can be designed.

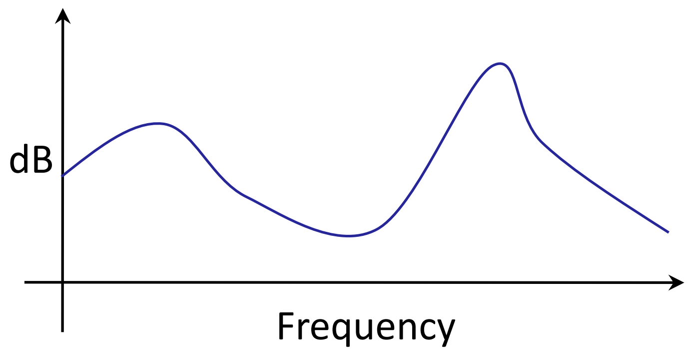


Christian-Albrechts-Universität zu Kiel

Filter Design – Part 2

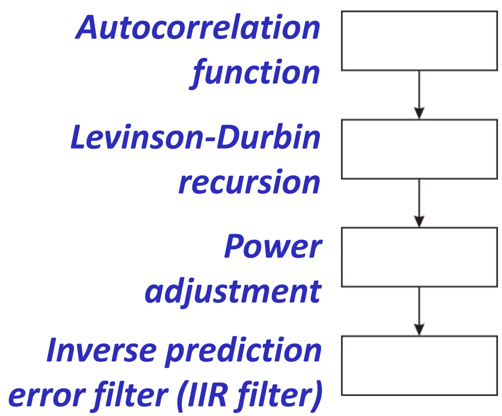
… but what to do, if e.g. …

- … a filter with arbitrary (known only at run-time) frequency response should be designed.



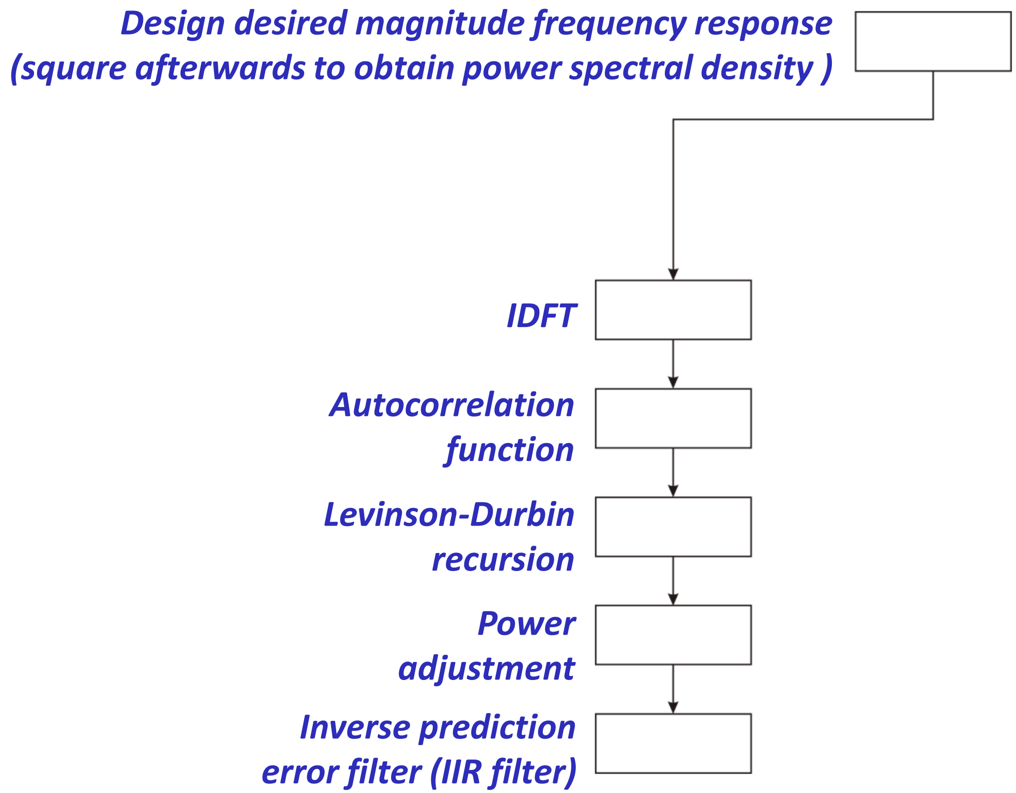
- … the filter should have either FIR or IIR structure (or a mix of both).
- … a mininum-phase filter should be designed (minimum group delay).
- … only limited computational power and memory are available for the design process.

Filter Design for Prediction Filters – Part 1



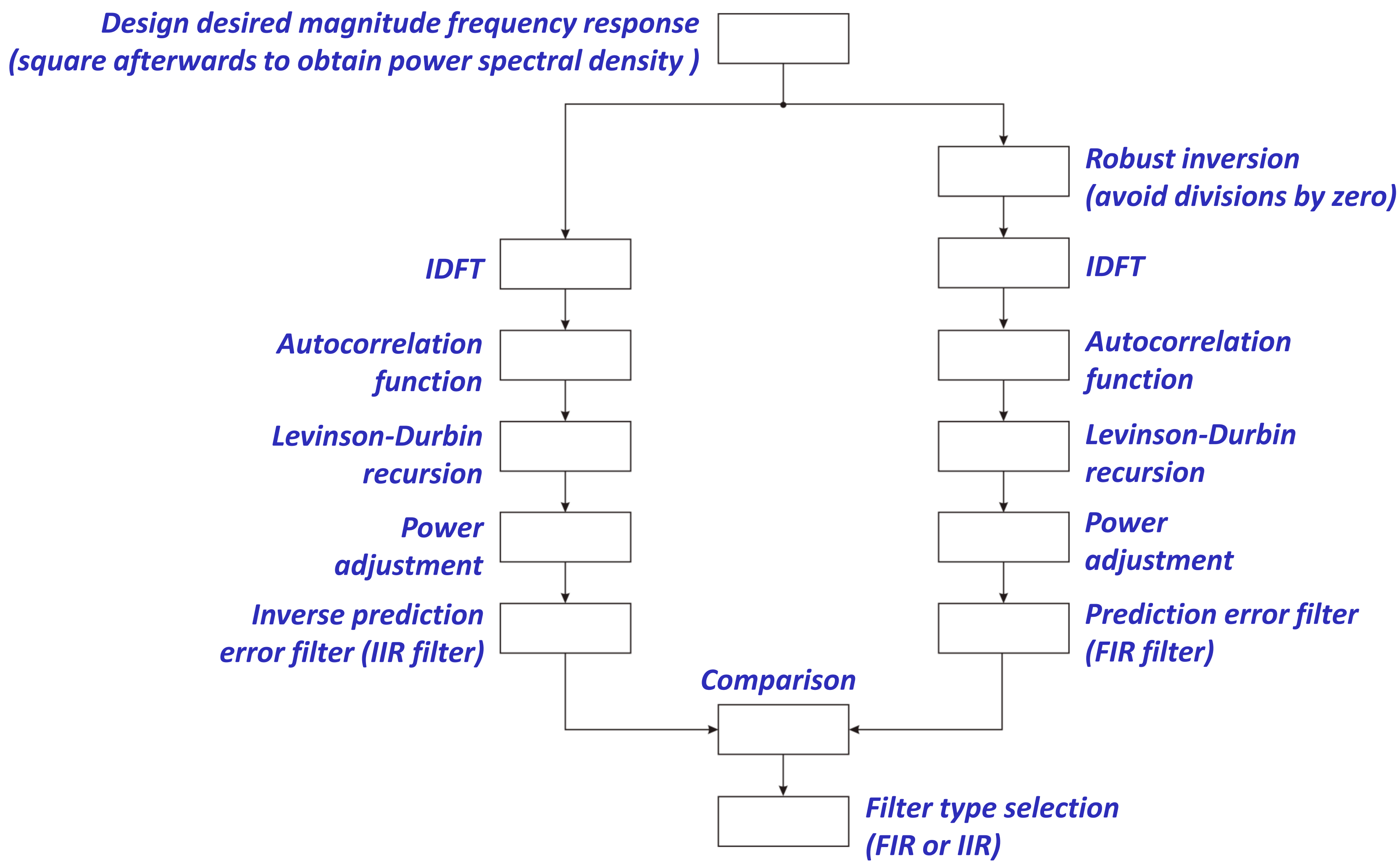
Christian-Albrechts-Universität zu Kiel

Filter Design for Prediction Filters – Part 2

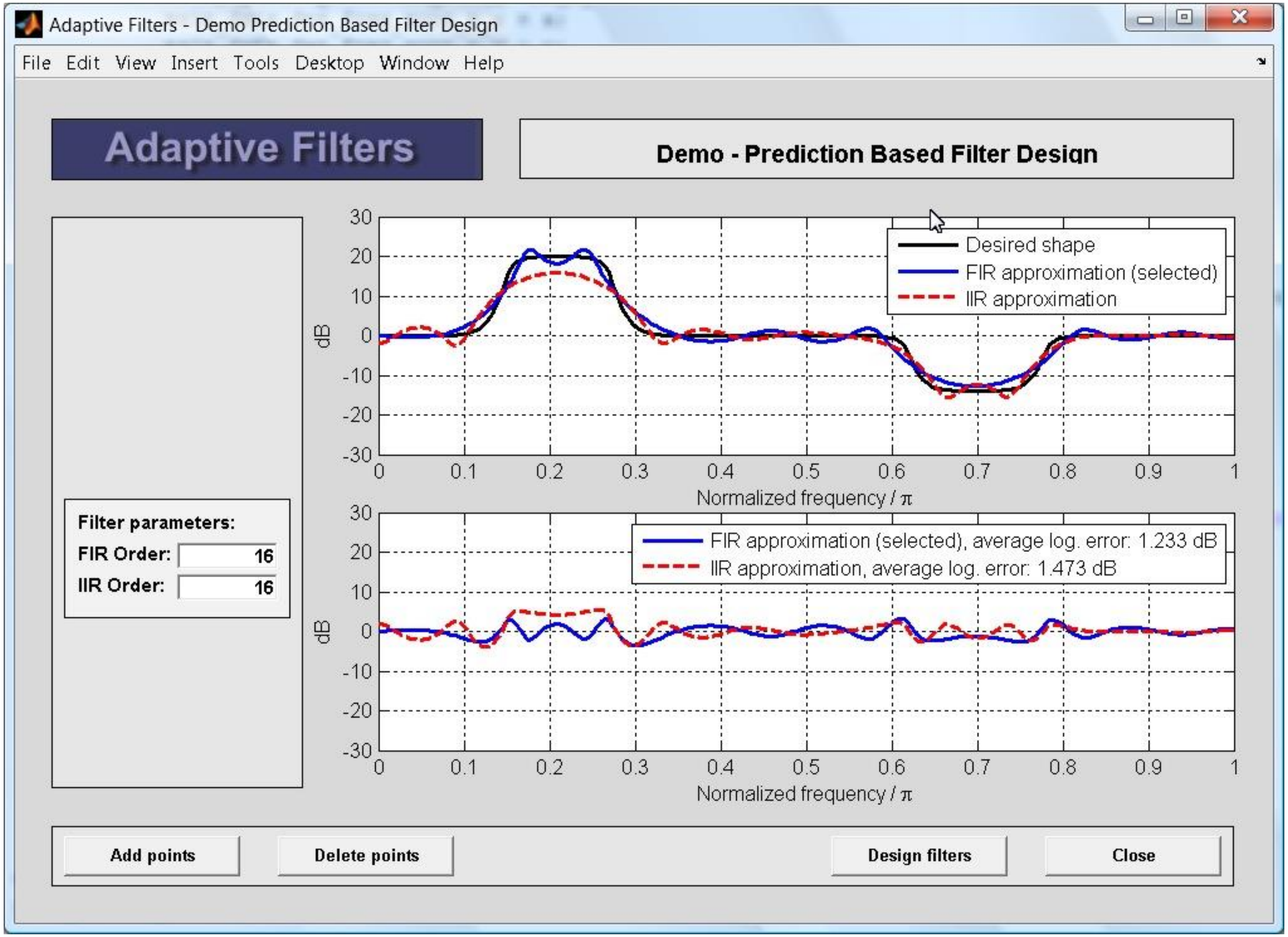


C | A | U
Christian-Albrechts-Universität zu Kiel

Filter Design for Prediction Filters – Part 3



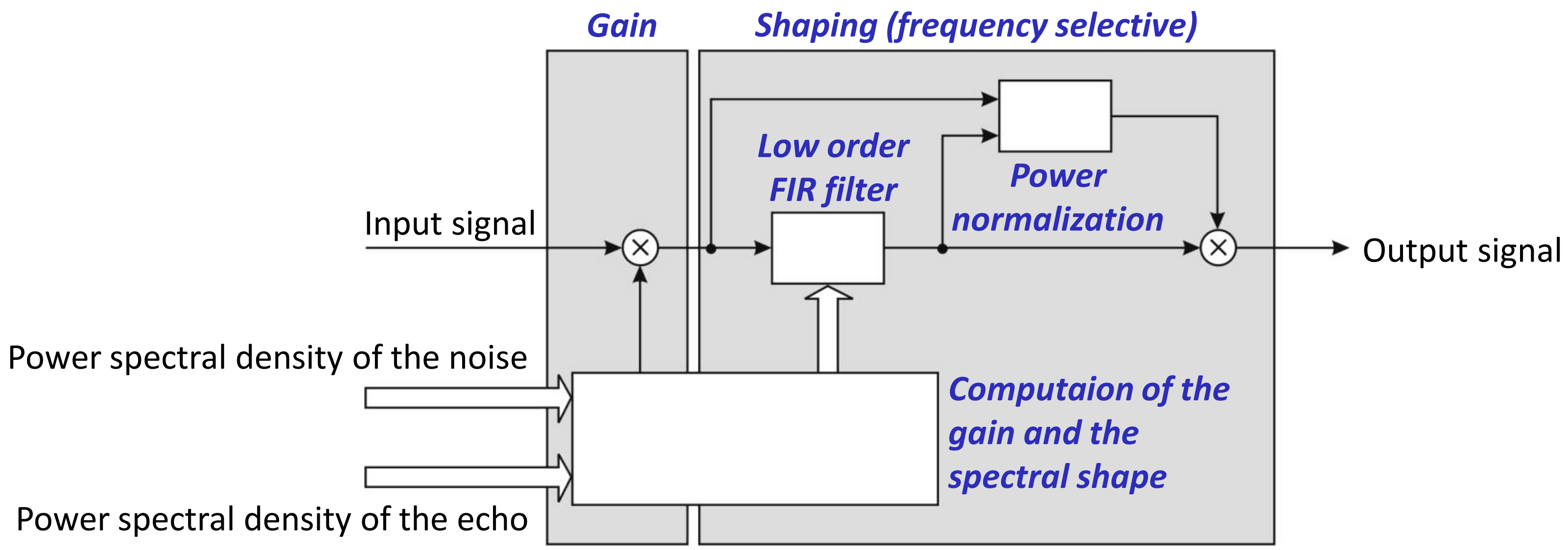
Design Example



Applications of Prediction-based Filter Design – Part 1

Application examples:

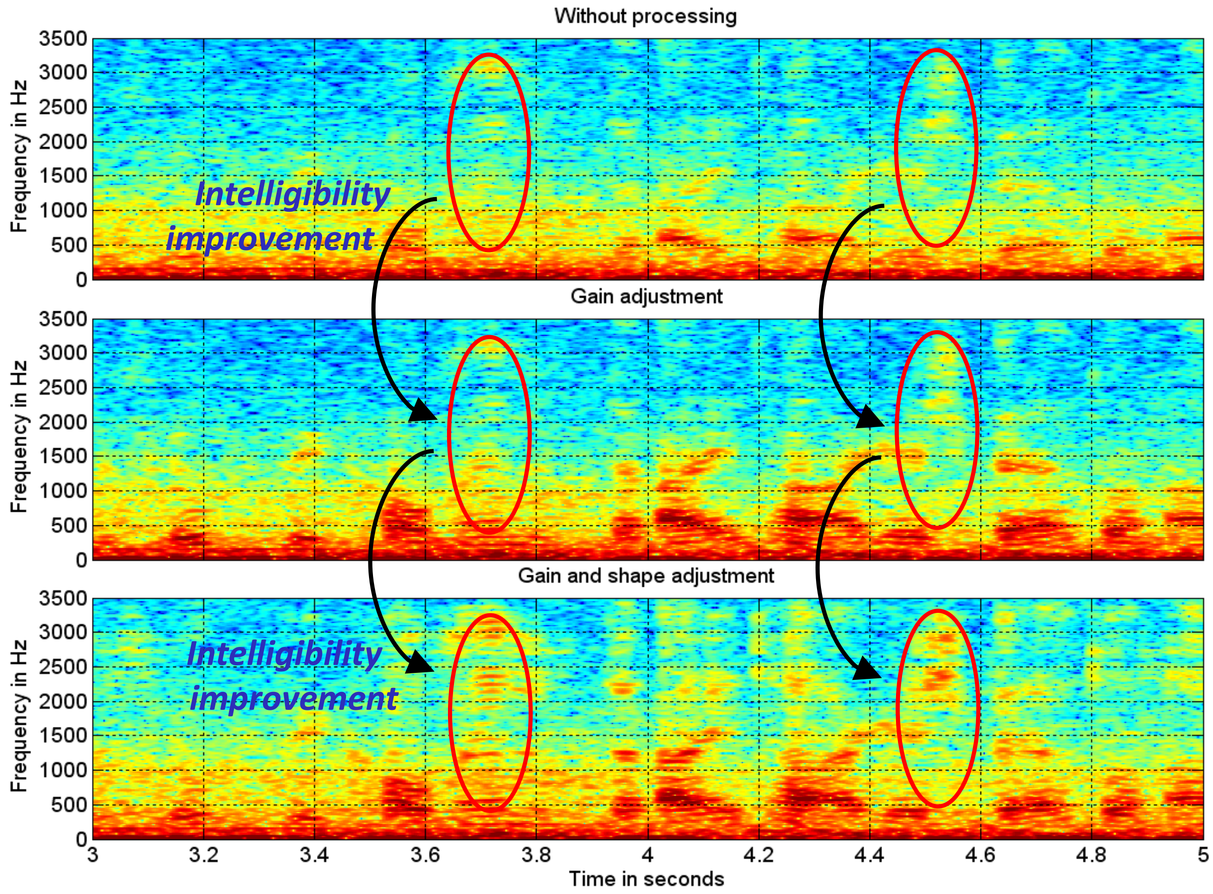
- For adaptively adjusting limiters.
- For low-delay noise reduction filters.
- For frequency selective gain adjustment of the output of speech prompters and hands-free systems (loudspeaker output).



Applications of Prediction-based Filter Design – Part 2

Measurement:

Binaural recording while acceleration of a car (left ear signal depicted).



Details: B. Iser, G. Schmidt: Receive Side Processing in a Hands-Free Application, Proc. HSCMA, 2008

Summary and Outlook

This week:

- Repetition of linear prediction
- Properties of prediction filters
- Application examples
  - Improving the convergence speed of adaptive filters
  - Speech and speaker recognition
  - Filter design