Christian-Albrechts-Universität zu Kiel

Pattern Recognition and Machine Learning

Part 2: Cost Functions and Single-channel Noise Suppression

Gerhard Schmidt

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

Contents

- Cost functions
  - Data/sample-based cost functions
  - Distribution-based cost functions
- Enhancement of speech signals
  - Generation and properties of speech signals
  - Wiener filter
  - Frequency-domain solution
  - Extensions of the gain rule
  - Extensions of the entire framework
  - Outlook to neural net based approaches
- Enhancement of EEG signals
  - Empirical mode decomposition

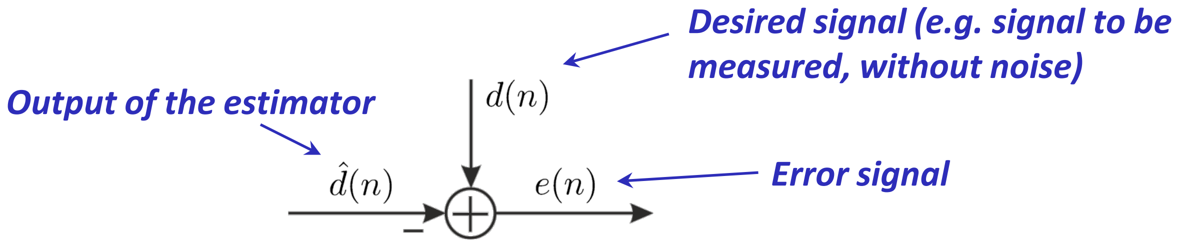
Contents

Cost Functions – Part 1

Signal-based error criteria – Part 1:

- An ***error signal*** $e(n)$ specifies often the difference between a ***desired signal*** $d(n)$ and its ***estimation*** $\hat{d}(n)$

$$e(n) = d(n) - \hat{d}(n).$$



- This results often in a ***cost function*** with the following ***properties***:
  - Necessary

$$f\big(e_2(n)\big) \geq f\big(e_1(n)\big) \quad \text{for} \quad \big|e_2(n)\big| \geq \big|e_1(n)\big|.$$

  - Desired:

$$f\big(e(n)\big) = f\big(-e(n)\big).$$

Signal-based error criteria – Part 2:

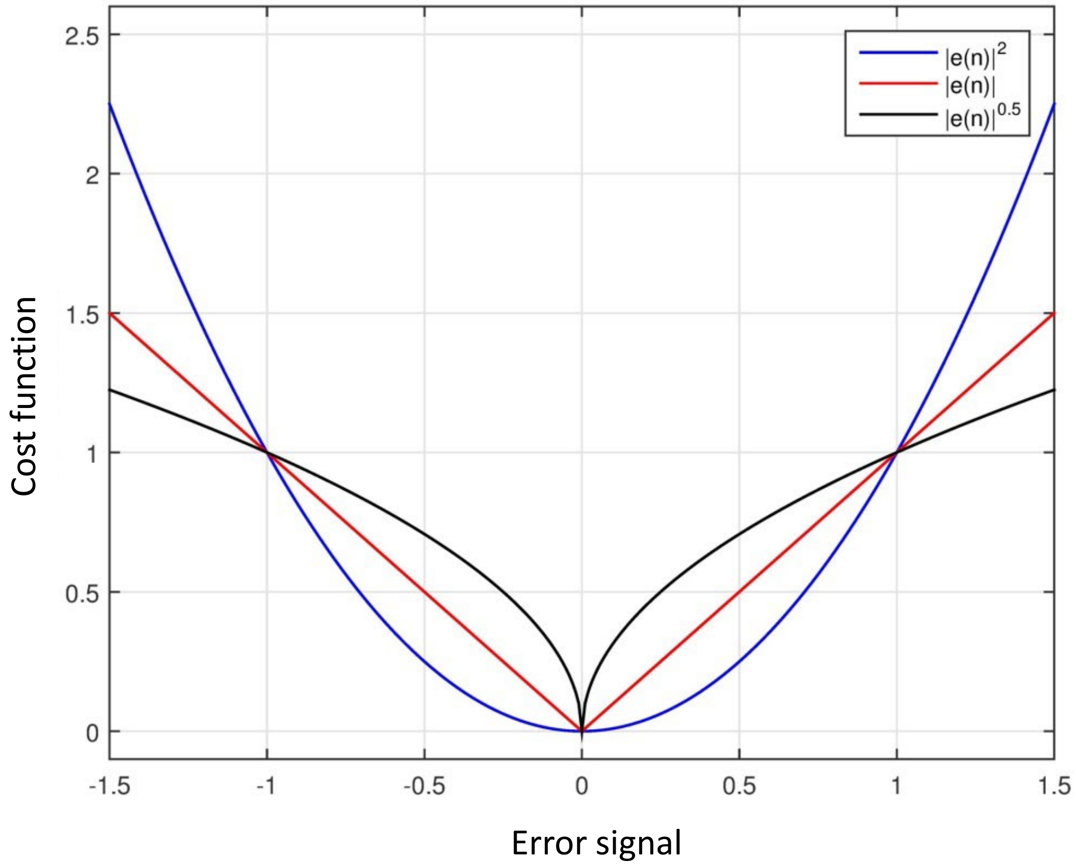
- Often used (instantaneous) ***cost functions***:
  - $$f\big(e(n)\big) \;=\; \big|e(n)\big|,$$
  - $$f\big(e(n)\big) \;=\; \big|e(n)\big|^2.$$
- More ***generically***, one can use

$$f\big(e(n)\big) \;=\; \big|e(n)\big|^\alpha.$$

For $\alpha > 1$ large errors will be amplified and small ones attenuated. For $\alpha < 1$ it is vice versa.

- For derivations often the ***mean squared error*** is used

$$f\big(e(n)\big) \;=\; \mathrm{E}\Big\{\big|e(n)\big|^2\Big\}.$$



Cost Functions – Part 3

Signal-based error criteria – Part 3:

- In machine learning so-called ***batches*** or ***mini-batches*** are computed and corresponding gradients are averaged. This leads to the following cost functions:

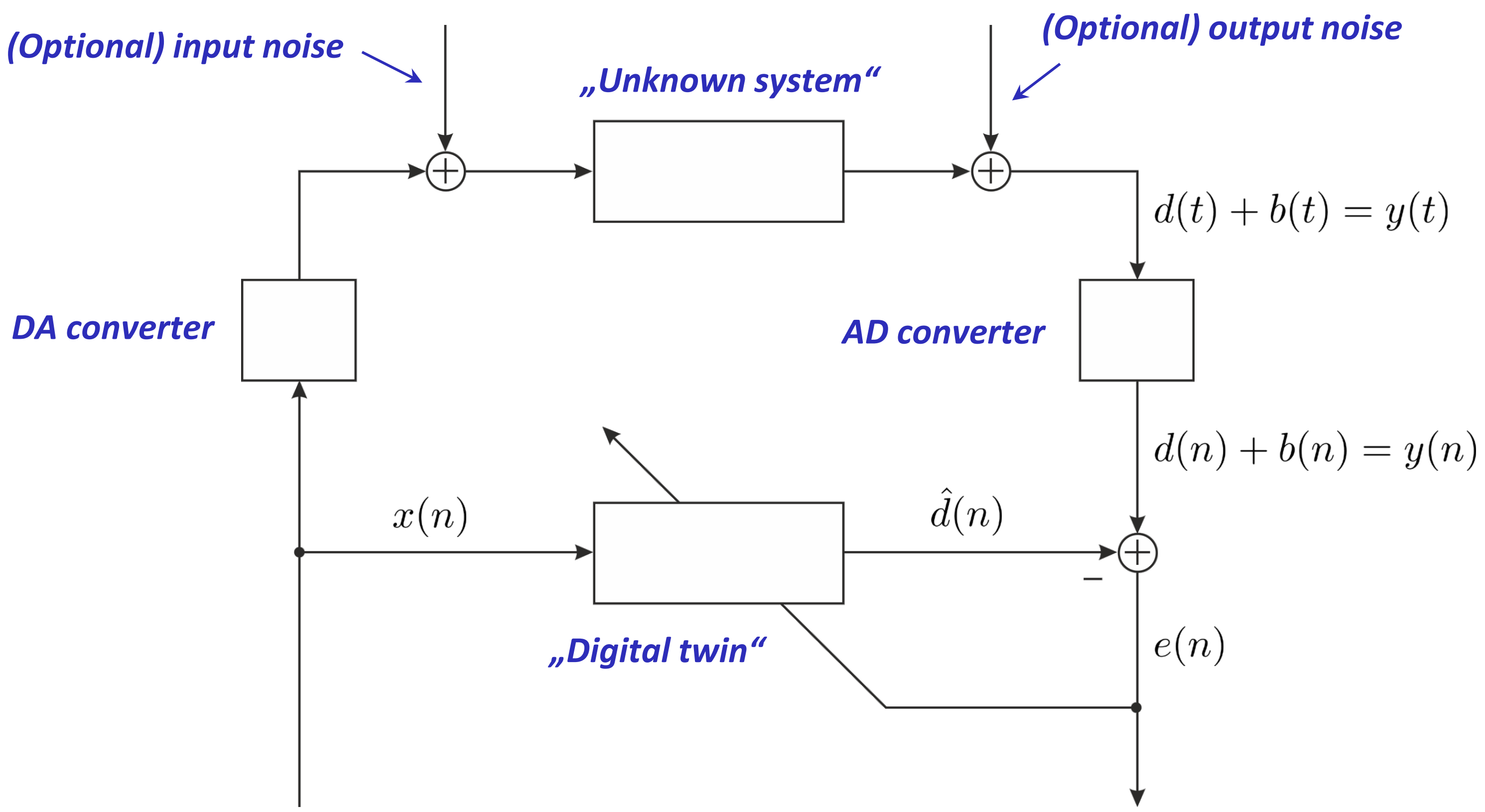
$$f\big(e(n),\, \ldots,\, e(n-N+1)\big) \;=\; \frac{1}{N}\sum_{i=0}^{N-1} e^2(n-i).$$

- Here often a compromise between average performance and memory consumption has to be found.
- Furthermore, after updating one batch the following ones can be computed with the updated parameters.
- Often the input data is randomized in temporal order, but attention is required if more than one data frame is contributing to the output.

Cost Functions – Part 4

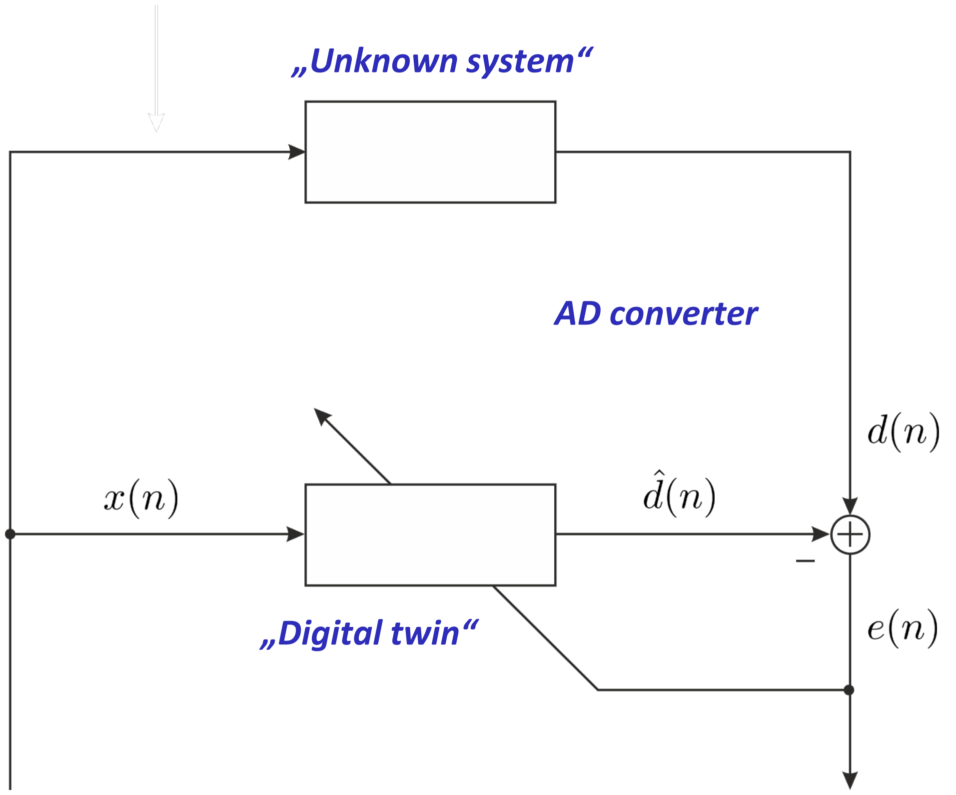
Signal-based error criteria – Part 4:

- Basic setup with access to excitation signals (not always given)



Signal-based error criteria – Part 5:

- Basic setup with access to excitation signals (not always given)
- Completely digital and noise-free version



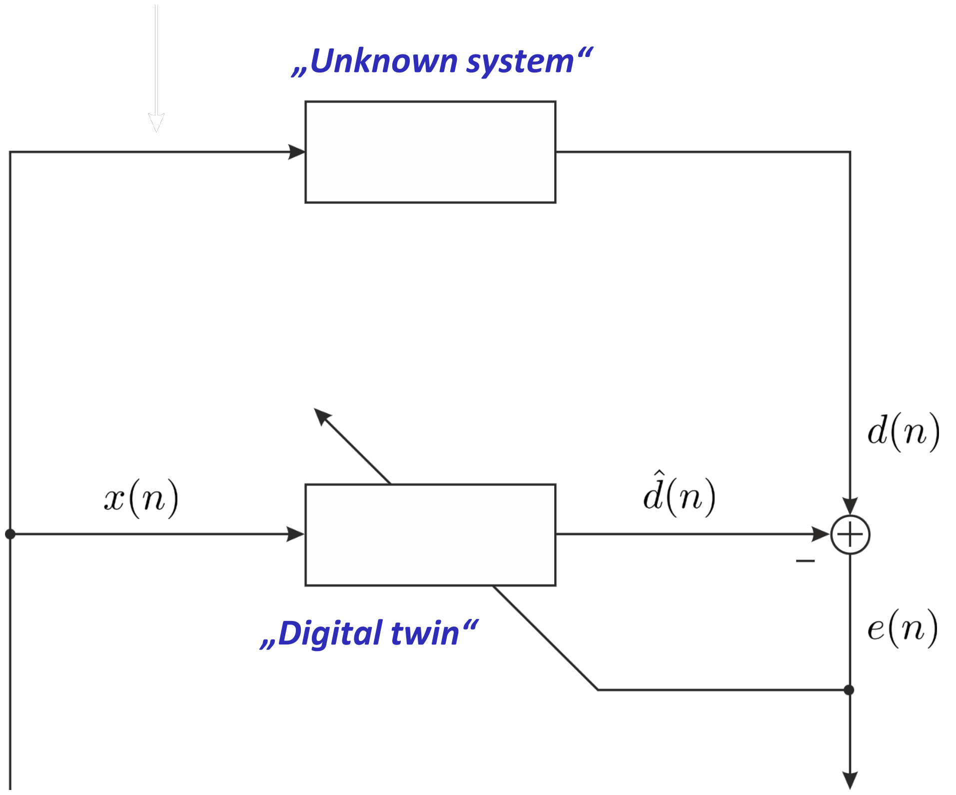
Cost Functions – Part 6

Signal-based error criteria – Part 6:

- Basic setup with access to excitation signals (not always given)
- Completely digital and noise-free version
- Linear systems of order one (for the unknown system and for the twin)

$$d(n) = \sum_{i=0}^{1} h_i \, x(n-i)$$

$$\hat{d}(n) = \sum_{i=0}^{1} \hat{h}_i \, x(n-i)$$



Cost Functions – Part 6

Signal-based error criteria – Part 6:

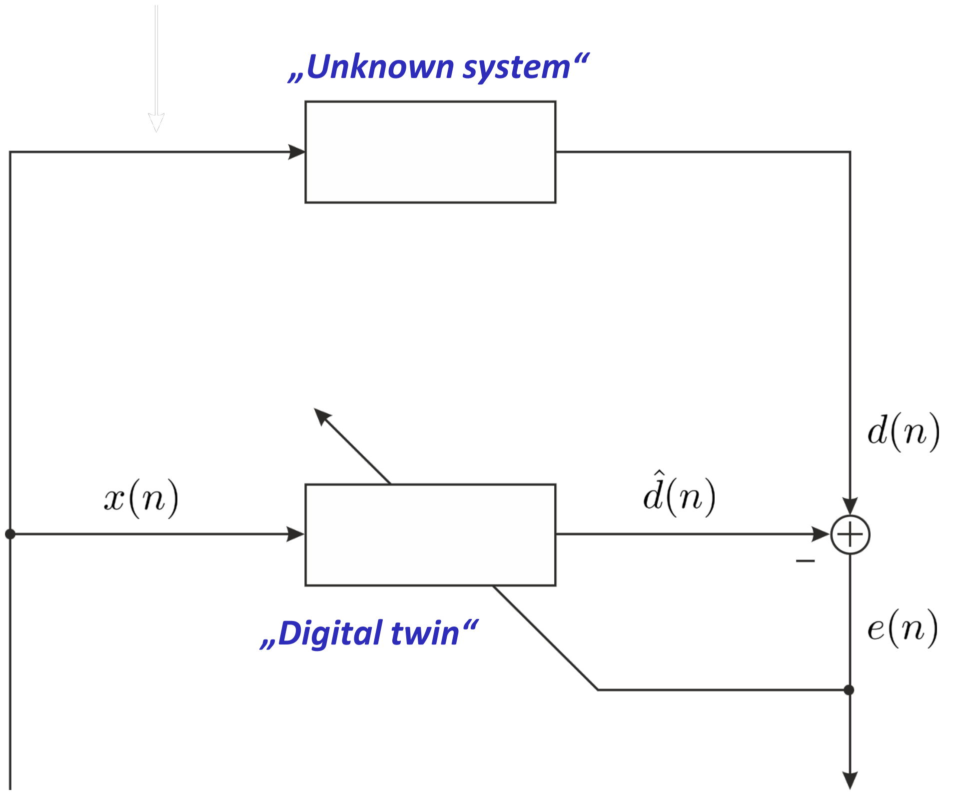
- Basic setup with access to excitation signals (not always given)
- Completely digital and noise-free version
- Linear systems of order one (for the unknown system and for the twin)

$$d(n) = \sum_{i=0}^{1} h_i\, x(n-i)$$

$$\hat{d}(n) = \sum_{i=0}^{1} \hat{h}_i\, x(n-i)$$

- For the average error power we get

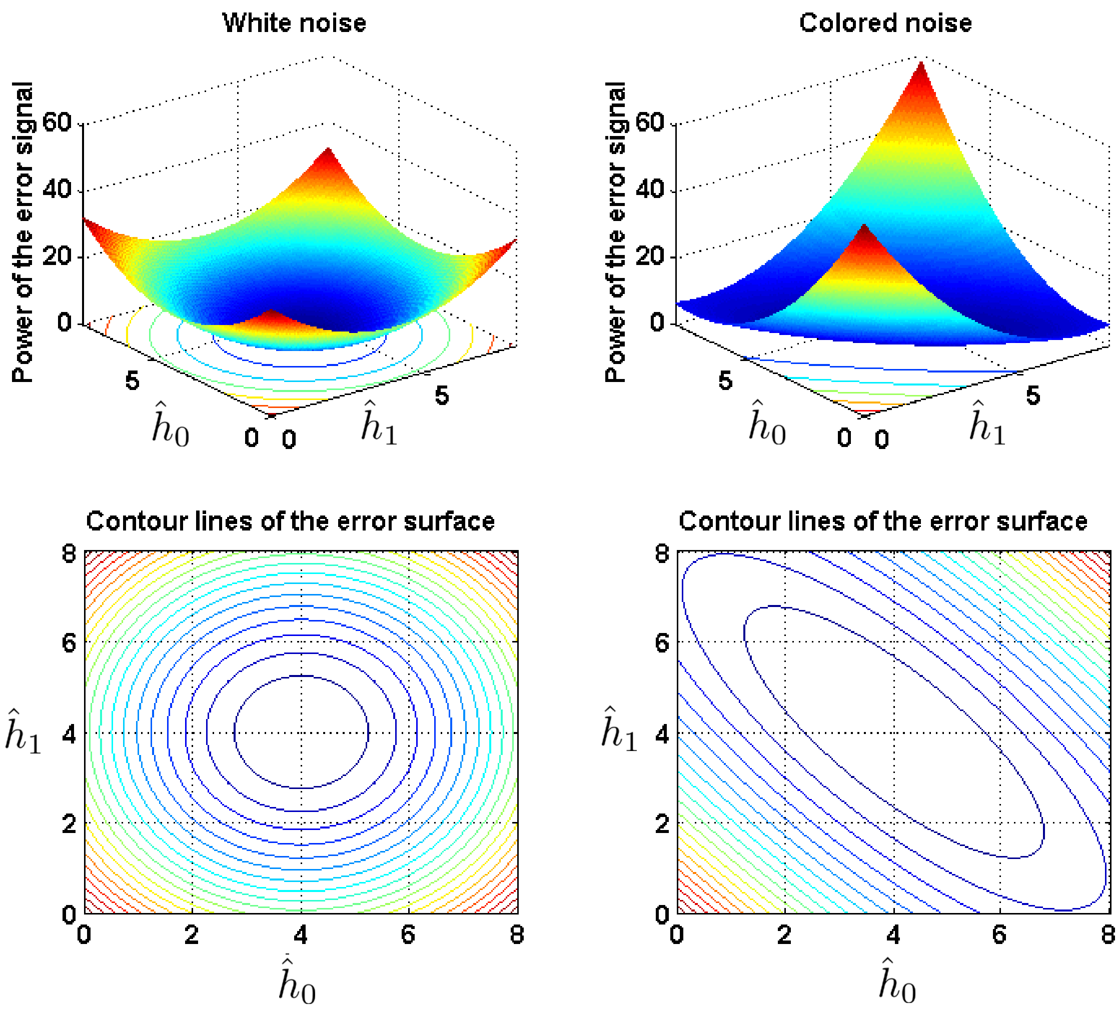
$$\mathrm{E}\{e^2(n)\} = \big[\boldsymbol{h} - \hat{\boldsymbol{h}}\big]^{\mathrm{T}} \boldsymbol{R}_{xx} \big[\boldsymbol{h} - \hat{\boldsymbol{h}}\big].$$



Cost Functions – Part 7

Signal-based error criteria – Part 6:

- Error surface for
  - $\boldsymbol{R}_{xx} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
  - $\boldsymbol{R}_{xx} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$
- Properties
- Unique minimum (no local minima).
- Error surface depends on the correlation properties of the input signal.



Cost Functions – Part 8

Density-based error criteria – Part 1:

- When processes and their properties should be identified or estimated, cost functions based on error signal are not always appropriate. Instead densities can be approximated by ***histograms*** and density-based cost functions such as the ***Kullback-Leibler divergence*** can be used:

$$D_{\mathrm{KL}}\Big(\{p_i\},\,\{q_i\}\Big) \;=\; \sum_i p_i\,\log_{10}\left\{\frac{p_i}{q_i}\right\}$$

$$D_{\mathrm{KL}}\Big(f_p(\boldsymbol{x}),\,f_q(\boldsymbol{x})\Big) \;=\; \int\limits_{\boldsymbol{x}} f_p(\boldsymbol{x})\,\log_{10}\left\{\frac{f_p(\boldsymbol{x})}{f_q(\boldsymbol{x})}\right\} d\boldsymbol{x}$$

- Here p_i and q_i are the discrete probabilities of the processes $\boldsymbol{p}(n)$ and $\boldsymbol{q}(n)$. $f_p(\boldsymbol{x})$ and $f_q(\boldsymbol{x})$ are the probability density functions of the processes $\boldsymbol{p}(n)$ and $\boldsymbol{q}(n)$.
- Besides the logarithm to the basis 10 sometimes also the ***natural logarithm*** (to the basis e) is used.
- Next these above mentioned distances will be explained by a ***simple discrete example***.

Cost Functions – Part 9

Density-based error criteria – Part 2:

- An example measurement (N samples) of a binary source $b(n) \in \{0,\,1\}$ is given:

$$\{b(n)\} \;=\; \{0,\,1,\,0,\,1,\,1,\,0,\,0,\,1,\,1,\,0\}.$$

- We assume to have now two binary sources
  - $p(n)$ with $p_0 = \dfrac{1}{2}$ and $p_1 = 1 - p_0 = \dfrac{1}{2}$,
  - $q(n)$ with $q_0 = \dfrac{3}{4}$ and $q_1 = 1 - q_0 = \dfrac{1}{4}$,

  and we want to compute the (normalized logarithmic) ratio of the probabilities that the sequence above is generated by the individual sources:

$$R \;=\; \frac{P\big(\{b(n)\,\big|\,p(n)\,\text{created the sequence}\}\big)}{P\big(\{b(n)\,\big|\,q(n)\,\text{created the sequence}\}\big)} \quad \text{and} \quad R_{\log} \;=\; \frac{1}{N}\log_{10}\left\{\frac{P\big(\{b(n)\,\big|\,p(n)\,\text{created the sequence}\}\big)}{P\big(\{b(n)\,\big|\,q(n)\,\text{created the sequence}\}\big)}\right\}.$$

Cost Functions – Part 10

Density-based error criteria – Part 3:

- The example measurement (N samples) again:

$$\{b(n)\} = \{0,\,1,\,0,\,1,\,1,\,0,\,0,\,1,\,1,\,0\}.$$

- The probability that source $p(n)$ has created the sequence:

$$\begin{aligned} P\big(\{b(n)\}\,|\,p(n)\,\text{created the sequence}\big) &= p_0\,p_1\,p_0\,p_1\,p_1\,p_0\,p_0\,p_1\,p_1\,p_0 \\ &= p_0^{N_0}\,p_1^{N_1}. \end{aligned}$$

- In a similar manner we can compute the probability that $q(n)$ has created the observation:

$$\begin{aligned} P\big(\{b(n)\}\,|\,q(n)\,\text{created the sequence}\big) &= q_0\,q_1\,q_0\,q_1\,q_1\,q_0\,q_0\,q_1\,q_1\,q_0 \\ &= q_0^{N_0}\,q_1^{N_1}. \end{aligned}$$

- This leads to the probability ratio

$$R = \frac{P\big(\{b(n)\,\big|\,p(n)\,\text{created the sequence}\}\big)}{P\big(\{b(n)\,\big|\,q(n)\,\text{created the sequence}\}\big)} = \frac{p_0^{N_0}\,p_1^{N_1}}{q_0^{N_0}\,q_1^{N_1}}.$$

Cost Functions – Part 11

Density-based error criteria – Part 4:

- Again the result of the last slide:

$$R = \frac{P\big(\{b(n)\,|\,p(n)\,\text{created the sequence}\}\big)}{P\big(\{b(n)\,|\,q(n)\,\text{created the sequence}\}\big)} = \frac{p_0^{N_0}\,p_1^{N_1}}{q_0^{N_0}\,q_1^{N_1}}.$$

- Now we can compute the normalized logarithmic ratio

$$R_{\log} = \frac{1}{N}\log_{10}\left\{\frac{P\big(\{b(n)\,|\,p(n)\,\text{created the sequence}\}\big)}{P\big(\{b(n)\,|\,q(n)\,\text{created the sequence}\}\big)}\right\}$$

… inserting the result from above …

$$= \frac{1}{N}\log_{10}\left\{\frac{p_0^{N_0}\,p_1^{N_1}}{q_0^{N_0}\,q_1^{N_1}}\right\}$$

… simplifying the logarithm …

$$= \frac{1}{N}\left[\log_{10}\left\{p_0^{N_0}\right\} + \log_{10}\left\{p_1^{N_1}\right\} - \log_{10}\left\{q_0^{N_0}\right\} - \log_{10}\left\{q_1^{N_1}\right\}\right]$$

Cost Functions – Part 12

Density-based error criteria – Part 5:

- Again the result of the last slide:

$$R_{\log} = \frac{1}{N}\Big[\log_{10}\big\{p_0^{N_0}\big\} + \log_{10}\big\{p_1^{N_1}\big\} - \log_{10}\big\{q_0^{N_0}\big\} - \log_{10}\big\{q_1^{N_1}\big\}\Big]$$

… simplifying the powers within the logarithms …

$$= \frac{N_0}{N}\log_{10}\{p_0\} + \frac{N_1}{N}\log_{10}\{p_1\} - \frac{N_0}{N}\log_{10}\{q_0\} - \frac{N_1}{N}\log_{10}\{q_1\}$$

… combining the terms with the same weighting …

$$= \frac{N_0}{N}\log_{10}\left\{\frac{p_0}{q_0}\right\} + \frac{N_1}{N}\log_{10}\left\{\frac{p_1}{q_1}\right\}.$$

- If we assume that the source $p(n)$ has created the sequence $b(n)$, than we can approximate (especially for large N) the following terms:
  - $\frac{N_0}{N} \approx p_0,$
  - $\frac{N_1}{N} \approx p_1,$

Cost Functions – Part 13

Density-based error criteria – Part 6:

- Again the result of the last slide:

$$R_{\log} = \frac{N_0}{N} \log_{10}\left\{\frac{p_0}{q_0}\right\} + \frac{N_1}{N} \log_{10}\left\{\frac{p_1}{q_1}\right\}$$

… inserting the assumption/approximation from the last slide …

$$\approx p_0 \log_{10}\left\{\frac{p_0}{q_0}\right\} + p_1 \log_{10}\left\{\frac{p_1}{q_1}\right\}$$

… writing the two terms as a sum …

$$= \sum_{i=0}^{1} p_i \log_{10}\left\{\frac{p_i}{q_i}\right\}.$$

- Comparing this result with the definition of the Kullback-Leibler divergence shows (hopefully) the meaning the cost function:

$$D_{\mathrm{KL}}\Big(\{p_i\},\, \{q_i\}\Big) = \sum_{i} p_i \log_{10}\left\{\frac{p_i}{q_i}\right\}, \qquad D_{\mathrm{KL}}\Big(f_p(\boldsymbol{x}),\, f_q(\boldsymbol{x})\Big) = \int_{\boldsymbol{x}} f_p(\boldsymbol{x}) \log_{10}\left\{\frac{f_p(\boldsymbol{x})}{f_q(\boldsymbol{x})}\right\} d\boldsymbol{x}.$$

Cost Functions – Part 14

Density-based error criteria – Part 7:

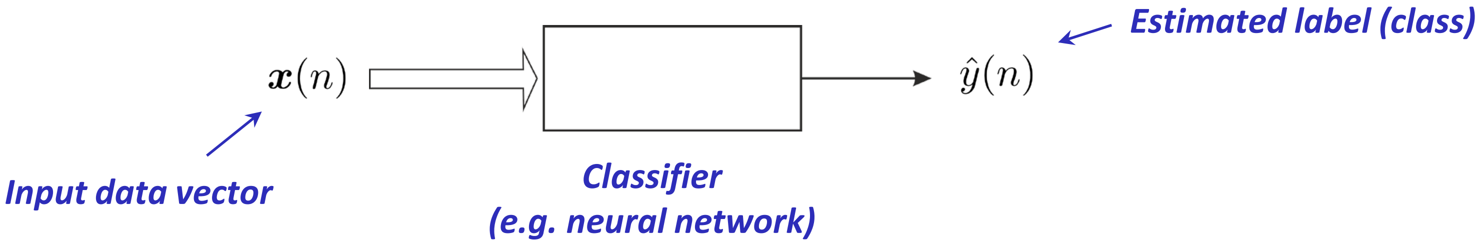
- One last item – cross entropy as a cost function and its relation with the Kullback-Leibler divergence.
- First of all the definition of ***cross entropy loss***:

$$D_{\text{CEL}}\Big(\{p_i\},\,\{q_i\}\Big) \;=\; -\sum_i p_i\,\log_{10}\{q_i\}$$

$$D_{\text{CEL}}\Big(f_p(\boldsymbol{x}),\,f_q(\boldsymbol{x})\Big) \;=\; -\int_{\boldsymbol{x}} f_p(\boldsymbol{x})\,\log_{10}\big\{f_q(\boldsymbol{x})\big\}\,d\boldsymbol{x}$$

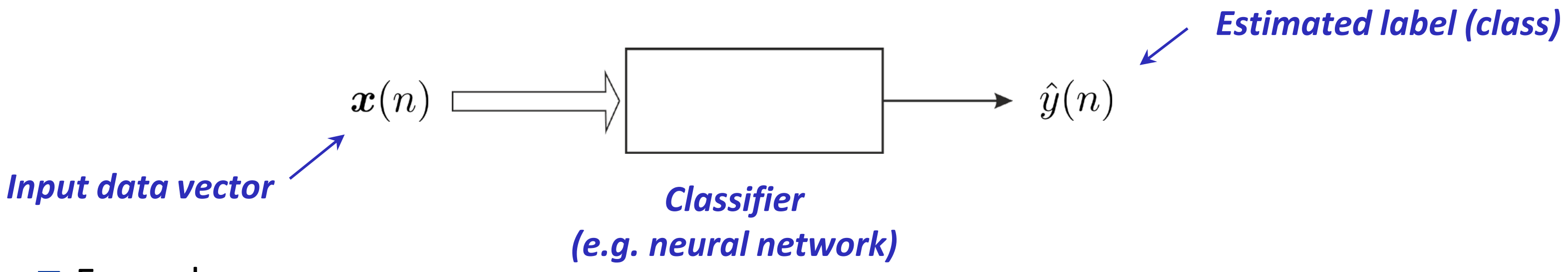
Usually cross entropy is defined with the minus. However, here we will use it as a loss function and therefore the minus was "added".

- In order to understand the application of cross entropy as a cost function, we need to introduce some classification structure (e.g. based on neural networks) that we will discuss later in more detail:



Density-based error criteria – Part 7:

- Again the basic structure:



- Example

$\boldsymbol{x}(0)$

$y(0) = 1$

$\boldsymbol{x}(1)$

$y(1) = 0$

0 for lizard
1 for hippo
2 for cat
3 for dog

Cost Functions – Part 15

Density-based error criteria – Part 8:

- Instead of a hard classification often soft classifications are used

$$\boldsymbol{x}(n) \Rightarrow \boxed{\phantom{\text{Classifier}}} \Rightarrow \hat{\boldsymbol{y}}(n)$$

Input data vector

Classifier (e.g. neural network)

Estimated label vector („one-hot encoded“ or probabilities)

- Example

$\boldsymbol{x}(0)$ 

$$\boldsymbol{y}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{x}(1)$ 

$$\boldsymbol{y}(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

0 for lizard
1 for hippo
2 for cat
3 for dog

Cost Functions – Part 16

Density-based error criteria – Part 9:

- Instead of a hard classification often soft classifications are used

Input data vector

Classifier (e.g. neural network)

Estimated label vector („one-hot encoded“ or probabilities)

- From “hard” to “soft”

The entries of the label vectors (as well as the ones oft the estimated leabels) respresent probabilities.

$\boldsymbol{x}(0)$ 

$$\boldsymbol{y}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{x}(1)$ 

$$\boldsymbol{y}(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

0 for lizard
1 for hippo
2 for cat
3 for dog

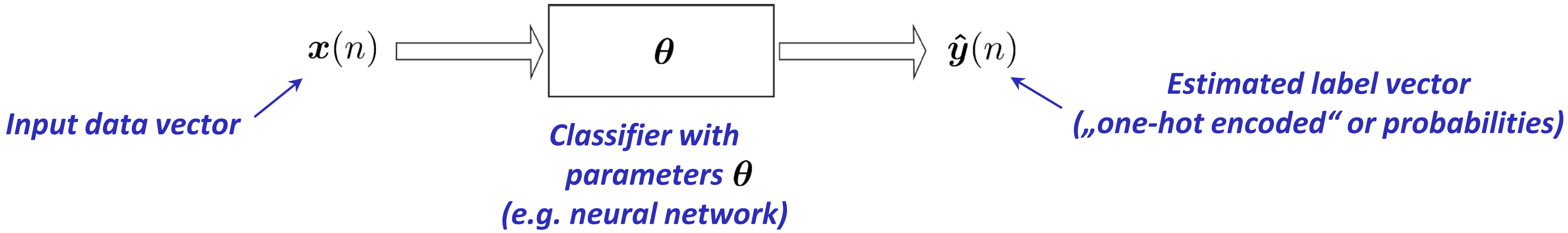
$\boldsymbol{x}(2)$ 

$$\boldsymbol{y}(2) = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}$$

Cost Functions – Part 17

Density-based error criteria – Part 10:

- Again our structure



- We have the conditional output distribution (during training) and we would like to learn the parameters for generating a similar conditional output distribution for the classification approach:

$$p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big),$$

$$p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big).$$

- Therefore, we can minimize the Kullback-Leibler divergence

$$D_{\mathrm{KL}}\Big(\Big\{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\Big\},\,\Big\{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)\Big\}\Big) \;=\; \sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\left\{\frac{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)}{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)}\right\}.$$

Cost Functions – Part 18

Density-based error criteria – Part 11:

- We start with the Kullback-Leibler divergence

$$D_{\mathrm{KL}}\Big(\big\{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\big\},\,\big\{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)\big\}\Big) = \sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\left\{\frac{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)}{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)}\right\}.$$

and look for the optimal parameters of the classifier

$$\boldsymbol{\theta}_{\mathrm{opt}} = \underset{\boldsymbol{\theta}}{\arg\min}\left\{\sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\left\{\frac{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)}{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)}\right\}\right\}$$

… converting the ration within the log to a difference of logs …

$$= \underset{\boldsymbol{\theta}}{\arg\min}\left\{\sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\big\{p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\big\} - \sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\big\{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)\big\}\right\}$$

… using that the first term does not depend on the parameter that should be optimized …

$$= \underset{\boldsymbol{\theta}}{\arg\min}\left\{-\sum_i p\big(y_i(n)\,\big|\,\boldsymbol{x}_i(n)\big)\,\log_{10}\big\{p\big(\hat{y}_i(n)\,\big|\,\boldsymbol{x}_i(n),\,\boldsymbol{\theta}\big)\big\}\right\}$$

Cost Functions – Part 19

Density-based error criteria – Part 12:

- Starting with the definition of our optimization

$$\boldsymbol{\theta}_{\mathrm{opt}} = \underset{\boldsymbol{\theta}}{\arg\min} \left\{ D_{\mathrm{KL}}\Big(\big\{ p\big(y_i(n) \,\big|\, \boldsymbol{x}_i(n)\big) \big\},\, \big\{ p\big(\hat{y}_i(n) \,\big|\, \boldsymbol{x}_i(n),\, \boldsymbol{\theta}\big) \big\} \Big) \right\}$$

… inserting the result of the last slide …

$$= \underset{\boldsymbol{\theta}}{\arg\min} \left\{ -\sum_i p\big(y_i(n) \,\big|\, \boldsymbol{x}_i(n)\big) \, \log_{10} \big\{ p\big(\hat{y}_i(n) \,\big|\, \boldsymbol{x}_i(n),\, \boldsymbol{\theta}\big) \big\} \right\}$$

… inserting the definition of the cross entropy loss …

$$= \underset{\boldsymbol{\theta}}{\arg\min} \left\{ D_{\mathrm{CEL}}\Big(\big\{ p\big(y_i(n) \,\big|\, \boldsymbol{x}_i(n)\big) \big\},\, \big\{ p\big(\hat{y}_i(n) \,\big|\, \boldsymbol{x}_i(n),\, \boldsymbol{\theta}\big) \big\} \Big) \right\}$$

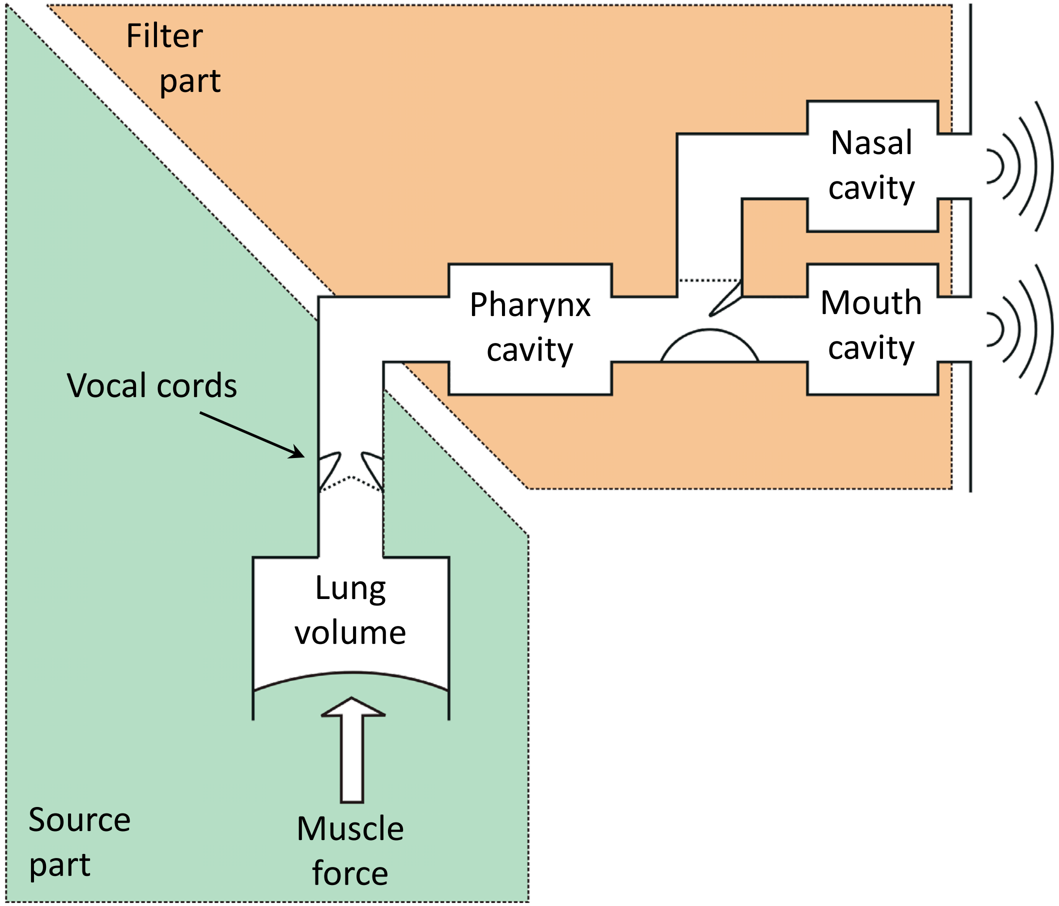
This means that optimizing the cross entropy loss is the same as optimizing the Kullback-Leibler divergence.

Cross Entropy (loss) is an often used cost function for network-based classifiers.

The previous derivations were based on the explanations of Adrian Liusie from Cambridge University. Thanks for the nice videos!

Contents

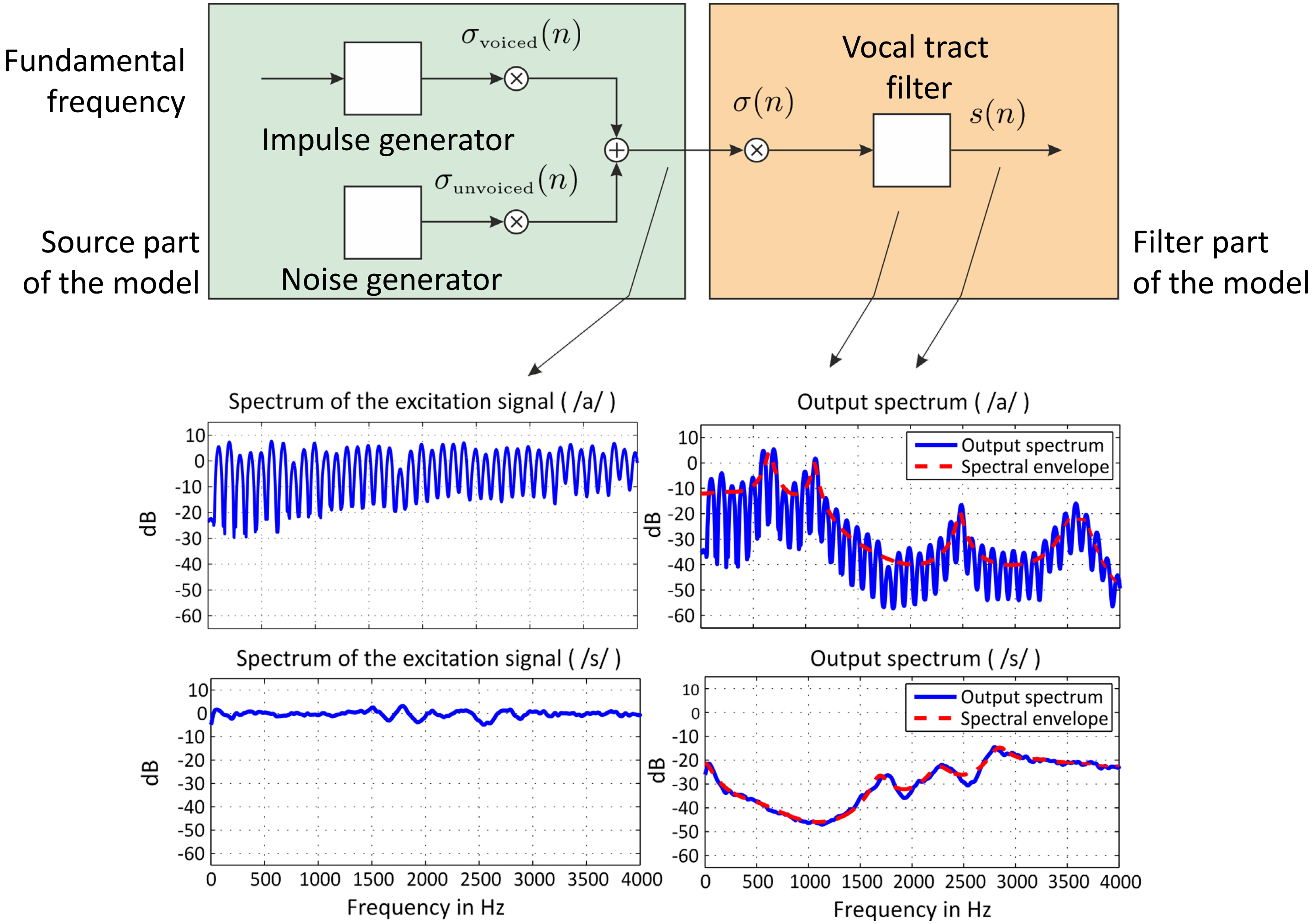
Generation of Speech Signals



Source- filter principle:

- An airflow, coming from the lungs, excites the vocal cords for voiced ***excitation*** or causes a noise-like signal (opened vocal cords).
- The mouth, nasal, and pharynx cavity are behaving like controllable resonators and only a few frequencies (called ***formant frequencies***) are not attenuated.

Source-Filter Model for Speech Generation



Properties of Speech Signals

Some basics:

- Speech signals can be modeled for short periods (about 10 ms to 30 ms) as ***weak stationary***.
  This means that the statistical properties up to second order are invariant versus temporal shifts.
- Speech contains a lot of ***pauses***. In these pauses the statistical properties of the background noise can be estimated.
- Speech has ***periodic signal components*** (fundamental frequency about 70 Hz [deep male voices up to 400 Hz [voices of children]) and ***noise-like components*** (e.g. fricatives).
- Speech signals have ***strong correlation*** at ***small lags*** on the one hand and ***around the pitch period*** (and multitudes of it) on the other hand.
- In various application the ***short-term spectral envelope*** is used for determining what is said (speech recognition) and who said it (speaker recognition/verification).

Wiener Filter – Part 1

Filter design by means of minimizing the squared error (according to Gauß)

Independent development

1941: A. Kolmogoroff: *Interpolation und Extrapolation von stationären zufälligen Folgen*, Izv. Akad. Nauk SSSR Ser. Mat. 5, pp. 3 – 14, 1941 (in Russian)

1942: N. Wiener: *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, J. Wiley, New York, USA, 1949 (originally published in 1942 as MIT Radiation Laboratory Report)

Assumptions / design criteria:

- Design of a filter that separates a desired signal optimally from additive noise
- Both signals are described as stationary random processes
- Knowledge about the statistical properties up to second order is necessary

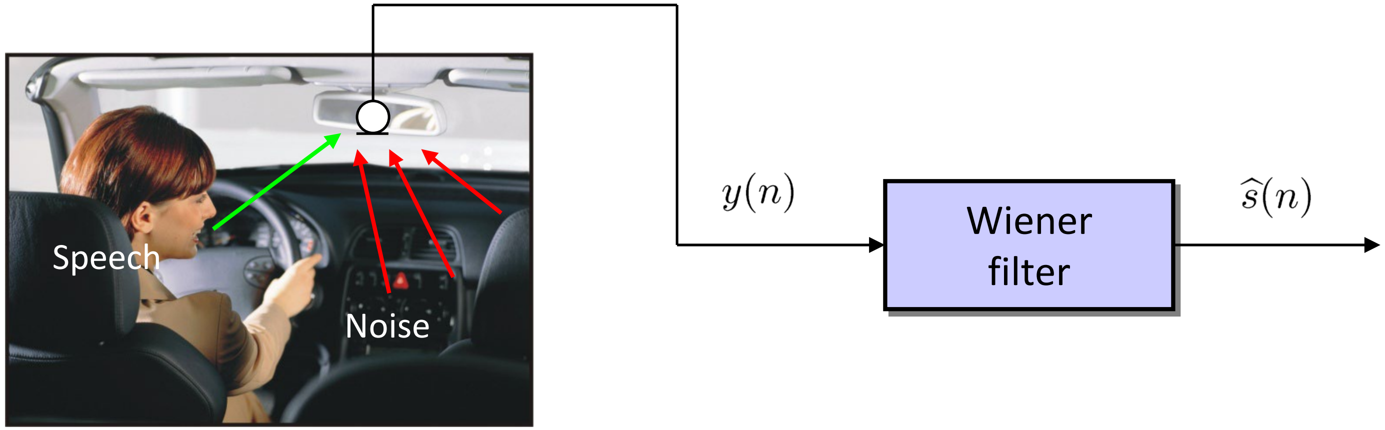
Literature about the Wiener Filter

Basics of the Wiener filter:

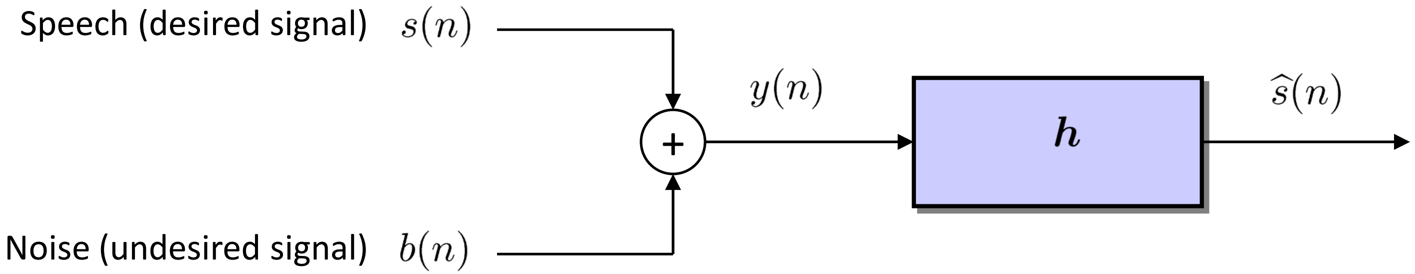
- E. Hänsler / G. Schmidt: ***Acoustic Echo and Noise Control – Chapter 5 (Wiener Filter)***, Wiley, 2004
- E. Hänsler: ***Statistische Signale: Grundlagen und Anwendungen – Chapter 8*** (Optimalfilter nach Wiener und Kolmogoroff), Springer, 2001 (in German)
- M. S.Hayes: ***Statistical Digital Signal Processing and Modeling – Chapter 7 (Wiener Filtering)***, Wiley, 1996
- S. Haykin: ***Adaptive Filter Theory – Chapter 2 (Wiener Filters)***, Prentice Hall, 2002

Wiener-Filter – Teil 2

Application example:



Model:

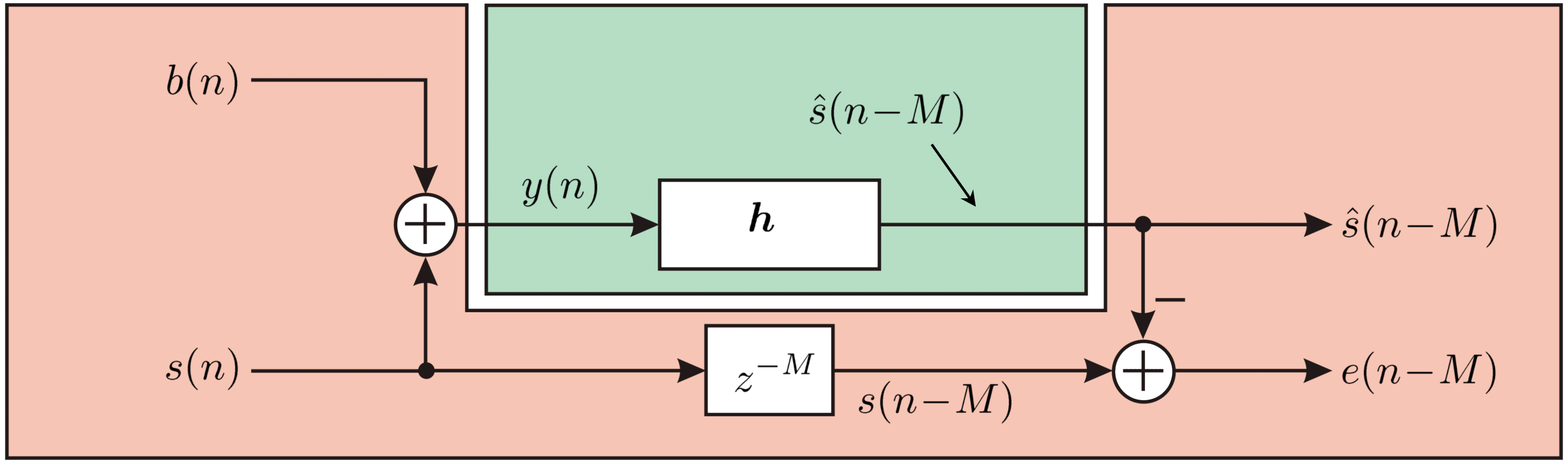


The Wiener solution if often applied in a "block-based fashion".

Christian-Albrechts-Universität zu Kiel

Wiener Filter – Part 3

Time-domain structure:



FIR structure:

$$\hat{s}(n-M) = \sum_{i=0}^{N-1} h_i \, y(n-i)$$

Optimization criterion:

$$\mathrm{E}\{e^2(n-M)\} \underset{h_i = h_{i,\mathrm{opt}}}{\longrightarrow} \min$$

This is only one of a variety of optimization criteria (topic for a talk)!

Wiener Filter – Part 4

Assumptions:

- The desired signal $s(n)$ and the distortion $b(n)$ are uncorrelated and have zero mean, i.e. they are orthogonal:

$$\mu_s = \mu_b = 0, \; s_{sb}(l) = \mu_s \, \mu_b = 0.$$

Computing the optimal filter coefficients:

$$\mathrm{E}\{e^2(n-M)\} \underset{h_i = h_{i,\mathrm{opt}}}{\longrightarrow} \min$$

$$\begin{aligned} \frac{d}{dh_i}\mathrm{E}\{e^2(n-M)\}\Big|_{h_i=h_{i,\mathrm{opt}}} &= 0 \\ 2\,\mathrm{E}\Big\{e(n-M)\,\frac{d}{dh_i}e(n-M)\Big\}\Big|_{h_i=h_{i,\mathrm{opt}}} &= 0 \end{aligned}$$

Wiener Filter – Part 5

Computing the optimum filter coefficients (continued):

$$2\,\mathrm{E}\left\{e(n-M)\,\frac{d}{dh_i}e(n-M)\right\}\bigg|_{h_i=h_{i,\mathrm{opt}}} = 0$$

Inserting the error signal: $e(n-M) = s(n-M) - \sum_{i=0}^{N-1} h_i\, y(n-i)$

$$2\,\mathrm{E}\left\{\left(s(n-M) - \sum_{j=0}^{N-1} h_j\, y(n-j)\right) y(n-i)\right\}\bigg|_{h_i=h_{i,\mathrm{opt}}} = 0$$

$$s_{sy}(i-M) - \sum_{j=0}^{N-1} h_{j,\mathrm{opt}}\, s_{yy}(i-j) = 0$$

Exploiting orthogonality of the input components: $s_{sy}(l) = s_{ss}(l) + \underbrace{s_{sb}(l)}_{=0} = s_{ss}(l)$

$$s_{ss}(i-M) - \sum_{j=0}^{N-1} h_{j,\mathrm{opt}}\, s_{yy}(i-j) = 0$$

True for i = 0 … N-1.

Wiener Filter – Part 6

Computing the optimum filter coefficients (continued):

$$\begin{bmatrix} s_{yy}(0) & s_{yy}(1) & \dots & s_{yy}(N-1) \\ s_{yy}(1) & s_{yy}(0) & \dots & s_{yy}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ s_{yy}(N-1) & s_{yy}(N-2) & \dots & s_{yy}(0) \end{bmatrix} \begin{bmatrix} h_{0,\text{opt}} \\ h_{1,\text{opt}} \\ \vdots \\ h_{N-1,\text{opt}} \end{bmatrix} = \begin{bmatrix} s_{ss}(-M) \\ s_{ss}(-M+1) \\ \vdots \\ s_{ss}(N-M-1) \end{bmatrix}$$

Problems:

- The autocorrelation of the undisturbed signal is not directly measurable.

  Solution: $s_{ss}(l) = s_{yy}(l) - s_{bb}(l)$ and estimation of the autocorrelation of the noise during speech pauses.

- The inversion of the autocorrelation matrix might lead to stability problems (because the matrix is only non-negative definite).

  Solution: Solution in the frequency domain (see next slides).

- The solution of the equation system is computationally complex (especially for large filter orders) and has to be computed quite often (every 1 to 20 ms).

  Solution: Solution in the frequency domain (see next slides).

Solution/Approximation in the Frequency Domain – Part 1

Solution in the time domain:

$$s_{ss}(i-M) - \sum_{j=0}^{N-1} h_{j,\mathrm{opt}}\, s_{yy}(i-j) \quad = \quad 0$$

Delayless solution:

$$s_{ss}(i) - \sum_{j=0}^{N-1} h_{j,\mathrm{opt}}\, s_{yy}(i-j) \quad = \quad 0$$

Removing the „FIR“ restriction:

$$s_{ss}(i) \; - \sum_{j=-\infty}^{\infty} h_{j,\mathrm{opt}}\, s_{yy}(i-j) \quad = \quad 0$$

Solution/Approximation in the Frequency Domain – Part 2

Solution in the time domain:

$$s_{ss}(i) - \sum_{j=-\infty}^{\infty} h_{j,\mathrm{opt}}\, s_{yy}(i-j) \quad = \quad 0$$

Solution in the frequency domain:

$$\begin{aligned} S_{ss}(\Omega) - H_{\mathrm{opt}}(e^{j\Omega})\, S_{yy}(\Omega) &= 0 \\ H_{\mathrm{opt}}(e^{j\Omega}) &= \frac{S_{ss}(\Omega)}{S_{yy}(\Omega)} \end{aligned}$$

Inserting orthogonality of the input components: $S_{ss}(\Omega) = S_{yy}(\Omega) - S_{bb}(\Omega)$

$$H_{\mathrm{opt}}(e^{j\Omega}) \quad = \quad 1 - \frac{S_{bb}(\Omega)}{S_{yy}(\Omega)}$$

Solution/Approximation in the Frequency Domain – Part 3

Solution in the frequency domain:

$$H_{\text{opt}}(e^{j\Omega}) \quad = \quad 1 - \frac{S_{bb}(\Omega)}{S_{yy}(\Omega)}$$

Approximation using short-term estimators:

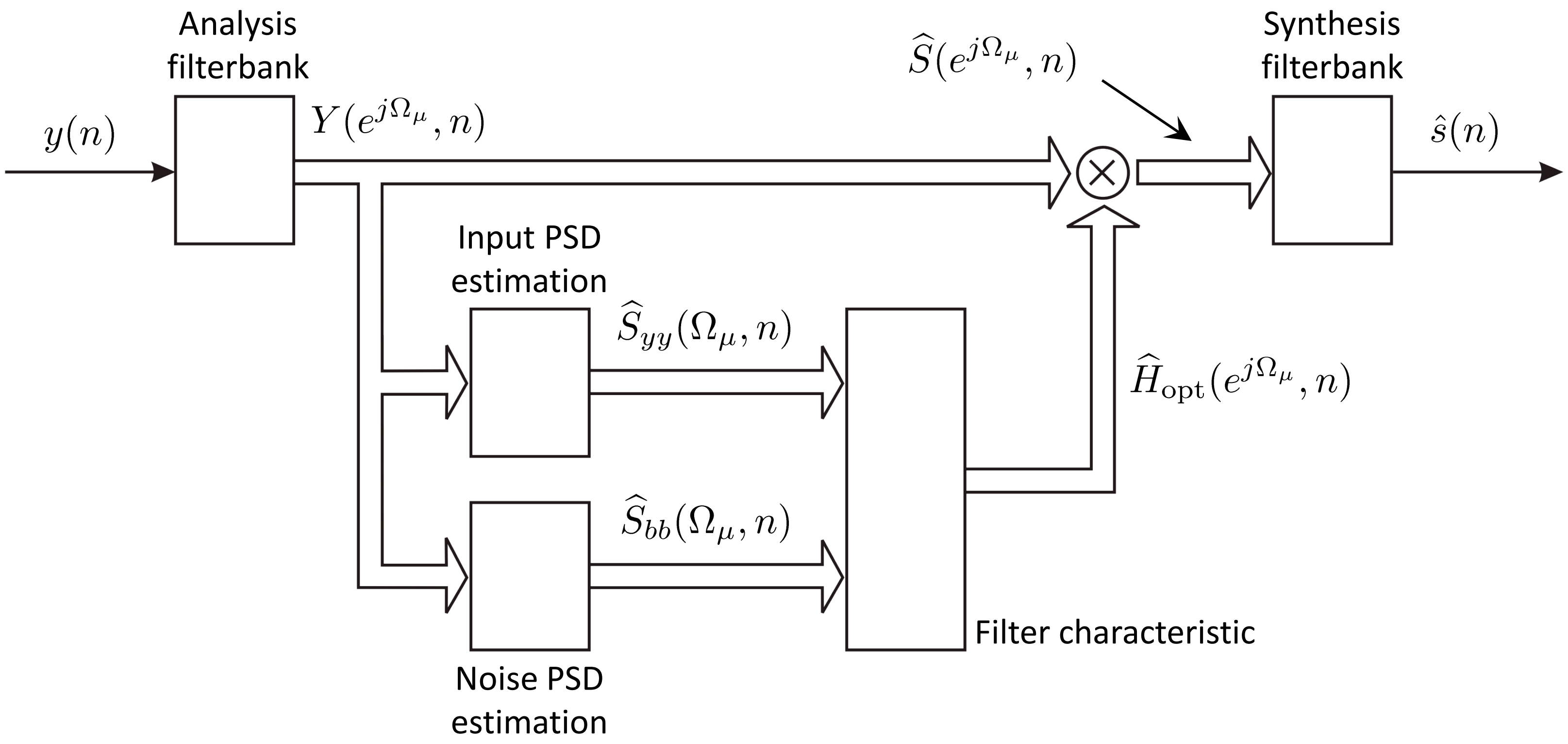
$$\widehat{H}_{\text{opt}}(e^{j\Omega}, n) \quad = \quad \max\left\{0,\, 1 - \frac{\widehat{S}_{bb}(\Omega, n)}{\widehat{S}_{yy}(\Omega, n)}\right\}$$

Typical setups:

- Realization using a filterbank system (attenuation in the subband domain).
- The analysis windows of the analysis filterbank are usually about 15 ms to 100 ms long.
  The synthesis windows are often of the same length, but sometimes also shorter.
- The frame shift is often set to 1 … 20 ms (depending on the application).
- The basic characteristic is often extended (adaptive overestimation, adaptive maximum attenuation, etc..

Solution/Approximation in the Frequency Domain – Part 4

Frequency-domain structure:



PSD = power spectral density

Solution/Approximation in the Frequency Domain – Part 5

Estimation of the (short-term) power spectral density of the input signal:

$$\widehat{S}_{yy}(\Omega_\mu, n) = \left|Y(e^{j\Omega_\mu}, n)\right|^2$$

Estimation of the (short-term) power spectral density of the background noise:

Schemes based on speech activity/pause destection (VAD)

Tracking of temporal minima

Solution/Approximation in the Frequency Domain – Part 6

Scheme with speech activity/pause detection

$$\widehat{S}_{bb}(\Omega_\mu, n) = \begin{cases} \beta\, \widehat{S}_{bb}(\Omega_\mu, n-1) + (1-\beta)\, \widehat{S}_{yy}(\Omega_\mu, n), & \text{during speech pauses,} \\ \widehat{S}_{bb}(\Omega_\mu, n-1), & \text{else.} \end{cases}$$

Temporal minima tracking:

$$\overline{S_{yy}(\Omega_\mu, n)} = \beta\, \overline{S_{yy}(\Omega_\mu, n-1)} + (1-\beta)\, \widehat{S}_{yy}(\Omega_\mu, n)$$

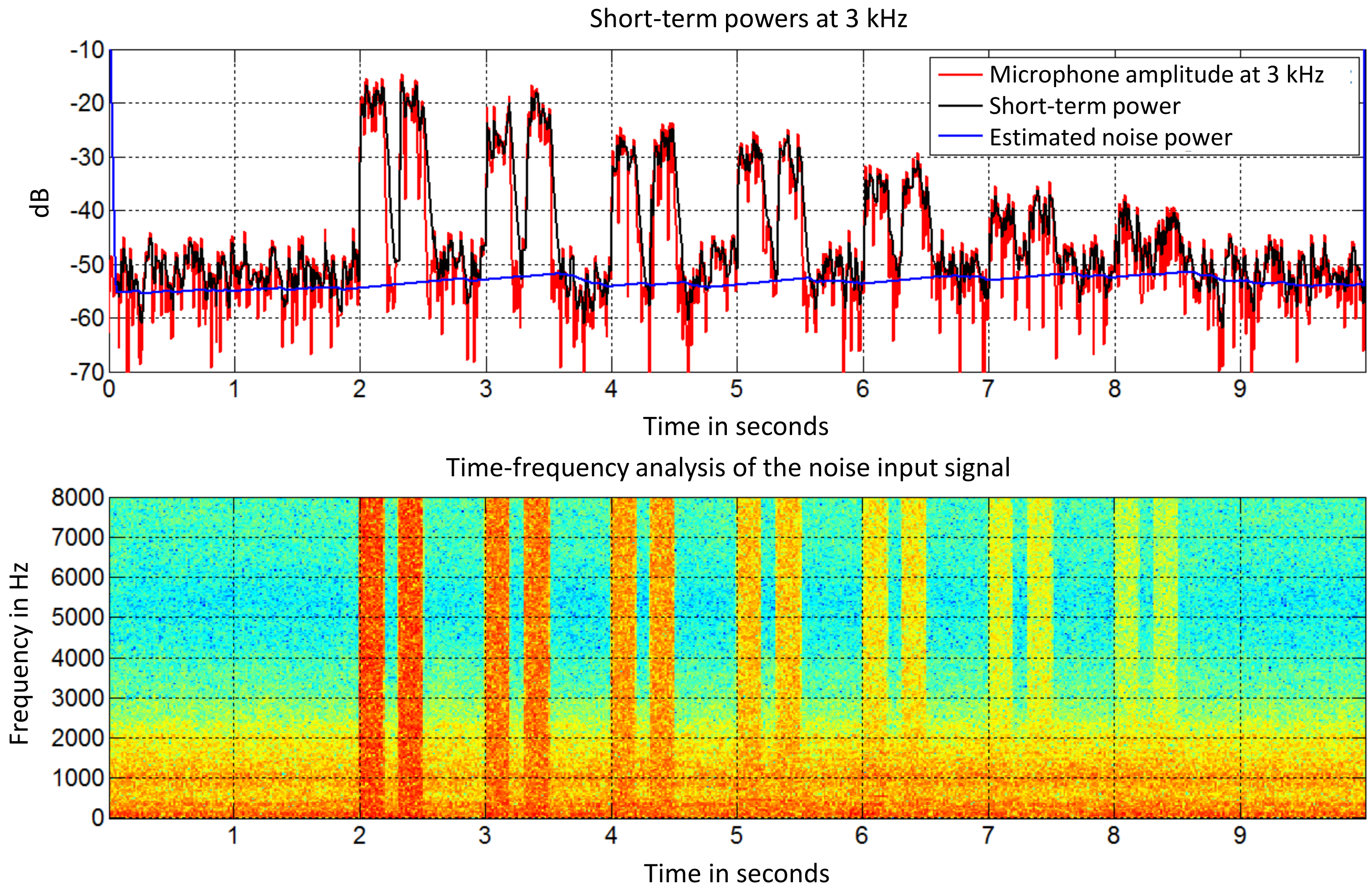
Constant slighty larger than 1

Bias correction

$$\widehat{S}_{bb}(\Omega_\mu, n) = K \begin{cases} \max\left\{S_{\text{min}},\, \widehat{S}_{bb}(\Omega_\mu, n-1)\right\} \Delta_{\text{inc}}, \\ \qquad \text{if } \overline{S_{yy}(\Omega_\mu, n)} > \widehat{S}_{bb}(\Omega_\mu, n-1), \\ \max\left\{S_{\text{min}},\, \widehat{S}_{bb}(\Omega_\mu, n-1)\right\} \Delta_{\text{dec}}, \\ \qquad \text{else.} \end{cases}$$

Constant slighty smaller than 1

Solution/Approximation in the Frequency Domain – Part 7



Extensions for the Wiener Characteristic – Overestimation of the Noise (Part 1)

Problem:

- In most estimation algorithms the estimated power spectral density of noise input signal will have ***more fluctuations*** than the corresponding estimated power spectral density of the noise. This leads to so-called ***musical noise*** (explanation in the next slides).

First solution:

- By introducing a so-called fixed ***overestimation***

$$\widehat{S}_{bb}(\Omega_\mu, n) \longrightarrow K_{\text{over}}\, \widehat{S}_{bb}(\Omega_\mu, n)$$

the undesired "opening" during speech pauses of the noise suppression filter can be avoided. However, this leads to a ***lower signal quality during speech activity***.

Extensions for the Wiener Characteristic – Overestimation of the Noise (Part 2)

Second solution:

- By replacing the fixed ***overestimation*** with an ***adaptive*** one (strong overestimation during speech pauses, no overestimation during speech activity), the drawbacks of the fixed overestimation can be avoided.
- An adaptive overestimation can be computed in a simple manner by ***using the filter coefficients of the previous frame***:

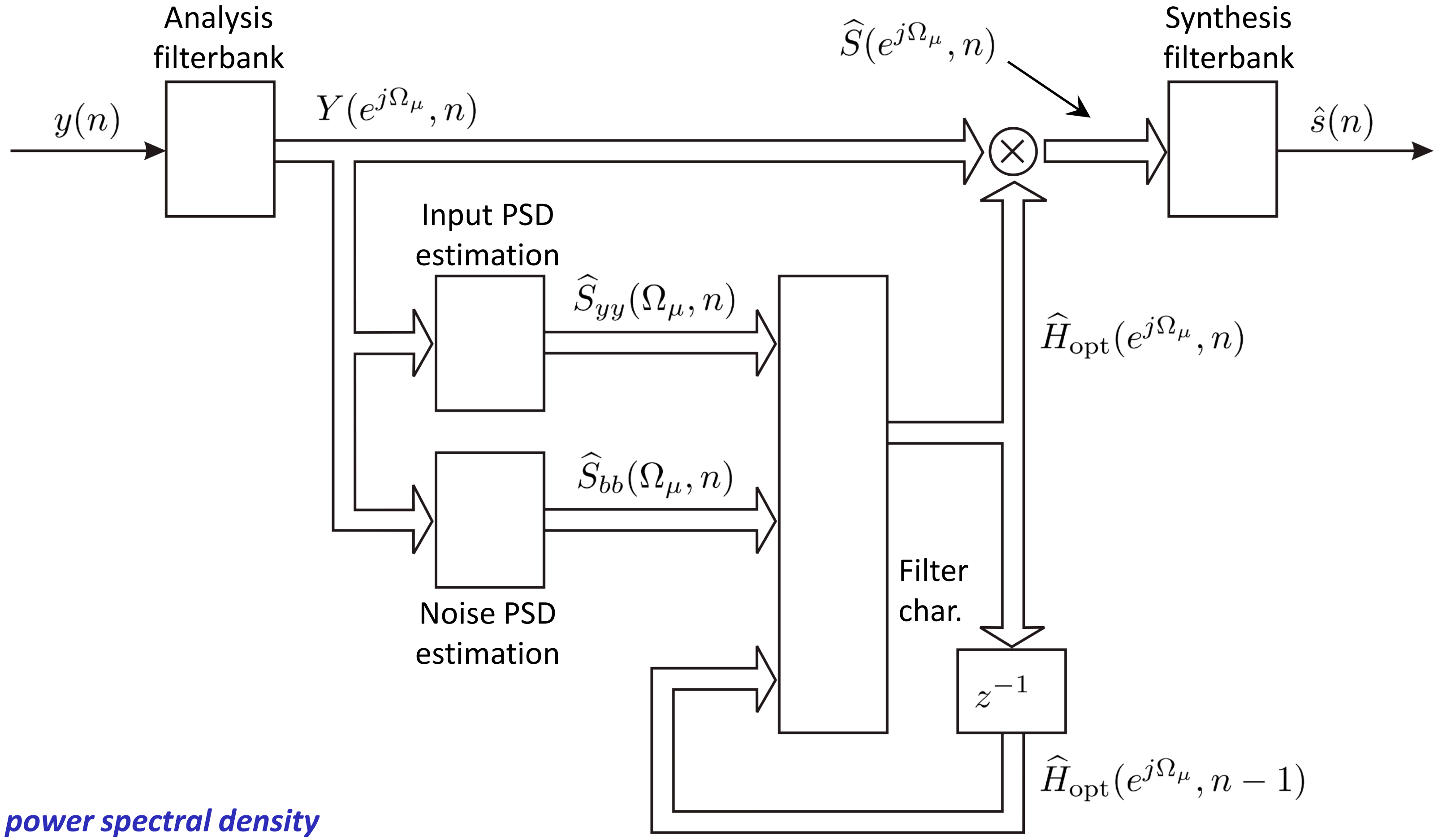
$$\widehat{S}_{bb}(\Omega_\mu, n) \longrightarrow \frac{1}{\widetilde{H}(e^{j\Omega_\mu}, n-1)}\,\widehat{S}_{bb}(\Omega_\mu, n).$$

- In addition the filter coefficients should be ***limited*** prior to their usage (otherwise the overestimation might be to strong):

$$\widetilde{H}(e^{j\Omega_\mu}, n) \quad = \quad \max\left\{\frac{1}{K_{\text{over}}},\, \widehat{H}_{\text{opt}}(e^{j\Omega_\mu}, n)\right\}.$$

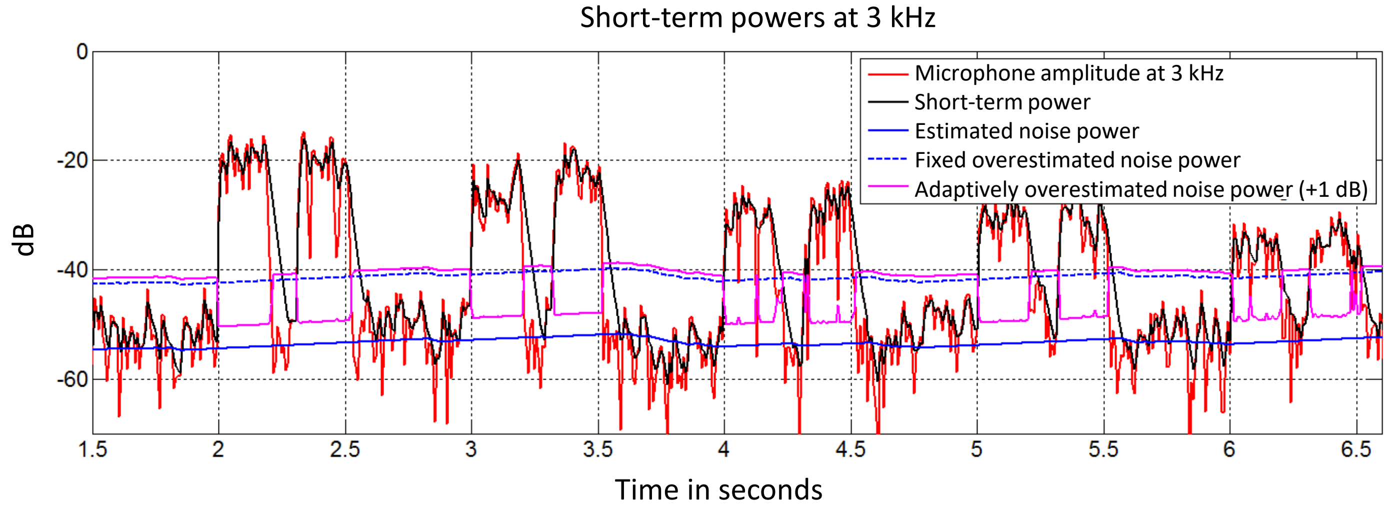
Extensions for the Wiener Characteristic – Overestimation of the Noise (Part 3)

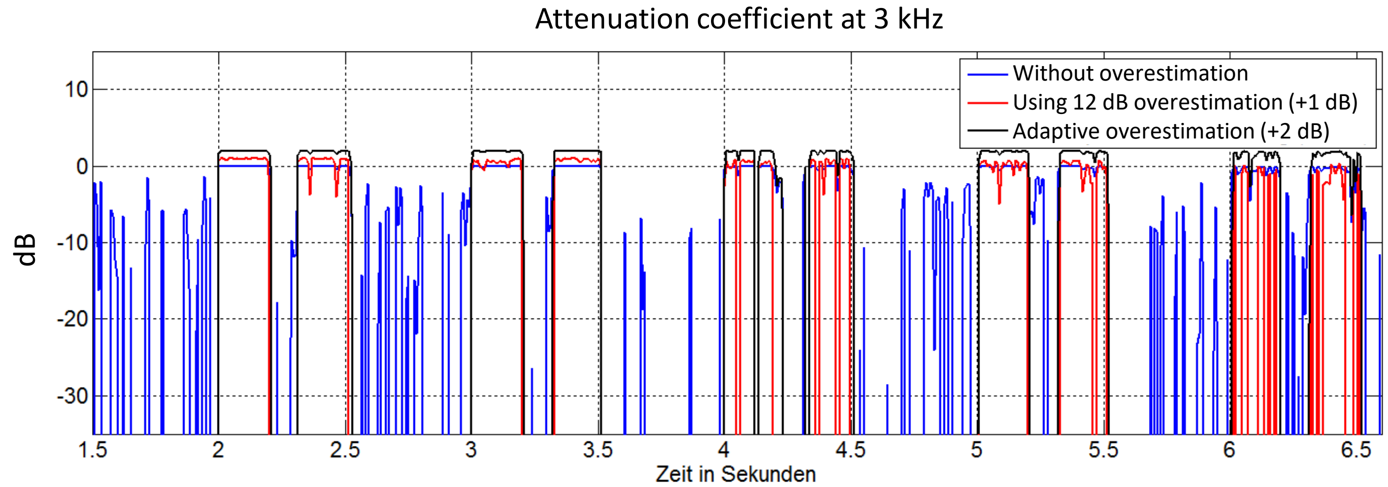
„Rekursives“ Wiener-Filter:



PSD = power spectral density

Extensions for the Wiener Characteristic – Overestimation of the Noise (Part 4)





: Microphone signal

: Output without overestimation

: Output with fixed over estimation

: Output with adaptive over estimation

Extensions for the Wiener Characteristic – Maximum Attenuation (Part 1)

Problem:

- If we would try to get rid of the noise completely, we would also loose the (acoustic) ***information about the environment*** in which the person is speaking. As a result it turned out that a ***noise reduction is better than a complete removal***.
- In addition, it's very ***complicated*** to design a high quality noise suppression that removes all noise.

Solution – Limiting the maximum filter attenuation:

Introducing a „desired“ noise (power spectral density)	Inserting an attenuation limit

Extensions for the Wiener Characteristic – Maximum Attenuation (Part 2)

Specification of a „desired noise“:

- We can try to ***specify*** or design one (or more) ***desired background noise*** types.
- If we specify more than one type of noise (e.g. train noise, car noise, “party” noise, or noises of different cars to “transform“ one car into another) we have to ***classify*** first the original noise type.
- The filter coefficients can be ***limited*** according to:

$$\widehat{H}_{\mathrm{opt}}(e^{j\Omega_\mu}, n) \quad = \quad \max\left\{H_{\min}(e^{j\Omega_\mu}, n),\, 1 - \frac{\widehat{S}_{bb}(\Omega_\mu, n)}{\widehat{S}_{yy}(\Omega_\mu, n)}\right\}.$$

- In the simplest case we chose the ***maximum attenuation*** as follows:

$$H_{\min}(e^{j\Omega_\mu}, n) \quad = \quad \min\left\{1,\, \sqrt{\frac{S_{bb,\mathrm{des}}(\Omega_\mu)}{\left|Y(e^{j\Omega_\mu}, n)\right|^2}}\right\}.$$

$$\left(\; H_{\min}(e^{j\Omega}, n)\left|Y(e^{j\Omega_\mu}, n)\right| \quad = \quad \sqrt{S_{bb,\mathrm{des}}(\Omega_\mu)}\;\right)$$

Extensions for the Wiener Characteristic – Maximum Attenuation (Part 3)

Specification of a „desired noise“ (continued):

- Problem: If we would use the procedures of the last slide, we would get a ***constant magnitude output spectrum*** (during speech pauses). Only the phase would vary from frame to frame. This sounds very unpleasant.
- Solution: If we add (or multiply) a ***random component*** to the attenuation limit, e.g. as

$$H_{\min}(e^{j\Omega_\mu}, n) \quad = \quad \min\left\{1, \sqrt{\frac{S_{bb,\mathrm{des}}(\Omega_\mu)}{\left|Y(e^{j\Omega_\mu}, n)\right|^2}} + H_{\mathrm{rand}}(n)\right\},$$

  we can avoid this effect.
- The advantage of this type of limiting the attenuation factors is to have ***control over the remaining background noise***. If we use such an add-on in speech recognition systems (as part of a pre-processing unit), the recognition engine can reduce the amount of parameters that are used for modelling the remaining noise (only one noise type remains).

Extensions for the Wiener Characteristic – Maximum Attenuation (Part 4)

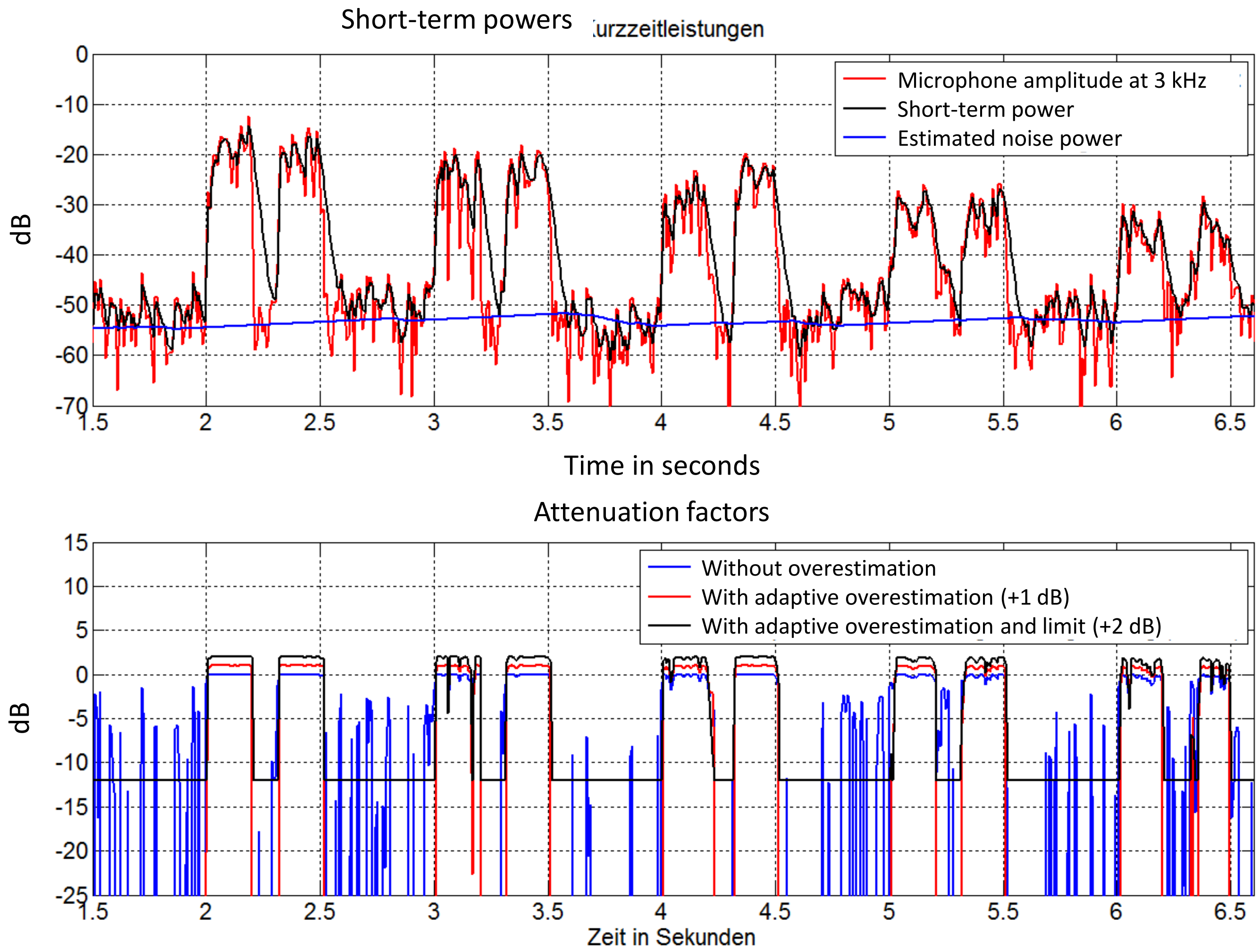
Controlling the attenuation limit:

- If we want ***to keep the original noise type*** (reduced by some decibels), we can use a fixed attenuation limit:

$$H_{\min}(e^{j\Omega_\mu}, n) \quad = \quad H_{\min}.$$

- In addition to that we can ***slowly modify the attenuation limit*** (over time).
  This means a lower amount of (maximum) attenuation during periods containing speech activity and a larger attenuation maximum (more attenuation) during speech pauses.

Extensions for the Wiener Characteristic – Maximum Attenuation (Part 5)



: Microphone signal

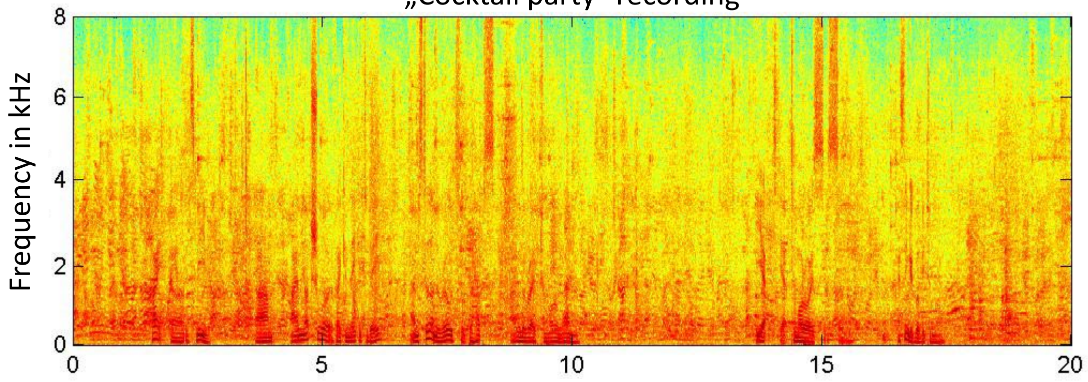
: Output without attenuation limit

: Output with attenuation limit

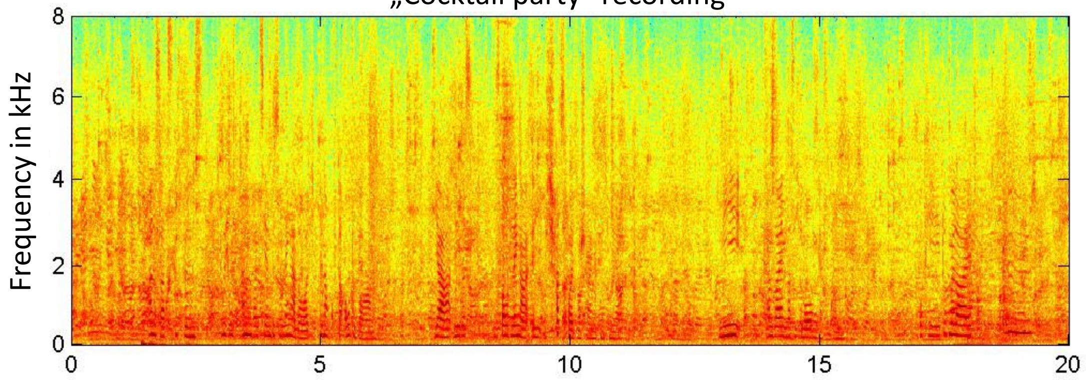
Extensions for the Wiener Characteristic – Maximum Attenuation (Part 6)

Examples for a noise transformation:

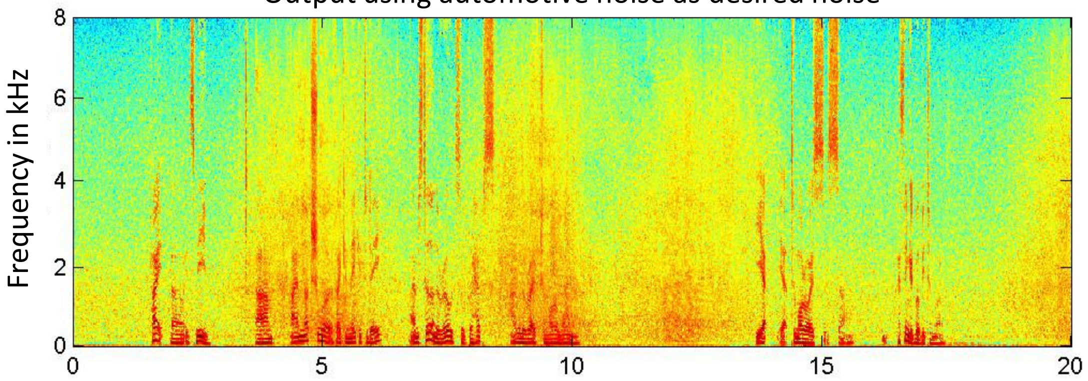
„Cocktail party“ recording



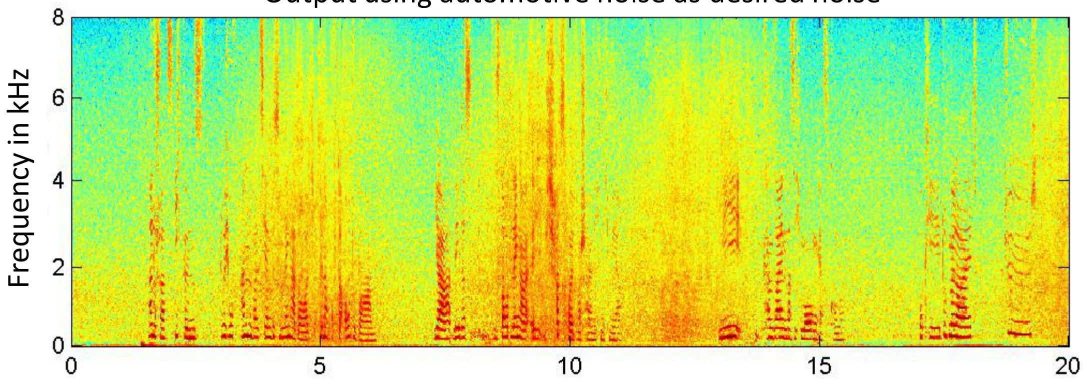
„Cocktail party“ recording



Output using automotive noise as desired noise



Output using automotive noise as desired noise



Extensions of Basis Noise Suppression Schemes – Reducing Reverberation (Part 1)

Dereverberation:

- When recording speech signal (with some distance between the microphone and the mouth of the speaker) in medium or large rooms the ***signals sound reverberant***. This leads to ***reduced speech quality*** on the one hand and to ***larger word error rates of speech dialog systems*** on the other hand.
- However, reverberation can also contribute in a positive sense to speech quality. ***Early reflections*** (duration up to 30 to 50 ms) lead to a better sounding of speech signals. ***Late reflections*** lead to the opposite effect and degrade usually the perceived quality.
- With the same approach that was used for noise suppression also reverberation can be reduced. We can ***modify the power spectral density of the distortion and filter characteristic*** according to

$$\widehat{S}_{bb}(\Omega_\mu, n) \longrightarrow \widehat{S}_{bb}(\Omega_\mu, n) + \widehat{S}_{rr}(\Omega_\mu, n)$$

$$\widehat{H}_{\text{opt}}(e^{j\Omega_\mu}, n) = \max\left\{ H_{\min},\ 1 - \frac{K_{bb,\text{over}}\,\widehat{S}_{bb}(\Omega_\mu, n) + K_{rr,\text{over}}\,\widehat{S}_{rr}(\Omega_\mu, n)}{\widehat{S}_{yy}(\Omega_\mu, n)} \right\}.$$

Extensions of Basis Noise Suppression Schemes – Reducing Reverberation (Part 2)

Estimating the power spectral density of the "reverb" components:

- We assume that the reverb power ***decays exponentially***.
  In addition, we assume a ***fixed ratio of the direct sound and the reverberant components*** and that the direct sound is large in amplitude compared to the reverberant components. This leads to the following estimation rule:

$$\widehat{S}_{rr}(\Omega_\mu, n) \quad = \quad \left|Y(e^{j\Omega_\mu}, n-D)\right|^2 R(e^{j\Omega_\mu})\, A^D(e^{j\Omega_\mu}) + \widehat{S}_{rr}(\Omega_\mu, n-1)\, A(e^{j\Omega_\mu})$$

with:

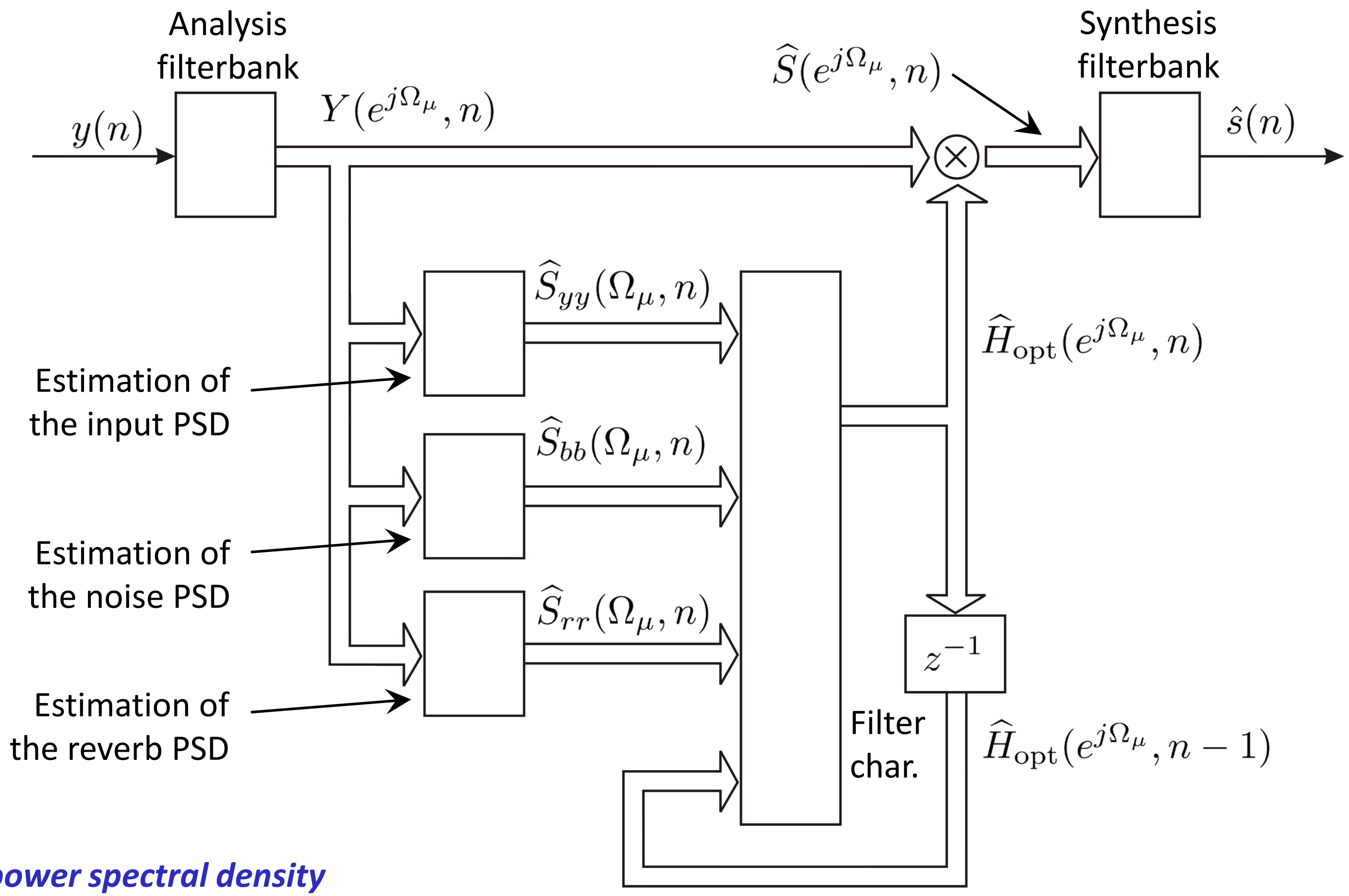
D : protection time in frames (reverberation with a delay lower than D frames is perceived as well-sounding, reverberation with a larger delay as disturbing)

$A(e^{j\Omega_\mu})$: attenuation parameter (reverb attenuation per frame)

$R(e^{j\Omega_\mu})$: direct-to-reverb ratio

Extensions of Basis Noise Suppression Schemes – Reducing Reverberation (Part 3)

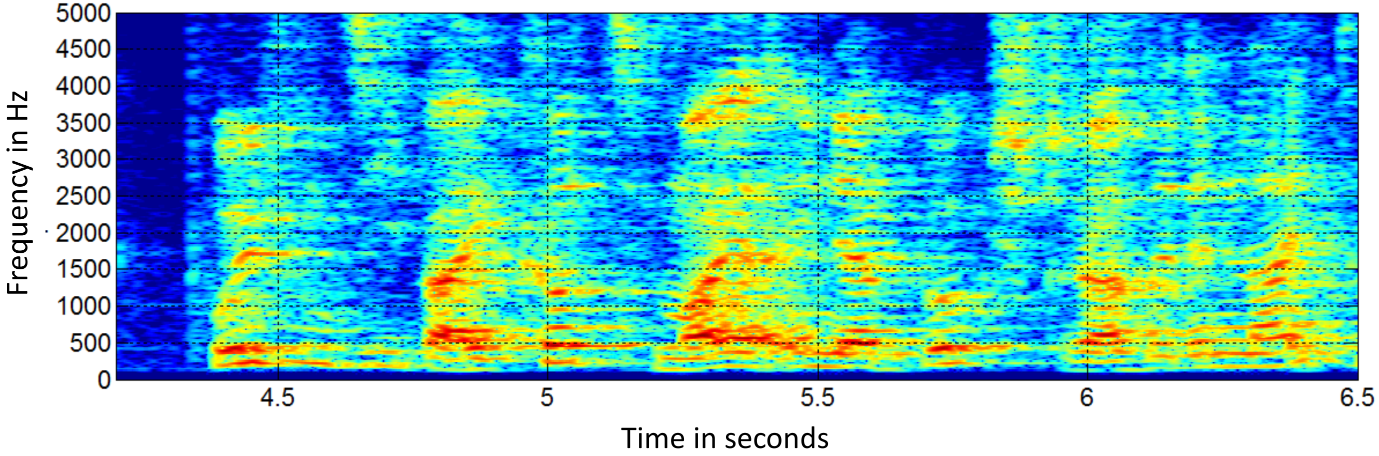
Combined reduction of noise and reverberation:



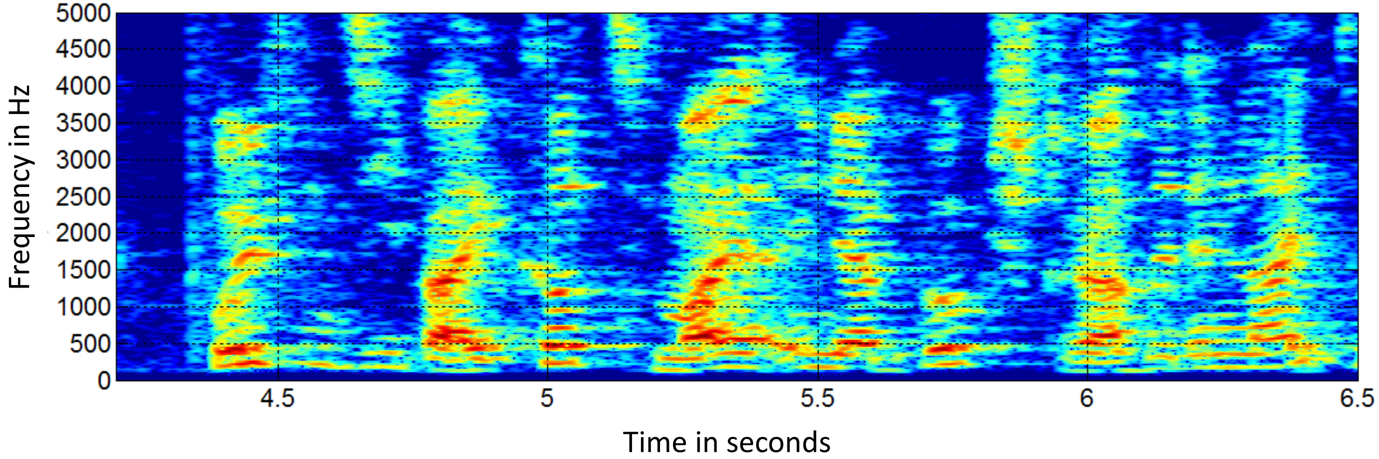
PSD = power spectral density

Extensions of Basis Noise Suppression Schemes – Reducing Reverberation (Part 4)

Time –frequency analysis of the input signal



Time –frequency analysis of the output signal



Partial Signal Reconstruction – Part 1

Conventional approach:

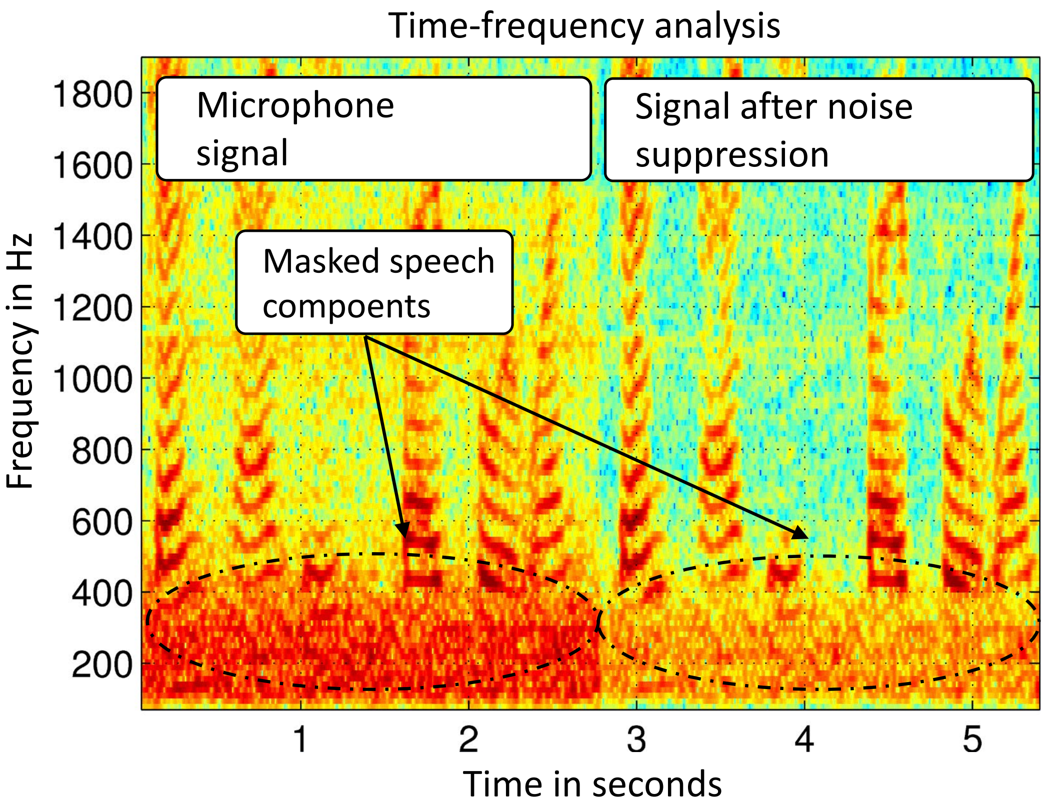
- Sufficient quality at medium and high SNRs

Problems:

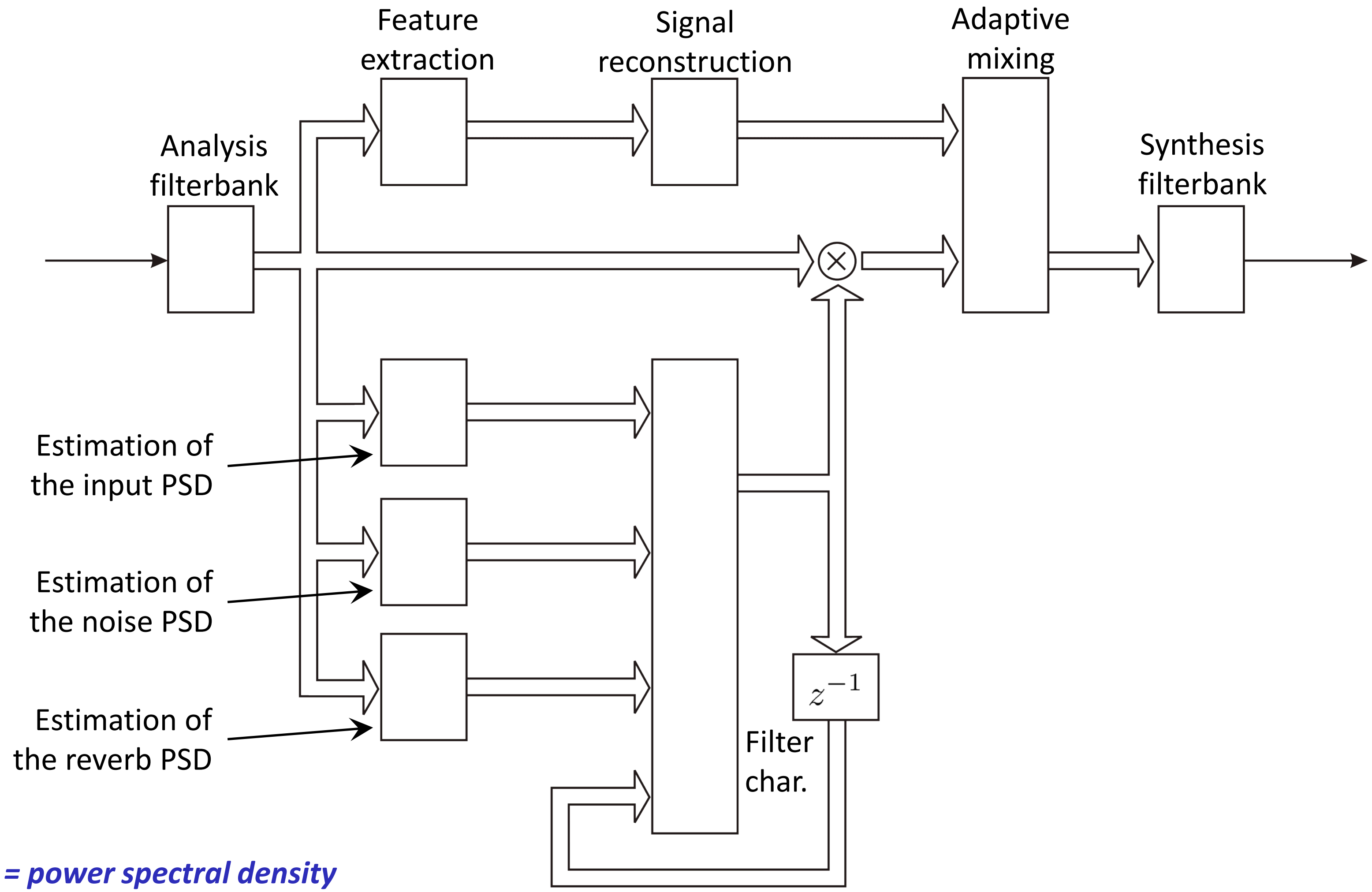
- Low quality at low SNRs (high noise)
- Some spectral components will be attenuated

Extension:

- Transition to ***model-based approaches***
- Extraction of relevant ***features*** out of the noisy input signal
- ***Reconstruction*** of the components with low SNR by using pre-trained models and extracted features (for appropriate model selection/adaption)

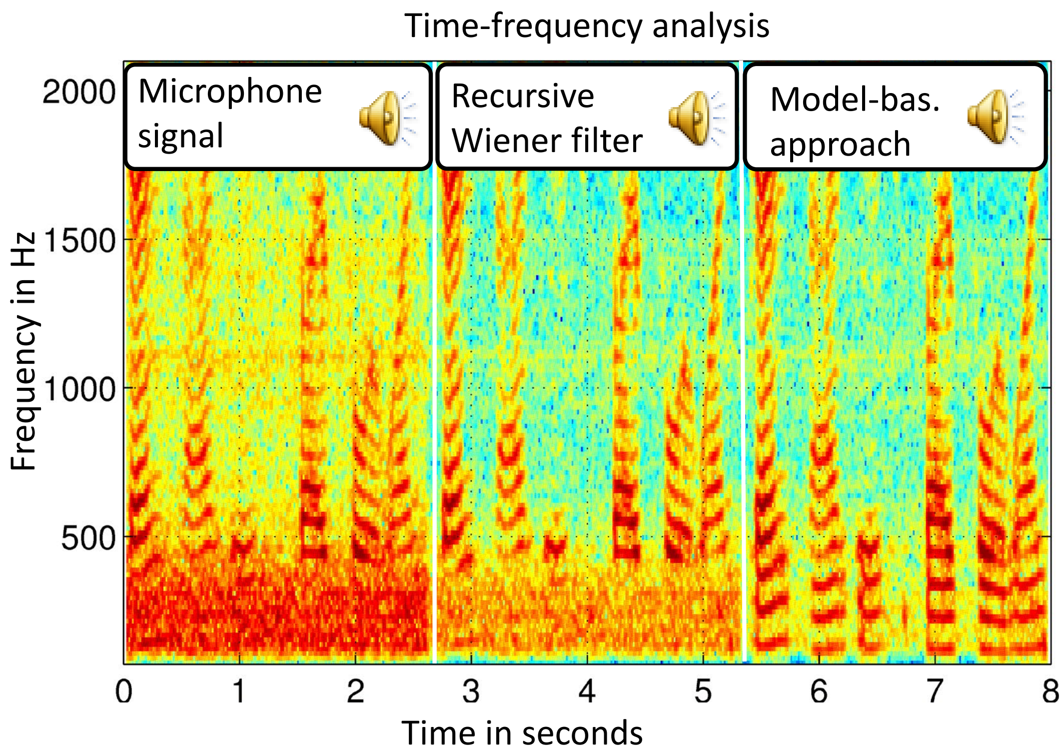


Partial Signal Reconstruction – Part 2

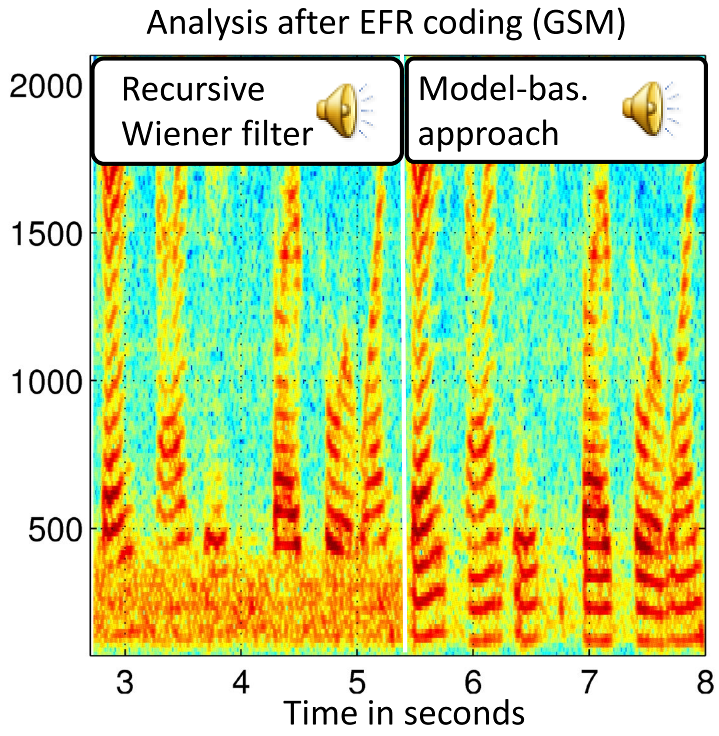


PSD = power spectral density

Partial Signal Reconstruction – Part 3

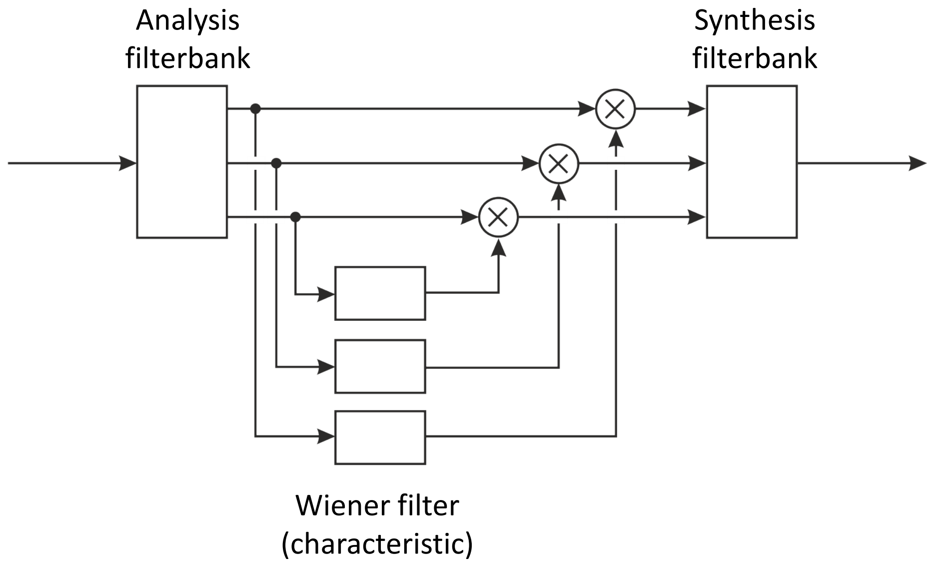


Noisy speech signal, measured in a car driving with 160 km/h

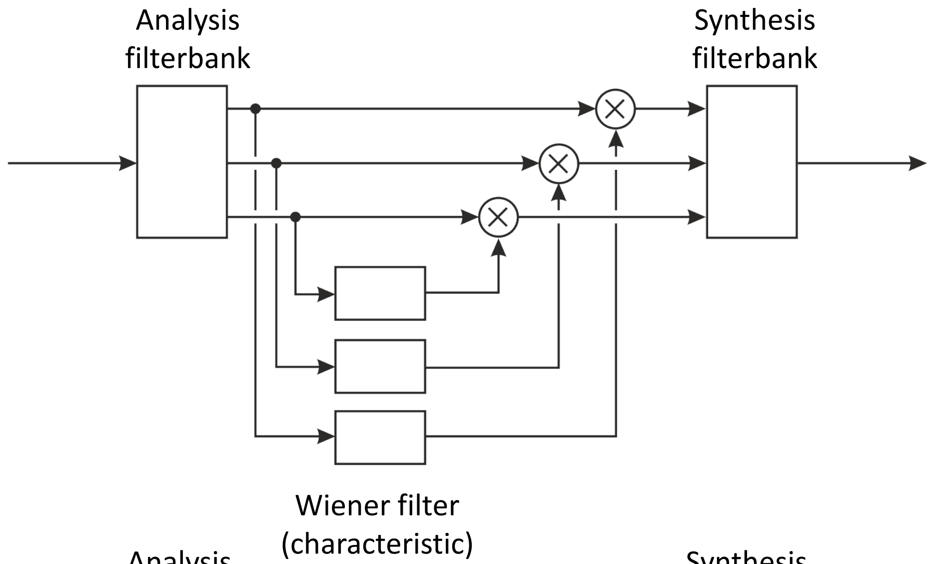


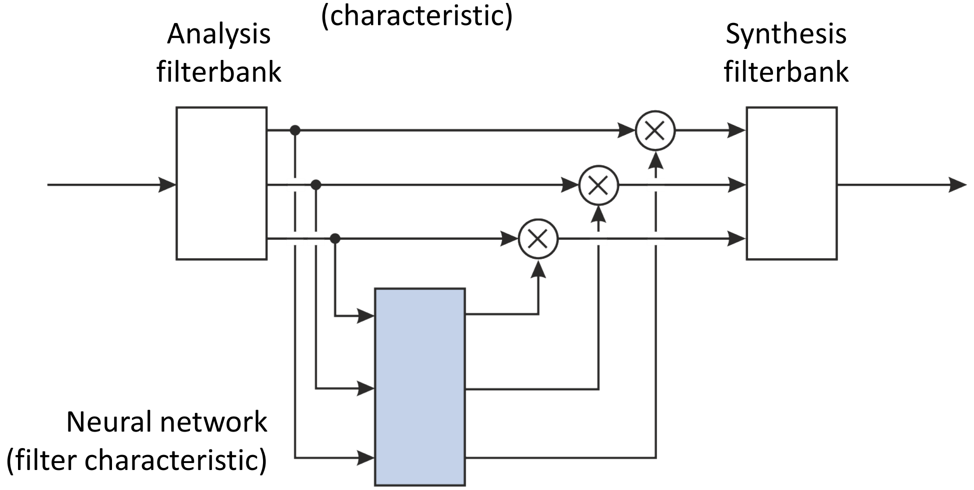
Source: Mohamed Krini, SVOX Deutschland, (Dissertation at TU Darmstadt)

Further Extensions

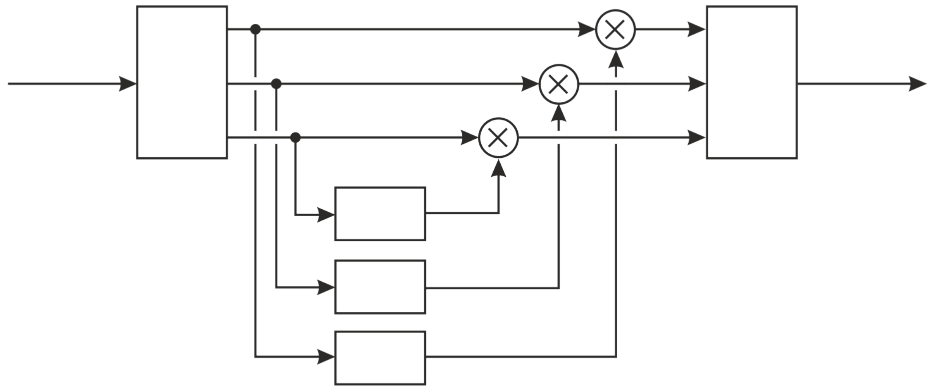


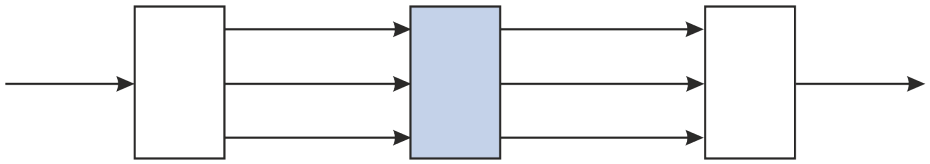
Further Extensions

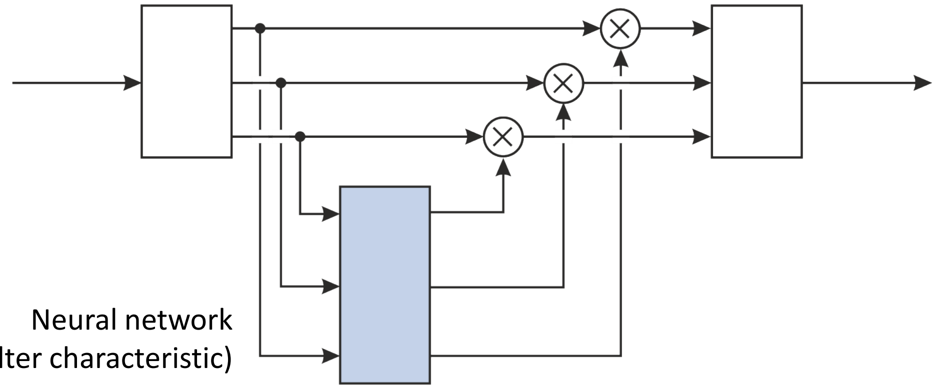




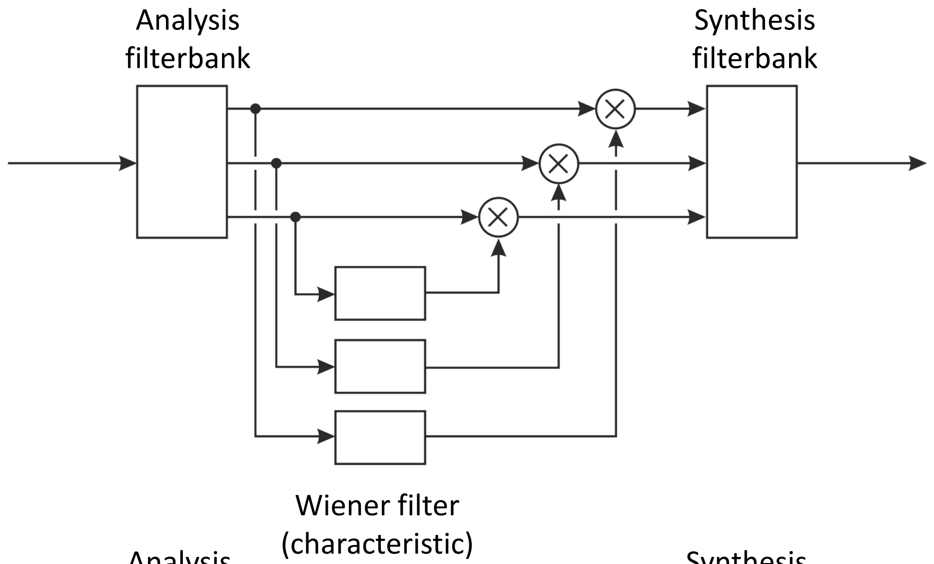
Further Extensions

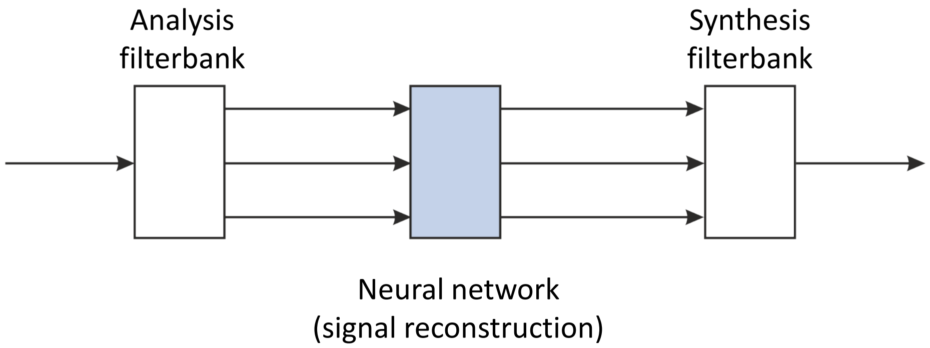


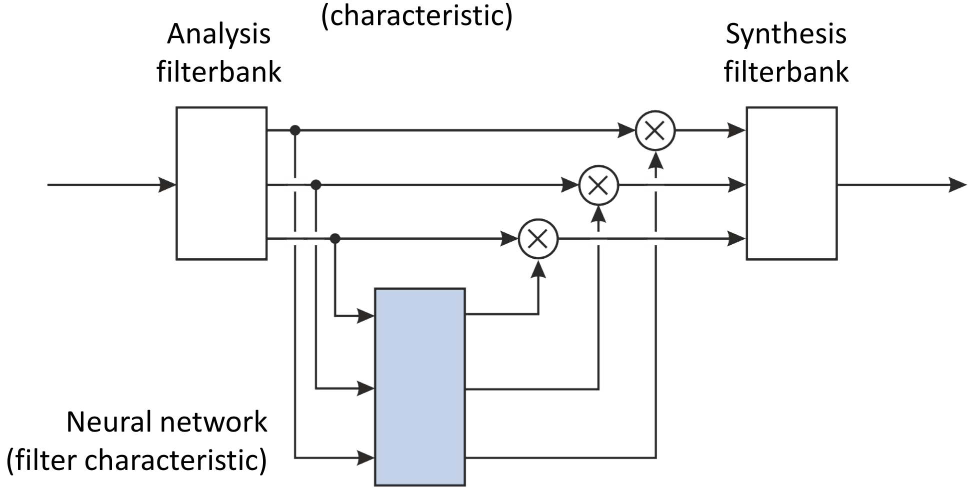


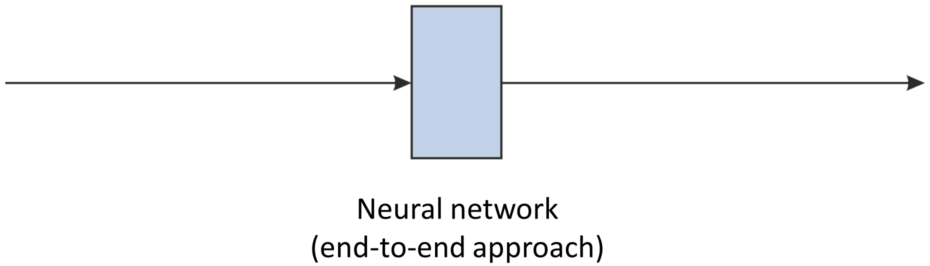


Further Extensions

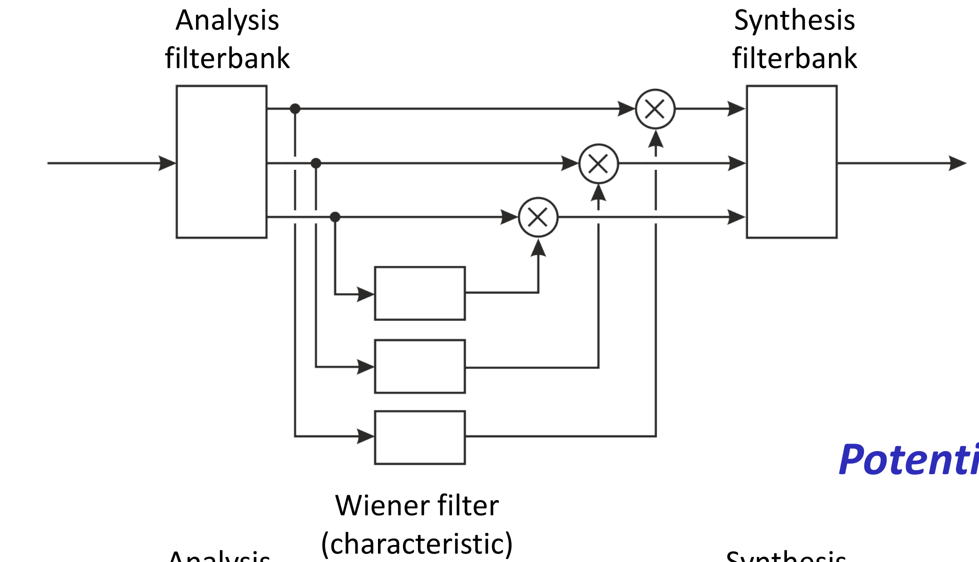


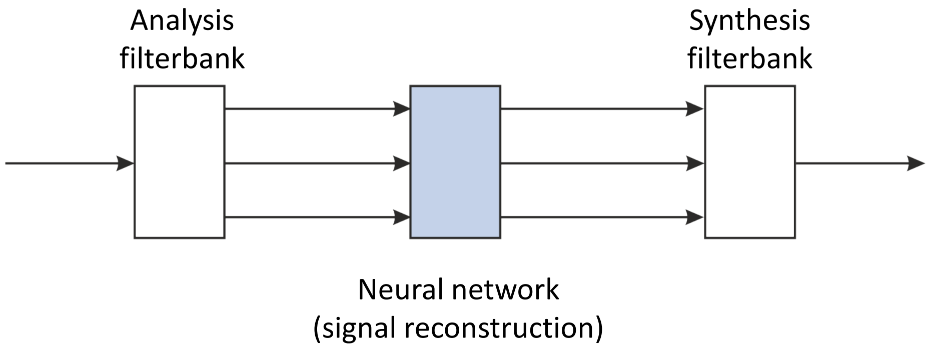




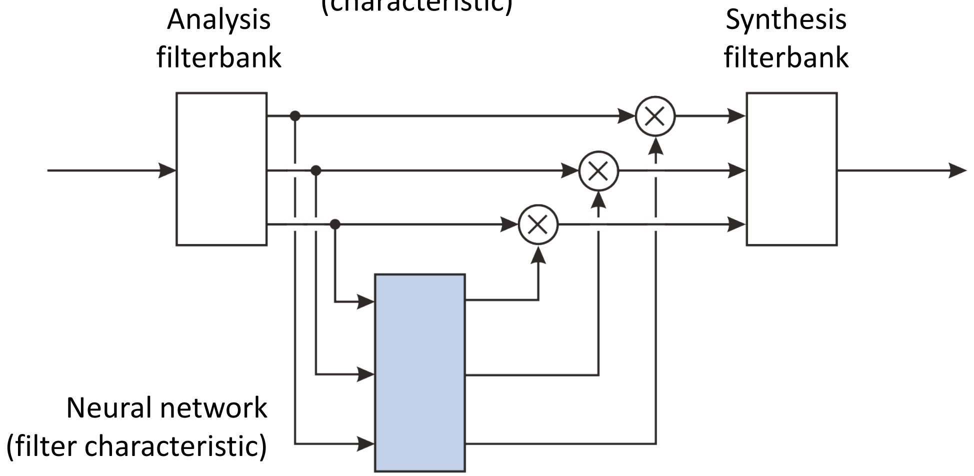


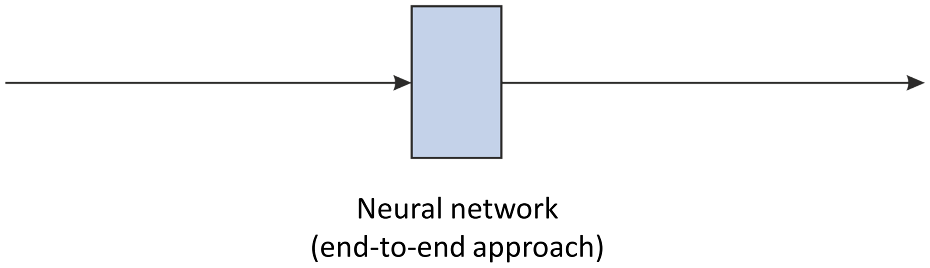
Further Extensions





Potential topics for your talks (current research)





Contents

Enhancement of EEG Signals – Background

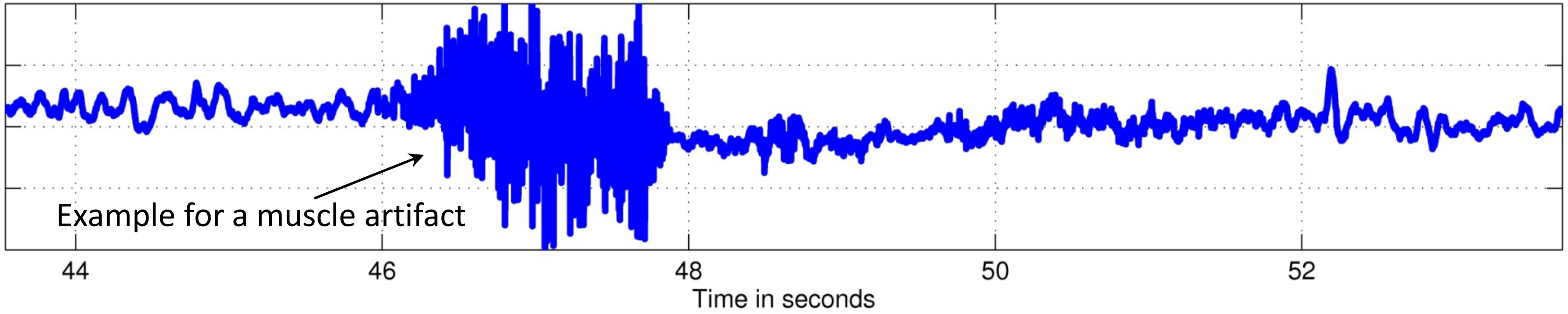
EEG (and MEG) signal enhancement:

- Channel-specific enhancement (without taking source [or network] localization into account)
- Mainly for the removal of artifacts

Artifacts can be:

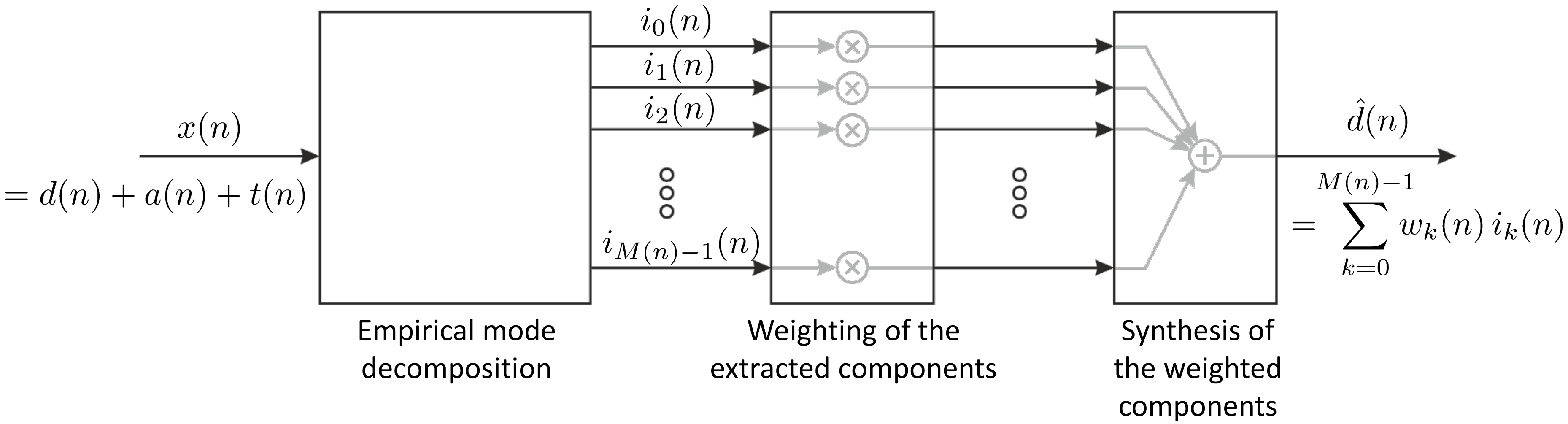
- Patient related (physiologic): eye movements, eye blinking, muscle artifacts, heart beating
- Technical: electrode popping, power supply

Example:



Signal Enhancement with Real-Time EMD

Basic structure:



Steps and objectives:

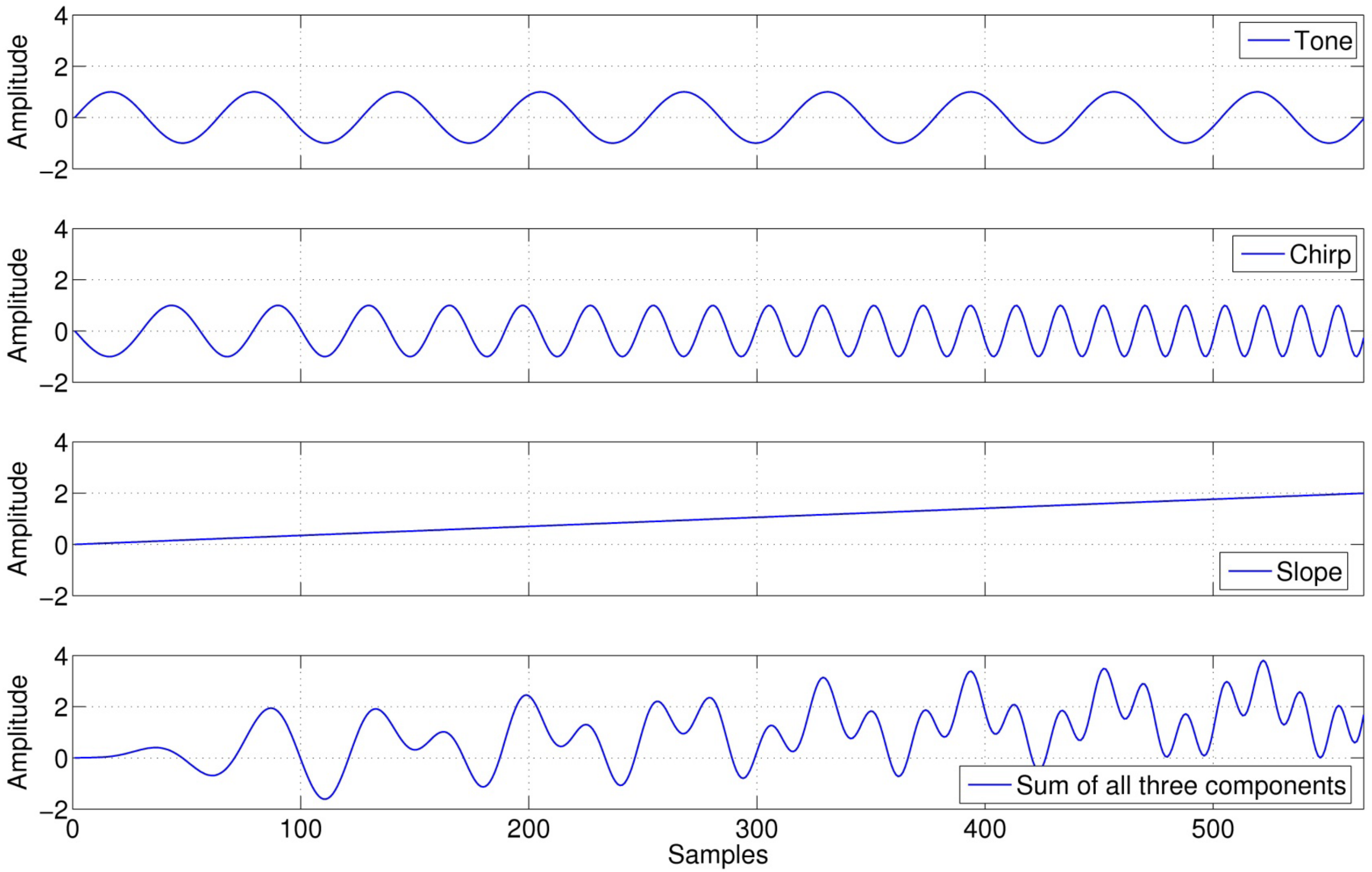
- Split the signal into (overlapping) blocks.
- Find signal-specific components (they sum up to the input signal) and find appropriate weights.
- The phase relations of the desired components should not be changed.

Empirical Mode Decomposition – Introduction

Objective and details of an empirical mode decomposition:

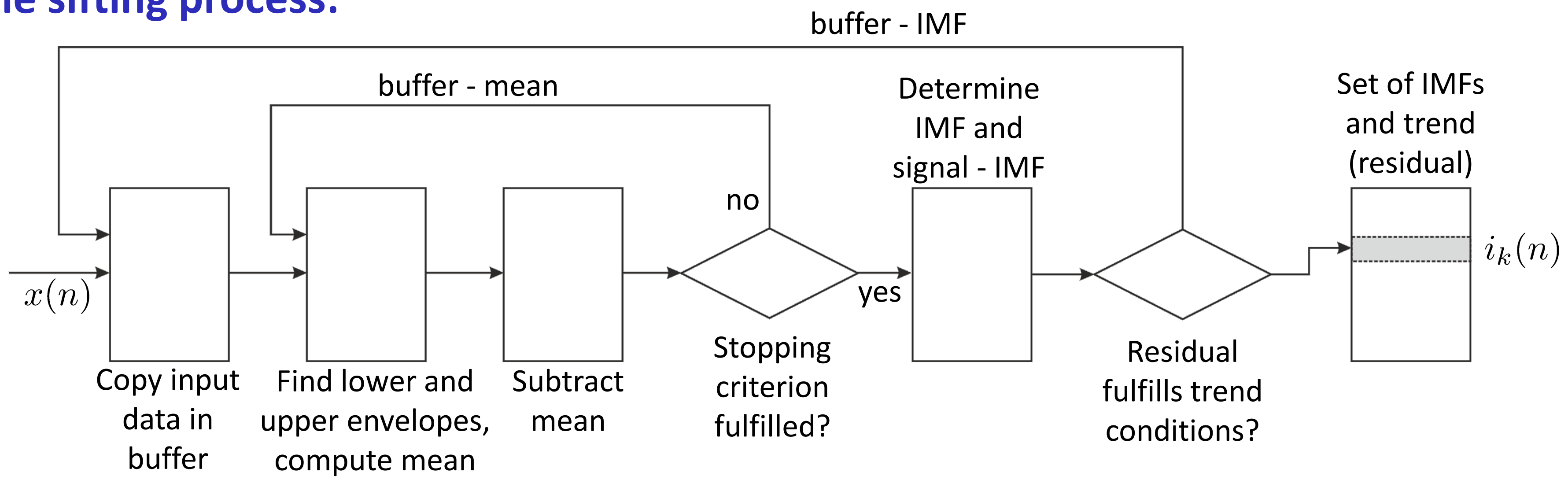
- ***Separate an arbitrary input signal*** into different components called ***intrinsic mode functions*** (IMFs).
- An IMF satisfies the following ***two conditions***:
  - The ***number of extrema*** and the ***number of zero crossings*** must either be ***equal*** or differ at most by one.
  - At any point, the ***mean value*** of the envelopes defined by the local maxima and the envelopes defined by the local minima ***is zero***
- The ***first IMF*** will contain the signal components with the ***highest frequency***. The next IMF will contain lower frequencies.

Empirical Mode Decomposition – An Example (Part 1)



Empirical Mode Decomposition – The Principle

Overview of the sifting process:



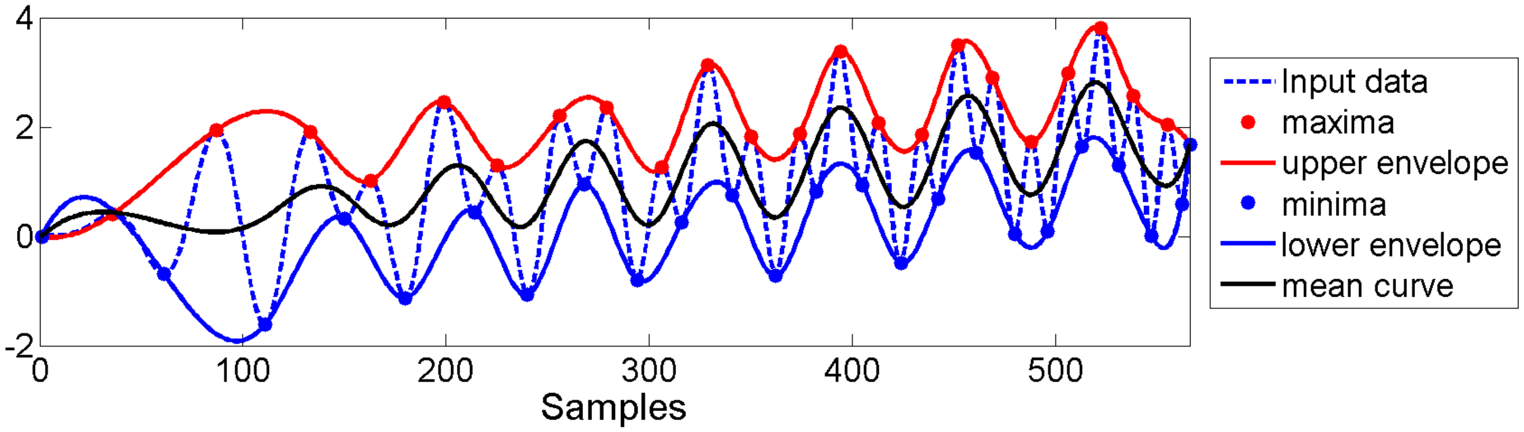
Stopping criteria for sifting process:

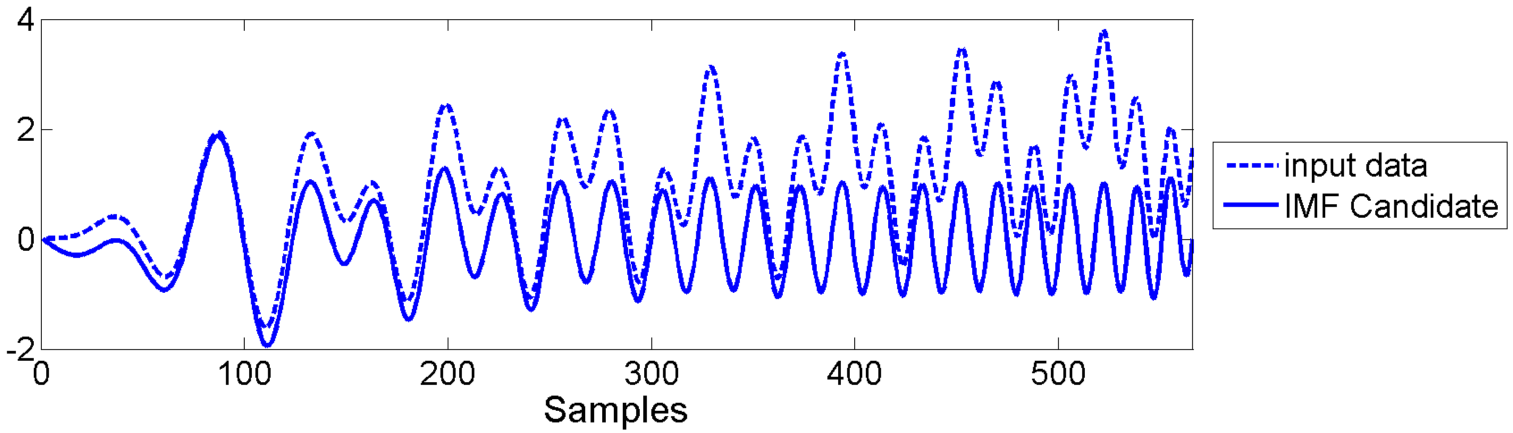
1. The IMF of the current iteration doesn‘t differ much from the previous iteration:

$$\frac{\sum\limits_n \big(i_{k,m}(n) - i_{k,m-1}(n)\big)^2}{\sum\limits_n i_{k,m-1}^2(n)} < T.$$

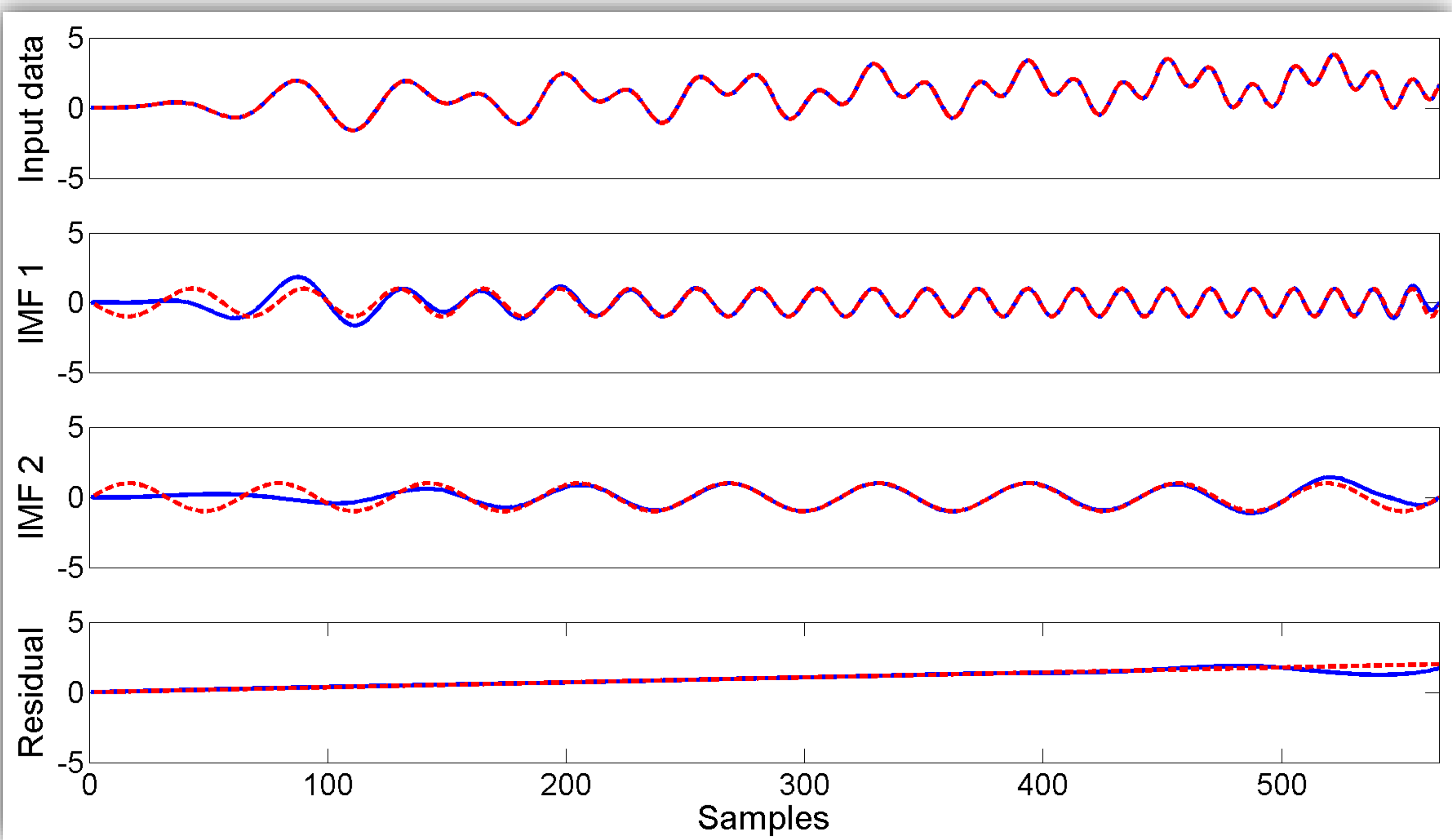
2. The maximum number of iterations is reached (for „real-time" reasons).

Empirical Mode Decomposition – An Example (Part 2)





Empirical Mode Decomposition – An Example (Part 3)

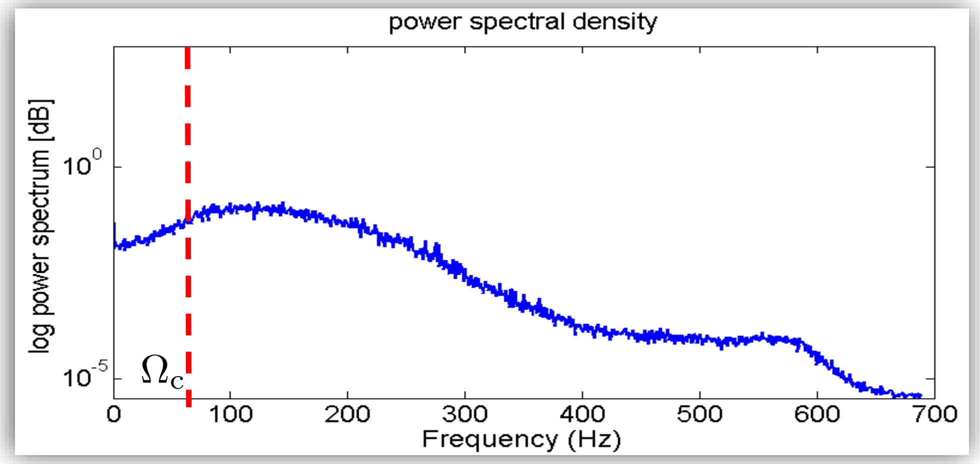


Christian-Albrechts-Universität zu Kiel

Empirical Mode Decomposition – Denoising

Assumption:

Nearly all noise components are in the higher frequency range.



Signal ***Noise***

Approximation for SNR:

$$r_k(n) = 10\,\log_{10}\left(\frac{\int\limits_{\Omega=0}^{\Omega_c} \hat{S}_{i_k i_k}(\Omega,n)\,d\Omega}{\int\limits_{\Omega=\Omega_c}^{\pi} \hat{S}_{i_k i_k}(\Omega,n)\,d\Omega}\right)$$

$\Downarrow$

IMF are dominated by noise, if

$$r_k(n) < T_{\text{noise}} \quad \Rightarrow$$

$$\boxed{w_k(n) = \begin{cases} 1, & \text{if } \; r_k(n) \geq T_{\text{noise}}, \\ 10^{\frac{r_k(n)-T_{\text{noise}}}{10}}, & \text{else.} \end{cases}}$$

Empirical Mode Decomposition – Detrending

Assumption:

The local trend is mostly represented by the residual.

Observation:

A comparison of the energy levels in the residual with the local trends has shown a proportional relationship.

Energy coefficient:

$$r_{\text{res}}(n) = \frac{\sum_{\ell} i^2_{M(n)-1}(n-\ell)}{\sum_{\ell} x^2(n-\ell)} \quad \Longrightarrow \quad \boxed{w_{M(n)-1}(n) = 1 - r_{\text{res}}(n)}$$

Empirical Mode Decomposition – Data Sets Processed

- ***Semi-simulated data:***

  Real EEG signals from the central and frontal lobes were contaminated with simulated muscle artifacts.

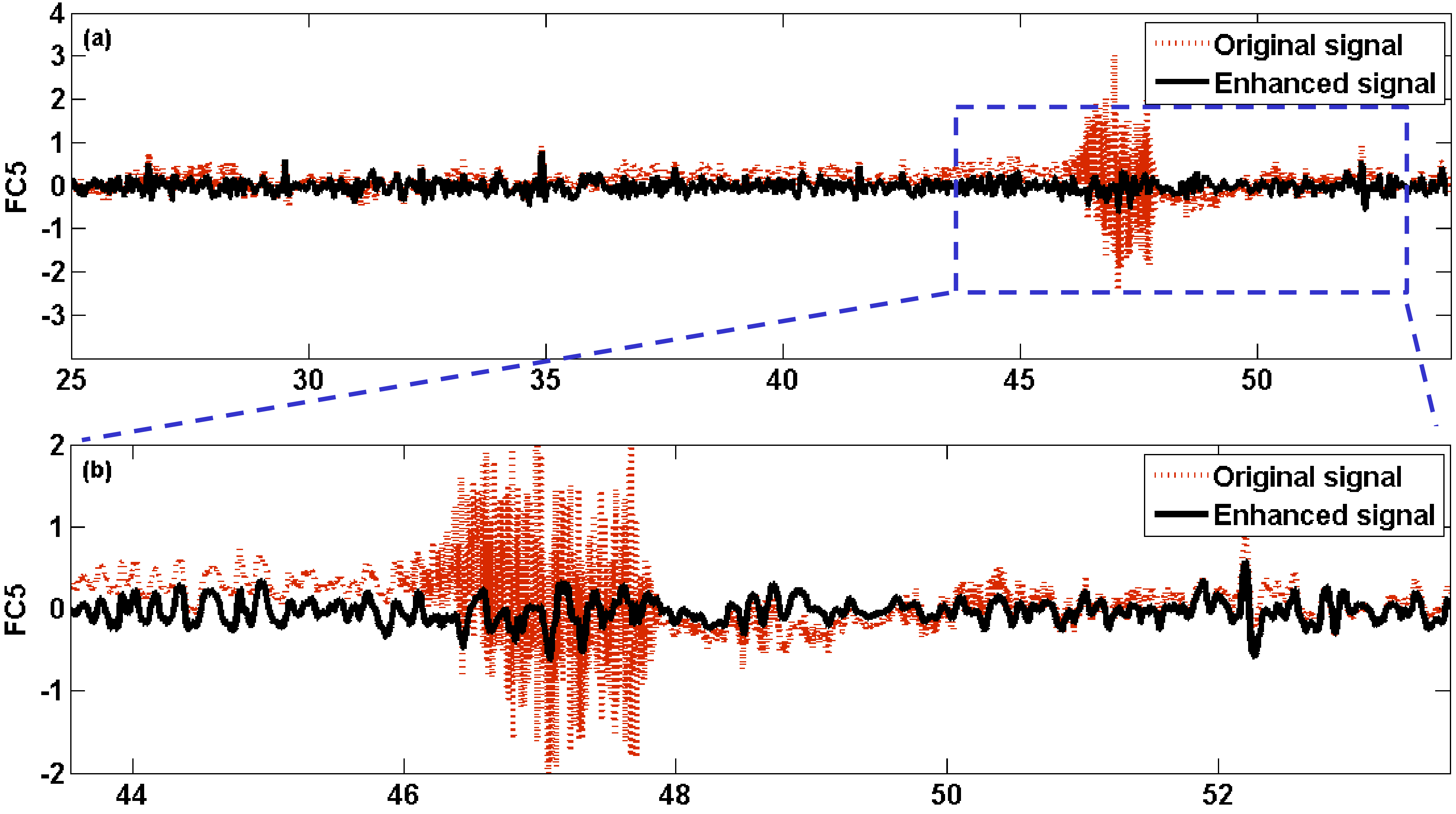
  - Length of the signals: 60 s.
  - Original sampling frequency: 5 kHz.
  - Input sampling frequency: 44.1 kHz.
  - Process sampling frequency: 1.378 kHz = 44.1 kHz / 32

- ***Real EEG signals:***

  Real data from an epilepsy patients with inherent muscle artifacts were processed.

  - Length of the signals: 60 s.
  - Number of channels: 30 channels.
  - Sampling frequencies: Same as for the simulated case

Real EEG Signals: Denoising



Literature – Part 2

Noise suppression:

- E. Hänsler, G. Schmidt: ***Acoustic Echo and Noise Control – Chap. 5 (Wiener Filter)***, Wiley, 2004
- M. S.Hayes: ***Statistical Digital Signal Processing and Modeling – Chapter 7 (Wiener Filtering),*** Wiley, 1996

Dereverberation:

- E. A. P. Habets, S. Gannot, I. Cohen: ***Dereverberation and Residual Echo Suppression in Noisy Environments***, in E. Hänsler, G. Schmidt (eds.), Speech and Audio Processing in Adverse Environments, Springer, 2008

Signal reconstruction:

- M. Krini, G. Schmidt: ***Model-based Speech Enhancement***, in E. Hänsler, G. Schmidt (eds.), Speech and Audio Processing in Adverse Environments, Springer, 2008

Empirical mode decomposition:

- E. Huang, Z. Shen, S.R. Long, M.L. Wu, H.H. Shih, Q. Zheng, N.C. Yen, C.C. Tung, and H.H. Liu: ***The Empirical Mode Decomposition and Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis***, Proc. Roy. Soc., vol. 454, pp. 903 – 995, 1998

Summary and Outlook

Summary:

- Cost functions
  - Data/sample-based cost functions
  - Distribution-based cost functions
- Enhancement of speech signals
  - Generation and properties of speech signals
  - Wiener filter
  - Frequency-domain solution
  - Extensions of the gain rule
  - Extensions of the entire framework
  - Outlook to neural net based approaches
- Enhancement of EEG signals
  - Empirical mode decomposition

Next part:

- Multi-channel noise suppression