Christian-Albrechts-Universität zu Kiel

Adaptive Filters – Linear Prediction

Gerhard Schmidt

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

Today

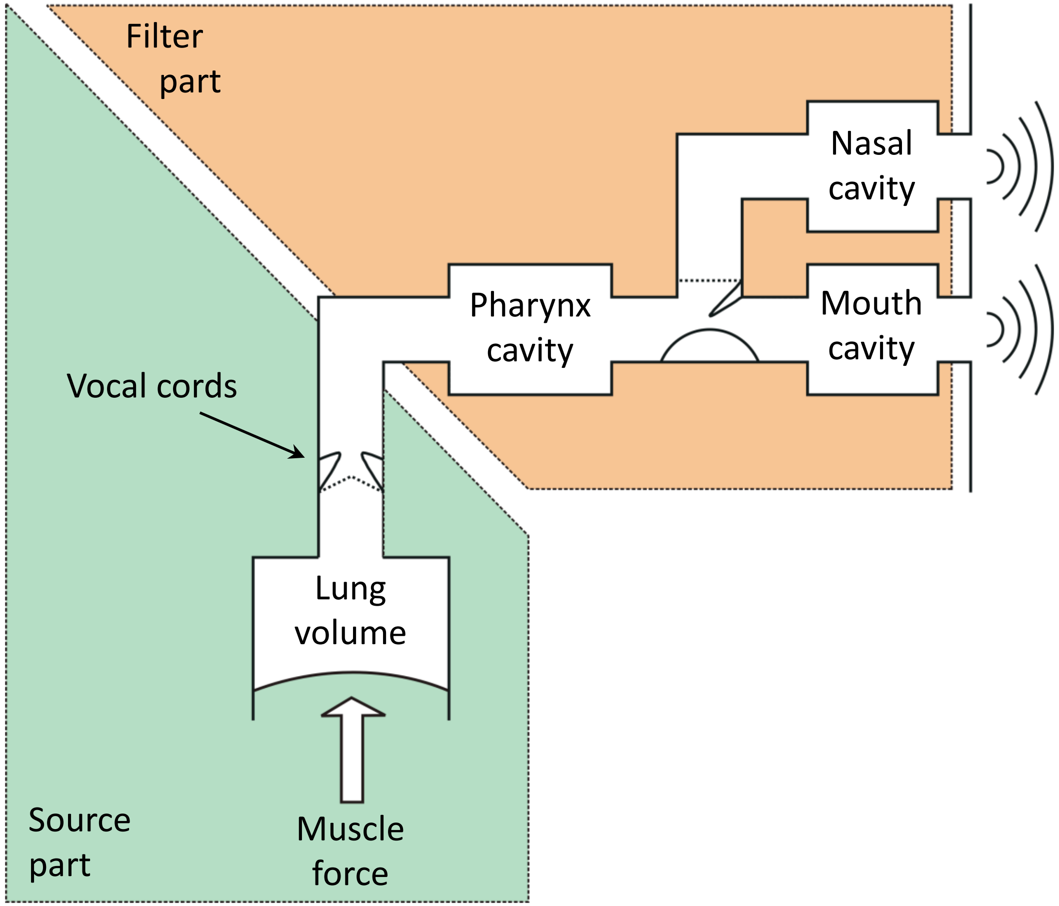
Contents of the Lecture:

- Source-filter model for speech generation
- Literature
- Derivation of linear prediction
- Levinson-Durbin recursion
- Application example

Christian-Albrechts-Universität zu Kiel

- ***Source-filter model for speech generation***
- *Literature*
- *Derivation of linear prediction*
- *Levinson-Durbin recursion*
- *Application example*

Speech Production

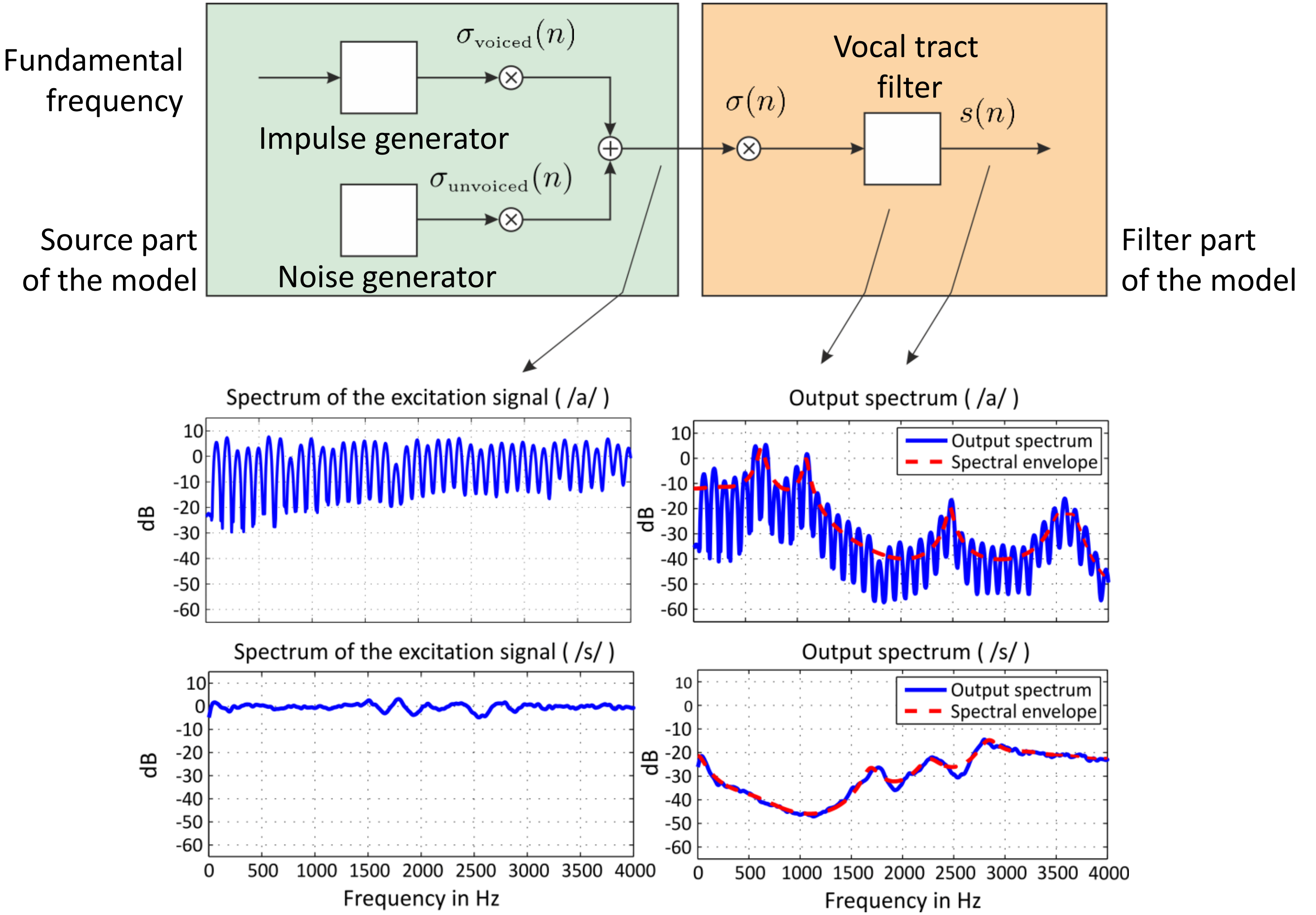


Principle:

- 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.

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

Source-filter Model



- *Source-filter model for speech generation*
- ***Literature***
- *Derivation of linear prediction*
- *Levinson-Durbin recursion*
- *Application example*

Books

Basic text:

- E. Hänsler / G. Schmidt: *Acoustic Echo and Noise Control – Chapter 6 (Linear Prediction),* Wiley, 2004

Speech processing:

- P. Vary, R. Martin: *Digital Transmission of Speech Signals – Chapter 2 (Models of Speech Production and Hearing)*, Wiley 2006
- J. R. Deller, J. H. l. Hansen, J. G. Proakis: *Discrete-Time Processing of Speech Signals – Chapter 3 (Modeling Speech Production)*, IEEE Press, 2000

Further basics:

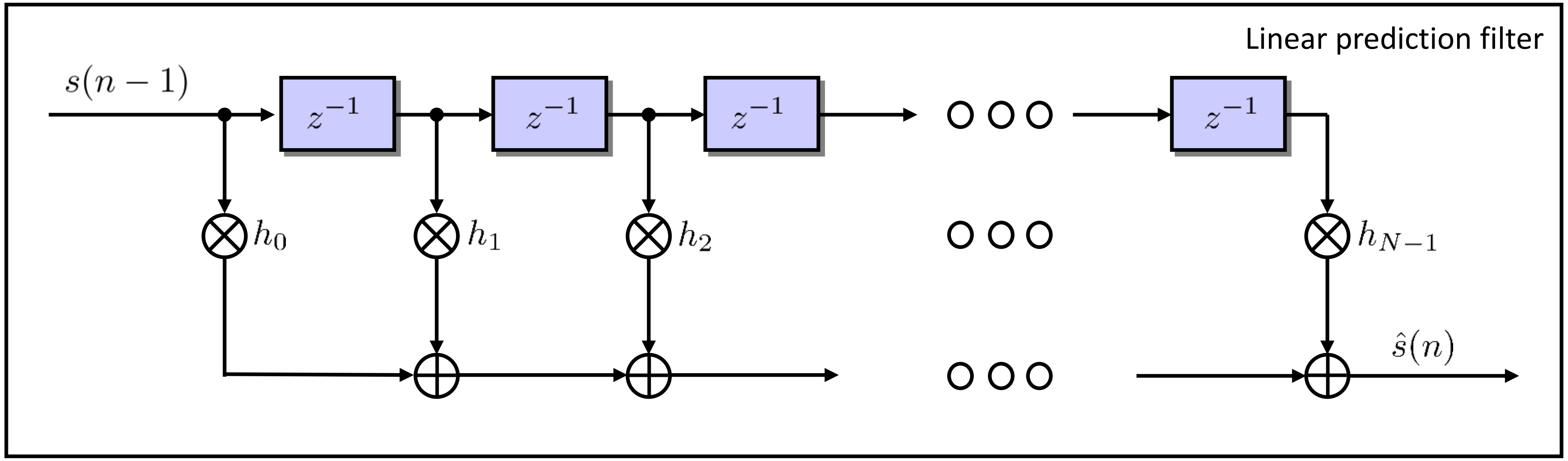
- E. Hänsler: *Statistische Signale: Grundlagen und Anwendungen – Chapter 6 (Linearer Prädiktor),* Springer, 2001 (in German)
- M. S. Hayes: *Statistical Digital Signal Processing and Modeling – Chapters 4 und 5 (Signal Modeling, The Levinson Recursion)*, Wiley, 1996

- *Source-filter model for speech generation*
- *Literature*
- ***Derivation of linear prediction***
- *Levinson-Durbin recursion*
- *Application example*

Basic Approach

Estimation of the current signal sample on the basis of the previous N samples:

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



With:

- $\hat{s}(n)$: estimation of $s(n)$
- h_i : predictor coefficients
- N : length / order of the predictor

Optimization Criterion

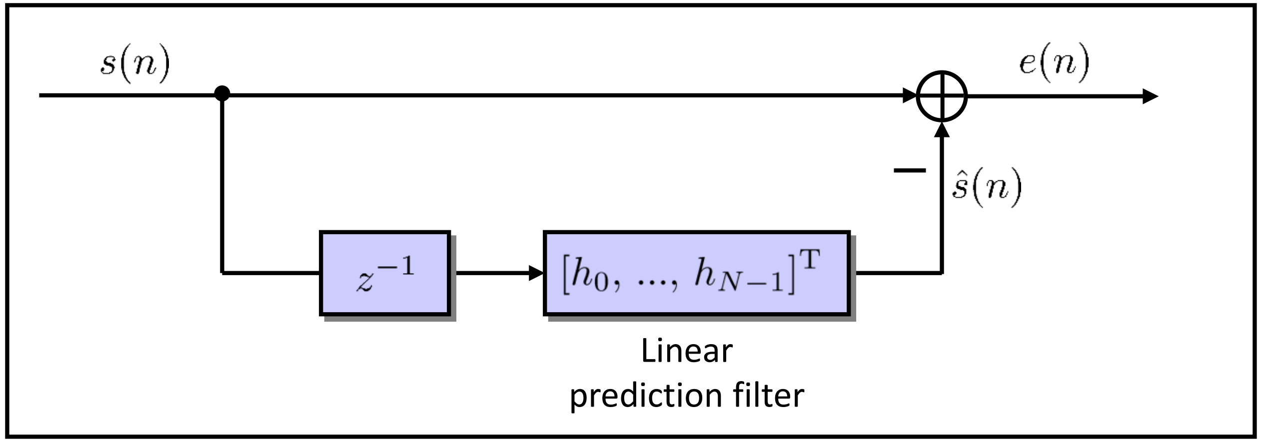
Optimization:

Estimation of the filter coefficients h_i such that a cost function is optimized.

Cost function:

$$\mathrm{E}\Big\{\big[\underbrace{s(n) - \hat{s}(n)}_{e(n)}\big]^2\Big\} \;\rightarrow\; \min$$

Structure:

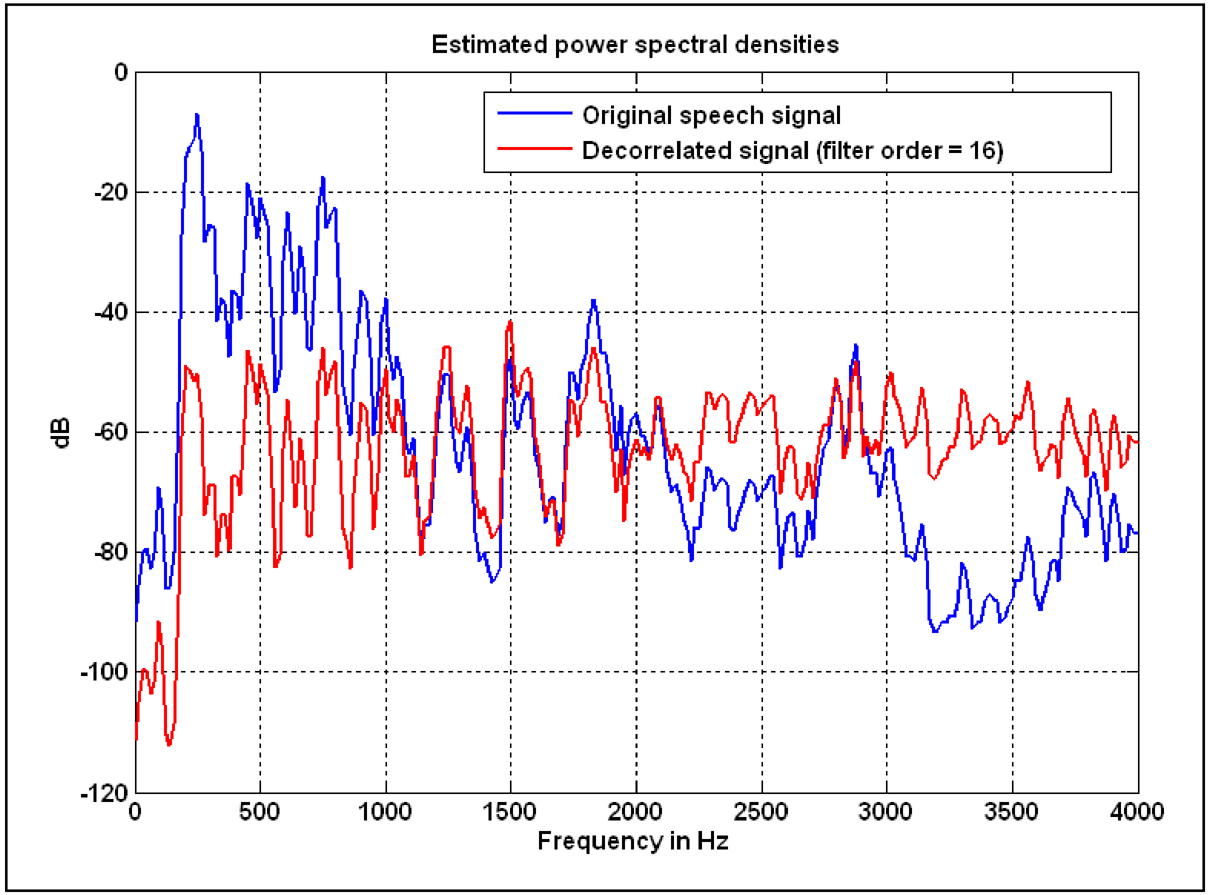


„Whitening“ Property

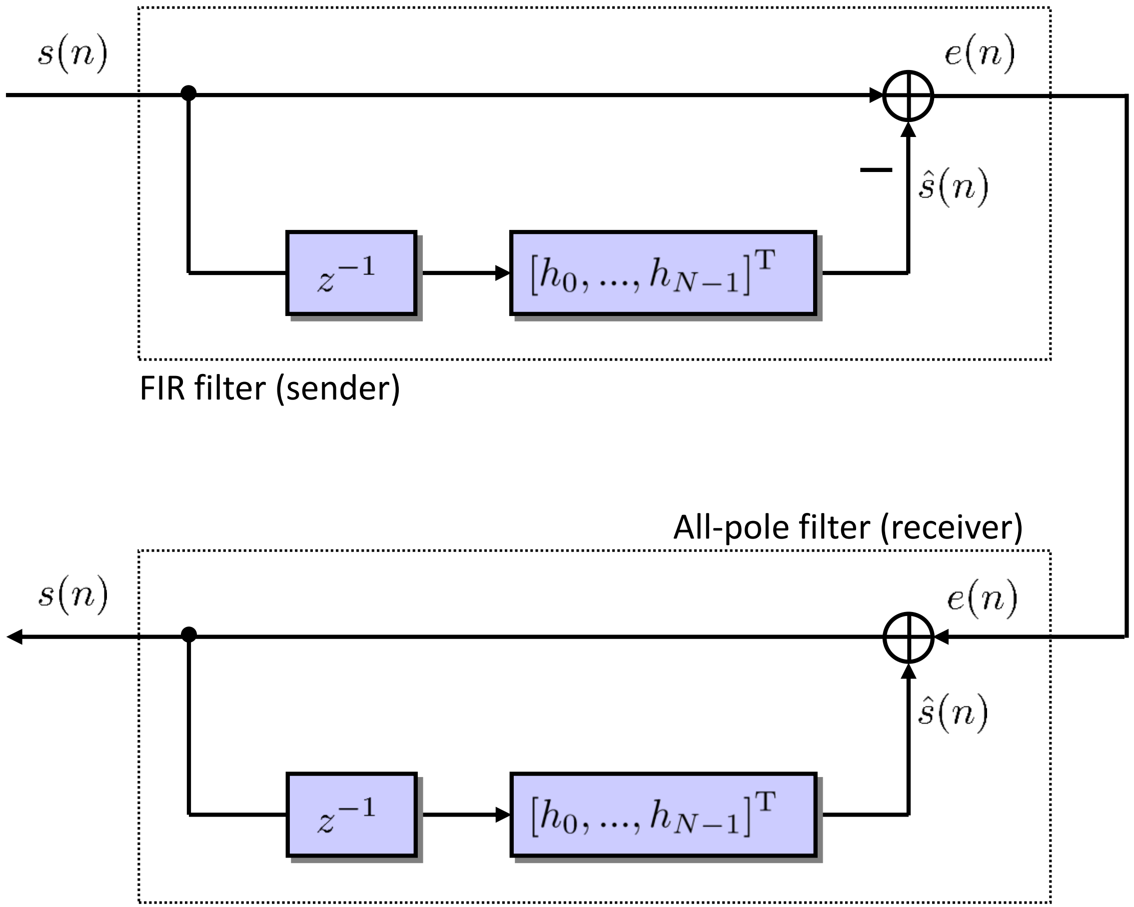
Cost function:

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

- Strong frequency components will be attenuated most (due to Parseval).
- This leads to a spectral „decoloring“ (whitening) of the signal.



Inverse Filter Structure



Properties:

- The inverse predictor error filter is an all-pole filter
- The cascaded structure - consisting of a predictor error filter and an inverse predictor error filter - can be used for lossless data compression and for sending and receiving signals.

Computing the Filter Coefficients

Derivation during the lecture …

Examples – Part 1

First example:

- Input signal $s(n)$: white noise with variance σ_0^2 (zero mean)
- Prediction order: $N = 3$
- Prediction of the next sample: $L = 1$

This leads to:

$$\boldsymbol{R}_{ss} = \begin{bmatrix} \sigma_0^2 & 0 & 0 \\ 0 & \sigma_0^2 & 0 \\ 0 & 0 & \sigma_0^2 \end{bmatrix}, \text{ respectively } \boldsymbol{R}_{ss}^{-1} = \frac{1}{\sigma_0^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{r}_{ss}(1) = \big[0,\, 0,\, 0\big]^{\mathrm{T}}$$

$$\boldsymbol{h} = \boldsymbol{R}_{ss}^{-1}\,\boldsymbol{r}_{ss}(1) = \big[0,\, 0,\, 0\big]^{\mathrm{T}}$$

, what means the no prediction is possible or – to be precise – the best prediction is the mean of the input signal which is zero.

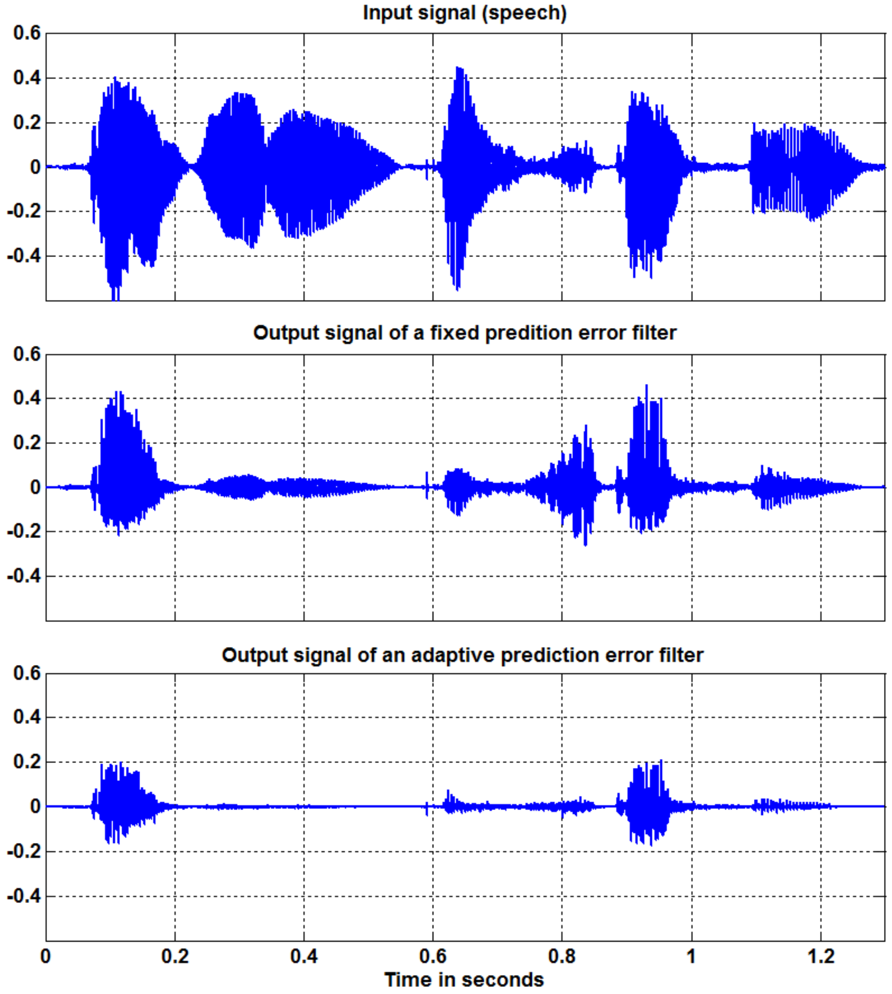
Examples – Part 2

Second example:

- Input signal $s(n)$: speech, sampled at $f_\mathrm{s} = 8$ kHz
- Prediction order: $N = 16$
- Prediction of the next sample: $L = 1$

Single optimization of the filter coefficients for the entire signal sequence

New adjustment of the filter coefficients every 64 samples



Estimation of the Autocorrelation Function – Part 1

Problem:

Ensemble averages are usually not known in most applications.

Solution:

Estimation of the ensemble averages by temporal averaging (ergodicity assumed):

$$\mathrm{E}\Big\{s(n)\,s(n+l)\Big\} \quad \longrightarrow \quad \frac{1}{L}\sum_{n} s(n)\,s(n+l)$$

Assumption:

$s(n)$ is a representative signal of the underlying random process.

Estimation schemes:

A few schemes for estimating an autocorrelation function exist. These scheme differ in the properties (such as unbiasedness or positive definiteness) that the resulting autocorrelation gets significantly.

Estimation of the Autocorrelation Function – Part 2

Example: „Autocorrelation method“:

Computed according to:

$$\widehat{r}_{ss}(l) = \begin{cases} \frac{1}{L} \sum\limits_{n=0}^{L-1-l} s(n)\, s(n+l), & \text{for } l \geq 0, \\ \frac{1}{L} \sum\limits_{n=l}^{L-1} s(n)\, s(n+l), & \text{for } l < 0 \end{cases}$$

Properties:

- The estimation is biased, we achieve: $\left| \mathrm{E}\left\{ \widehat{r}_{ss}(l) \right\} \right| \leq \left| r_{ss}(l) \right|$
- But we obtain:

$$\widehat{r}_{ss}(l) = 0, \text{ for } |l| > L$$

$$\widehat{r}_{ss}(l) = \widehat{r}_{ss}(-l)$$

$$\left| \widehat{r}_{ss}(l) \right| \leq \widehat{r}_{ss}(0)$$

- The resulting (estimated) autocorrelation matrix is positive definite.
- The resulting (estimated) autocorrelation matrix has Toeplitz structure.

Levinson-Durbin Recursion – Part 1

Problem:

The solution of the equation system

$$\boldsymbol{R}_{ss}\,\boldsymbol{h}_{\text{opt}} = \boldsymbol{r}_{ss}(L)$$

has – depending on how the autocorrelation matrix $\boldsymbol{R}_{ss}$ is estimated – a complexity proportional to N^2 or N^3, respectively. In addition numerical problems can occur if the matrix is ill-conditioned.

Goal:

A robust solution method that avoids direct inversion of the matrix $\boldsymbol{R}_{ss}$.

Solution

Exploiting the Toeplitz structure of the matrix $\boldsymbol{R}_{ss}$:

- Recursion over the filter order
- Combining forward and backward prediction

Literature:

- J. Durbin: *The Fitting of Time Series Models*, Rev. Int. Stat. Inst., no. 28, pp. 233 - 244, 1960
- N. Levinson: *The Wiener RMS Error Criterion in Filter Design and Prediction*, J. Math. Phys., no. 25, pp. 261 - 268, 1947

- *Source-filter model for speech generation*
- *Literature*
- *Derivation of linear prediction*
- ***Levinson-Durbin recursion***
- *Application example*

Levinson-Durbin Recursion – Part 2 (Backward Prediction)

Equation system of the forward prediction:

$$\begin{array}{ccccccccc}
r(1) & = & h_0\, r(0) & + & h_1\, r(1) & + & \dots & + & h_{N-1}\, r(N-1) \\
r(2) & = & h_0\, r(1) & + & h_1\, r(0) & + & \dots & + & h_{N-1}\, r(N-2) \\
r(3) & = & h_0\, r(2) & + & h_1\, r(1) & + & \dots & + & h_{N-1}\, r(N-3) \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
r(N) & = & h_0\, r(N-1) & + & h_1\, r(N-2) & + & \dots & + & h_{N-1}\, r(0)
\end{array}$$

$$\text{for } h_i = h_{\text{opt},i}$$

Changing the equation order:

$$\begin{array}{ccccccccc}
r(N) & = & h_0\, r(N-1) & + & h_1\, r(N-2) & + & \dots & + & h_{N-1}\, r(0) \\
r(N-1) & = & h_0\, r(N-2) & + & h_1\, r(N-3) & + & \dots & + & h_{N-1}\, r(1) \\
r(N-2) & = & h_0\, r(N-3) & + & h_1\, r(N-4) & + & \dots & + & h_{N-1}\, r(2) \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
r(1) & = & h_0\, r(0) & + & h_1\, r(1) & + & \dots & + & h_{N-1}\, r(N-1)
\end{array}$$

$$\text{for } h_i = h_{\text{opt},i}$$

Levinson-Durbin Recursion – Part 3 (Backward Prediction)

After rearranging the equations:

$$
\begin{array}{ccccccccc}
r(N) & = & h_0\,r(N-1) & + & h_1\,r(N-2) & + & \dots & + & h_{N-1}\,r(0) \\
r(N-1) & = & h_0\,r(N-2) & + & h_1\,r(N-3) & + & \dots & + & h_{N-1}\,r(1) \\
r(N-2) & = & h_0\,r(N-3) & + & h_1\,r(N-4) & + & \dots & + & h_{N-1}\,r(2) \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
r(1) & = & h_0\,r(0) & + & h_1\,r(1) & + & \dots & + & h_{N-1}\,r(N-1)
\end{array}
$$

$$\text{for } h_i = h_{\text{opt},i}$$

Changing the order of the elements on the right side:

$$
\begin{array}{ccccccccc}
r(N) & = & h_{N-1}\,r(0) & + & h_{N-2}\,r(1) & + & \dots & + & h_0\,r(N-1) \\
r(N-1) & = & h_{N-1}\,r(1) & + & h_{N-2}\,r(0) & + & \dots & + & h_0\,r(N-2) \\
r(N-2) & = & h_{N-1}\,r(2) & + & h_{N-2}\,r(1) & + & \dots & + & h_0\,r(N-3) \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
r(1) & = & h_{N-1}\,r(N-1) & + & h_{N-2}\,r(N-2) & + & \dots & + & h_0\,r(0)
\end{array}
$$

$$\text{for } h_i = h_{\text{opt},i}$$

Levinson-Durbin Recursion – Part 4 (Backward Prediction)

After changing the order of the elements on the right side:

$$\begin{array}{ccccccccc}
r(N) & = & h_{N-1}\, r(0) & + & h_{N-2}\, r(1) & + & \dots & + & h_0\, r(N-1) \\
r(N-1) & = & h_{N-1}\, r(1) & + & h_{N-2}\, r(0) & + & \dots & + & h_0\, r(N-2) \\
r(N-2) & = & h_{N-1}\, r(2) & + & h_{N-2}\, r(1) & + & \dots & + & h_0\, r(N-3) \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
r(1) & = & h_{N-1}\, r(N-1) & + & h_{N-2}\, r(N-2) & + & \dots & + & h_0\, r(0)
\end{array}$$

$$\text{for } h_i = h_{\text{opt},i}$$

Matrix-vector notation:

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

Levinson-Durbin Recursion – Part 5 (Backward Prediction)

Matrix-vector notation:

$$\tilde{\boldsymbol{r}}_{ss}(1) = \boldsymbol{R}_{ss}\,\tilde{\boldsymbol{h}}_{\text{opt}}$$

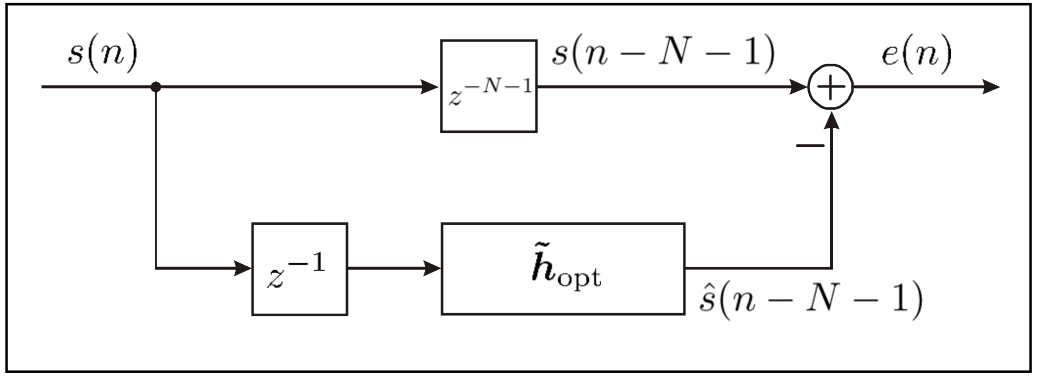
Due to symmetry of the autocorrelation function:

$$\begin{aligned}
\tilde{\boldsymbol{r}}_{ss}(1) &= \Big[r(N),\, r(N-1),\, \ldots,\, r(1)\Big]^{\mathrm{T}} \\
&\qquad \ldots\ldots\ldots \quad \text{inserting } r(l) = r(-l) \\
&= \Big[r(-N),\, r(-N+1),\, \ldots,\, r(-1)\Big]^{\mathrm{T}} \\
&= \boldsymbol{r}_{ss}(-N)
\end{aligned}$$

Backward prediction by N samples:

$$\begin{aligned}
\tilde{\boldsymbol{r}}_{ss}(1) &= \boldsymbol{R}_{ss}\,\tilde{\boldsymbol{h}}_{\text{opt}} \\
\boldsymbol{r}_{ss}(-N) &= \boldsymbol{R}_{ss}\,\tilde{\boldsymbol{h}}_{\text{opt}}
\end{aligned}$$

$$\boxed{\tilde{\boldsymbol{h}}_{\text{opt}} = \boldsymbol{R}_{ss}^{-1}\,\boldsymbol{r}_{ss}(-N)}$$



Levinson-Durbin Recursion – Part 6 (Derivation of the Recursion)

Derivation during the lecture …

Levinson-Durbin Recursion – Part 7 (Basic Structure of Recursive Algorithms)

Estimated signal using a prediction filter of length N:

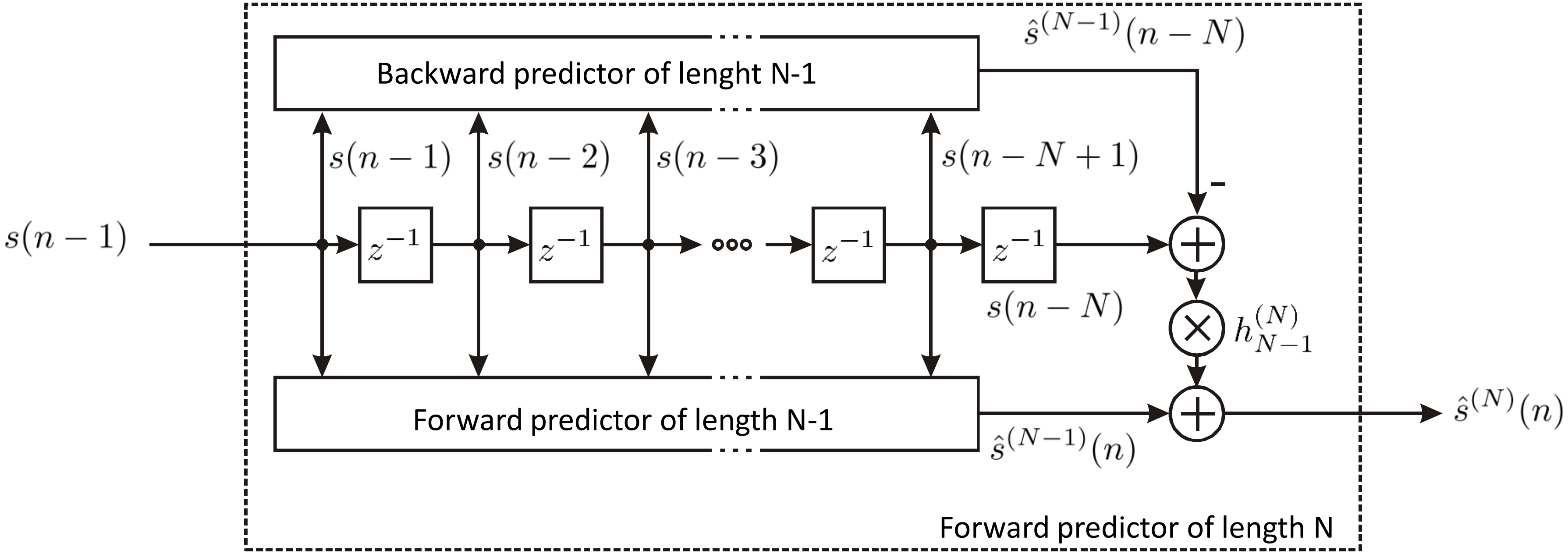
$$\hat{s}^{(N)}(n) = \sum_{i=0}^{N-1} h_i^{(N)} s(n-1-i) = \sum_{i=0}^{N-2} h_i^{(N)} s(n-1-i) + h_{N-1}^{(N)} s(n-N)$$

Inserting the recursion $h_i^{(N)} = h_i^{(N-1)} - h_{N-1}^{(N)}\, h_{N-2-i}^{(N-1)}$ ***:***

$$\begin{aligned}
\hat{s}^{(N)}(n) &= \sum_{i=0}^{N-2} \left(h_i^{(N-1)} - h_{N-1}^{(N)}\, h_{N-2-i}^{(N-1)}\right) s(n-1-i) + h_{N-1}^{(N)} s(n-N) \\
&= \sum_{i=0}^{N-2} h_i^{(N-1)} s(n-1-i) - h_{N-1}^{(N)} \sum_{i=0}^{N-2} h_{N-2-i}^{(N-1)} s(n-1-i) + h_{N-1}^{(N)} s(n-N) \\
&= \underbrace{\sum_{i=0}^{N-2} h_i^{(N-1)} s(n-1-i)}_{\text{Forward predictor of length N-1}} + h_{N-1}^{(N)} \overbrace{\left(\underbrace{s(n-N)}_{\text{Additional sample}} - \underbrace{\sum_{i=0}^{N-2} h_{N-2-i}^{(N-1)} s(n-1-i)}_{\text{Backward predictor of length N-1}} \right)}^{\text{Innovation}}
\end{aligned}$$

Levinson-Durbin Recursion – Part 8 (Basic Structure of Recursive Algorithms)

Structure that shows the recursion over the order:



In short form:

$$\hat{s}^{(N)}(n) \;=\; \hat{s}^{(N-1)}(n) \;+\; h_{N-1}^{(N)}\Big(s(n-N) - \hat{s}^{(N-1)}(n-N)\Big)$$

New estimation = old estimation + weighting * (new sample – estimated new sample)

Levinson-Durbin Recursion – Part 9 (Recursive Computation of the Error Power)

Minimal error power:

$$\begin{aligned}
\mathrm{E}\{e^2(n)\}\Big|_{\min} &= \mathrm{E}\left\{\left(s(n) - \sum_{j=0}^{N-1} h_{\mathrm{opt},j}\, s(n-1-j)\right)^2\right\} \\
&= r(0) - 2\sum_{j=0}^{N-1} h_{\mathrm{opt},j}\, r(j+1) + \sum_{j=0}^{N-1}\sum_{i=0}^{N-1} h_{\mathrm{opt},j}\, h_{\mathrm{opt},i}\, r(i-j) \\
&= r(0) - 2\,\boldsymbol{h}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{r}_{ss}(1) + \boldsymbol{h}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{R}_{ss}\,\boldsymbol{h}_{\mathrm{opt}}
\end{aligned}$$

Inserting $\boldsymbol{h}_{\mathrm{opt}} = \boldsymbol{R}_{ss}^{-1}\,\boldsymbol{r}_{ss}(1)$***:***

$$\begin{aligned}
\mathrm{E}\{e^2(n)\}\Big|_{\min} &= r(0) - 2\,\boldsymbol{h}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{r}_{ss}(1) + \boldsymbol{h}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{R}_{ss}\,\boldsymbol{R}_{ss}^{-1}\,\boldsymbol{r}_{ss}(1) \\
&= r(0) - \boldsymbol{h}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{r}_{ss}(1)
\end{aligned}$$

Order-recursive notation:

$$\begin{aligned}
E_{\min}^{(N)} &= r(0) - \left(\boldsymbol{h}_{\mathrm{opt}}^{(N)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N)}(1) \\
&= r(0) - r(N)\, h_{N-1}^{(N)} - \left[h_0^{(N)},\, \ldots,\, h_{N-2}^{(N)}\right] \boldsymbol{r}_{ss}^{(N-1)}(1)
\end{aligned}$$

Levinson-Durbin Recursion – Part 10 (Recursive Computation of the Error Power)

Minimal error power:

$$E_{\min}^{(N)} = r(0) - r(N)\, h_{N-1}^{(N)} - \left[h_0^{(N)},\, ...,\, h_{N-2}^{(N)}\right] \boldsymbol{r}_{ss}^{(N-1)}(1)$$

Inserting the Levinson recursion:

$$\begin{aligned}
E_{\min}^{(N)} &= r(0) - r(N)\, h_{N-1}^{(N)} - \left(\boldsymbol{h}_{\mathrm{opt}}^{(N-1)} - h_{N-1}^{(N)}\, \tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1) \\
&= r(0) - \left(\boldsymbol{h}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1) - h_{N-1}^{(N)} \left(r(N) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)\right) \\
&= E_{\min}^{(N-1)} - h_{N-1}^{(N)} \underbrace{\left(r(N) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)\right)}_{(1)}
\end{aligned}$$

Levinson-Durbin Recursion – Part 11 (Recursive Computation of the Error Power)

Recursion of the refection coefficient:

$$
\begin{aligned}
h_{N-1}^{(N)} &= \frac{r(N) - \left(\tilde{\boldsymbol{r}}_{ss}^{(N-1)}(1)\right)^{\mathrm{T}} \boldsymbol{h}_{\mathrm{opt}}^{(N-1)}}{r(0) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \tilde{\boldsymbol{r}}_{ss}^{(N-1)}(1)} \\
&\quad \text{Transpose numerator and denominator, mirror all vectors} \\
&= \frac{r(N) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)}{r(0) - \left(\boldsymbol{h}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)} \\
&= \frac{r(N) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)}{E_{\mathrm{min}}^{(N-1)}}
\end{aligned}
$$

Rearranging:

$$
\underbrace{h_{N-1}^{(N)} E_{\mathrm{min}}^{(N-1)}}_{(2)} = r(N) - \left(\tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)
$$

Levinson-Durbin Recursion – Part 12 (Recursive Computation of the Error Power)

Previous results:

$$E_{\min}^{(N)} = E_{\min}^{(N-1)} - h_{N-1}^{(N)} \underbrace{\left(r(N) - \left(\tilde{\boldsymbol{h}}_{\text{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)\right)}_{(1)}$$

$$\underbrace{h_{N-1}^{(N)}\, E_{\min}^{(N-1)}}_{(2)} = r(N) - \left(\tilde{\boldsymbol{h}}_{\text{opt}}^{(N-1)}\right)^{\mathrm{T}} \boldsymbol{r}_{ss}^{(N-1)}(1)$$

Inserting (2) in (1):

$$\boxed{E_{\min}^{(N)} = E_{\min}^{(N-1)} \left(1 - \left(h_{N-1}^{(N)}\right)^2\right)}$$

Remarks:

- Start of the recursion: $E_{\min}^{(0)} = r(0)$
- The error power should not increase when increasing the filter order. For that reason the error power is a suitable quantity for checking if the recursion should terminated due to rounding errors, etc.

Levinson-Durbin Recursion – Part 13 (Summary)

Initialization

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

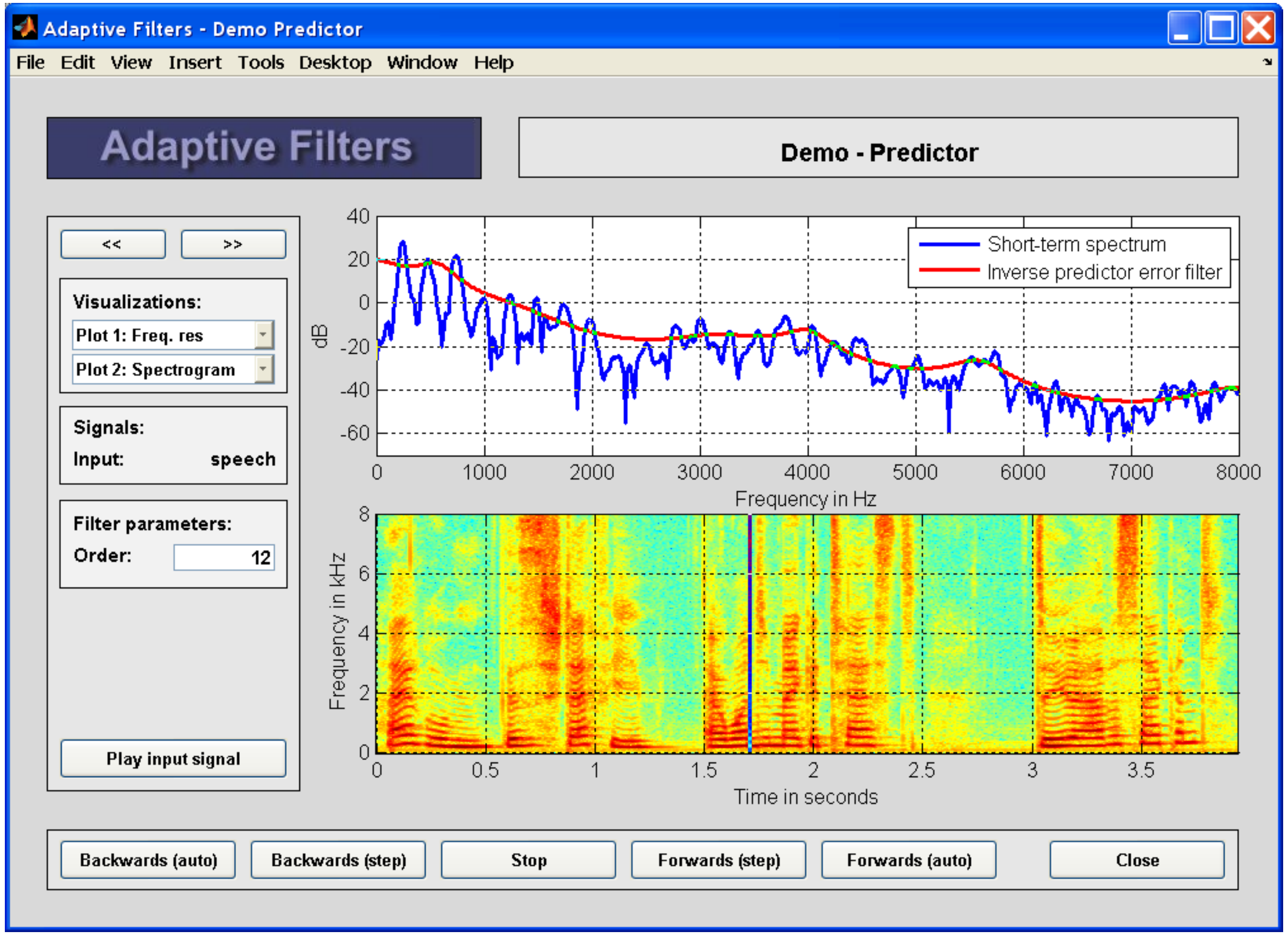
Recursion:

- Reflection 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: $$\left[h_0^{(N)}, h_1^{(N)}, ..., h_{N-2}^{(N)}\right]^{\mathrm{T}} = \boldsymbol{h}_{\mathrm{opt}}^{(N-1)} - h_{N-1}^{(N)} \tilde{\boldsymbol{h}}_{\mathrm{opt}}^{(N-1)}$$
- Backward predictor: $$\tilde{h}_i^{(N)} = h_{N-i-1}^{(N)}$$
- Error power (optional): $$E_{\min}^{(N)} = E_{\min}^{(N-1)} \left[1 - \left(h_{N-1}^{(N)}\right)^2\right]$$

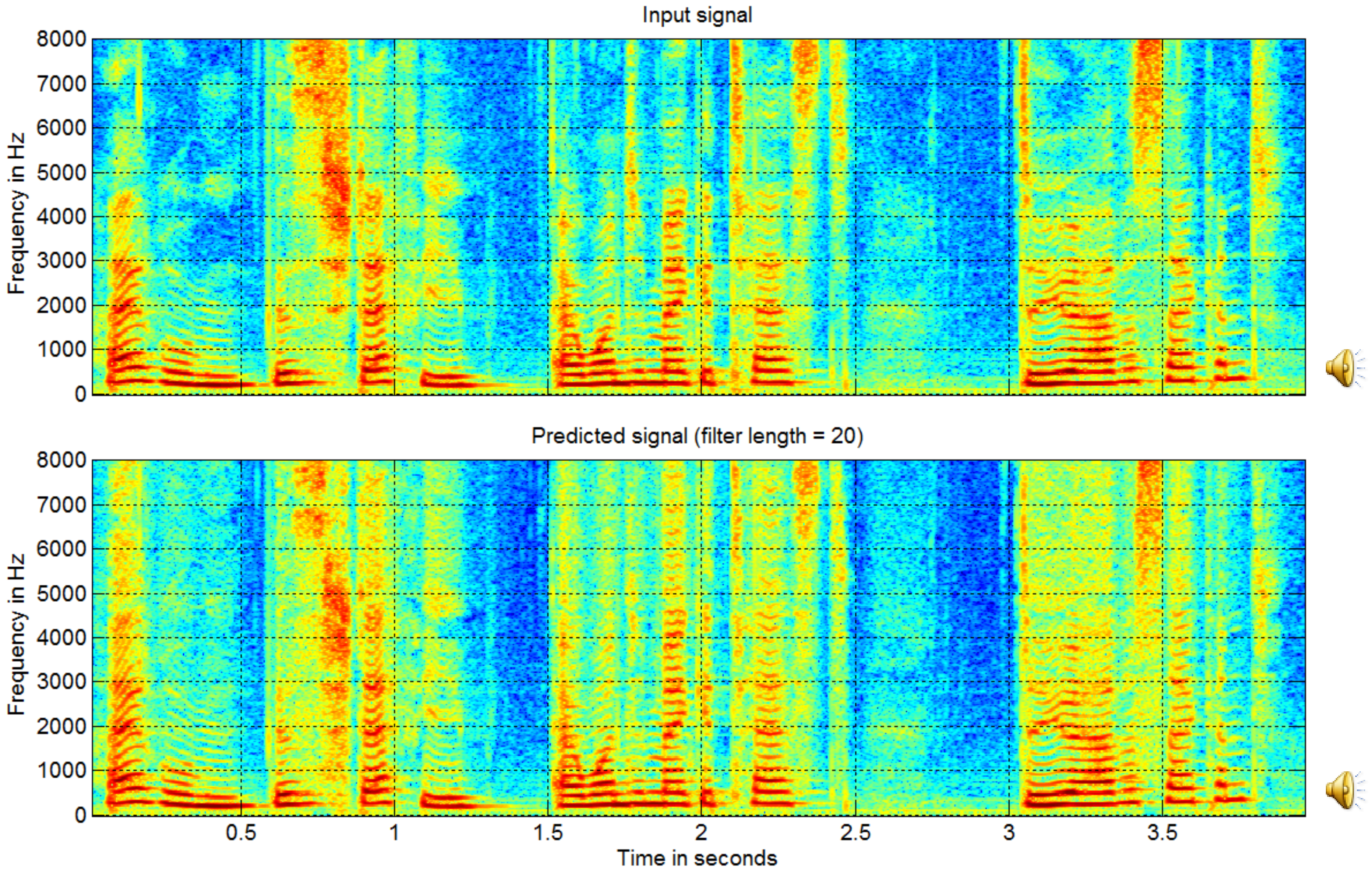
Condition for termination:

- Numerical problems: If $\left(h_{N-1}^{(N)}\right)^2 > 1 - \varepsilon$ is true, use the coefficients of the previous recursion and fill the missing coefficients with zeros.
- Order: If the desired filter order is reached, stop the recursion.

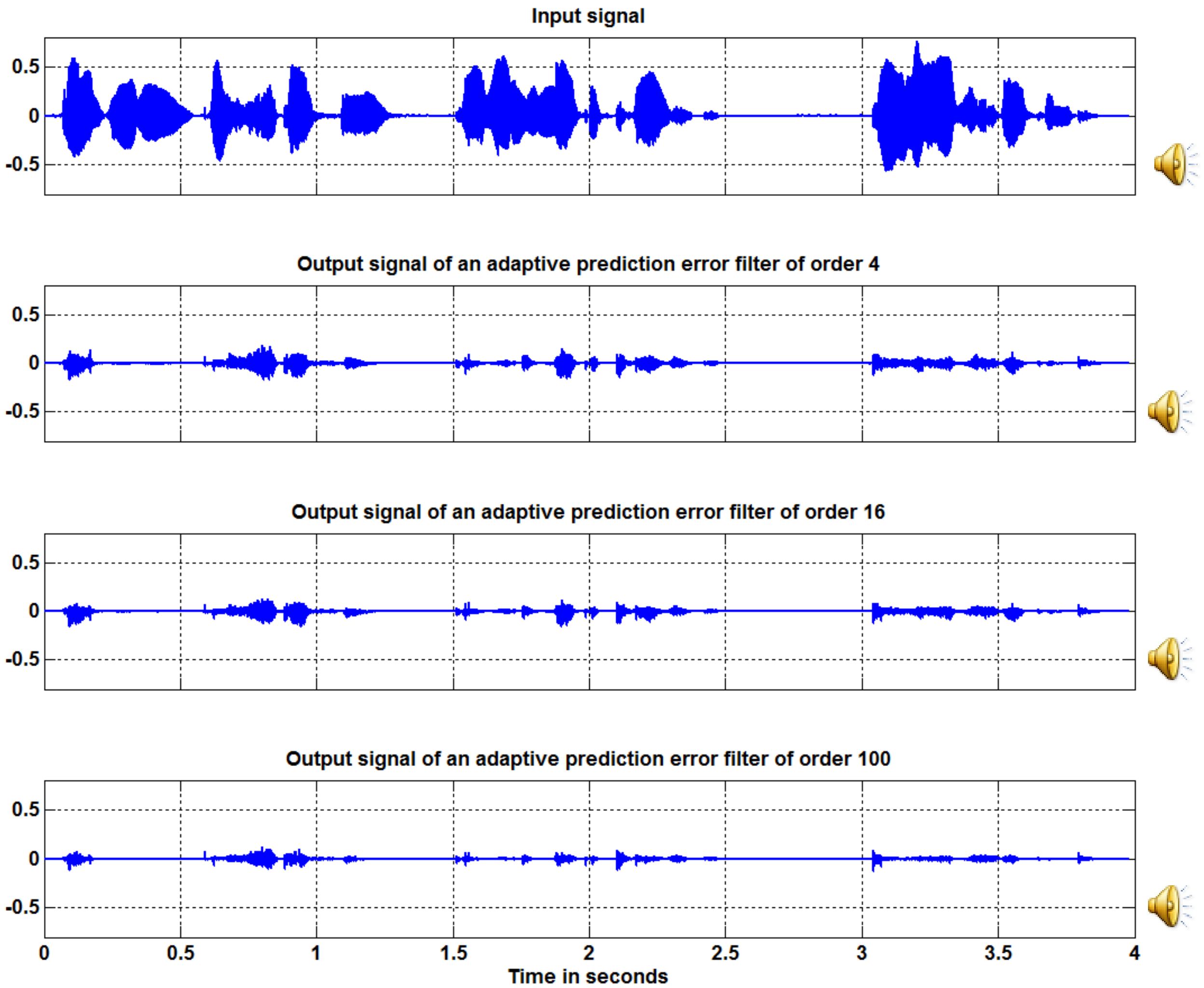
Matlab Demo


Adaptive Filters - Demo Predictor
File Edit View Insert Tools Desktop Window Help
Adaptive Filters
Demo - Predictor
<<
>>
Visualizations:
Plot 1: Freq. res
Plot 2: Spectrogram
Signals:
Input: speech
Filter parameters:
Order: 12
Play input signal
Short-term spectrum
Inverse predictor error filter
dB
Frequency in Hz
Frequency in kHz
Time in seconds
Backwards (auto)
Backwards (step)
Stop
Forwards (step)
Forwards (auto)
Close

Matlab Demo – Input Signal and Estimated Signal



Matlab Demo – Error Signals



Summary and Outlook

This week:

- Source-filter model for speech generation
- Derivation of linear prediction
- Levinson-Durbin recursion
- Application example

Next week:

- Adaptation algorithms – part 1