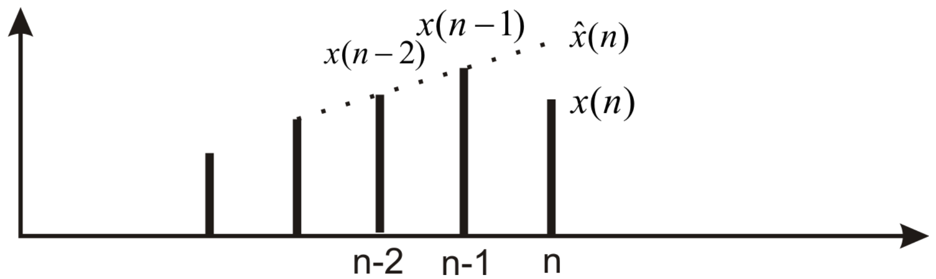
ADAPTIVE FILTER
an example: linear predictor



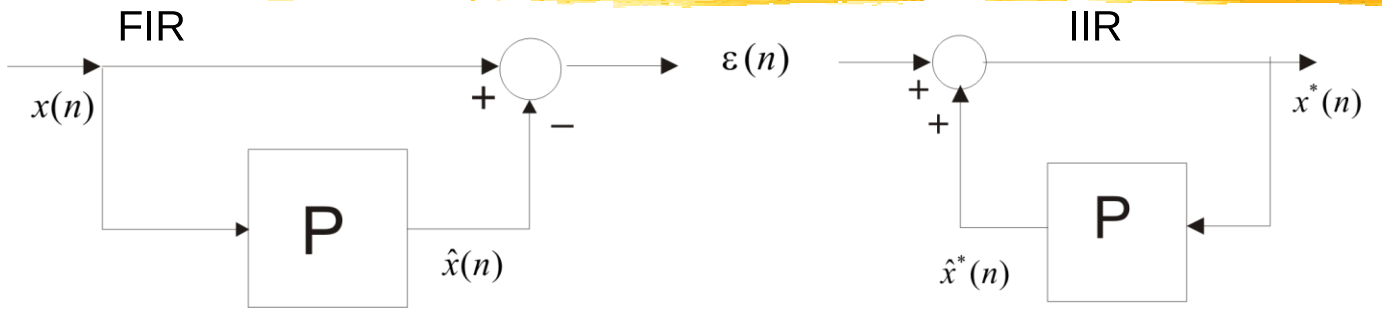
Prediction of the current sample x(n) is linear combination of M previous samples

$$\hat{x}(n) = \sum_{i=1}^{M} a_i \, x(n-i) = \bar{a}^T \bar{x}(n)$$

$\bar{a} = [a_1, \ldots, a_M]^T$ - prediction coefficients

If coefficients depend on time, we have an adaptive filter: $\bar{a} = \bar{a}(n)$

FIR and IIR adaptive filters



$$\varepsilon(n) = x(n) - \hat{x}(n) = x(n) - \sum_{i=1}^{M} a_i(n)x(n-i)$$ - prediction error

IIR predictive filter is used for synthesis of speech signal in cellular telephony, VOIP etc.
If prediction error signal is not quantized, then at the output of IIR filter undistorted signal x(n) is obtained.

Analysis of adaptive filters

Description of linear adaptive filter: **difference equations**, e.g. FIR filter:

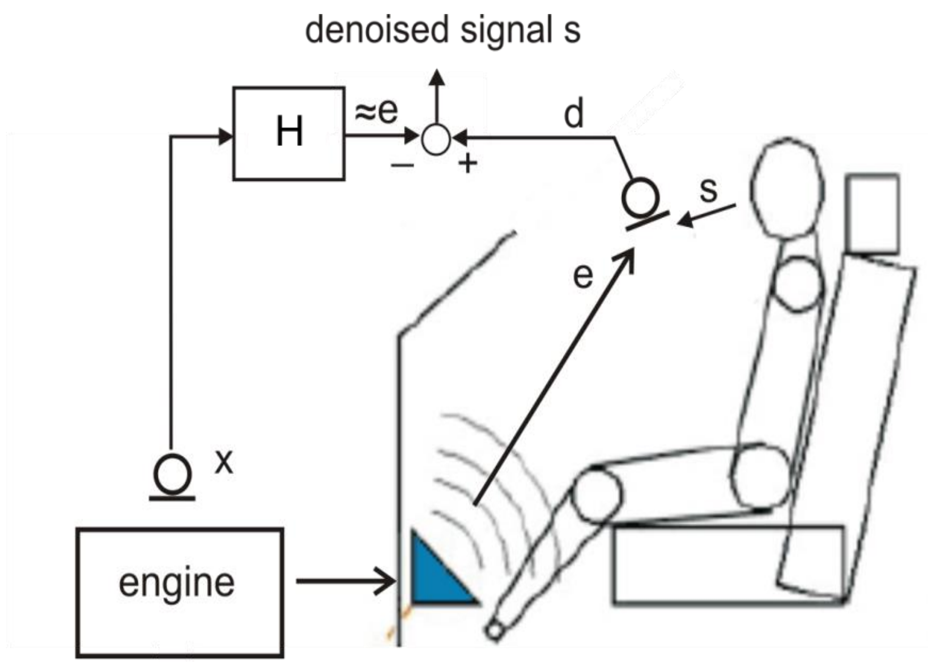
$$\hat{x}(n) = \sum_{i=0}^{M-1} h_i(n)x(n-i) = \bar{h}^T(n)\bar{x}(n) \qquad \bar{h}(n) = \begin{matrix} h_0(n) \\ \vdots \\ h_{M-1}(n) \end{matrix} \qquad \bar{x}(n) = \begin{matrix} x\,(n) \\ \vdots \\ x\,(n-M+1) \end{matrix}$$

Filter coefficients are varying in time according to adaptation algorithm, e.g.

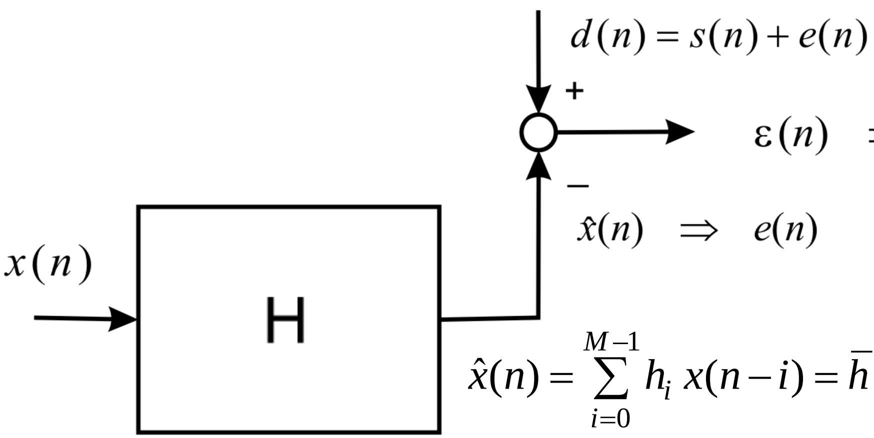
$$\bar{h}(n+1) = \bar{h}(n) + \Delta\bar{h}(n)$$

Impulse response is not very useful; Kronecker delta at the input $x(n) = \delta(n)$ yields at the output $y(n) = h_n(n)$. If coefficients vary slowly, then $y(n) \approx h_n(0)$ and $\bar{h}(n)$ may be regarded as time varying impulse response.
Zet transform shouldn't be used for adaptive filters, because it requires constant values of filter coefficients (time invariance). However, if filter coefficients are constant for certain time, Zet transform is locally used to obtain frequency response, perform stability analysis etc. This approach is used in analysis of speech coders, e.g. CELP coders.

TASKS TO BE SOLVED USING ADAPTIVE FILTERING: 1. DENOISING



1. DENOISING



$\varepsilon(n) \Rightarrow s(n)$

$\hat{x}(n) \Rightarrow e(n)$

$$\hat{x}(n) = \sum_{i=0}^{M-1} h_i \, x(n-i) = \bar{h}^T \bar{x}(n)$$

adaptive filter

$h_i = h_i(n)$

$s(n)$ - speech or music

$e(n)$ - noise

$d(n)$ - signal with noise (microphone 1)

$x(n)$ - noise recorded with the second microphone, correlated with *e*(*n*), not correlated with *s*(*n*)

$\hat{x}(n)$ - filtered noise

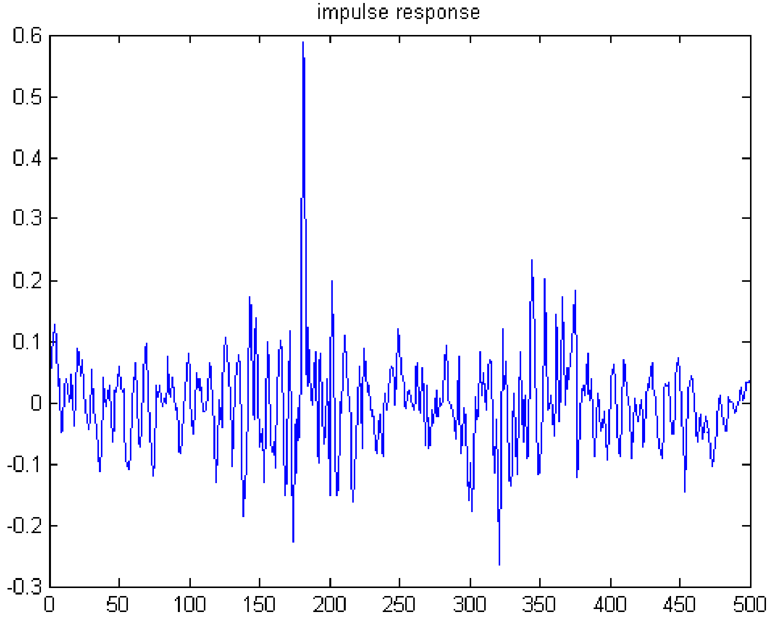
$\varepsilon(n) = s(n) + e(n) - \hat{x}(n) \approx s(n)$ - signal with noise suppressed

transposition

Searched: $\bar{h} = [h_0, h_1, \ldots, h_{M-1}]^T$ - impulse response of the adaptive FIR filter

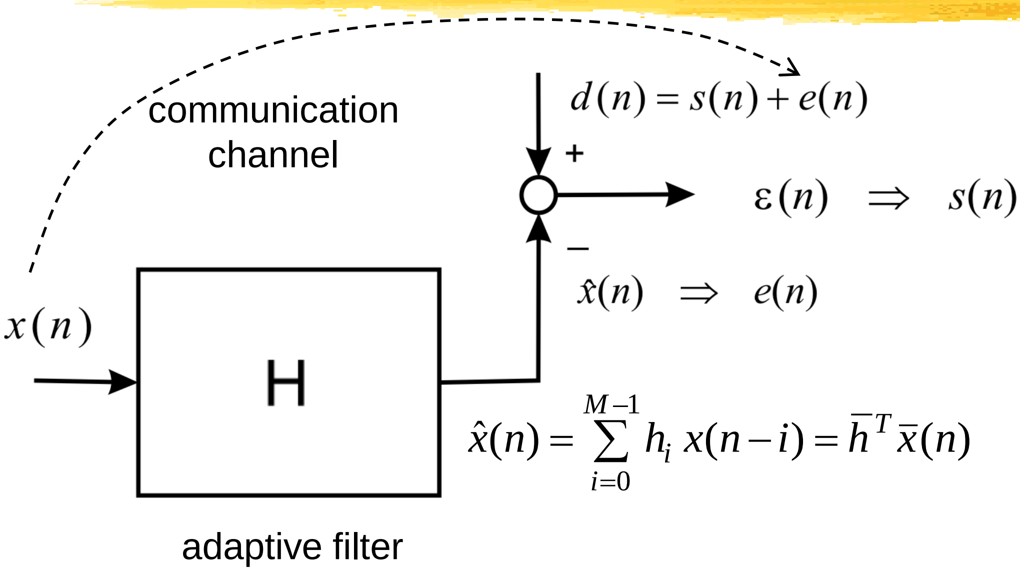
Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

1. DENOISING



Example: coefficients of an adaptive denoising filter

2. ECHO CANCELLATION in telephonic speech



$$\hat{x}(n) = \sum_{i=0}^{M-1} h_i\, x(n-i) = \bar{h}^T \bar{x}(n)$$

$s(n)$ - speech of the remote user

$x(n)$ - speech of the local user

$e(n)$ - echo of the local speaker (signal x(n), delayed and filtered in the communication channel)

$d(n)$ - signal received by the local user

$\hat{x}(n)$ - filtered speech of the local speaker

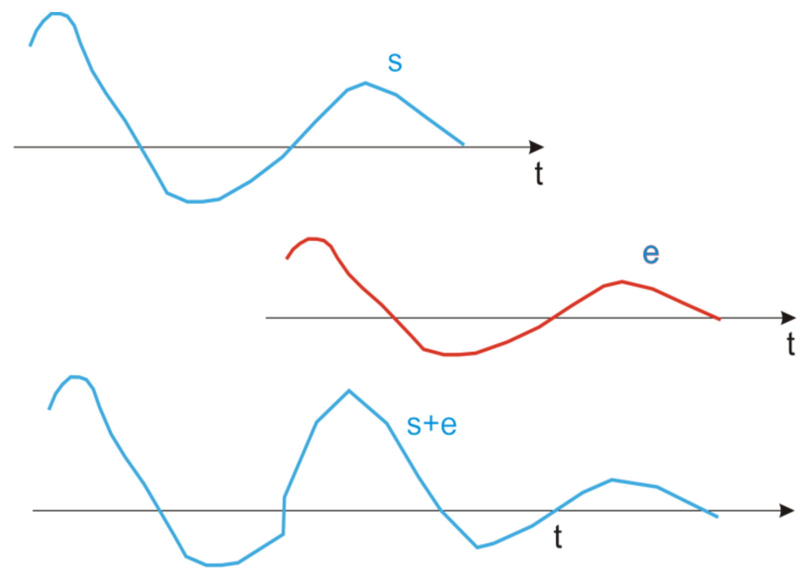
$\varepsilon(n) = s(n) + e(n) - \hat{x}(n) \approx s(n)$ - speech with echo suppressed

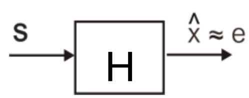
Searched: $\bar{h} = [h_0, h_1, ..., h_{M-1}]^T$ - impulse response of the adaptive FIR filter

transposition

Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

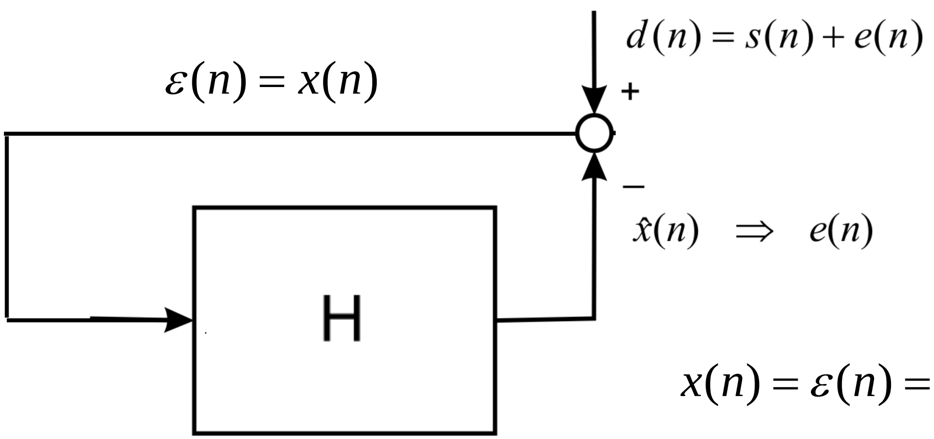
3. ECHO CANCELLATION
in audio file





Signal s is unknown, so $s+e-\hat{x}$ is filtered

3. ECHO CANCELLATION in audio file



adaptive filter

$s(n)$ - original speech or music

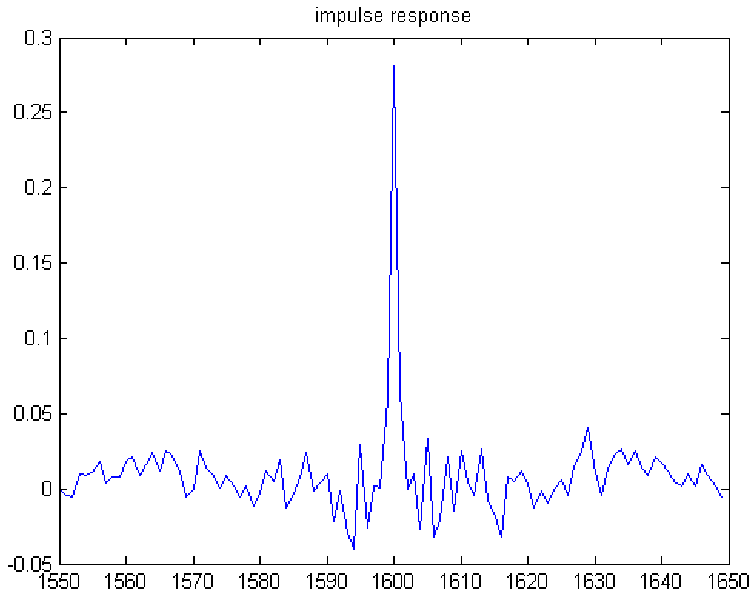
$e(n)$ - echo, e.g. $e(n)$ = a s(n-del)

$d(n)$ - signal with echo (audio file)

$\hat{x}(n) \approx e(n)$ - approximation of echo signal

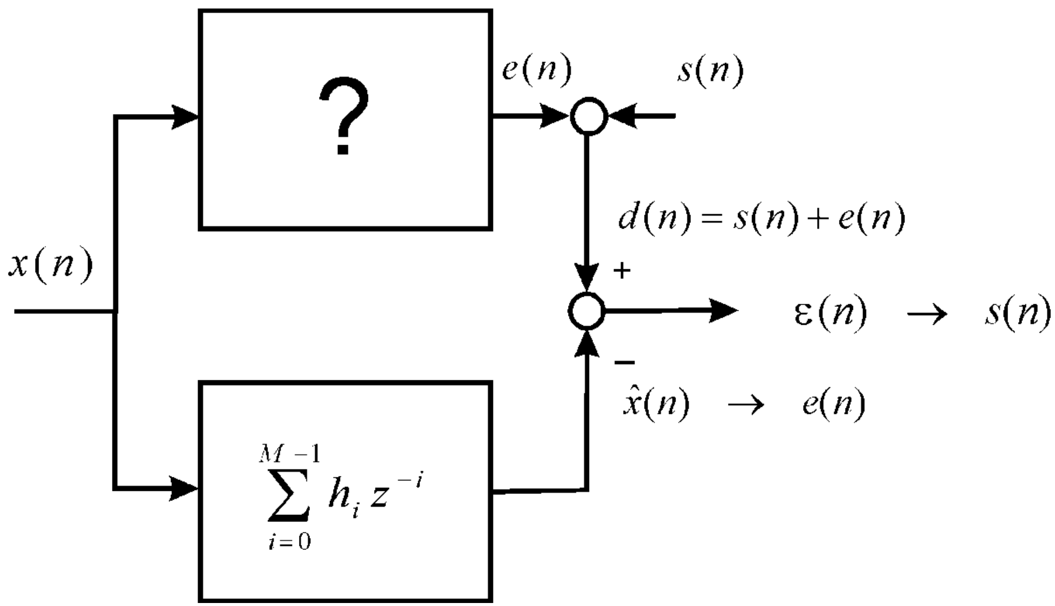
$x(n) = \varepsilon(n) = s(n) + e(n) - \hat{x}(n) \approx s(n)$ - audio signal with echo suppressed

3. ECHO CANCELLATION
in audio file



Example: coefficients of an adaptive echo suppressing filter

4. IDENTIFICATION OF A DYNAMIC OBJECT



$s(n)$ - measurement noise or zero

$x(n)$ - input signal

$e(n)$ - output signal

$d(n) \approx e(n)$ - observed signal

$\hat{x}(n)$ - filtered signal

$\varepsilon(n)$ - error signal

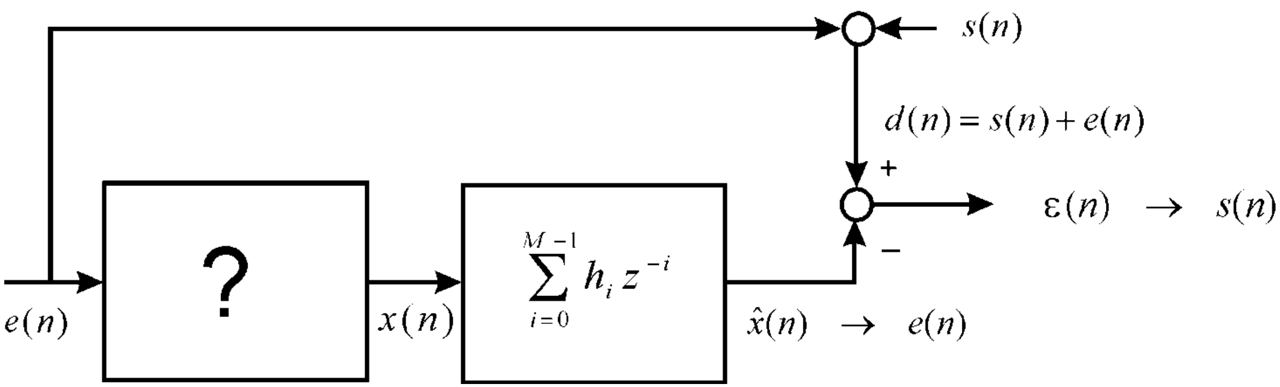
$H(z) = \sum_{i=0}^{M-1} h_i z^{-i}$ - model of „?"

transposition

Searched: $\bar{h} = [h_0, h_1, \ldots, h_{M-1}]^T$ - impulse response of the adaptive FIR filter

Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

5. CORRECTOR OF TRANSMISSION CHANNEL



$s(n)$ – measurement noise or zero

$e(n)$ – transmitted signals (known)

$x(n)$ – output of the channel

$\hat{x}(n)$ - output of the corrector

$s(n)+e(n)-\hat{x}(n)$ - error signal

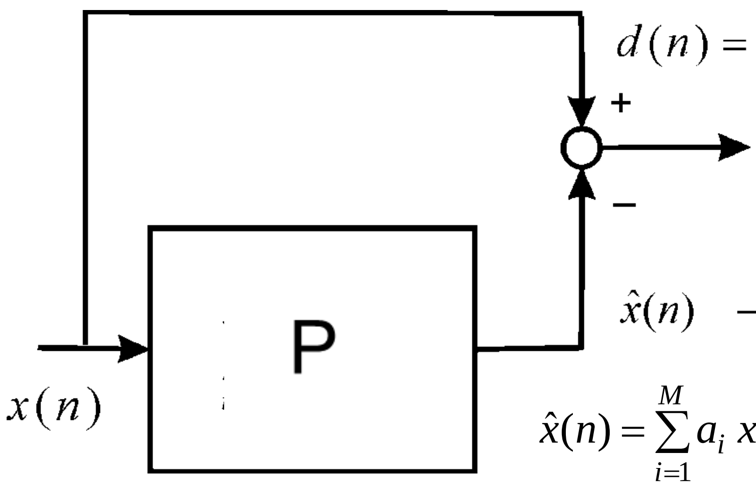
$$\hat{x}(n)=\sum_{i=0}^{M-1} h_i\, x(n-i)=\overline{h}^T\overline{x}(n)$$

transposition

Searched: $\overline{h}=[h_0, h_1, \ldots, h_{M-1}]^T$ - impulse response of the adaptive FIR filter

Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

6. PREDICTION



$$d(n) = s(n) + x(n)$$

$$\varepsilon(n) \rightarrow s(n)$$

$$\hat{x}(n) \rightarrow x(n)$$

$$\hat{x}(n) = \sum_{i=1}^{M} a_i \, x(n-i) = \bar{a}^T \bar{x}(n)$$

$x(n)$ - current signal sample

$s(n)$ - measurement noise or zero

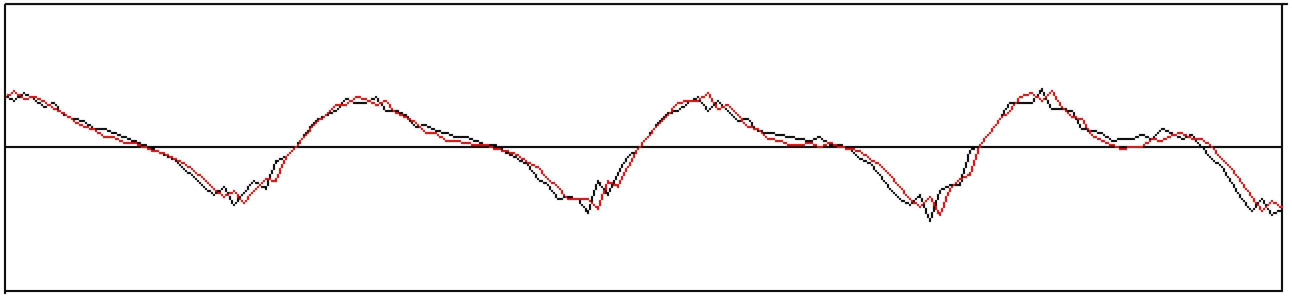
$\hat{x}(n)$ -prediction of $x(n)$

$\varepsilon(n) = s(n) + x(n) - \hat{x}(n)$ - prediction error

Searched: $a = [a_1, \ldots, a_M]^T$ - prediction coefficients

Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

6. PREDICTION



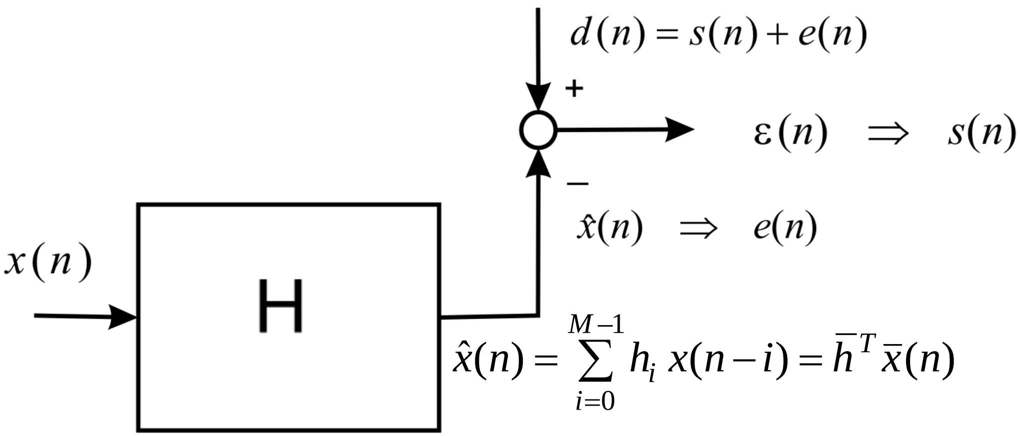
Example: speech signal (black) and its prediction (red)
-10 prediction coefficients

Prediction gain

$$G_p = \frac{\sum x^2(n)}{\sum[x(n) - \hat{x}(n)]^2} = \frac{\sum x^2(n)}{\sum \varepsilon^2(n)}$$

$$G_p[dB] = 10log_{10}(G_p)$$

GENERAL SCHEME



$s(n)$ - signal 1

$e(n)$ - signal 2

$d(n)$ - observed signal

$x(n)$ - signal correlated with $e(n)$, not correlated with $s(n)$

$\hat{x}(n)$ - filtered signal

$\varepsilon(n) = s(n) + e(n) - \hat{x}(n)$ - error signal

transposition

Searched: $\bar{h} = [h_0, h_1, \ldots, h_{M-1}]^T$ - impulse response of the adaptive FIR filter

Criterion: minimum energy of the error signal $\sum_n \varepsilon^2(n)$

FILTER ADAPTATION ALGORITHM

Sequential adaptation

For every signal sample new filter coefficients are calculated, by adding a correction $\Delta\bar{h}(n)$ to the previous coefficients:

$$\bar{h}(n+1) = \bar{h}(n) + \Delta\bar{h}(n)$$

Corrections should minimize power of the error signal: $\min\,[\varepsilon(n)]^2$

This is the **LMS** (least mean square) criterion.
Filter adaptation algorithms using this criterion are called LMS methods or SG (stochastic gradient) methods.

LMS ALGORITHM

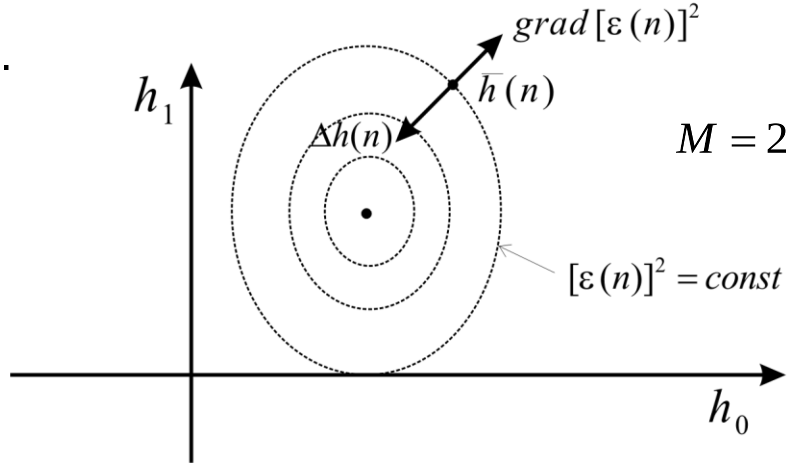
Gradient $grad[\varepsilon(n)]^2 = \dfrac{\partial}{\partial \bar{h}(n)}[\varepsilon(n)]^2$

shows direction of maximum growth of $[\varepsilon(n)]^2$.

Therefore the correction $\Delta\bar{h}(n)$

should have the opposite direction:

$$\Delta\bar{h}(n) \propto -grad[\varepsilon(n)]^2$$



LMS ALGORITHM

Instantaneous error power: $[\varepsilon(n)]^2 = [d(n) - \bar{h}^T(n)x(n)]^2$

is a function of filter coefficients $\bar{h}(n)$.

Gradient (vector of derivatives) is calculated as follows:

$$grad[\varepsilon(n)]^2 = \frac{\partial}{\partial \bar{h}}[d(n) - \bar{h}^T(n)x(n)]^2 = -2\varepsilon(n)x(n)$$

Correction (the opposite direction): $\Delta\bar{h}(n) = -\frac{\beta}{2} grad[\varepsilon(n)]^2 = \beta\varepsilon(n)x(n)$

LMS (or SG) algorithm: $\bar{h}(n+1) = \bar{h}(n) + \Delta\bar{h}(n) = \bar{h}(n) + \beta\varepsilon(n)x(n)$

β – adaptation speed

LMS (or SG) with normalization: $\beta = \frac{L_\beta}{\sigma_n^2 + M}$ σ_n^2 - signal power