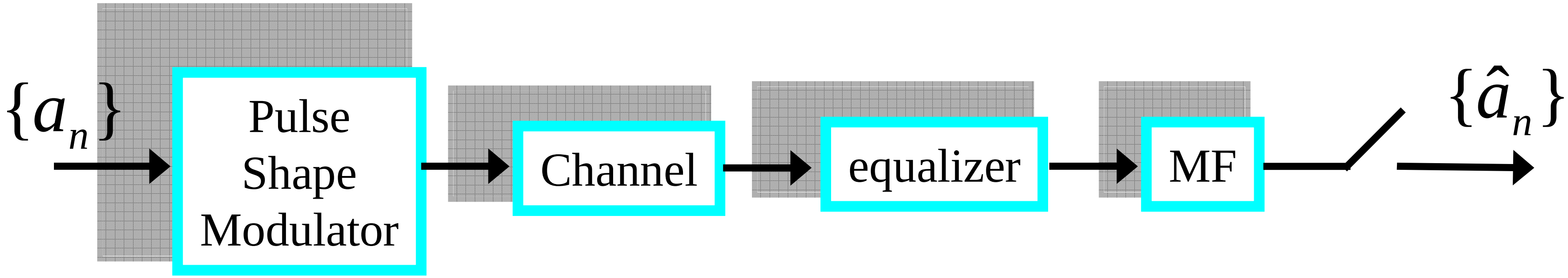
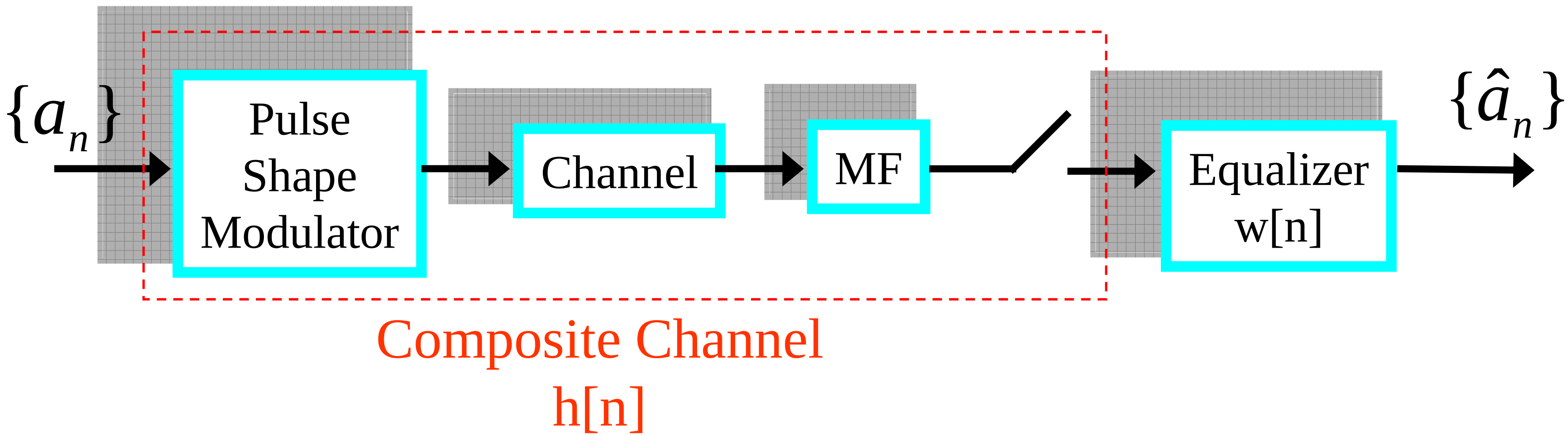
Blind Adaptive Filters & Equalization

Table of contents:

Equalization





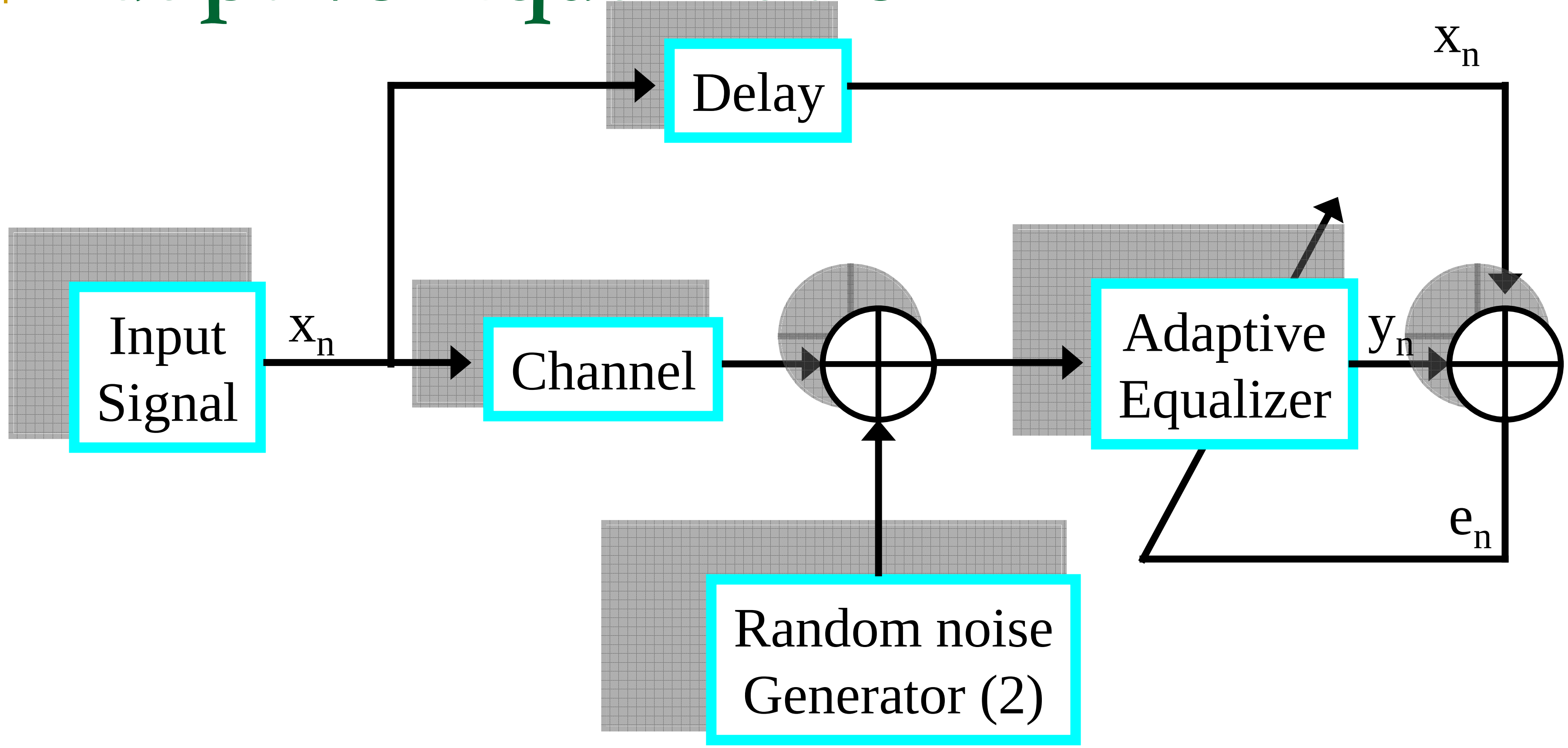
$$h[n]*w[n]=\delta[n]\Rightarrow\sum_{k=-N}^{N}w_k h_{n-k}=\begin{cases}1 & n=0\\ 0 & n=\pm1,\pm2,\ldots\end{cases}$$

$$w[n]=\sum_{i=-N}^{N}w_i\delta[n-i]$$

Example:

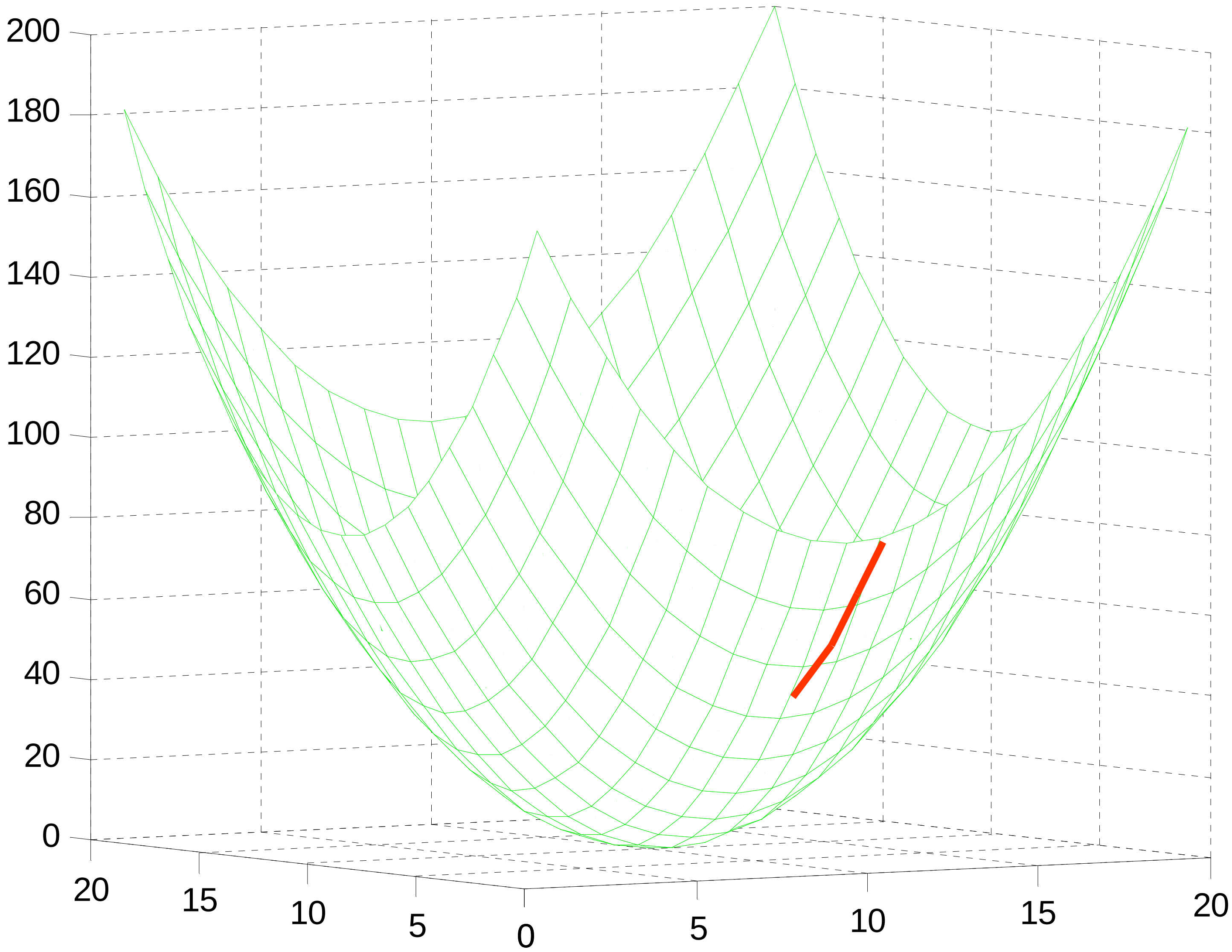
$$\begin{bmatrix} h_0 & h_{-1} & h_{-2} & h_{-3} & h_{-4} \\ h_1 & h_0 & h_{-1} & h_{-2} & h_{-3} \\ h_2 & h_1 & h_0 & h_{-1} & h_{-2} \\ h_3 & h_2 & h_1 & h_0 & h_{-1} \\ h_4 & h_3 & h_2 & h_1 & h_0 \end{bmatrix}\begin{bmatrix} w_{-2} \\ w_{-1} \\ w_0 \\ w_1 \\ w_{+2} \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Adaptive Equalization



$$e_n = a_n - y_n, \quad error = E[e_n^2]$$

$$= E[a_n{}^2 + (\sum w_k x_{n-k})^2 - 2a_n \sum w_k x_{n-k}]$$


200
180
160
140
120
100
80
60
40
20
0
20
15
10
5
0
5
10
15
20

$$y_n = \sum_{k=-N}^{N} w_k x_{n-k}$$

$$e_n = a_n - y_n$$

$$error = E[e_n^2]$$

$$w_k(n+1) = w_k(n)a_n - \frac{1}{2}\mu \frac{\partial\, error}{\partial\, w_k}$$

$$\frac{\partial\, error}{\partial\, w_k} = 2E[e_n \frac{\partial\, e_n}{\partial\, w_k}]$$

$$= -2E[e_n \frac{\partial\, y_n}{\partial\, w_k}] = -2E[e_n x_{n-k}]$$

$$X_n = [x_{n+N}, \ldots, x_{n+1}, x_n, x_{n-1}, \ldots, x_{n-N}]^T$$

$$W_n = [w_{-N}, \ldots, w_{-1}, w_0, w_1, \ldots, w_N]^T$$

$$y_n = \sum_{k=-N}^{N} w_k x_{n-k}$$

$$Y_n = X_n^T W_n$$

$$e_n = a_n - y_n$$

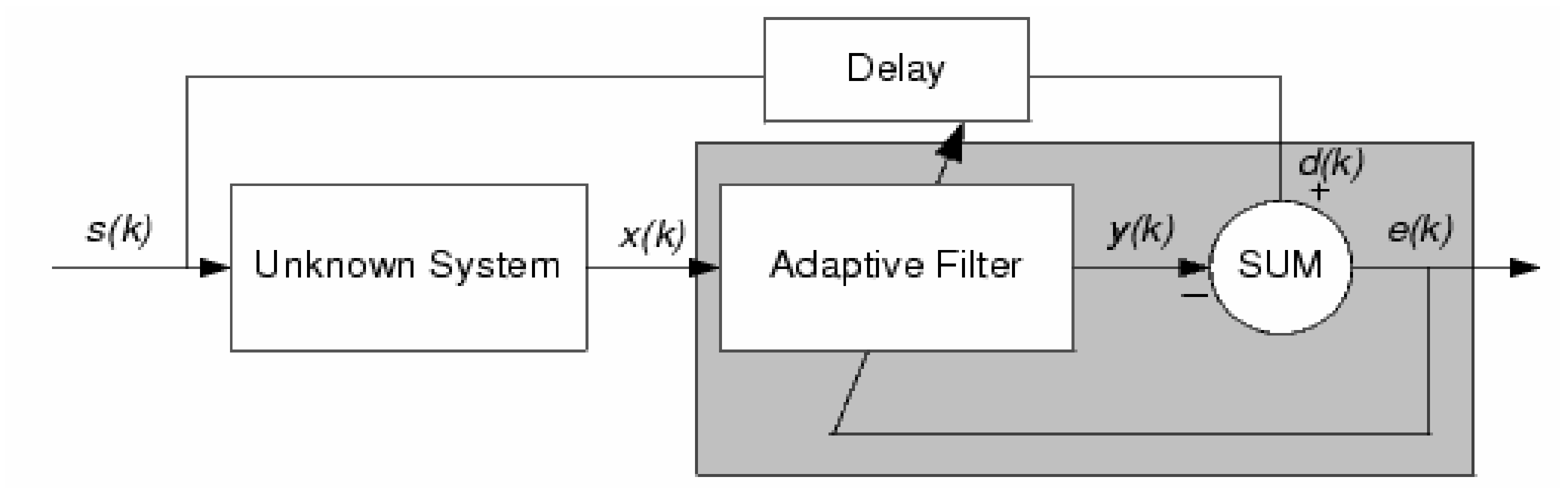
$$e_n = a_n - y_n$$

$$w_k(n+1) = w_k(n) a_n - \frac{1}{2}\mu \frac{\partial\, error}{\partial\, w_k}$$

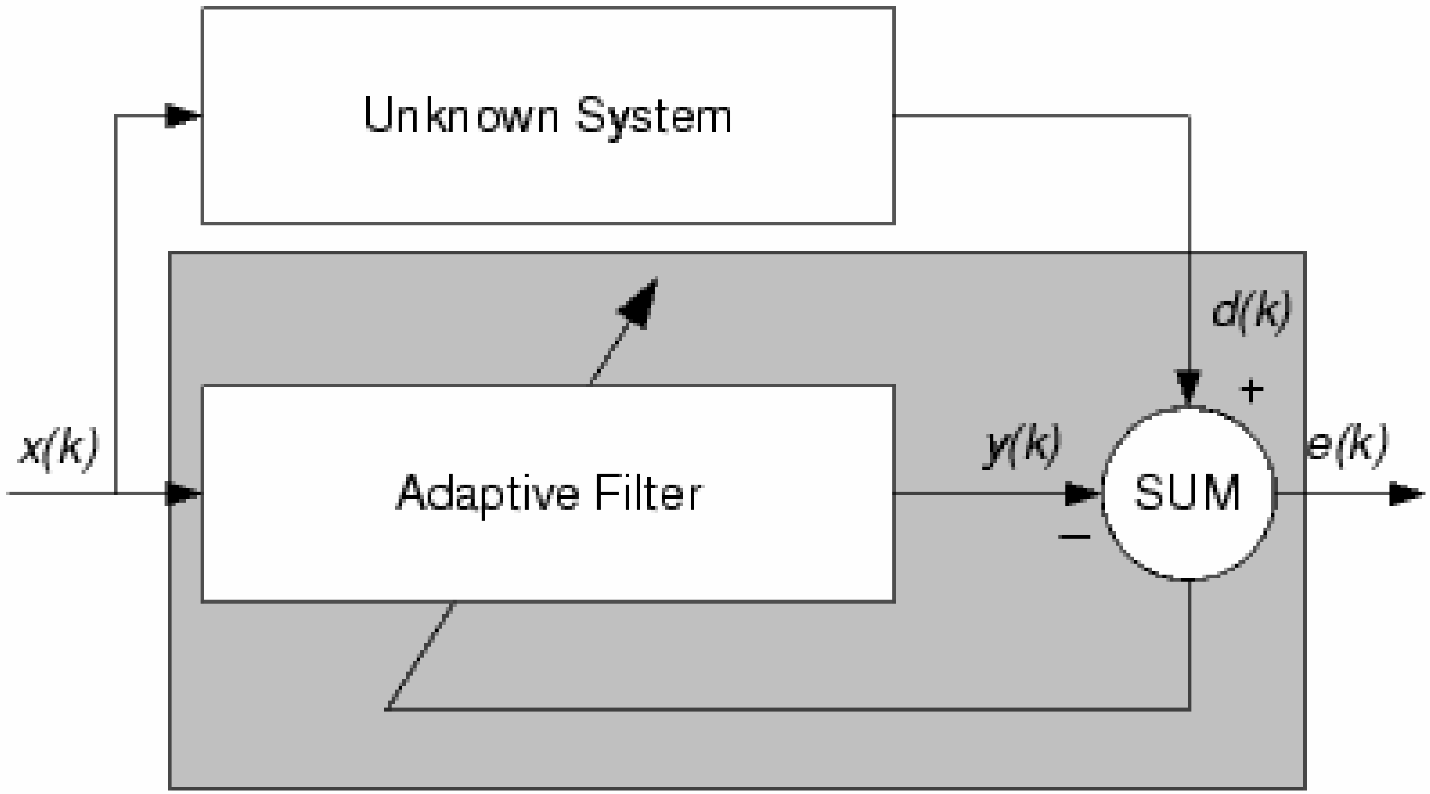
$$\frac{\partial\, error}{\partial\, w_k} = -2E[e_n x_{n-k}]$$

$$w_{n+1} = w_n + \mu\, e_n x_n$$

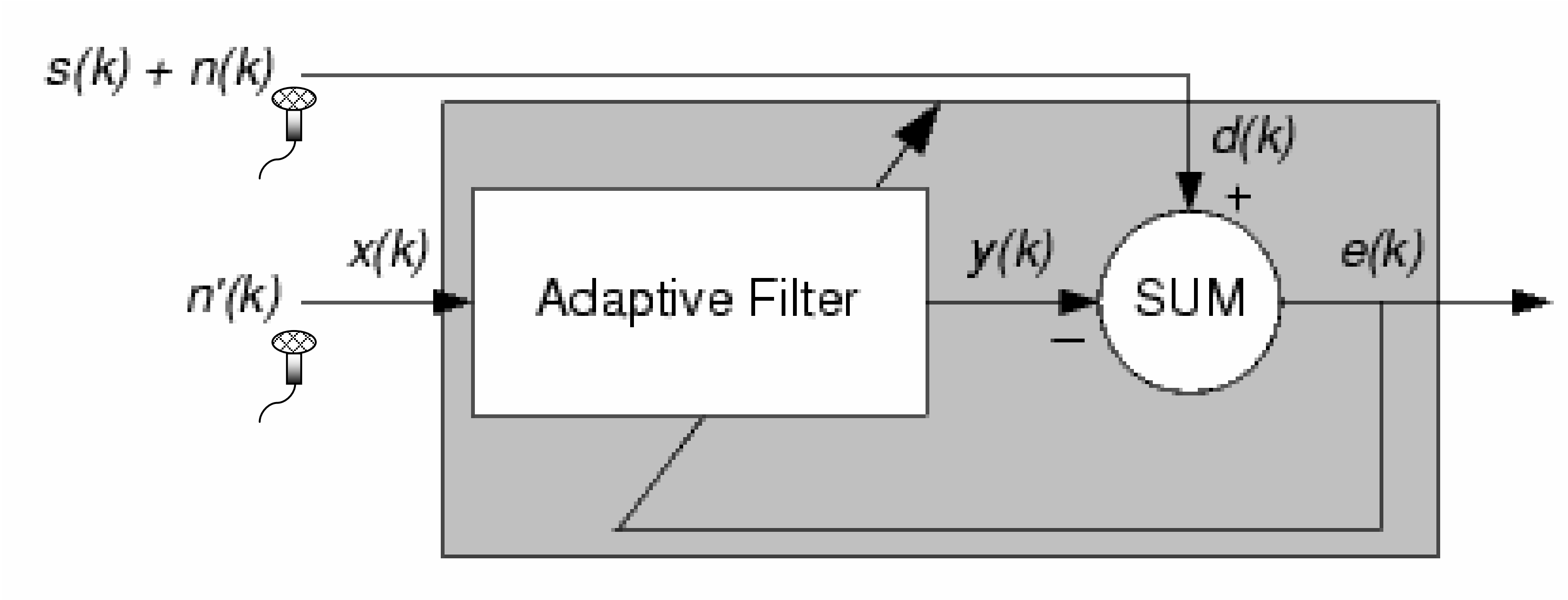
Equalization, Deconvolution, System compenstation



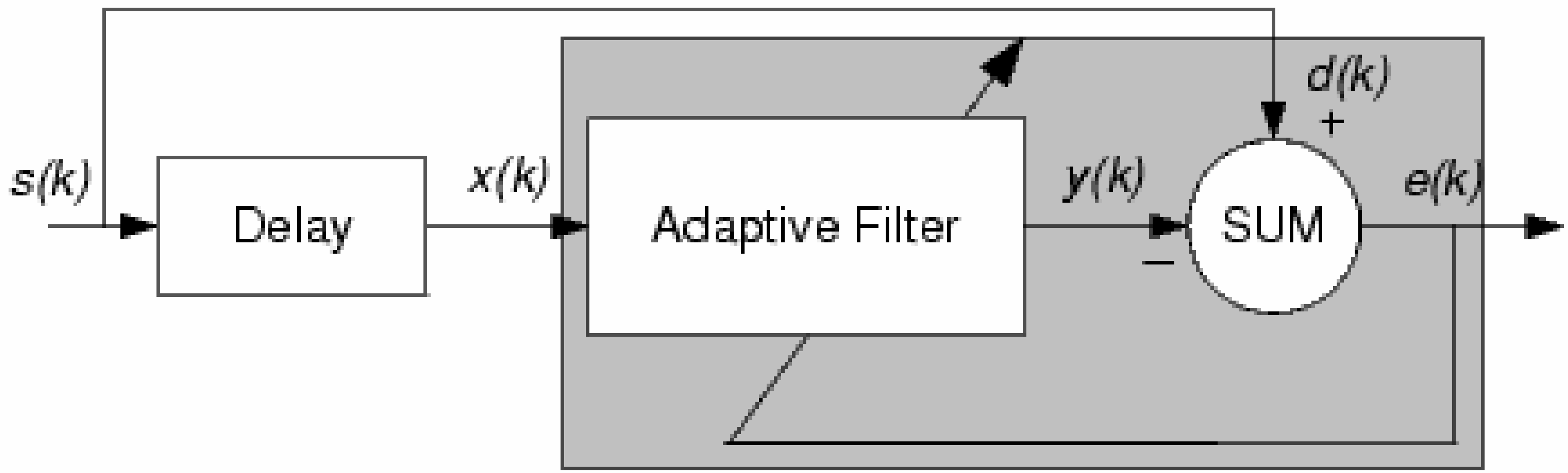
System Identification



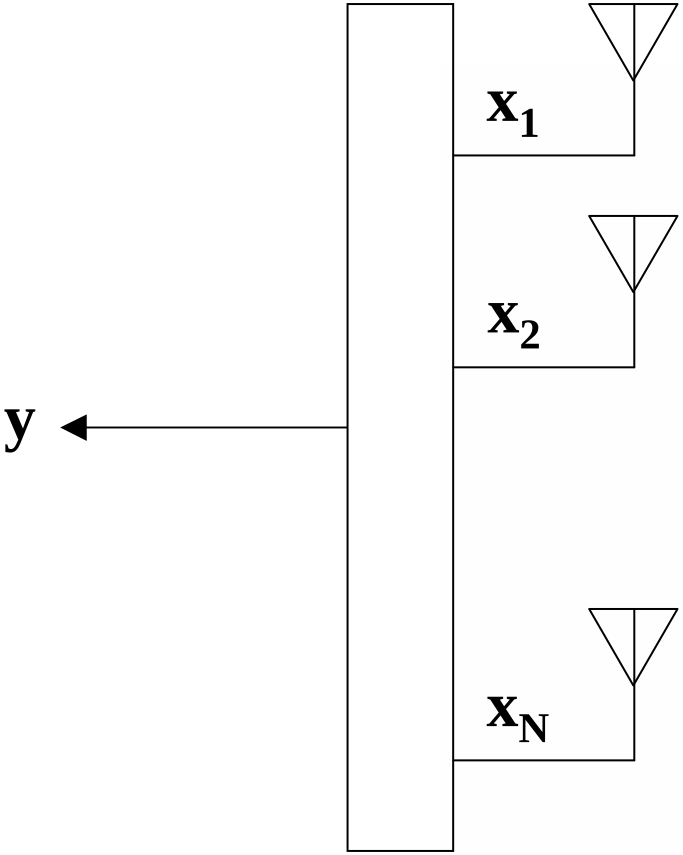
Noise Cancellation

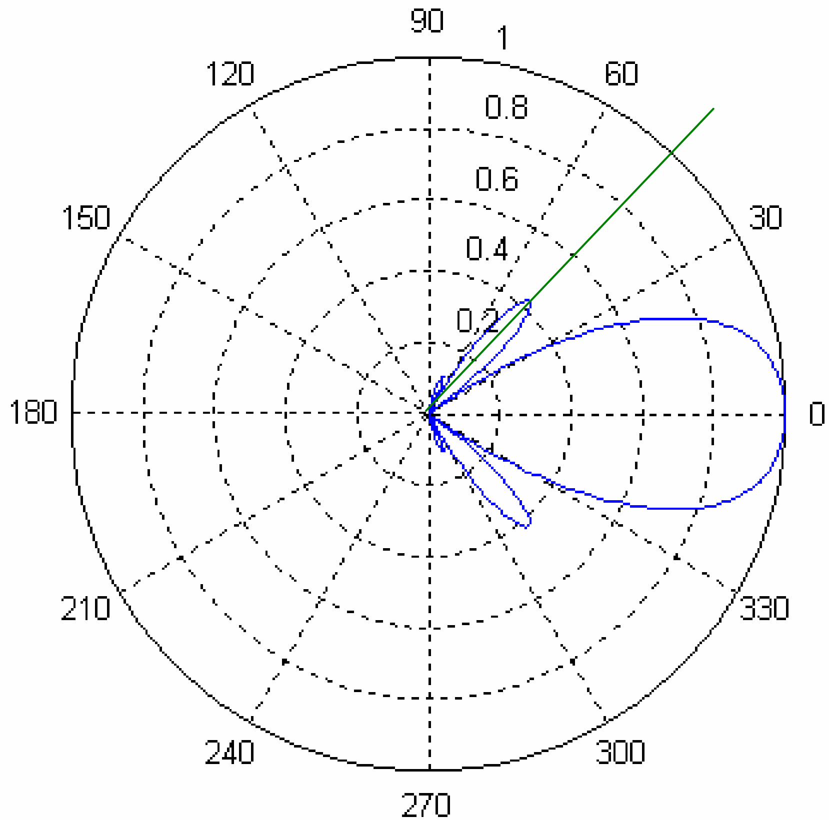


Prediction

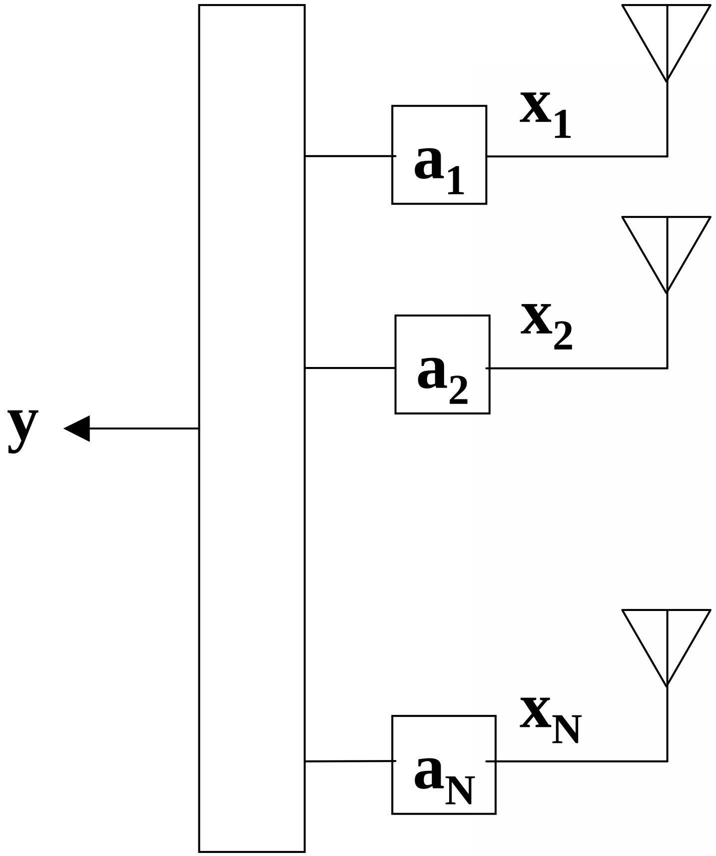


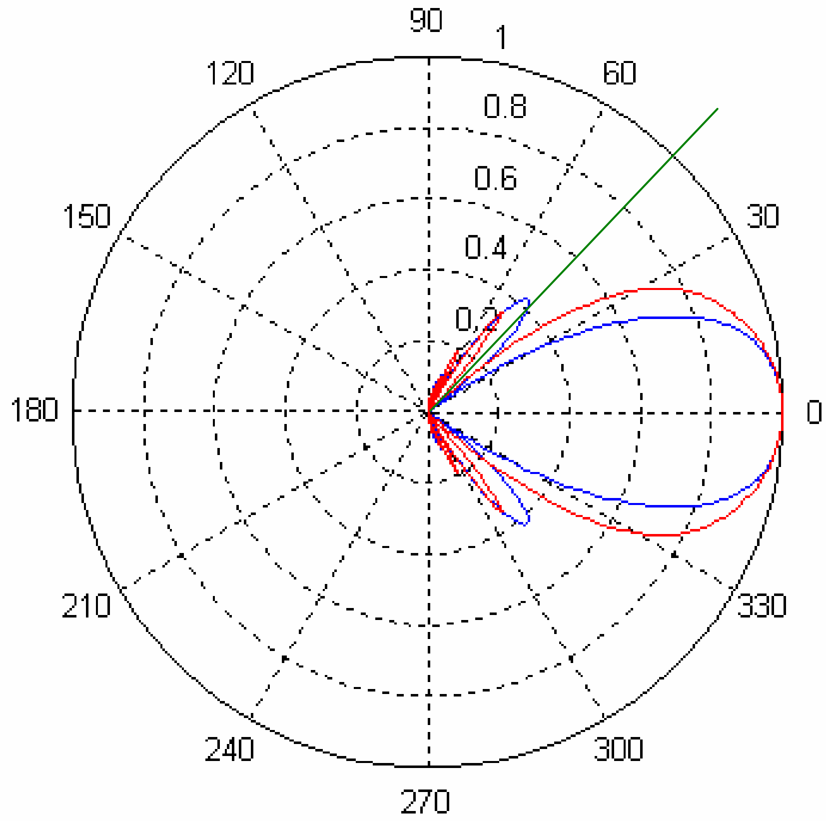
Sidelobe cancellation





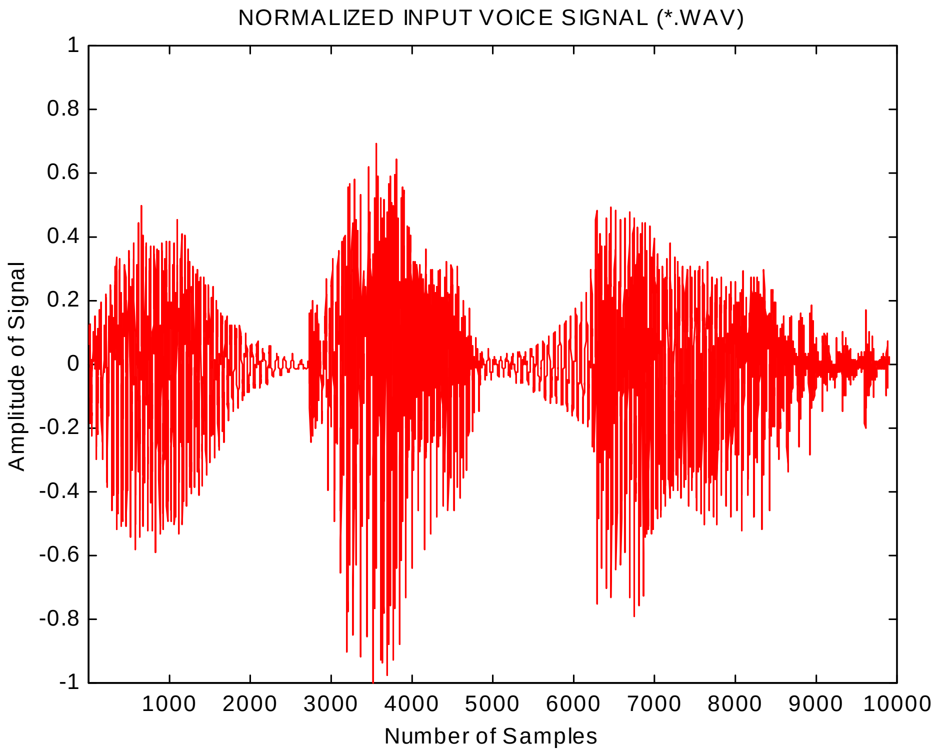
$y(t)=\Sigma\, x_i(t)$





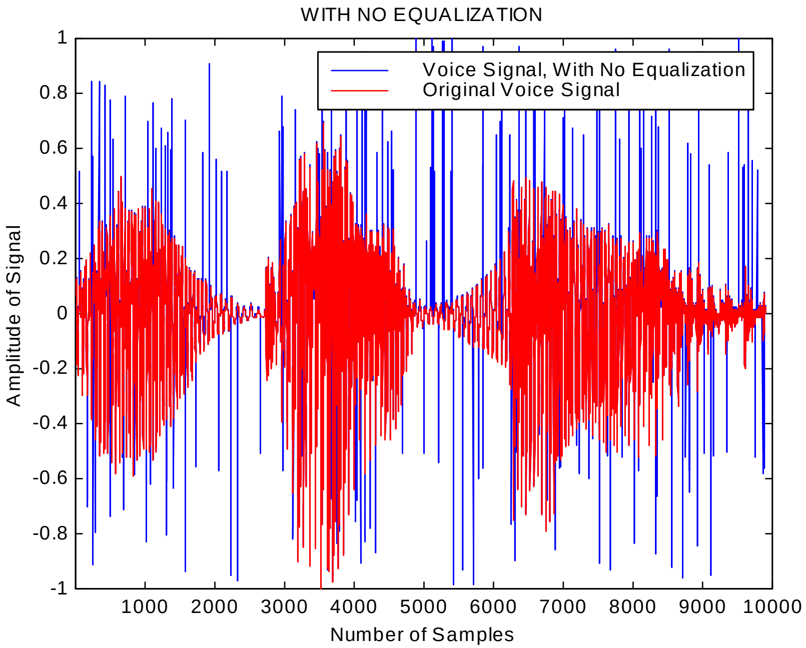
$$y(t)=\Sigma a_i\, x_i(t)$$

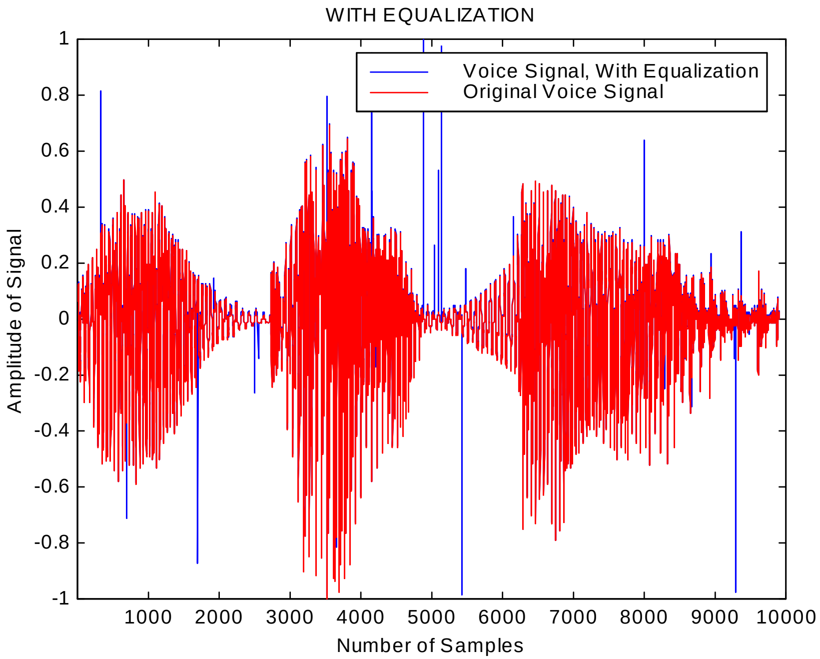
Sound Clip



- Normalized *.wav file (microsoft format)
- 9,946 bytes
- click here

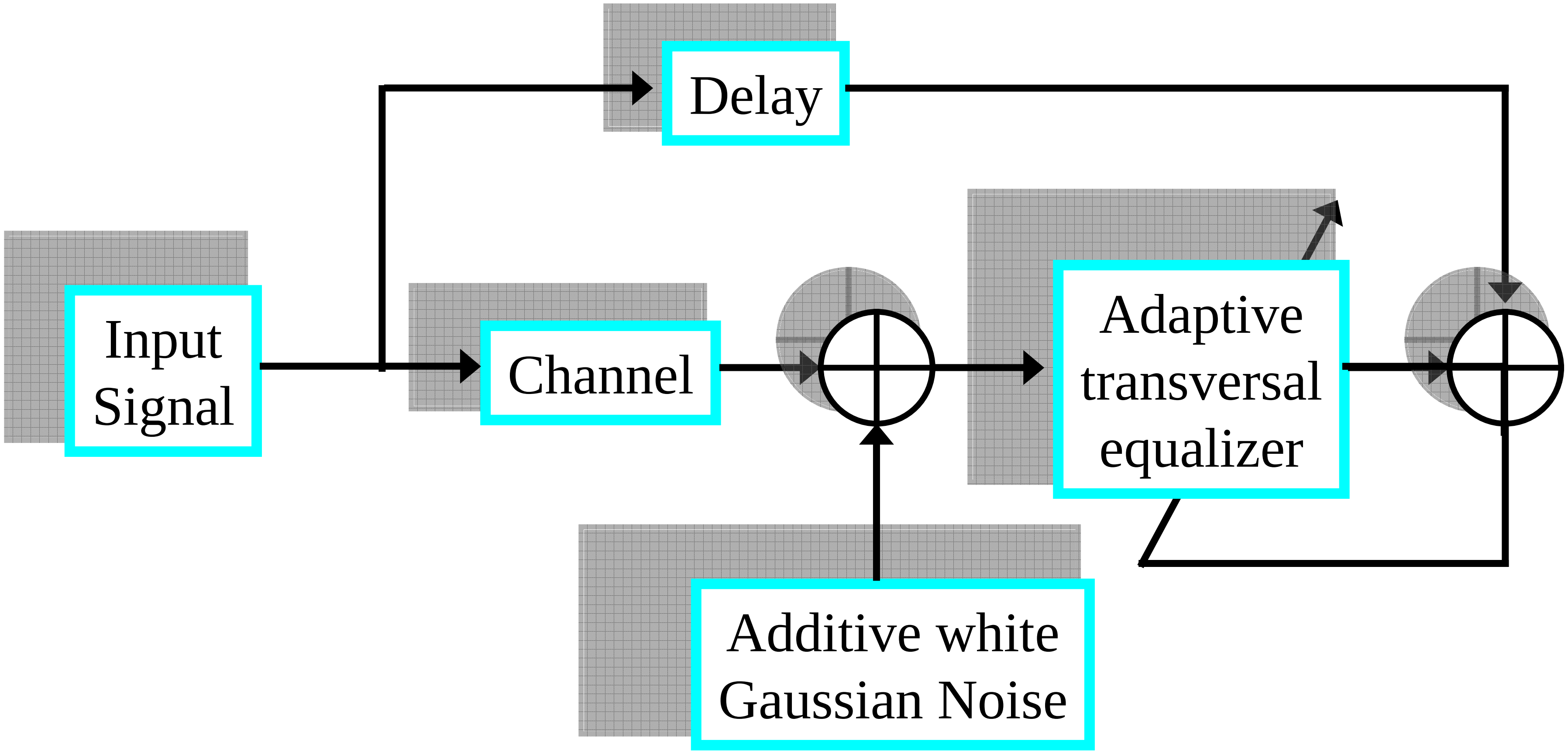
Graphs – w/o and w/ equalization





Simulation under the advisory of Prof. Fontaine (downloaded)

Blind Equalization



Blind equalization approaches

- Stochastic gradient descent approach that minimizes a chosen cost function over all possible choices of the equalizer coefficients in an iterative fashion
- Higher Order Statistics (HOS) method that is using the higher order cumulants spread of the underlying process, and hence to the flatness
- Approaches that exploit statistical cyclostationarity information coefficients toward their optimum value, at a given frequency
- Algorithms that are based on the maximum likelihood criterion. depends on the value of the power spectral density of the

Blind Equalization: HOS

$$C_x^{(1)} = E\{x(t)\} = \int x(t)\, f_X(x)\, dx \quad : \text{First Order Statistics}$$

Second Order Statistics :

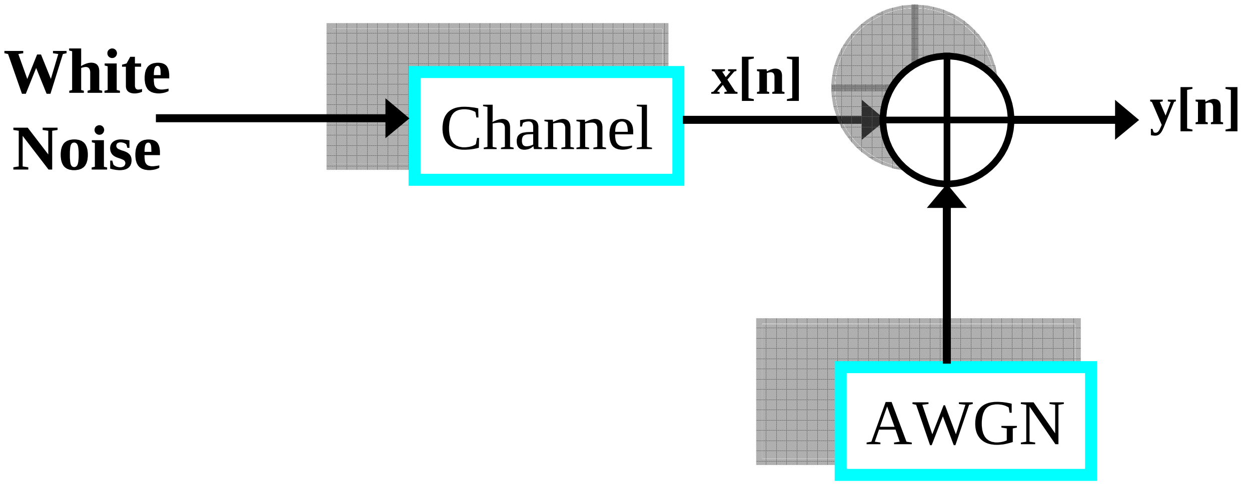
$$C_x^{(2)}(\tau) = E\{x(t)x(t+\tau)\} = \int x(t)x(t+\tau)\, f_X(x(t), x(t+\tau))\, dx$$

Third Order Statistics :

$$C_x^{(3)}(\tau_1, \tau_2) = E\{x(t)x(t+\tau_1)x(t+\tau_2)\}$$

$$= \int x(t)x(t+\tau_1)x(t+\tau_2)\, f_X(x(t), x(t+\tau_1), x(t+\tau_2))\, dx$$

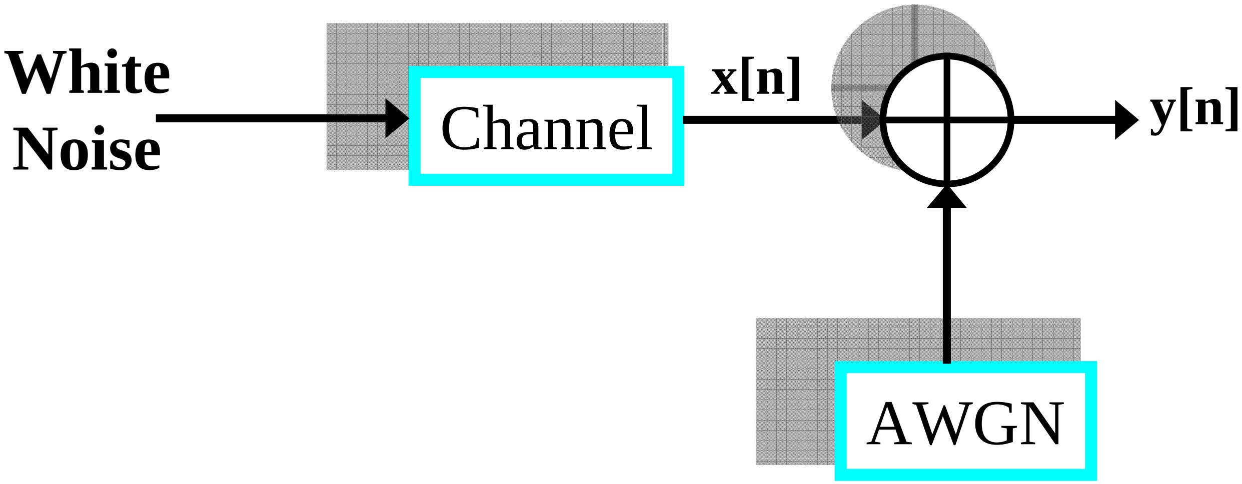
Blind Equalization: HOS



$$C_y^{(2)} \Rightarrow S_y(f) = |H(f)|^2 S_{input}(f) + S_n(f)$$

$$C_y^{(2)} \Rightarrow S_y(f) = |H(f)|^2 K$$

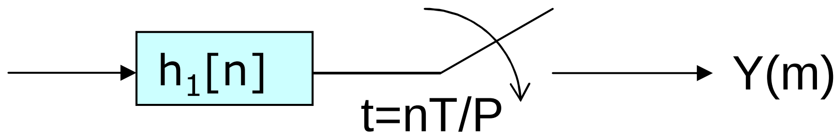
Blind Equalization: HOS



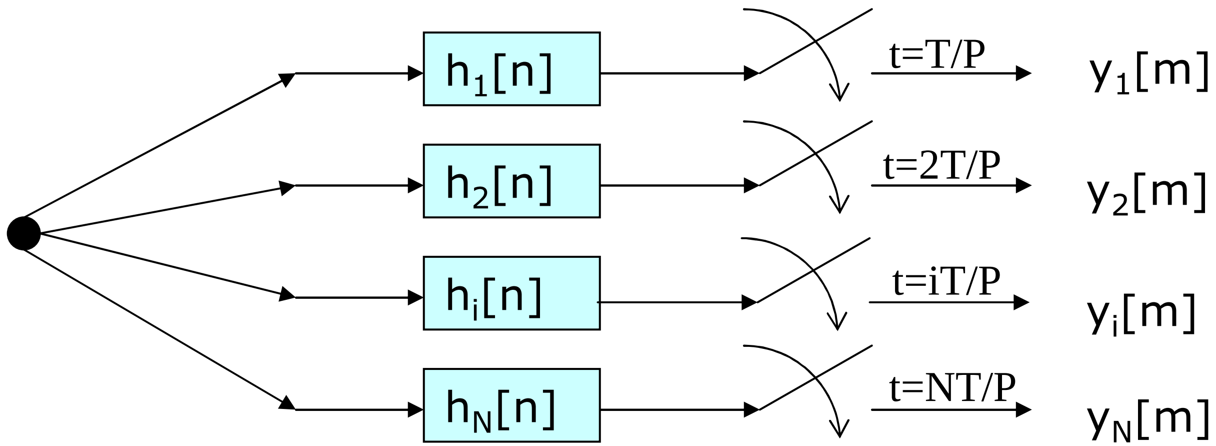
- For Gaussian signals:

$$C_x^{(n)} = 0 \quad \text{for n} > 2 \qquad C_y^{(4)} = C_h^{(4)} + C_n^{(4)}$$

Fractionally Spaced Equalizer



$$y(m) = \sum_l W(l)h(m - lP) + n(m)$$



$$y_i(m) = \sum_l W(l)h_i(m - l) + n_i(m)$$

$$Y_N(n) = HW_{N+L_h}(n) + N_N(n)$$

Fractionally Spaced Equalizer

$$\textit{Fractionally Spaced Sampling} \Rightarrow \textit{Cyclostationary Output}$$

$$Y_N(n) = HW_{N+L_h}(n) + N_N(n)$$

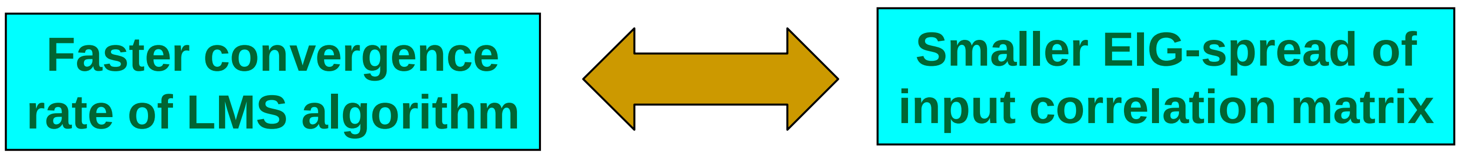
$$H = \begin{bmatrix} h(0) & h(1) & \cdots & h(L_h) & \cdots & 0 \\ 0 & h(0) & \cdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & h(L_h-1) & h(L_h) \end{bmatrix}$$

Formula:

- Channel model $$Y_N(n) = HW_{N+L_h}(n) + V_N(n)$$
- Cost function definition $$J = E\left\{\left|\hat{W} - W(n)\right|^2\right\}$$
- Updating equalizer coefficients $$\hat{g}(T) = \hat{g}(T-1) - 0.5\,\mu\,\nabla\hat{J}(T)$$
- Our proposed Wavelet domain gradient $$\nabla\hat{J}(T) = Y_N(n)Y_N'(n)g_0 - \sigma_W^2 H(:,1)$$

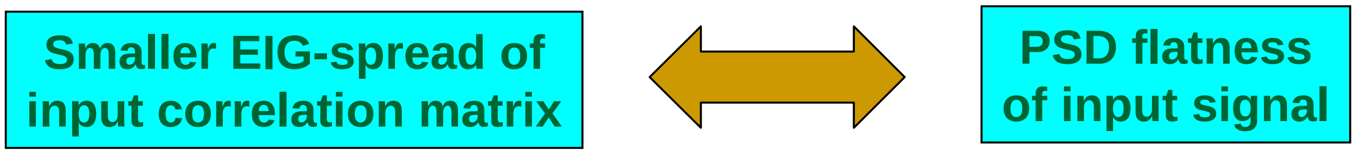
Convergence rate of LMS algorithm

- It is well known that the convergence behavior of conventional LMS algorithm depends on the eigenvalue spread of input process

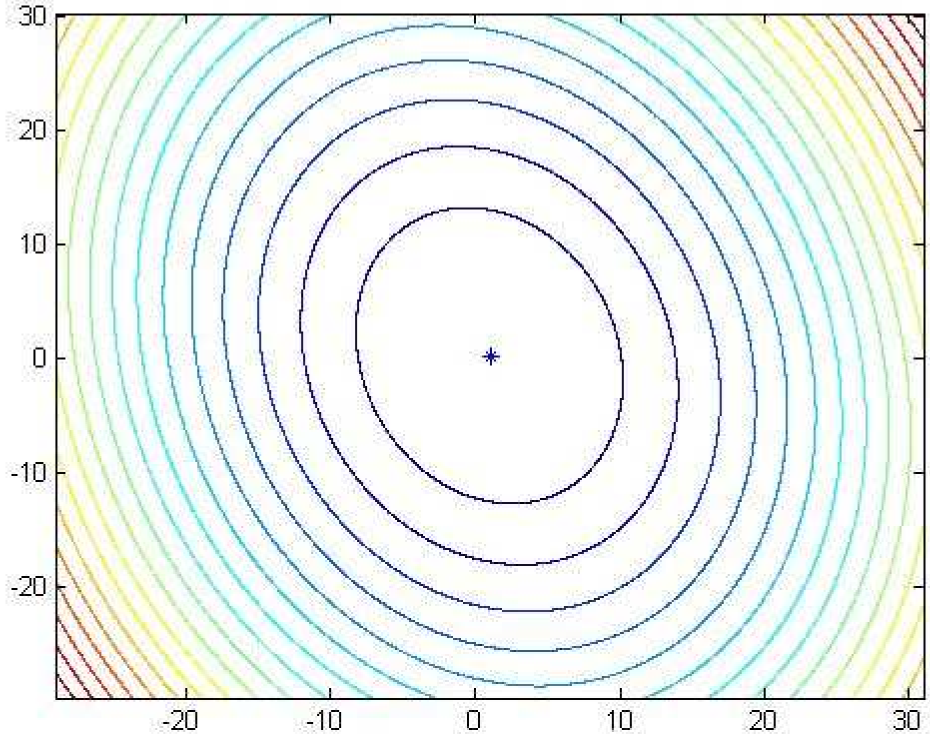


EIG-spread of input correlation matrix (R) vs. flatness of its PSD

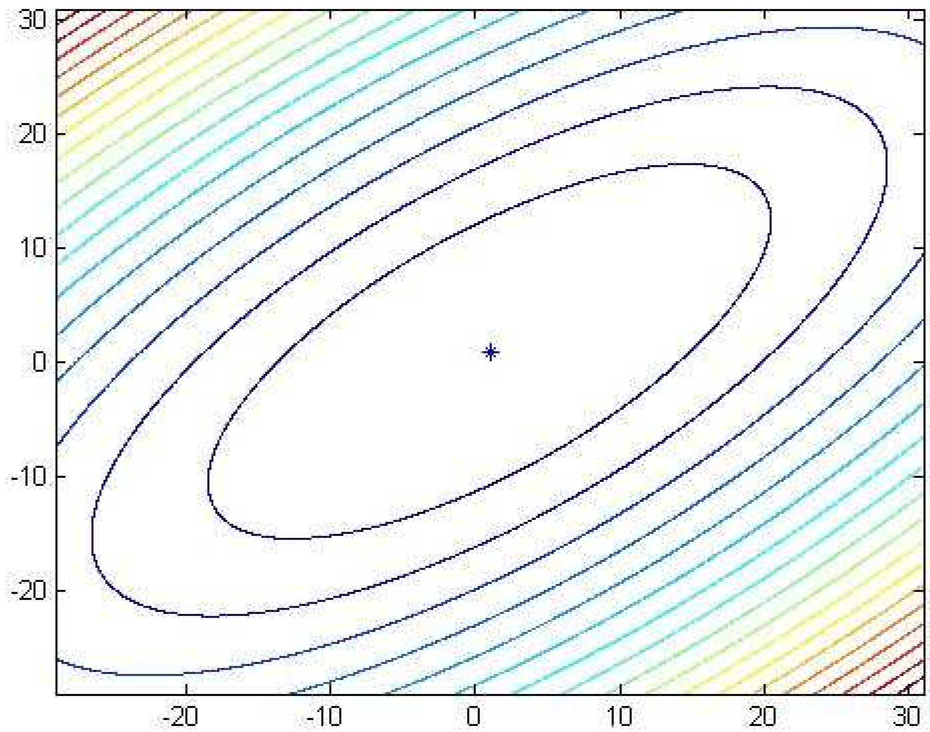
- Convergence rate of filter coefficients toward their optimum value, at a given frequency depends on the value of the power spectral density of the underlying process at that frequency relative to all other frequencies.



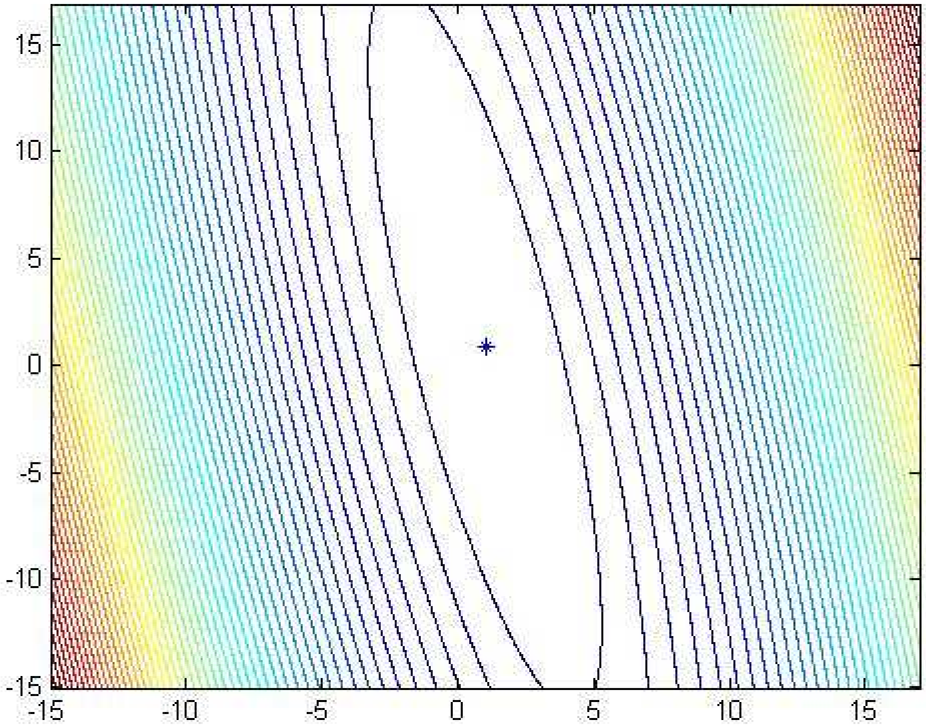
EIG-spread of R vs. shape of error surface: Example 1, EIG = 1.22



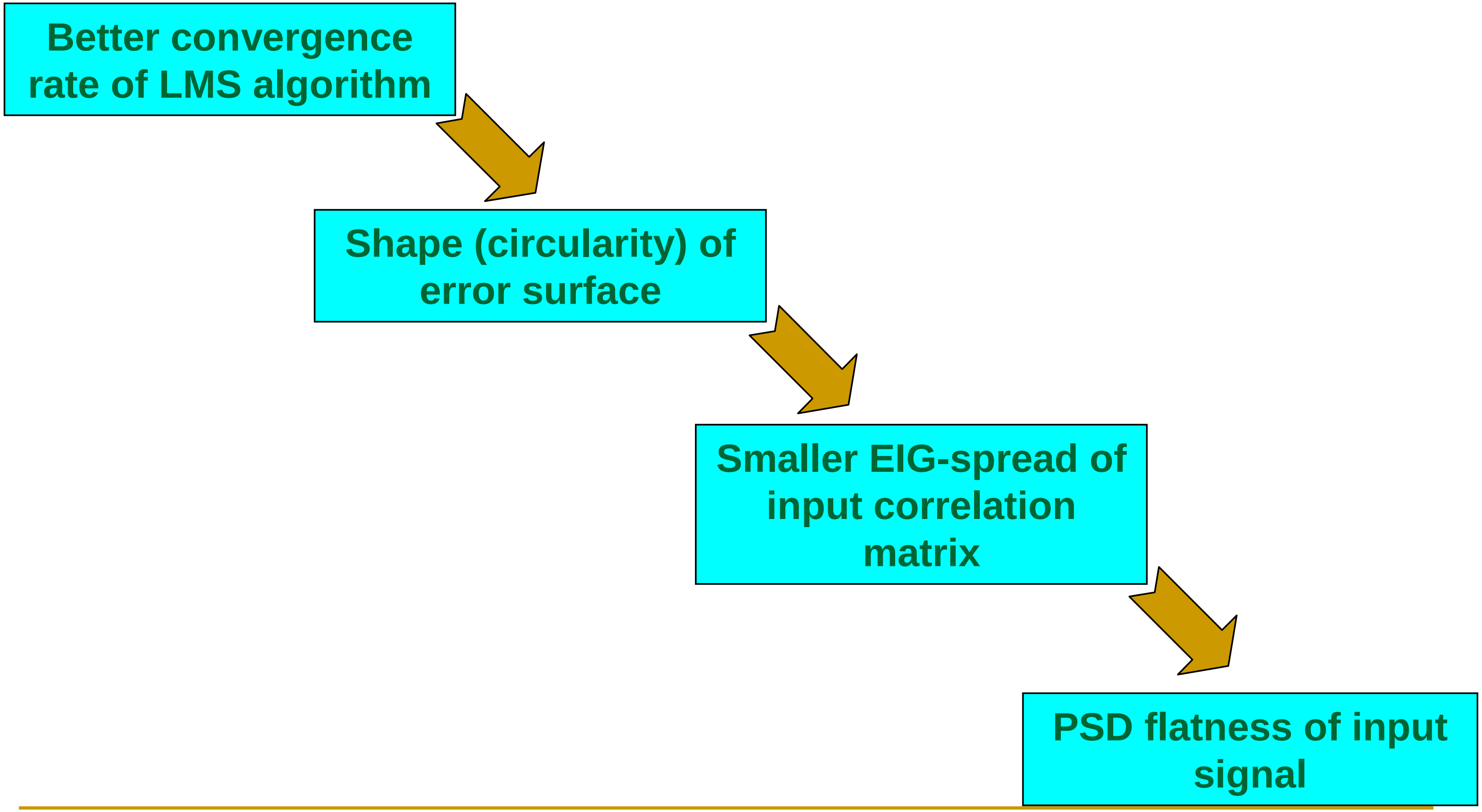
Example 2, EIG = 3



Example 3, EIG = 100



Summary:


Better convergence rate of LMS algorithm
Shape (circularity) of error surface
Smaller EIG-spread of input correlation matrix
PSD flatness of input signal

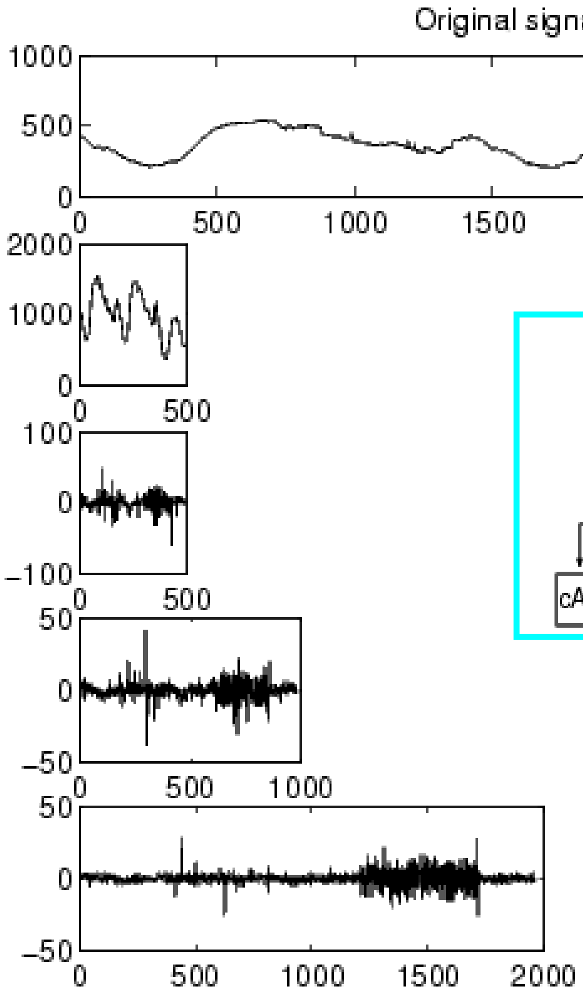
How transforms improve the convergence rate of LMS?

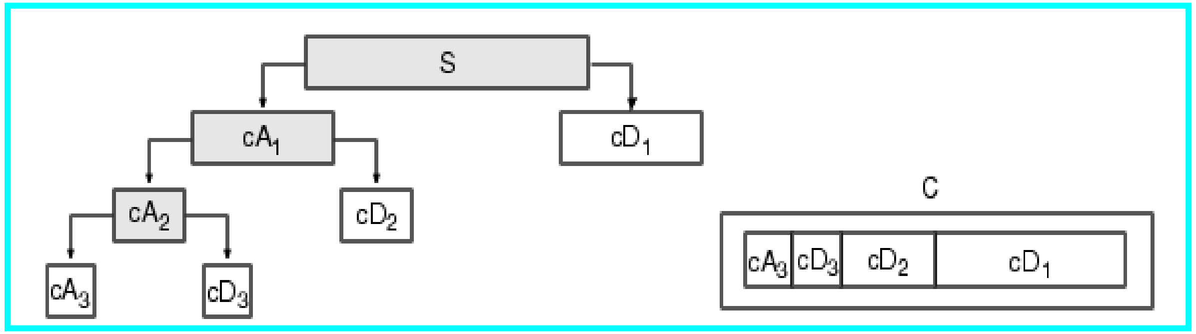
- Band-partitioning property of Wavelet transform
- ➔ Transformed elements are (at least) approximately uncorrelated with one another
- ➔ Correlation matrix is closer to a diagonal matrix
- ➔ An appropriate normalization can convert the result to a normalized matrix whose EIG spread will be much smaller

Advantages of using Wavelet transform

- Efficient transform algorithms exist (e.g. the Mallat algorithm)
- Transforms can be implemented as filter banks with FIR filters
- Strong mathematical foundations allow the possibility of custom designing the wavelets e.g. the lifting scheme

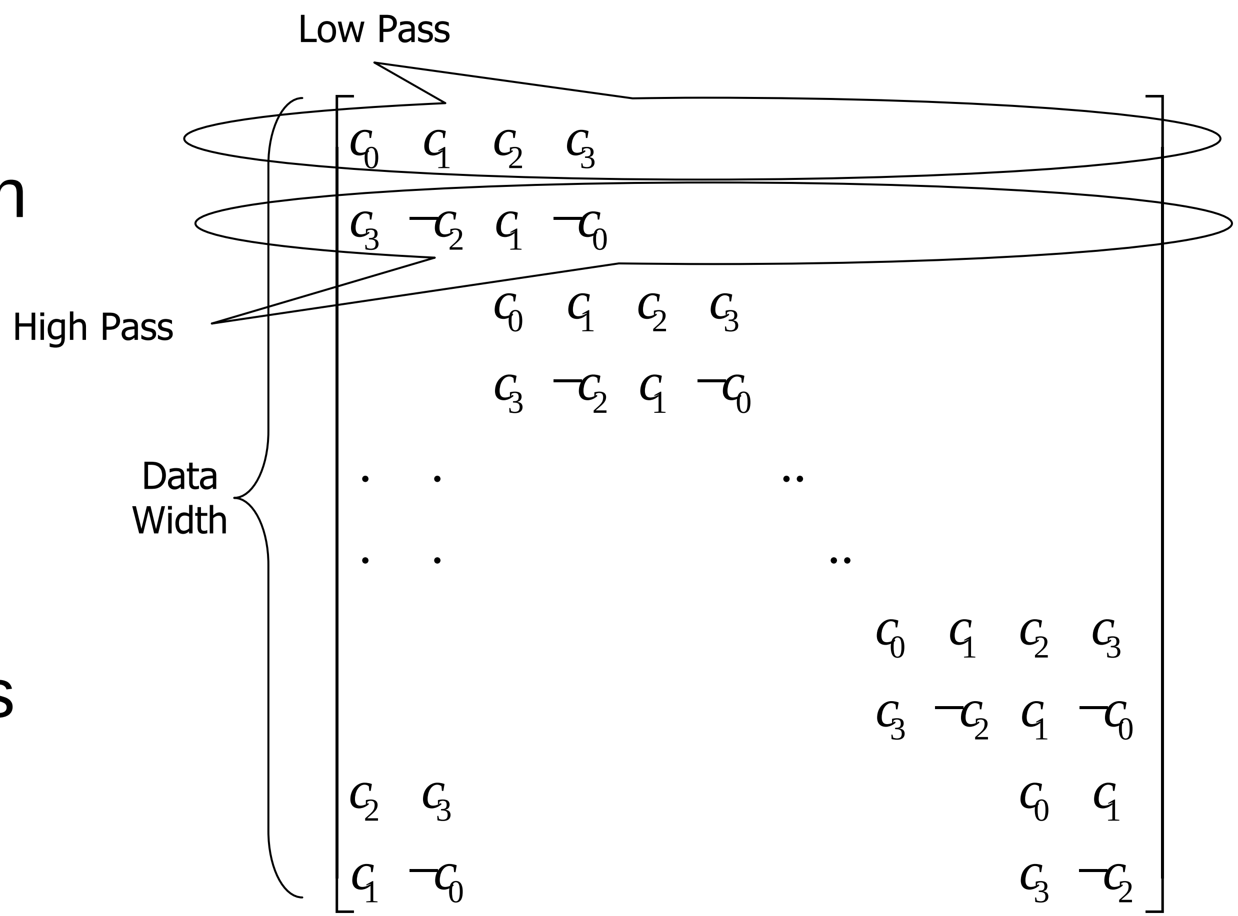
Wavelet transform algorithm:





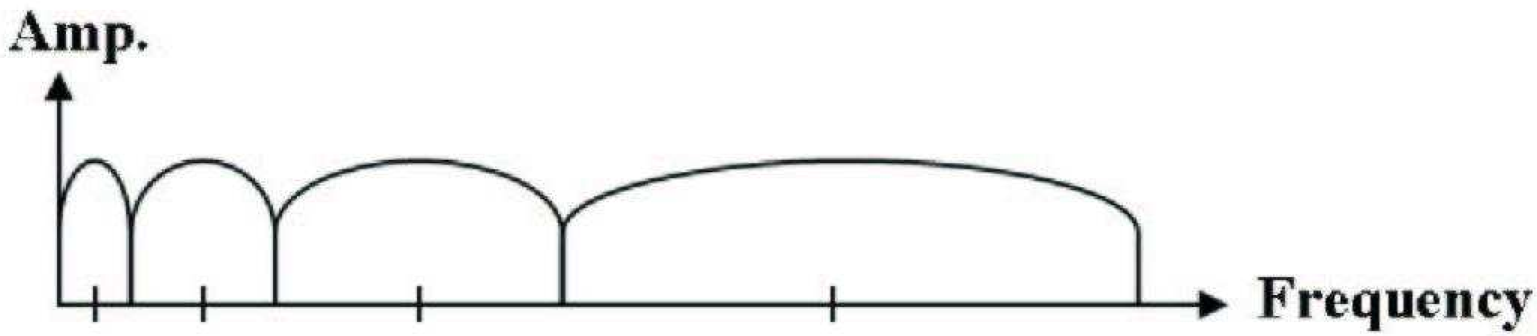
Matrix form implementation of Wavelet transform

- As mentioned in the previous slide, Wavelet transform consists of two low-pass and high-pass filters

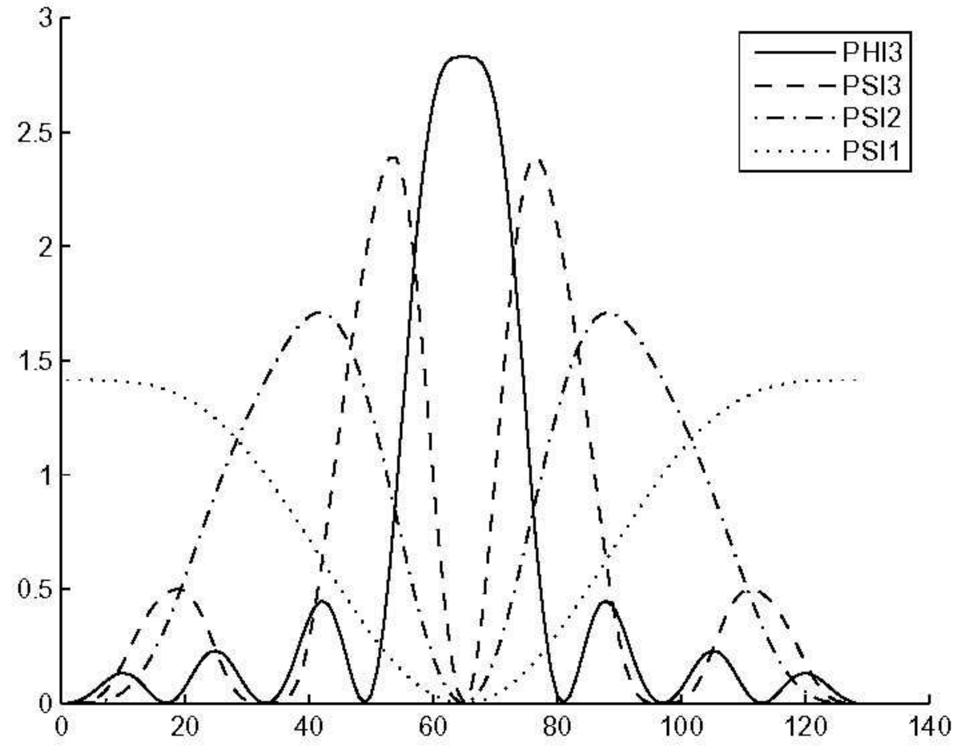


Why Wavelet transform?

- wavelet analysis filters are "constant-Q" filters; i.e., the ratio of the bandwidth to the center frequency of the band is constant



band-partitioning property of Daubechies filters



After transformation, each coefficient shows the amount of energy passed from one of above filters

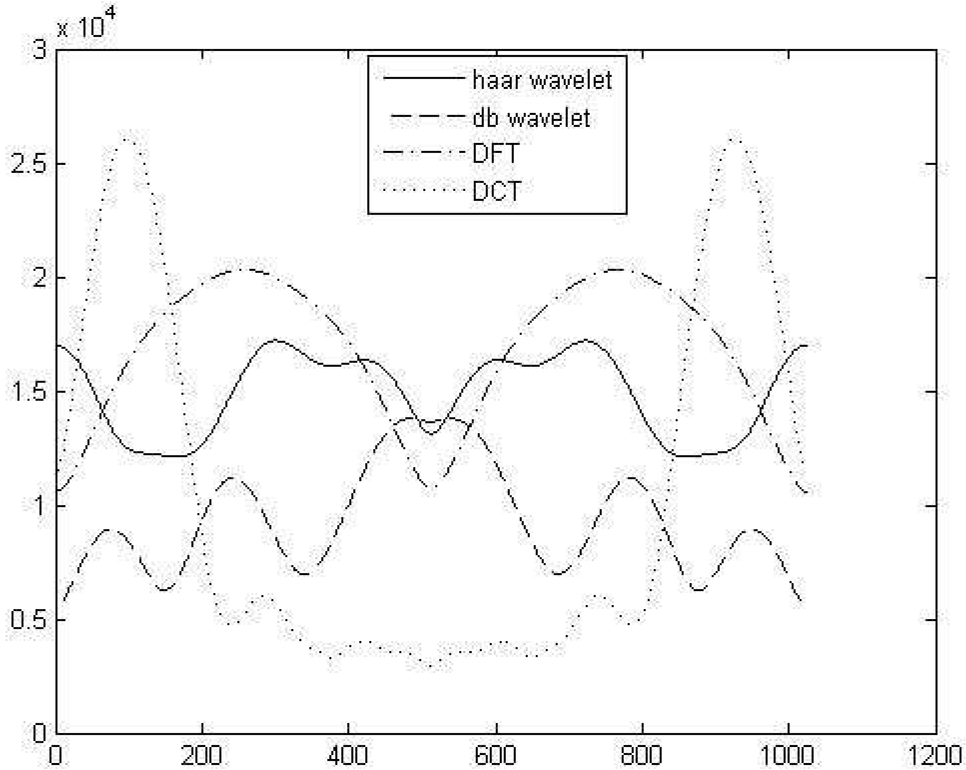
The bandwidth of the filters in low frequencies is narrow compared to the bandwidth of the filters in higher frequencies

Most communication signals have a low-pass nature

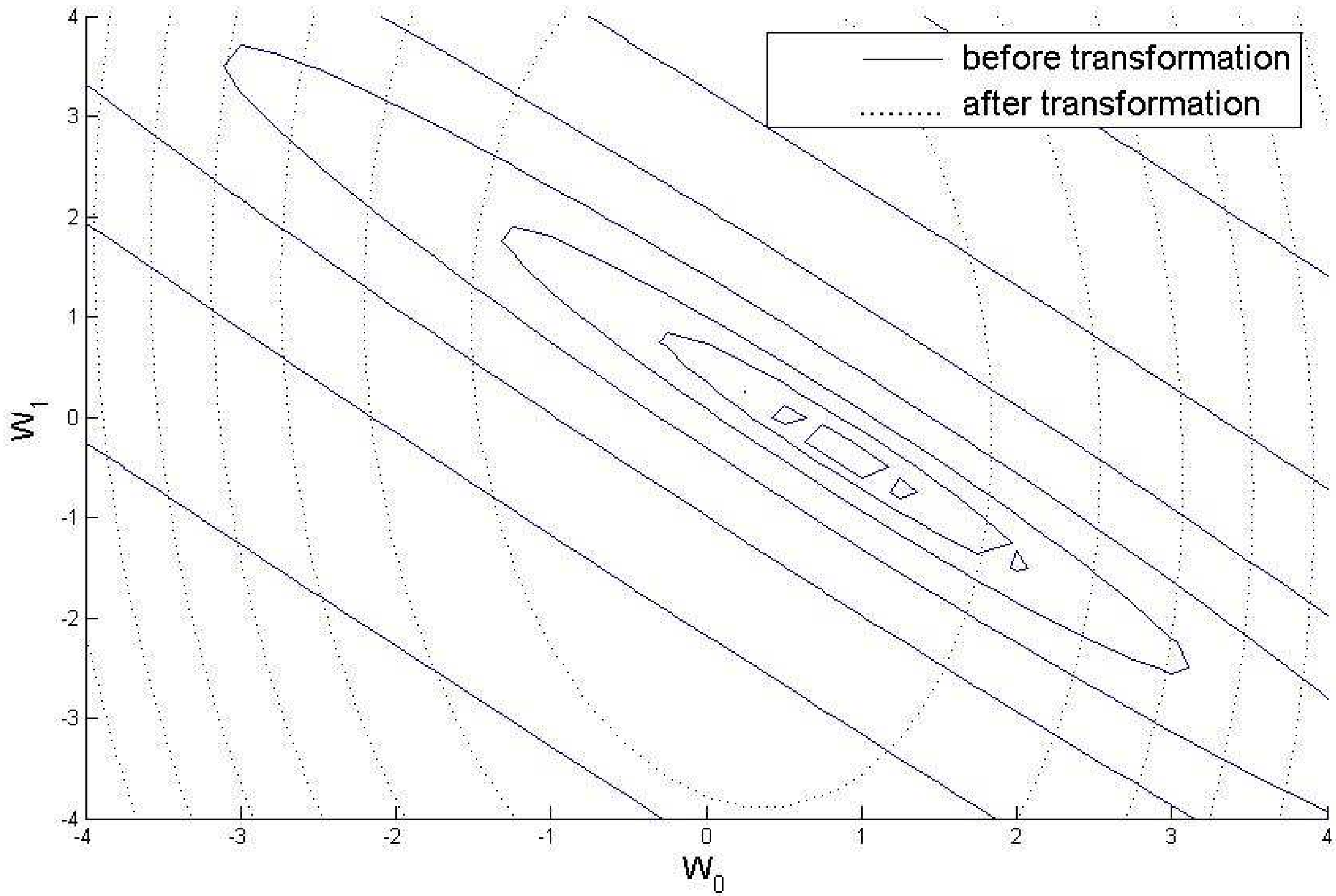
It’s more probable that the output of filters contain the same amount of energy

more likely to obtain a flat spectrum.

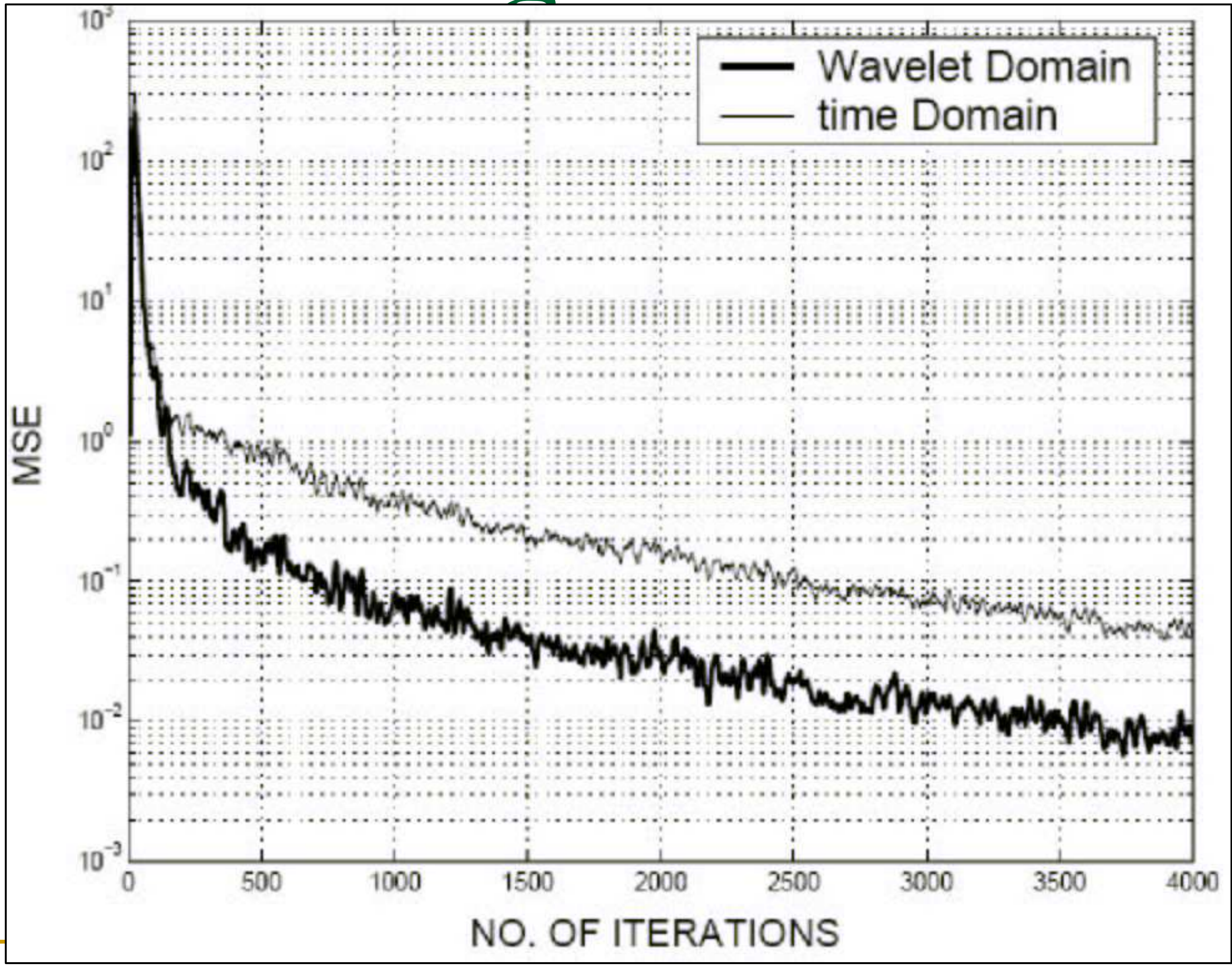
PSD of a typical communication signal after different transforms



Effect of Wavelet transform on error surface



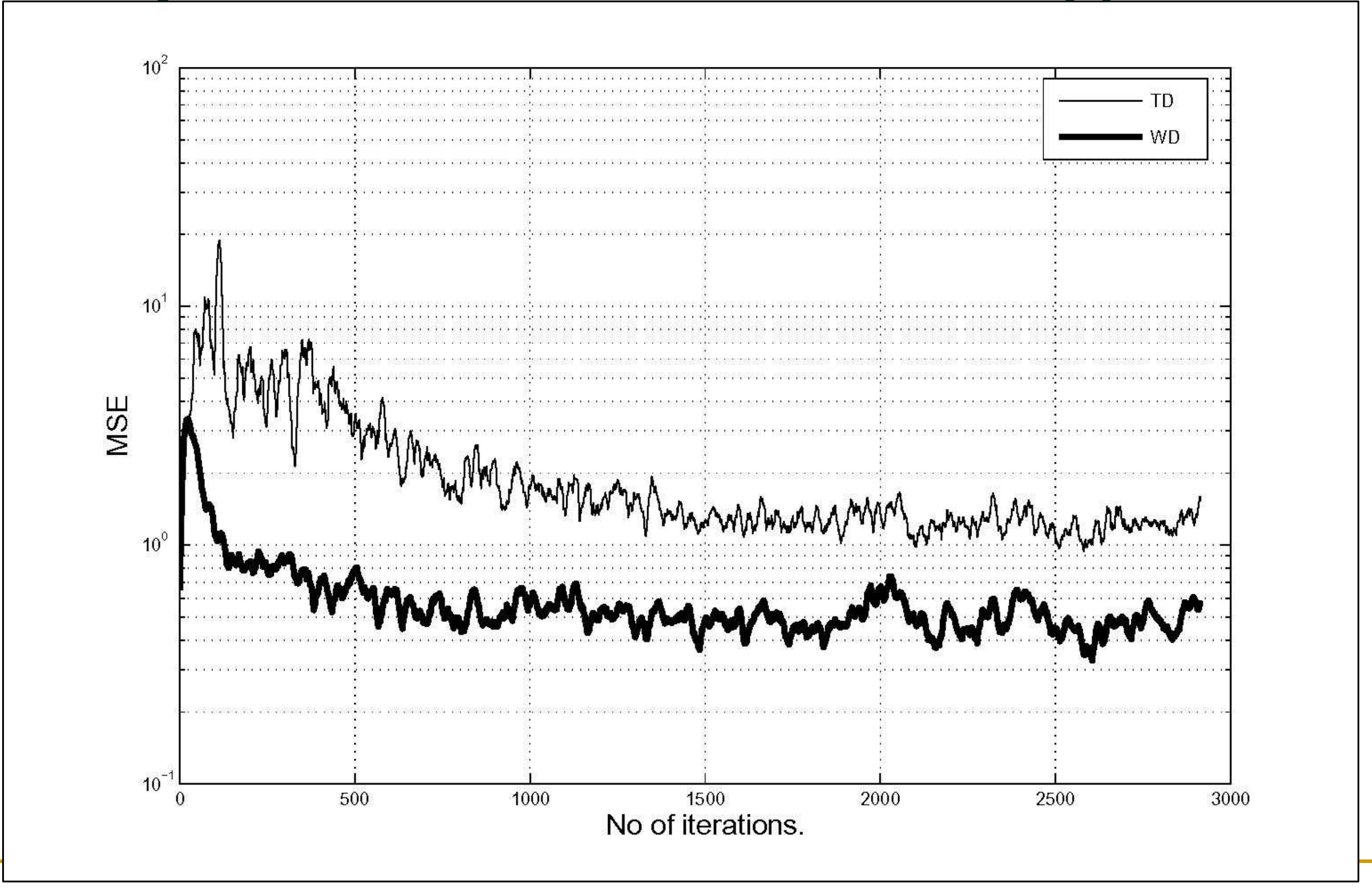
MSE of a TD-Godard algorithm vs. WD-Godard algorithm



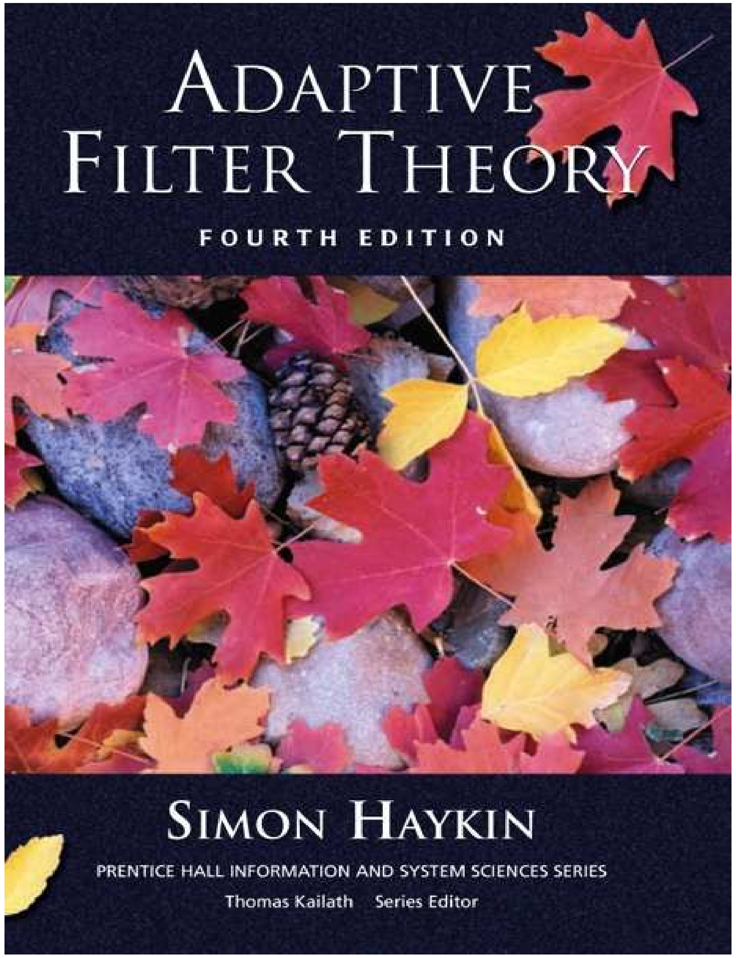
Formula:

- Channel model $$Y_N(n) = HW_{N+L_h}(n) + V_N(n)$$
- Cost function definition $$J = E\left\{\left|\hat{W} - W(n)\right|^2\right\}$$
- Updating equalizer coefficients $$\hat{g}(T) = \hat{g}(T-1) - 0.5\,\mu\,\nabla\hat{J}(T)$$
- Our proposed Wavelet domain gradient $$\nabla\hat{J}(T) = \mathcal{T}Y_N(n)Y'_N(n)\mathcal{T}'g_0 - \sigma_w^2\mathcal{T}H(:,1)$$

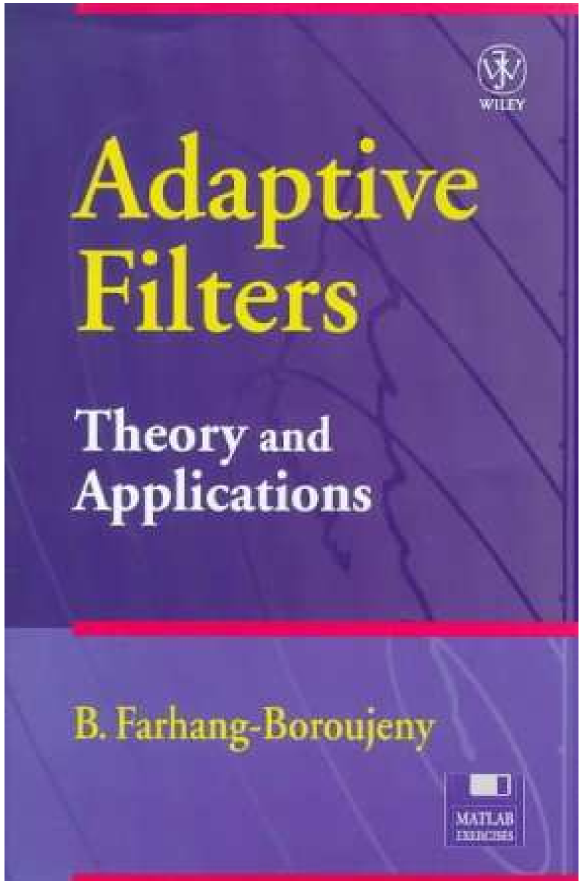
MSE of a TD-FSE algorithm compared with WDFSE algorithm



References



- **Adaptive Filter Theory**
- **Simon Haykin**
- **Prentice Hall**



- **Adaptive Filters**
  Theory and Applications
- **B. Farhang-Boroujeny**
- **wiley**