Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

Consortium leader

PETER PAZMANY CATHOLIC UNIVERSITY

Consortium members

SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Digital- and Neural Based Signal Processing & Kiloprocessor Arrays

Digitális- neurális-, és kiloprocesszoros architektúrákon alapuló jelfeldolgozás

Adaptive Signal Processing

(Adaptív Jelfeldolgozás)

János Levendovszky, András Oláh, Dávid Tisza, Gergely Treplán

Outline

- Introduction to adaptive signal processing
- Motivation and historical review
- Applications
- Wiener-filtering
- The Levinson-Durbin algorithm
- The Robinson-Monroe stochastic approximation
- The LMS algorithm
- Adaptive-predictive coding
- Radio channel equalization

Introduction

Digital signal processing is becoming more and more important in various kinds of fields.

Many types of signals, which were processed formerly by analog techniques, are now usually being processed using VLSI processors such as digital signal processors.

Digital television and digital mobile communications are becoming very popular owing to the development of digital techniques.

Historical overview (1)

- Theory of linear approximation (Galilei-1632, Gauss-1795)
- Approximation with minimum mean square error (Wiener-1930, Kolgomorov-1939)
- Levinson-Durbin algorithm (1947)
- Wiener filter (Norbert Wiener, 1949)
- LMS algorithm (Hoff-1960, Widrow-1970)
- Digital filter (1960 – 1965, J. F. Kaiser 1965)
- Kalman filter (Kalman-1960)
- Minimax criteria (Zames-1981)

Historical overview (2)

- First linear digital filter for solving difference equations in fifties
- Digital filter (1960 – 1965, J. F. Kaiser 1965), at first, it was called “numerical filter” or “sampled-data filter”.
- Tapped delay line for equalization in the digital communication technologies in the seventies
- Filters and adaptive architectures implemented on DSP in the eighties
- Array signal processing in the nineties

Practical applications of adaptive signal processing (1)

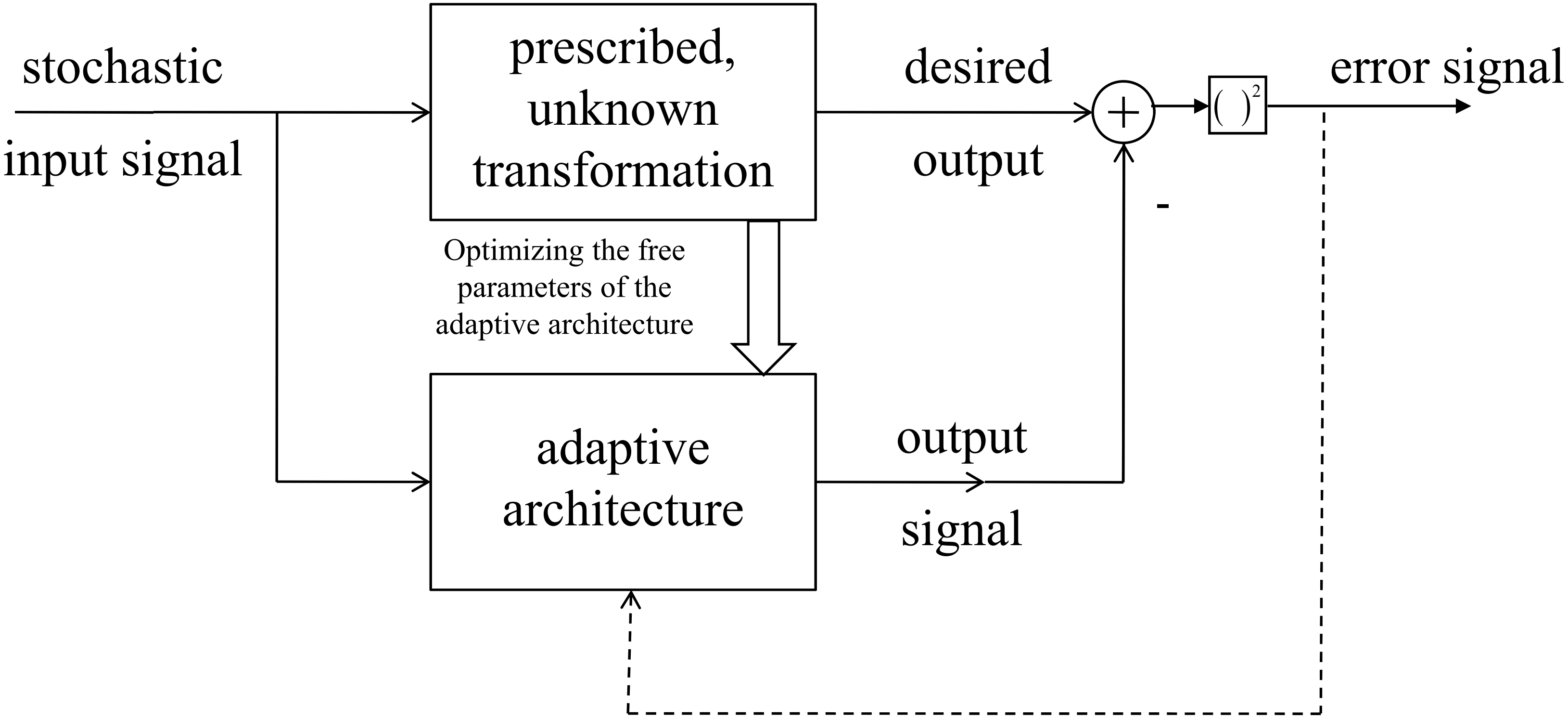
- **Communication technology**: equalization, adaptive modulation, etc.
- **Information and computer technology**: encoding, decoding, error correction, data compression, etc.
- **Multimedia technology**: still and moving image processing, image compression, data transmission, human interface, etc.
- **Mechanical engineering**: vibration control/analysis, noise control/analysis, mechanical systems control/analysis, etc.
- **Control engineering:** control of many kinds of dynamical systems

Practical applications of adaptive signal processing (2)

- **Systems engineering**: modeling and optimization of practical systems
- **Image technology:** image processing, pattern recognition, medical imaging, remote sensing
- **Architectural engineering**: architectural acoustics, vibration control (for earthquakes), noise control, etc.
- **Civil engineering:** underwater estimation, flood prediction, fluid control, environmental, data processing, bridge engineering, soil engineering, etc.

Motivation

Based on observed examples (inputs and desired outputs) learn the desired signal transformation


stochastic
input signal
prescribed, unknown transformation
desired
output
+
-
( )2
error signal
Optimizing the free parameters of the adaptive architecture
adaptive architecture
output
signal

Notations and definitions (1)

- Input signal: x_k weakly stationary process
- Expected value: (zero mean process) $E\left[x_k\right]=0$
- Correlation function:

$$R(l)=E\left[\left(x_k-E\left[x_k\right]\right)\left(x_{k-l}-E\left[x_{k-l}\right]\right)\right]=E\left[x_k x_{k-l}\right]$$

- Correlation matrix: $R(i-j)=E\left[x_{k-i}x_{k-j}\right]\Rightarrow$

$$\Rightarrow \mathbf{R}: R_{ij}=R(i-j),\ i,j=0,\dots,J$$

Notations and definitions (2)

- Output signal: d_k weakly stationary process
- Expected value:

$$E\left[d_k\right] = M = 0$$

- Correlation function:

$$V\left(l\right) = E\left[d_k d_{k-l}\right]$$

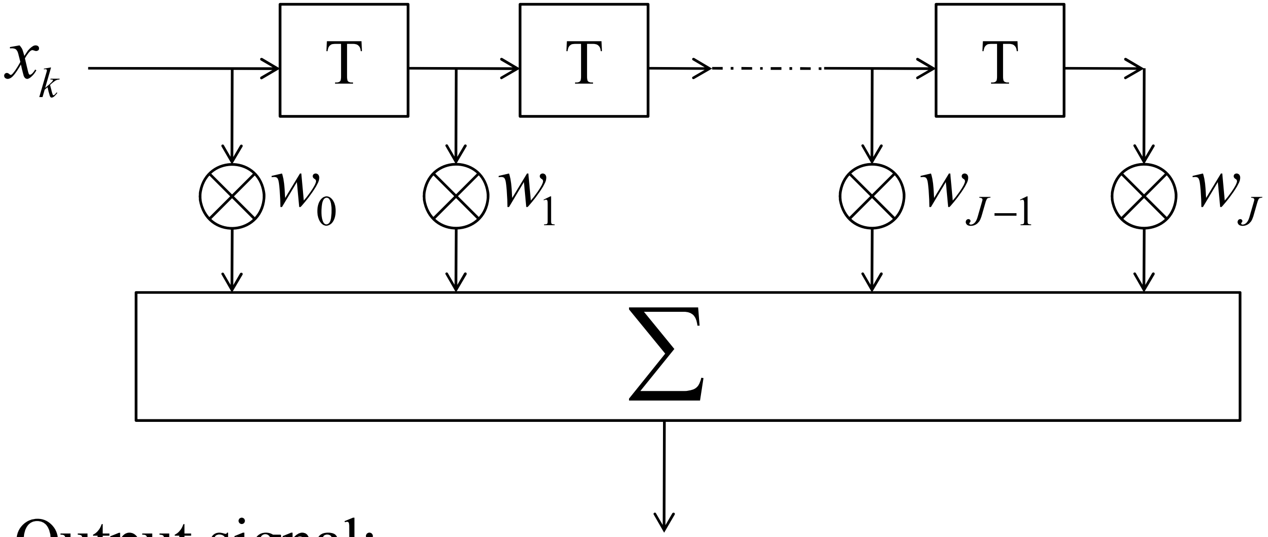
- Cross correlation:

$$r\left(l\right) = E\left[d_k x_{k-l}\right] \Rightarrow$$

$$\Rightarrow \mathbf{r} : r_l = R\left(l\right),\ l = 0, \ldots, J$$

Notations and definitions (3)

- Adaptive linear architecture (FIR):



Adaptation: changing **w**

- Output signal:

$$y_k = \sum_{j=0}^{J} w_j x_{k-j}$$

Notations and definitions (4)

- Error signal: $e_k = y_k - d_k$

- Error function: $\mathrm{E}\left[\left(d_k - y_k\right)^2\right] = J\left(\mathbf{w}\right)$

- Empirical error: $\frac{1}{K}\sum_{k=1}^{K}\left(d_k - y_k\right)^2 = J_{emp}\left(\mathbf{w}\right)$

- Offline-algorithm: $\mathbf{w}_{\mathrm{opt}} := \min_{\mathbf{w}} J_{\mathrm{emp}}\left(\mathbf{w}\right)$

Notations and definitions (5)

- Online-algorithm (recursive solution):

$$\mathbf{w}(k+1)=\Psi\left(\mathbf{w}(k),d_k,y_k\right)$$

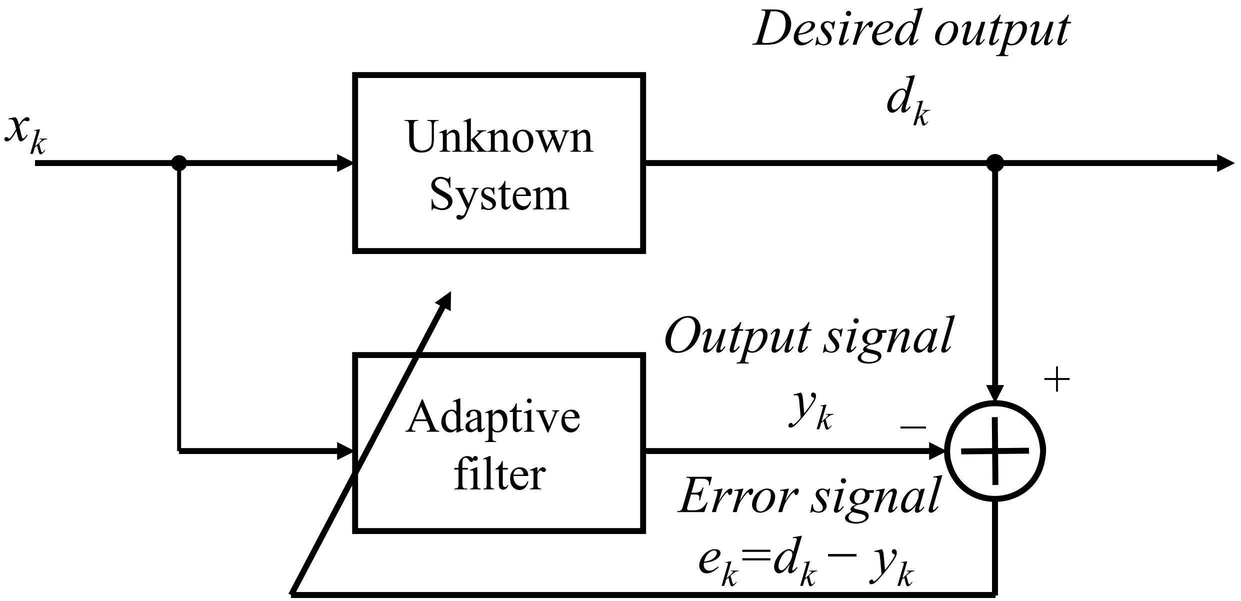
- Objective (convergence):

$$\lim_{k\to\infty}\mathbf{w}(k)=\mathbf{w}_{\text{opt}}$$

Main application classes

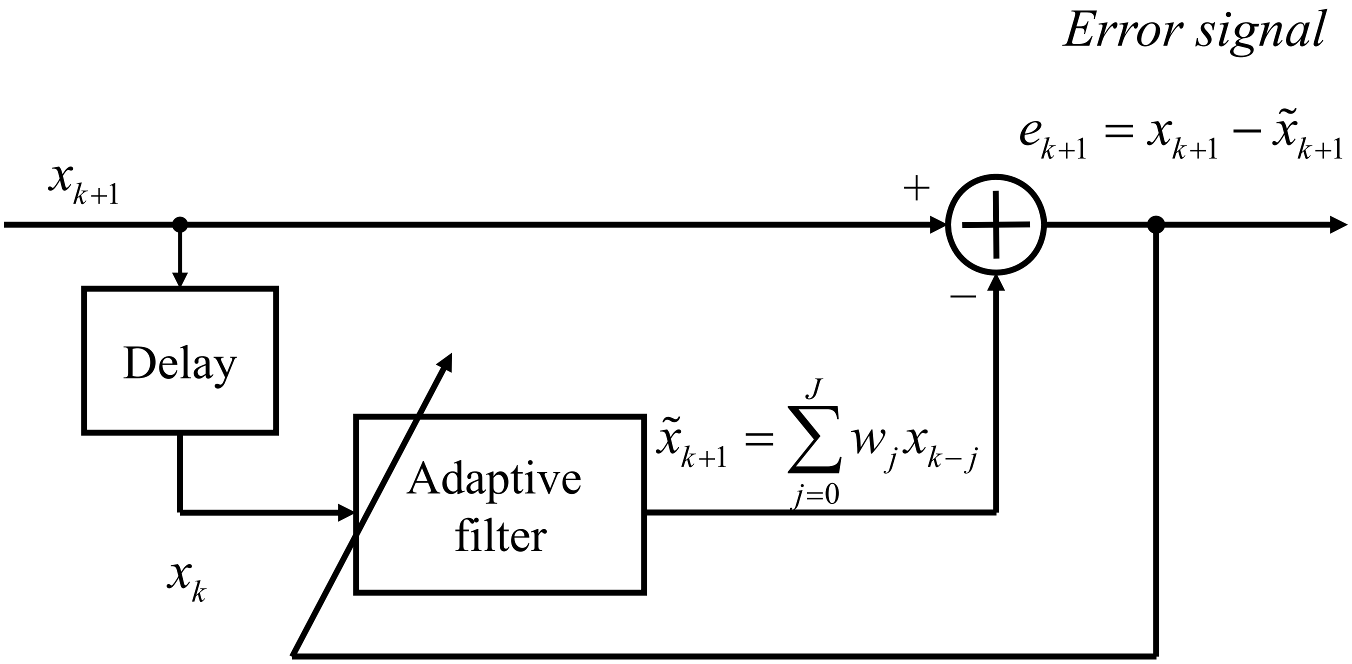
- System identification and modeling
  - (E.g.: modeling of unknown channel distortion)
- Prediction
  - (E.g.: **adaptive-predictive coding** in speech communication)
  - Linear time series prediction (E.g. financial time series)
- Inverse identification
  - (E.g.: **equalization of communication channels**)
- Noise cancelation

System identification and modeling

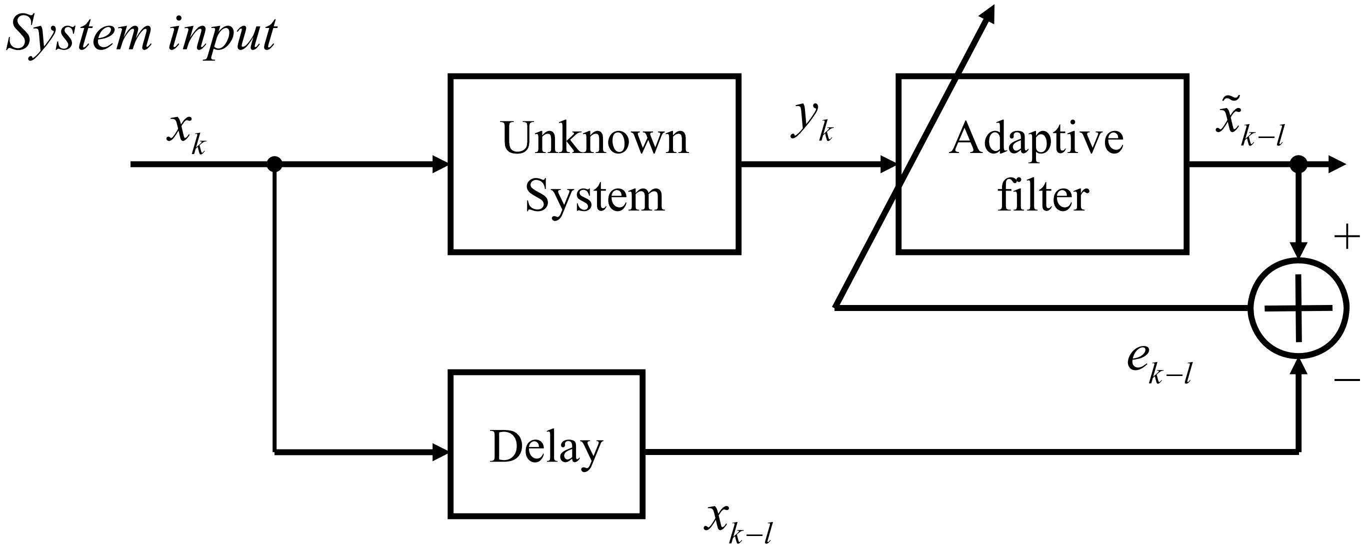


For more details see the results of BAUER, P. – SPAGNUOLO, G. – BOKOR, J (2007).

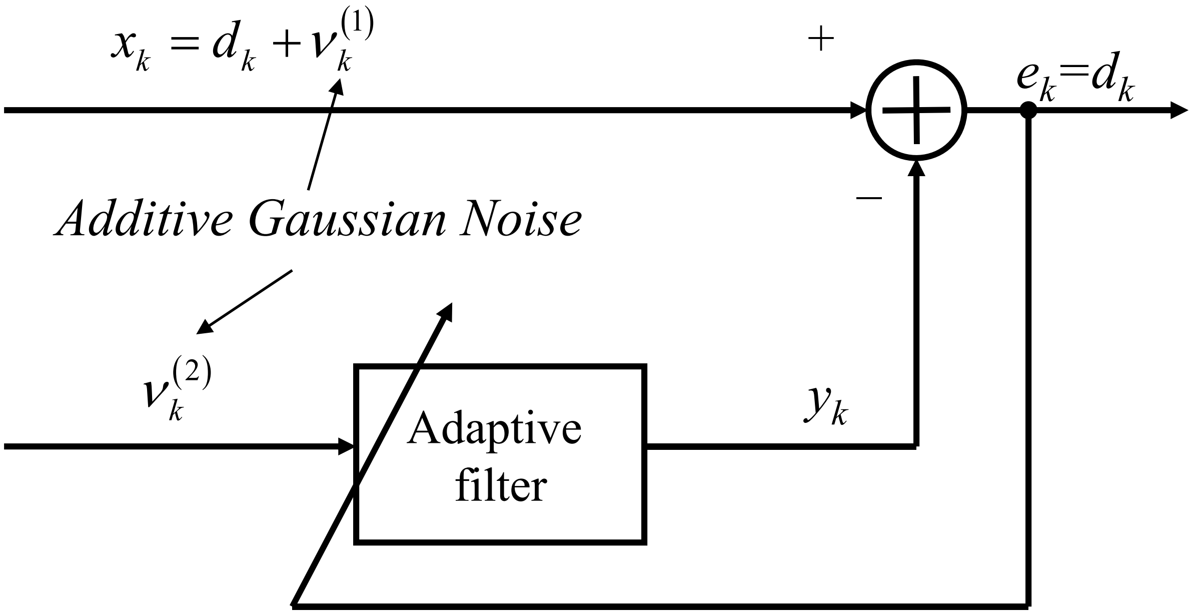
Linear time series prediction


Error signal
$e_{k+1} = x_{k+1} - \tilde{x}_{k+1}$
x_{k+1}
+
−
Delay
Adaptive filter
$\tilde{x}_{k+1} = \sum_{j=0}^{J} w_j x_{k-j}$
x_k

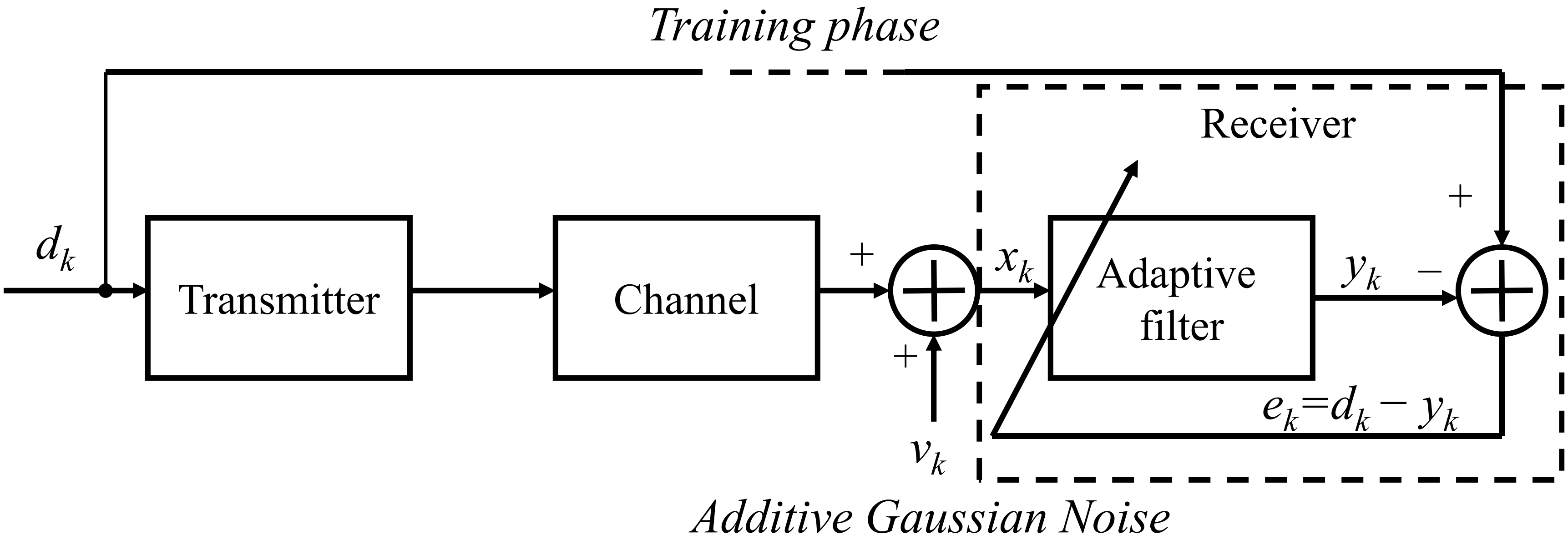
Inverse identification



Noise cancelation with a reference signal

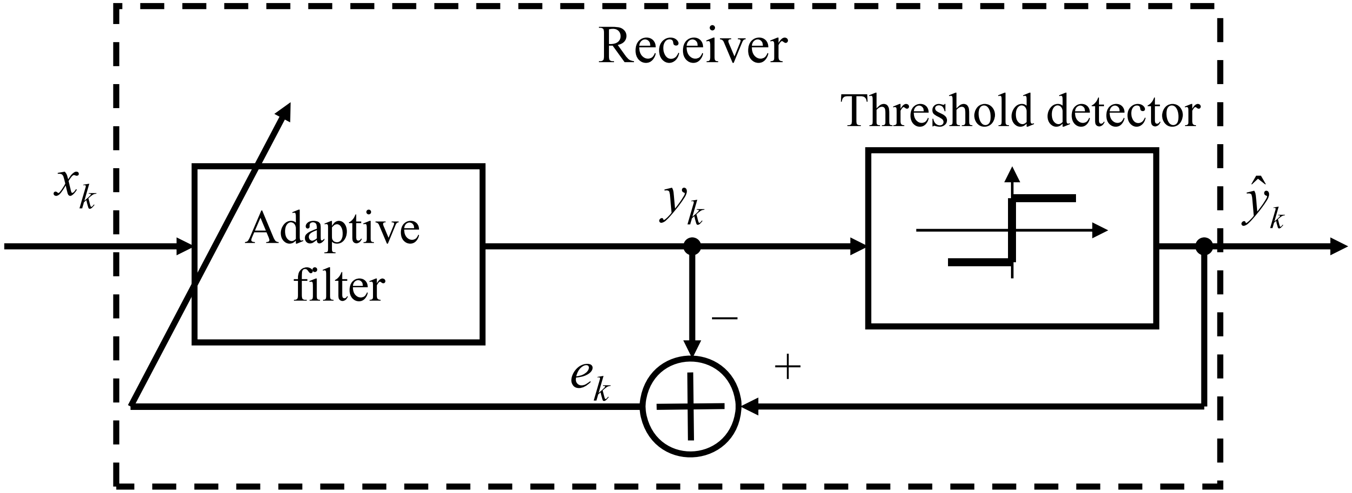


Some other adaptive filter configurations (1)



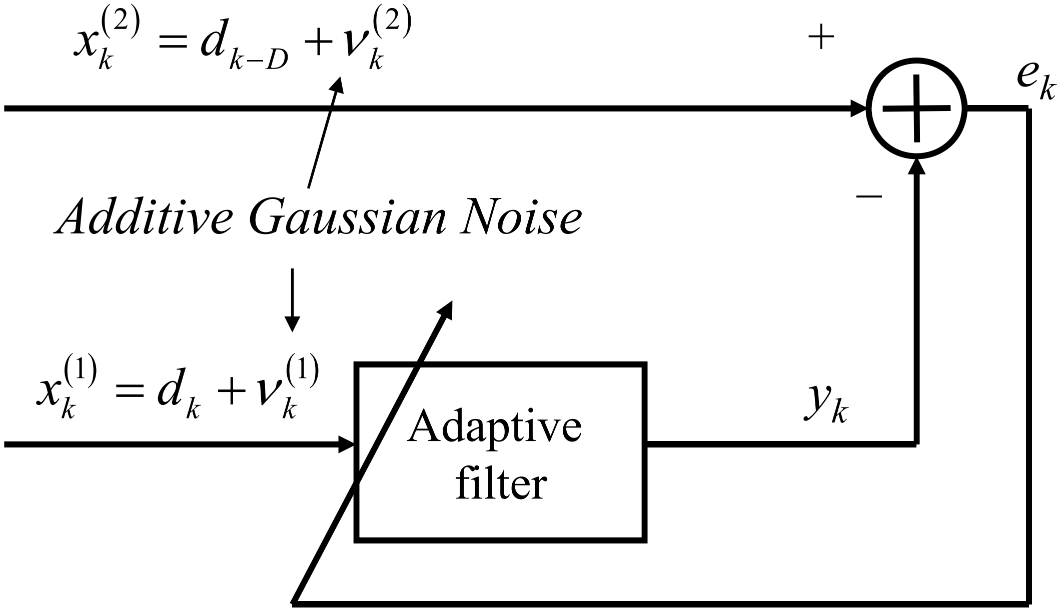
Channel equalization using training sequence (see later)

Some other adaptive filter configurations (2)



Equalization in decision-directed (decision-feedback/IIR) mode (without training sequence)

Some other adaptive filter configurations (3)


$x_k^{(2)} = d_{k-D} + \nu_k^{(2)}$
+
e_k
Additive Gaussian Noise
–
$x_k^{(1)} = d_k + \nu_k^{(1)}$
Adaptive filter
y_k

Time-delay estimator: filter cancels the delay between $x^1{}_k$ and $x^2{}_k$. The peak in **w** gives the *D* delay, which is taken as multiple of the sample interval.

Wiener filter (1)

$$J(\mathbf{w}) = \mathrm{E}\left[\left(d_k - \sum_{j=0}^{J} w_j x_{k-j}\right)^2\right] = \mathrm{E}\left[d_k^{\,2}\right] - 2\sum_{j=0}^{J} w_j \mathrm{E}\left[d_k x_{k-j}\right] + \sum_{j=0}^{J}\sum_{i=0}^{J} w_j w_i \mathrm{E}\left[x_{k-j} x_{k-i}\right] =$$

$$= V(0) - 2\sum_{j=0}^{J} w_j r(j) + \sum_{j=0}^{J}\sum_{i=0}^{J} w_j w_i R(i-j) = V(0) - 2\mathbf{w}^T\mathbf{r} + \mathbf{w}^T\mathbf{R}\mathbf{w}$$

constant

- Autocorrelation function:

$$\mathbf{R} = \left[R(i-j)\right]_{i,j=0}^{J},\ \dim(\mathbf{R}) = (J+1)\mathrm{x}(J+1)$$

- Cross correlation function:

$$\mathbf{r} = \left[r(j)\right]_{j=0}^{J},\ \dim(\mathbf{r}) = (J+1)$$

Objective:

$$\boxed{\mathbf{w}_{opt} := \min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{R}\mathbf{w} - 2\mathbf{w}^T\mathbf{r}\right\}}$$

Wiener filter (2)

- Properties of the correlation matrix (**R**):

1. Toeplitz: $R_{ij} = R_{i-j}$
2. Symmetric: $\mathbf{R} = \mathbf{R}^T$
3. Hermitian: $\forall \mathbf{a}, \mathbf{b}: \ \mathbf{a}^T \mathbf{R} \mathbf{b} = \mathbf{b}^T \mathbf{R} \mathbf{a}$
4. Positive semi definite: $\forall \mathbf{a}: \ \mathbf{a}^T \mathbf{R} \mathbf{a} \geq 0$
5. Eigenvectors are orthonormal, $\mathbf{R}\mathbf{s}_\mathbf{i} = \lambda_i \mathbf{s}_\mathbf{i}$

non-negative eigenvalues: $\mathbf{s}_i^T \mathbf{s}_j = \delta_{ij} = \begin{Bmatrix} 1, \text{ if} \\ 0, \text{ if} \end{Bmatrix}, \ \lambda_i \geq 0, \ \forall \mathrm{i}$

Wiener filter (3)

- Objective: $\mathbf{w}_{opt} := \min_{\mathbf{w}} \left\{ \mathbf{w}^T \mathbf{R} \mathbf{w} - 2\mathbf{w}^T \mathbf{r} \right\}$
- Global minimum exists because of Property 4.
- Solution: $\dfrac{\partial \mathbf{w}^T \mathbf{R} \mathbf{w} - 2\mathbf{w}^T \mathbf{r}}{\partial \mathbf{w}} = 2\mathbf{R}\mathbf{w} - 2\mathbf{r} = 0$
- Wiener-Hopf normal equation: $\boxed{\mathbf{R}\mathbf{w}_{opt} = \mathbf{r}}$
- Problems (why we should go further into adaptive signal processing) :

1. matrix inversion is not cost efficient: $\cancel{\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{r}}$
2. statistical features are not known: $\cancel{\mathbf{R}, \mathbf{r}}$

The need for recursive and adaptive solutions

In the case of real information processes **R**(k) and **r**(k) are not known, furthermore are changing in time because of only quasi stationer property of dealt processes (e.g.: voice can be treated as a stationer process only for a 30 ms time window)

Recursive solution: gradient descent algorithm:

$$\boxed{\mathbf{w}(k+1)=\mathbf{w}(k)-\Delta\left\{\mathbf{R}\mathbf{w}(k)-\mathbf{r}\right\}}$$

Speed of convergence

without matrix inversion.

$$\Delta_{\text{opt}}=\frac{2}{\lambda_{\min}+\lambda_{\max}}$$

Steady state: $\mathbf{w}=\mathbf{w}(k+1)=\mathbf{w}(k):\mathbf{R}\mathbf{w}=\mathbf{r}$

Recursive solution of Wiener filter (1)

$$\mathbf{w}(k+1)=\mathbf{w}(k)-\Delta\{\mathbf{R}\mathbf{w}(k)-\mathbf{r}\}$$

Eigenvector basis transformation: $\mathbf{R}\mathbf{s_i}=\lambda_i\mathbf{s_i},\ \forall i=0,\dots J$

$$\mathbf{w}(k)=\sum_{j=0}^{J}v_j(k)\mathbf{s}_j,\qquad v_j(k)=\mathbf{w}^T(k)\mathbf{s}_j,\qquad \mathbf{r}(k)=\sum_{j=0}^{J}\varsigma_j(k)\mathbf{s}_j$$

$$\mathbf{w}(k+1)=\mathbf{w}(k)-\Delta\{\mathbf{R}\mathbf{w}(k)-\mathbf{r}\}$$

$$\sum_{i=0}^{J}v_i(k+1)\mathbf{s_i}=\sum_{i=0}^{J}v_i(k)\mathbf{s_i}-\Delta\left\{\mathbf{R}\left(\sum_{i=0}^{J}v_i(k)\mathbf{s_i}\right)-\sum_{i=0}^{J}\varsigma_i(k)\mathbf{s_i}\right\}$$

Recursive solution of Wiener filter (2)

After rewriting the formula without **R**:

$$\sum_{i=0}^{J} v_i(k+1)\mathbf{s}_i = \sum_{i=0}^{J}\left(v_i(k) - \Delta\lambda_i v_i(k) + \Delta\varsigma_i\right)\mathbf{s}_i$$

Two vector can be only equal, if each of the components are equal:

$$v_i(k+1)\mathbf{s}_i = (1-\Delta\lambda_i)v_i(k) + \Delta\varsigma_i$$

This differential equation has the following homogeneous solution:

$$v_i(k) = c_i(1-\Delta\lambda_i)^k + \frac{\varsigma_i}{\lambda_i}$$

Recursive solution of Wiener filter (3)

After replacing it into the original formula:

$$\mathbf{w}(k)=\sum_{i=0}^{J}c_i\left(1-\Delta\lambda_i\right)^k\mathbf{s}_i+\sum_{i=0}^{J}\frac{\varsigma_i}{\lambda_i}\mathbf{s}_i$$

Transient component: $\sum_{i=0}^{J}c_i\left(1-\Delta\lambda_i\right)^k\mathbf{s}_i$

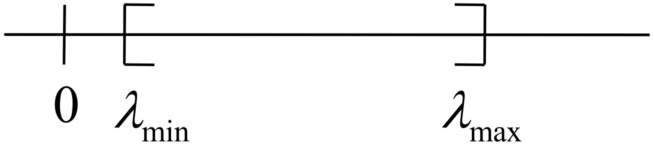
Wiener solution: $\sum_{i=0}^{J}\frac{\varsigma_i}{\lambda_i}\mathbf{s}_i$

What is the optimal step size in order to maximize the speed of convergence? $\Delta_{\text{opt}}=?$

Recursive solution of Wiener filter (5)

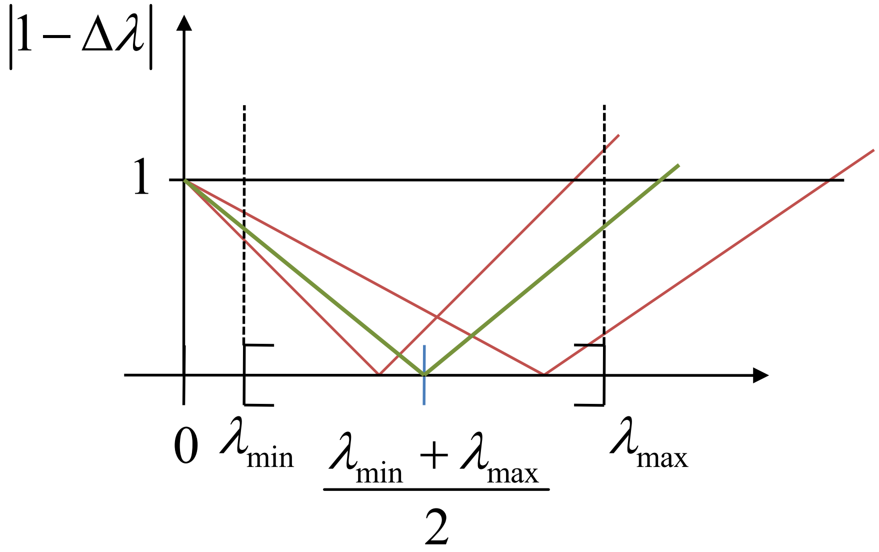
$$\Delta_{\text{opt}} = \min_{\Delta} \max_{i} \{1 - \Delta\lambda_i\}$$

$$\lambda_{\min} = \min(\lambda_i) \qquad \lambda_{\max} = \max(\lambda_i) \qquad \lambda_i \in [\lambda_{\min}, \lambda_{\max}],\ \forall i$$



Relaxation: $$\Delta_{\text{opt}} = \min_{\Delta} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \{1 - \Delta\lambda_i\}$$

Recursive solution of Wiener filter (6)



The step size can be optimal iff,

$$\Delta_{\text{opt}} = \min_{\Delta} \max_{i} \{1-\Delta\lambda_i\}$$

$$1-\Delta\lambda_{\min} = \Delta\lambda_{\max} - 1$$

$$\Delta_{\text{opt}} = \frac{2}{\lambda_{\min}+\lambda_{\max}}$$

Recursive solution of Wiener filter (7)

$$\mathbf{w}(k)=\sum_{i=0}^{J}c_i\left(1-\Delta_{\text{opt}}\lambda_i\right)^k\mathbf{s}_i+\sum_{i=0}^{J}\frac{\varsigma_i}{\lambda_i}\mathbf{s}_i$$

$$\mathbf{w}(k+1)=\mathbf{w}(k)-\Delta_{\text{opt}}\left\{\mathbf{R}\mathbf{w}(k)-\mathbf{r}\right\}$$

Problems: $\lambda_{\min},\lambda_{\max}=?\quad\lambda_i=?$

$\mathbf{R}\mathbf{s}_\mathbf{i}=\lambda_i\mathbf{s}_\mathbf{i},\ \forall i=0,\ldots J$ should be solved, but due to the complexity of eigenvalue decomposition it is impossible to be done in real-time!

$$\det\left(\mathbf{R}-\lambda\mathbf{I}\right)=0$$

Recursive solution of Wiener filter (8)

Estimation of eigenvalues: Gersgorin circles

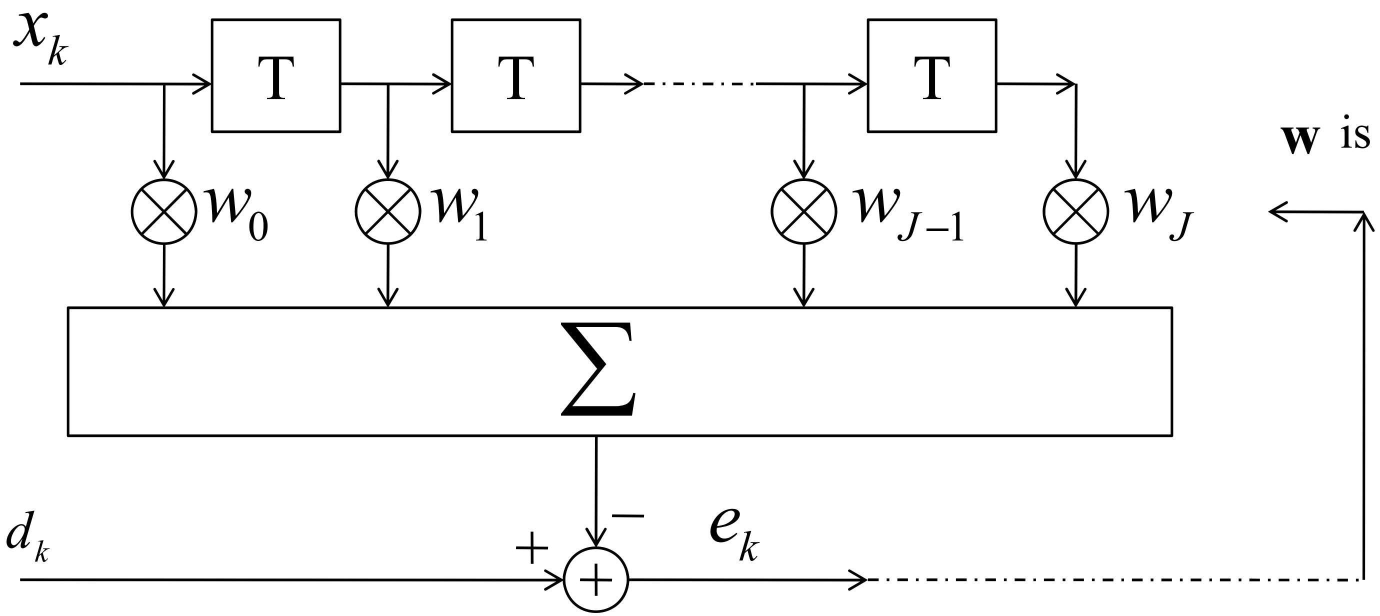
$$\lambda_i \in \left[R_{ii} - \sum_{j,j\neq i} \left| R_{ij} \right|, R_{ii} + \sum_{j,j\neq i} \left| R_{ij} \right|, \right]$$

$$\lambda_{\min} \geq R_{ii} - \sum_{j,j\neq i} \left| R_{ij} \right|,\ \lambda_{\max} \leq R_{ii} + \sum_{j,j\neq i} \left| R_{ij} \right|$$

R(0) can be observable! A near optimal step size can be implemented!

$$\Delta_{\text{opt}} = \frac{2}{\lambda_{\min} + \lambda_{\max}} \approx \frac{2}{2R_{ii}} = \frac{1}{R(0)}$$

Adaptive filter solution



$\mathbf{w}$ is changing in time

$$\mathbf{w}_{opt} := \min J(\mathbf{w})$$

$$J(\mathbf{w}) = \mathrm{E}\left[(d_k - y_k)^2\right] = \mathrm{E}\left[\left(d_k - \sum_{j=0}^{J} w_j x_{k-j}\right)^2\right] = \text{how to minimize this}$$

without the knowledge of $\mathbf{R}$ and $\mathbf{r}$

Adaptive filtering with unknown statistical parameters(1)

Problem: **R** and **r** are not given!

$$\mathbf{R} : R_{ij} = \mathrm{E}\left[x_{k-i} x_{k-j} \right]$$

$$\mathbf{r} : r_i = \mathrm{E}\left[d_k x_{k-i} \right]$$

However a learning set can be known (a set of input-output pairs):

$$\tau^{(K)} = \left\{ \left(x_k , d_k \right), k = 1,\dots, K \right\}$$

Empirical error function:

$$\mathbf{w}_{\text{opt}}^{(K)} = \min_{\mathbf{w}} \frac{1}{K} \sum_{k=1}^{K} \left(d_k - \sum_{j=0}^{J} w_j x_{k-j} \right)^2$$

Unknown statistical parameters(2)

Does it converge to the analytical solution?

$$\mathbf{d}=\left(d_{1,}d_{2,}\ldots,d_{K,}\right),\ \mathbf{x}^{(j)}=\left(\underbrace{0,0,\ldots,0}_{j},x_{1},x_{2,}\ldots,d_{K-j,}\right)$$

Notations:

$$J\left(\mathbf{w}\right)=\frac{1}{K}\sum_{k=1}^{K}\left(d_k-\sum_{j=0}^{J}w_j x_{k-j}\right)^2\sim\left\|\mathbf{d}-\sum_{j=0}^{J}w_j\mathbf{x}^{(j)}\right\|^2$$

Application of the projection theorem:

$$\mathbf{w}_{\text{opt}}:\left\langle\mathbf{d}-\sum_{j=0}^{J}w_j\mathbf{x}^{(j)},\mathbf{x}^{(i)}\right\rangle=0,\ \forall i=0,\ldots,J$$

Unknown statistical parameters(3)

After re-ordering:

$$\underbrace{\sum_{j=0}^{J}\frac{1}{K}\left(\sum_{k=1}^{K}x_{k-i}x_{k-j}\right)w_j}_{\widetilde{\mathbf{R}}\mathbf{w}}=\underbrace{\frac{1}{K}\sum_{k=1}^{K}d_k x_{k-i}}_{\tilde{\mathbf{r}}}\,,\ \forall i=0,\dots,J$$

It approximates to the optimal solution, in the sense that we use the consistent estimation of correlation matrix and vector, respectively!

A recursive solution: LD algorithm(1)

$$\mathbf{R}^{(K+1)}\mathbf{w}^{(K+1)} = \mathbf{r}^{(K+1)}$$

We can decompose the correlation matrix for consecutive diades in time.

$$\mathbf{w}^{(K+1)} = \left(\mathbf{R}^{(K+1)}\right)^{-1}\mathbf{r}^{(K+1)} = \left(\underbrace{\mathbf{R}^{(K)} + \mathbf{x}^{(K+1)}\mathbf{x}^{(K+1)\mathbf{T}}}_{\text{Matrix diade inversion lemma}}\right)^{-1}\left(\mathbf{r}^{(K)} + d_{K+1}\mathbf{x}^{(K+1)}\right)$$

The matrix diade inversion lemma states:

$$\left(\mathbf{R}^{(K+1)}\right)^{-1} = \frac{\left(\mathbf{R}^{(K)}\right)^{-1}\mathbf{x}^{(K+1)}\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\left(\left(\mathbf{R}^{(K)}\right)^{-1}\right)^{\mathrm{T}}}{1+\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\mathbf{R}^{(K)}\mathbf{x}^{(K+1)}}$$

A recursive solution: LD algorithm(2)

By using the decomposition we can write the optimal recursion, the Levinson-Durbin algorithm:

$$\mathbf{w}_{\text{opt}}^{(K+1)} = \mathbf{w}_{\text{opt}}^{(K)} - \frac{\left(\mathbf{R}^{(K)}\right)^{-1}\mathbf{x}^{(K+1)}\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\left(\left(\mathbf{R}^{(K)}\right)^{-1}\right)^{\mathrm{T}}}{1+\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\mathbf{R}^{(K)}\mathbf{x}^{(K+1)}}\left\{d_{K+1} - \left(\mathbf{w}^{(K)}\right)^{\mathrm{T}}\mathbf{x}^{(K)}\right\}\mathbf{x}^{(K)}$$

The RM algorithm

The computing complexity of the term is high:

$$\frac{\left(\mathbf{R}^{(K)}\right)^{-1}\mathbf{x}^{(K+1)}\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\left(\left(\mathbf{R}^{(K)}\right)^{-1}\right)^{\mathrm{T}}}{1+\left(\mathbf{x}^{(K+1)}\right)^{\mathrm{T}}\mathbf{R}^{(K)}\mathbf{x}^{(K+1)}}$$

Simpler solution with stochastic approximation is a substitution of a monotone decreasing function:

$$\underbrace{w_l(k+1)=w_l(k)-\Delta\left(k\right)\left\{d_k-\sum_{j=0}^{J}w_j(k)x_{k-j}\right\}x_{k-l},\quad l=0,\dots,J}$$

Robbins-Monroe algorithm

The solution yielded by the RM algorithm

Solution converges only in mean square, i.e.:

$$\lim_{k\to\infty} E\left\|\mathbf{w}(k)-\mathbf{w}_{\text{opt}}\right\|^2 = 0$$

The proof of this statement is based on the Kushner-Clark theorem (coming later), first we only demonstrate that in equilibrium indeed the optimal solution is obtained.

Steady state analysis of the RM algorithm (1)

In order to analyze the equilibrium, one must first keep in mind that this algorithm is a stochastic (random) recursion. As result, no changes will occur if the average of is going to be zero. As a result for reaching the equilibrium the following set of equations must be satisfied:

$$\left\{ d_k - \sum_{j=0}^{J} w_j(k) x_{k-j} \right\} x_{k-l}$$

$$E\left(\left\{ d_k - \sum_{j=0}^{J} w_j(k) x_{k-j} \right\} x_{k-l} \right) = 0, \quad l = 0, \ldots, J$$

Steady state solution of the RM algorithm (2)

which condition can be rewritten as follows:

$$E\left(\sum_{j=0}^{J} w_j x_{k-j} x_{k-l}\right) = E\left(d_k x_{k-l}\right),\ \ l = 0,\dots,J$$

$$\sum_{j=0}^{J} w_j E\left(x_{k-j} x_{k-l}\right) = E\left(d_k x_{k-l}\right),\ \ l = 0,\dots,J$$

$$\sum_{j=0}^{J} R_{lj} w_j = r_l,\ \ l = 0,\dots,J$$

or in vector notation:

$$\mathbf{Rw} = \mathbf{r}$$

which is just the solution of the Wiener filtering!

The LMS algorithm

Least Mean Squares algorithm: $\Delta(k) = \Delta, \ \forall \mathrm{k}$

$$w_l(k+1) = w_l(k) - \Delta\left\{d_k - \sum_{j=0}^{J} w_j(k) x_{k-j}\right\} x_{k-l}, \quad l = 0,\dots,J$$

LD algorithm: high computational complexity, optimal convergence

RM algorithm: lower complexity, only asymptotic convergence can be guaranteed

LMS algorithm(2)

LMS algorithm: very low complexity, no convergence guaranteed, but mostly it works! (compared to the RM algorithm)

Steepest-descent version of RM algorithm.

Performing the LMS recursion only the observed samples of random processes and are needed and the algorithm converge to the optimal solution of Wiener filtering without any a priori knowledge on the correlation properties of the processes.

$$w_l(k+1) = w_l(k) - \Delta\left\{d_k - \sum_{j=0}^{J} w_j(k)x_{k-j}\right\}x_{k-l}, \quad l = 0,\ldots,J$$

Applications of adaptive filtering

- Open questions:
  - What is the optimal degree of the model? *J*=?
  - Information theoretical solution (Akaike, Risamen)
  - VC dimension (Vapnik Chervonenkis)
- This algorithm has wide spread applications in
  - data compression,
  - adaptive channel equalization,
  - noise cancellation.

Linear prediction(1)

Past samples of an ergodic and weakly stationary process are given: $x_{k-J}, x_{k-(J-1)}, \ldots, x_{k-1}$

Let us predict the future: $x_{k-J}, x_{k-(J-1)}, \ldots, x_{k-1} \rightarrow \tilde{x}_k$

Prediction with linear filter: $\widetilde{x_k} = \sum_{j=1}^{J} w_j x_{k-j}$

Optimal linear predictive filter: $\mathbf{w}_{\text{opt}} = \min_{\mathbf{w}} \left\{ E\left[x_k - \widetilde{x_k} \right]^2 \right\}$

Task: $\mathbf{w}_{\text{opt}} = \min_{\mathbf{w}} \left\{ E\left[x_k - \sum_{j=1}^{J} w_j x_{k-j} \right]^2 \right\}$

Linear prediction (2)

This optimization task equals to a special Wiener-filtering problem, where:

$$d_k = x_k$$

$$R_{ij} = R(i-j) = E\left[x_{k-i} x_{k-j}\right] \qquad r_i = r(i) = E\left[d_k x_{k-i}\right]$$

$$\mathbf{Rw} = \mathbf{r}$$

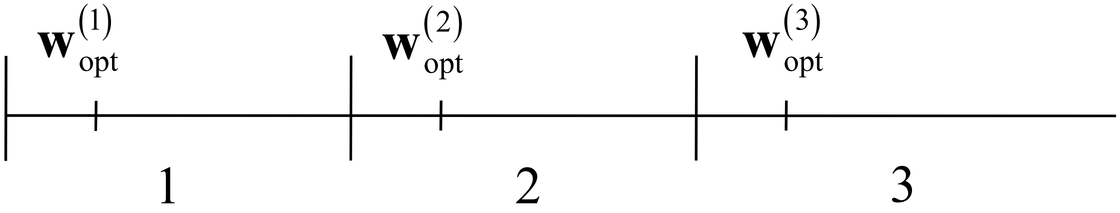
Note: Model degree is only J (not J+1).

If **R** is not known, use the RM algorithm:

$$w_l(k+1) = w_l(k) - \Delta(k)\left\{d_k - \sum_{j=0}^{J} w_j(k) x_{k-j}\right\} x_{k-l}, \quad l = 1, \ldots, J$$

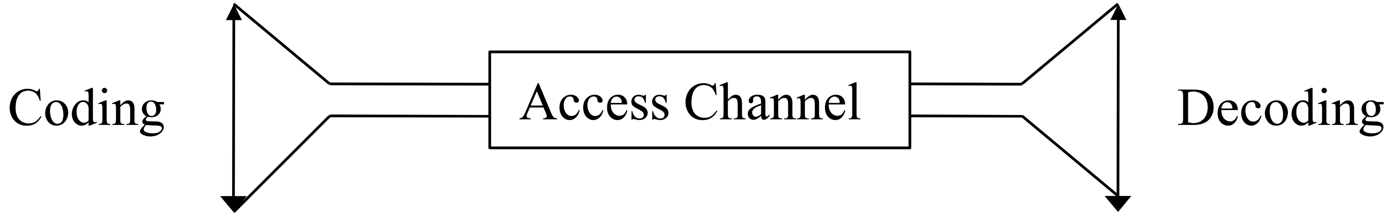
Implementation in real-life communication systems

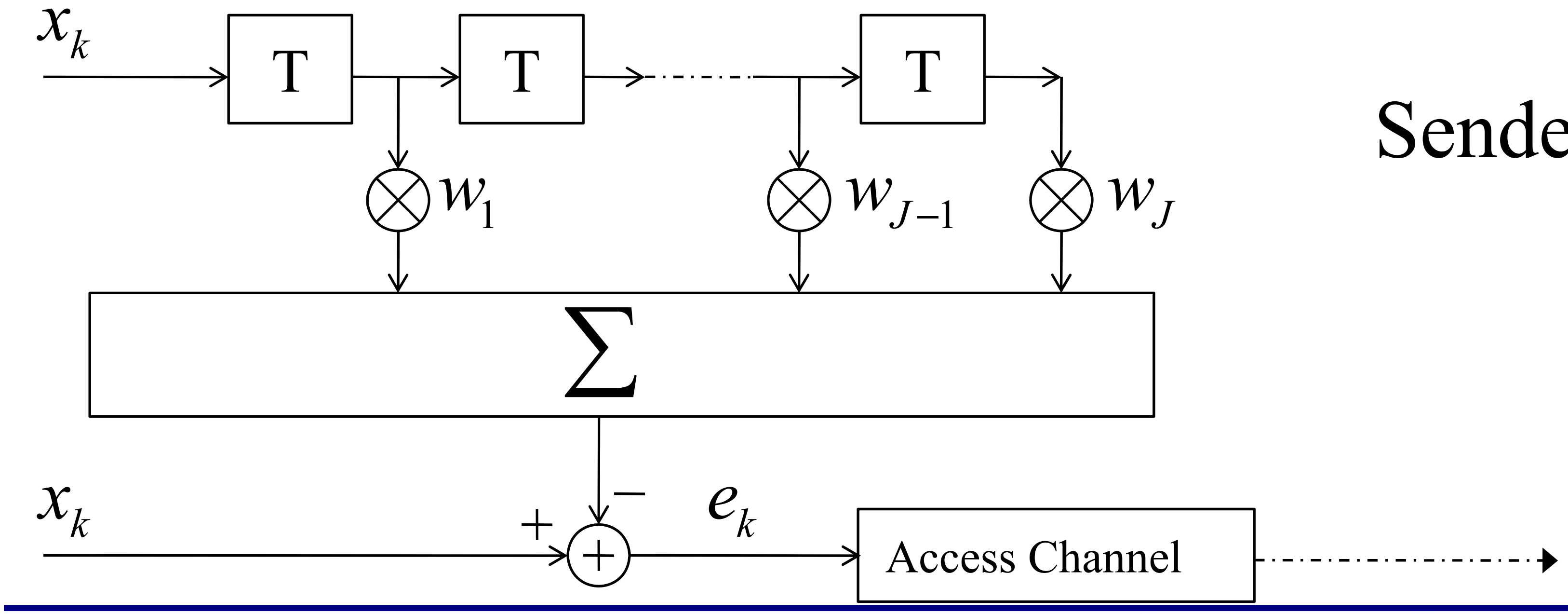
The source is not stationary, only in time slots, therefore filter parameters should be re-optimized all the time:


$\mathbf{w}_{\text{opt}}^{(1)}$
$\mathbf{w}_{\text{opt}}^{(2)}$
$\mathbf{w}_{\text{opt}}^{(3)}$
1
2
3

Optimal filter setting should be resent according to the statistical feature of the learning set!

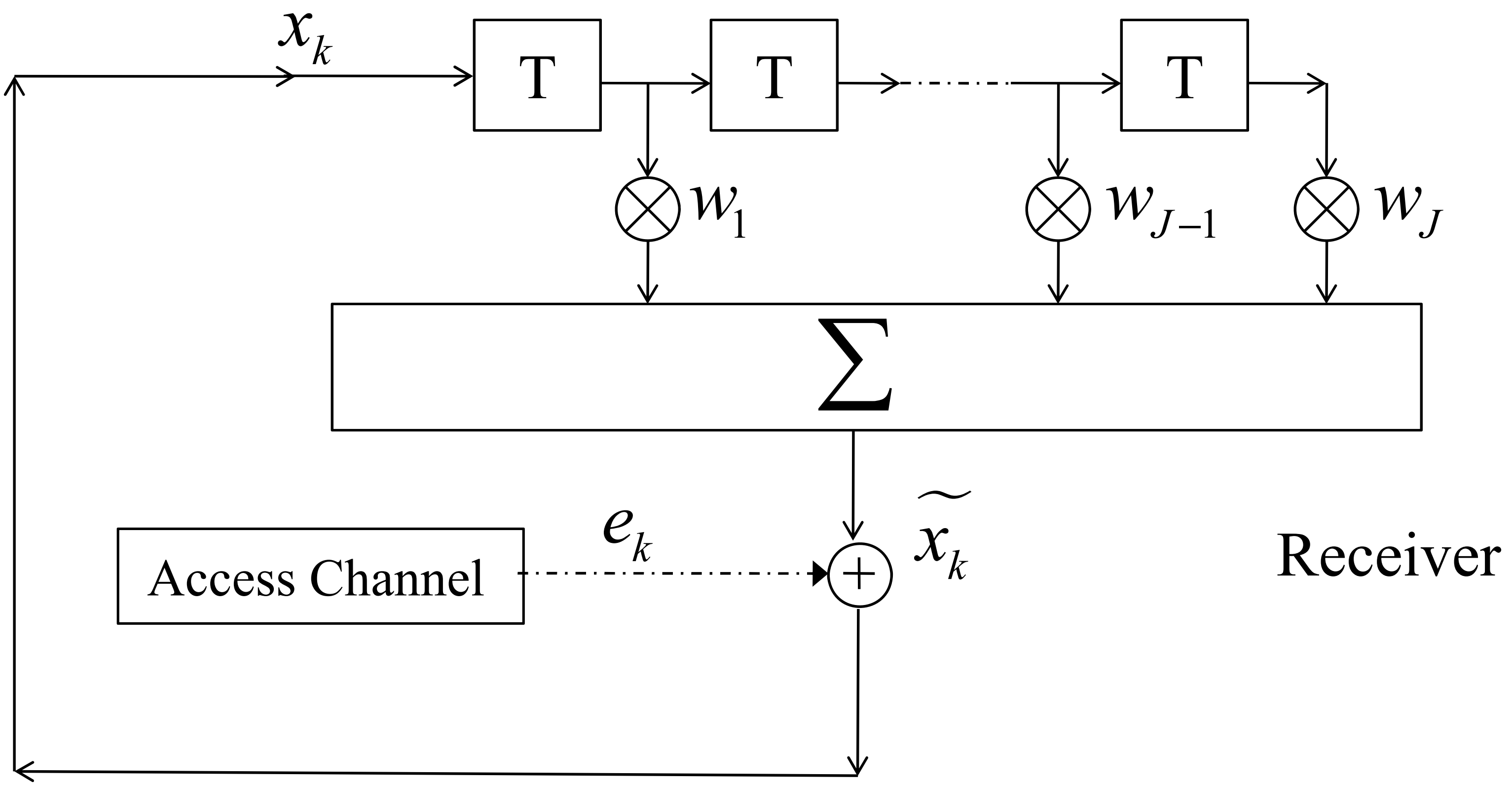
Applications: adaptive-predictive coding(1)





Sender

Applications: adaptive-predictive coding(2)



Applications: adaptive-predictive coding(3)

Real-time implementation:

$$\mathbf{w}_{\text{opt}} = \min_{\mathbf{w}} \mathrm{E}\left[e_k^2\right] \rightarrow \mathbf{R}\mathbf{w}_{\text{opt}} = \mathbf{r}$$

$$w_l(k+1) = w_l(k) - \Delta\left(k\right)\left\{d_k - \sum_{j=0}^{J} w_j(k) x_{k-j}\right\} x_{k-l}, \quad l = 1,\ldots,J$$

Applications: adaptive-predictive coding (4)

We are interested in data compression rate.

$$\mathrm{E}\left[e_k^2\right]=\mathrm{E}\left[\left(x_k-\sum_{j=1}^{J}w_j^{\mathrm{opt}}x_{k-j}\right)^2\right]=\mathrm{E}\left[x_k{}^2\right]-2\sum_{j=1}^{J}w_j\mathrm{E}\left[x_kx_{k-j}\right]+\sum_{j=1}^{J}\sum_{i=1}^{J}w_jw_i\mathrm{E}\left[x_{k-j}x_{k-i}\right]=$$

$$=V(0)-2\sum_{j=1}^{J}w_jr(j)+\sum_{j=1}^{J}\sum_{i=1}^{J}w_jw_iR(i-j)=V(0)-2\mathbf{w}^T\mathbf{r}+\mathbf{w}^T\mathbf{R}\mathbf{w}$$

with the optimal filter coefficients this becomes

$$\mathrm{E}\left[e_k^2\right]=V(0)-2\mathbf{w}_{\mathrm{opt}}^T\mathbf{r}+\mathbf{w}_{\mathrm{opt}}^T\mathbf{R}\mathbf{w}_{\mathrm{opt}}=R(0)-\mathbf{w}_{\mathrm{opt}}^T\mathbf{r}=R(0)-\mathbf{w}_{\mathrm{opt}}^T\mathbf{R}\mathbf{w}_{\mathrm{opt}}$$

Applications: adaptive-predictive coding (5)

R is hermitian, therefore it has orthonormal eigenvectors only positive eigenvalues:

$$\mathbf{R}\mathbf{s}_\mathbf{i} = \lambda_i \mathbf{s}_i,\ \mathbf{s}_i^T \mathbf{s}_j = 0, i \neq j,\ \lambda_i > 0,\ \forall i = 1,\ldots J$$

Optimal filter can be represented in the eigenvector space:

$$\mathbf{w}_{\mathrm{opt}} = \sum_{i=1}^{J} v_i^{\mathrm{opt}} \mathbf{s}_i$$

$$\mathrm{E}\left[e_k^2\right] = \mathrm{E}\left[x_k^2\right] - \sum_{i=1}^{J}\sum_{j=1}^{J} v_i^{\mathrm{opt}} v_j^{\mathrm{opt}} \mathbf{s}_i^T \mathbf{R}\mathbf{s}_j \mathrm{E}\left[x_{k-j} x_{k-i}\right] =$$

$$= \mathrm{E}\left[x_k^2\right] - \sum_{i=1}^{J}\sum_{j=1}^{J} v_i^{\mathrm{opt}} v_j^{\mathrm{opt}} \lambda_i \mathbf{s}_i \mathbf{s}_j^T \mathrm{E}\left[x_{k-j} x_{k-i}\right] = \mathrm{E}\left[x_k^2\right] - \sum_{i=1}^{J} \lambda_i v_i^{\mathrm{opt}}$$

Applications: adaptive-predictive coding (6)

The energy of the input signal is much higher than the energy of the compressed signal, which yields better quantization possibilities!

$$\sum_{i=1}^{J} \lambda_i v_i^{\text{opt}} << 0 \rightarrow \mathrm{E}\left[e_k^2\right] << \mathrm{E}\left[x_k^2\right]$$

$$H\left[e_k^2\right] << H\left[x_k^2\right]$$

The entropy of the predicted signal is smaller than the original one, thus source coding can be more efficient!

Applications: adaptive-predictive coding (7)

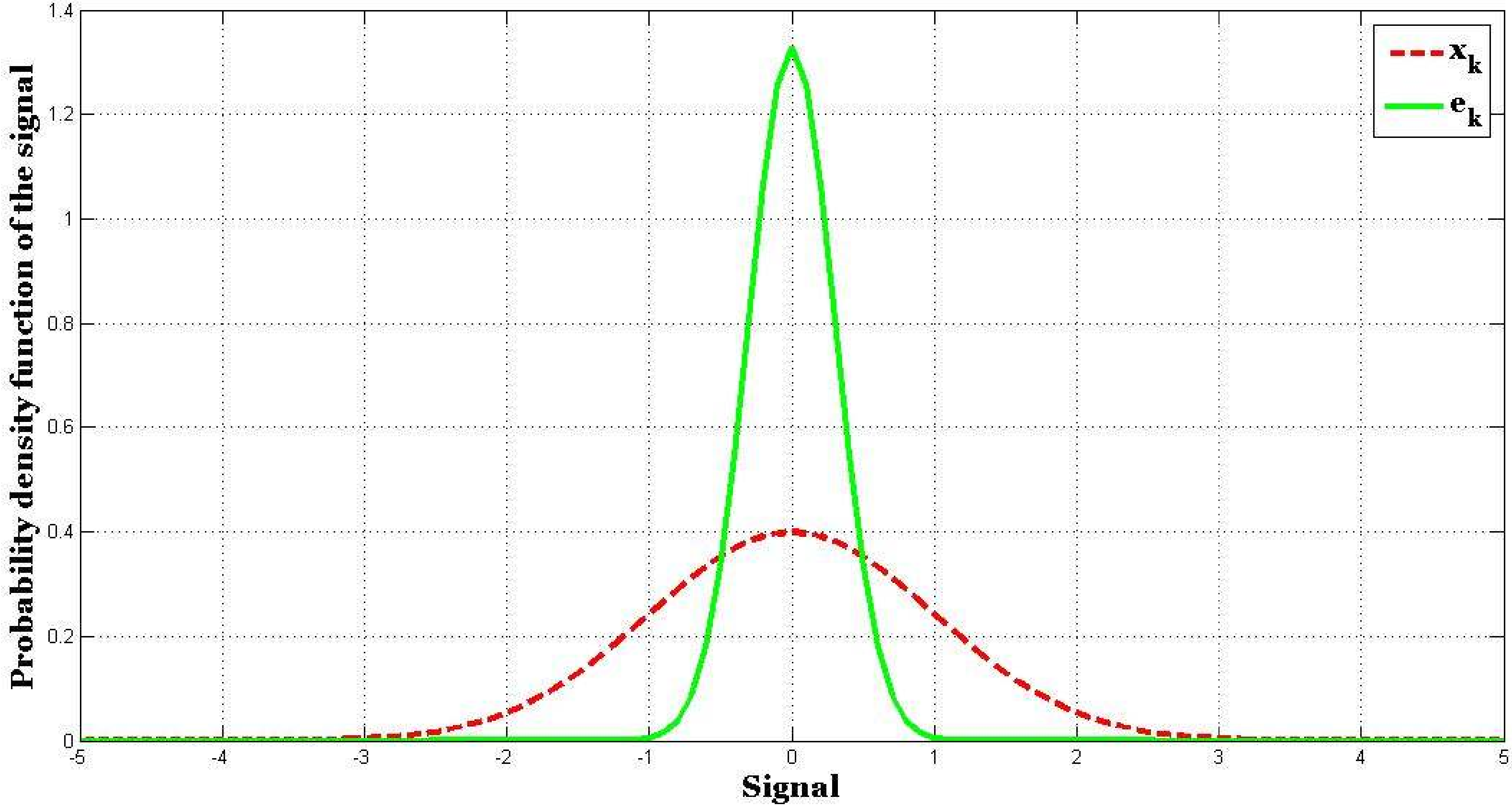
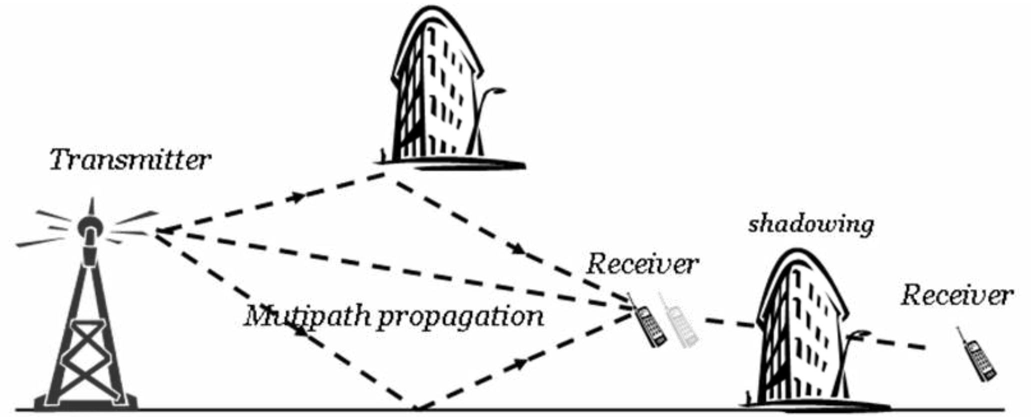


Illustration of the effect of data compression

Applications: adaptive-predictive coding (8)

- The adaptive-predictive coding (or linear predictive coding) is used as a form of voice compression by phone companies (e.g.. in the GSM standard).
  - GSM coder uses this approach and achieves 6.5 Kbit/s instead of 64 Kbit/s in the case of voice!
- It is also used for secure wireless, where voice must be digitized, encrypted and sent over a narrow voice channel (e.g.. Navajo I).
- LPC synthesis can be used to construct vocoders where musical instruments are used as excitation signal to the time-varying filter estimated from a singer's speech.
- LPC predictors are used in Shorten, MPEG-4 ALS, FLAC, and other lossless audio codecs.

Applications: channel equalization (1)

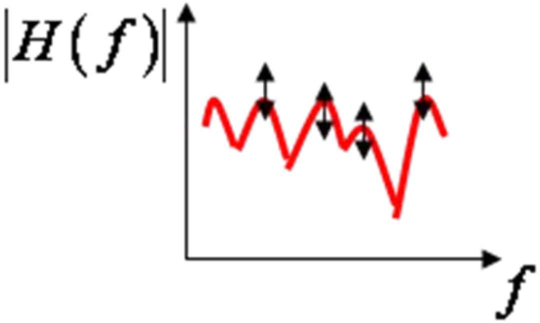


Wireless channel impairments:

–Shadowing, large-scale path loss

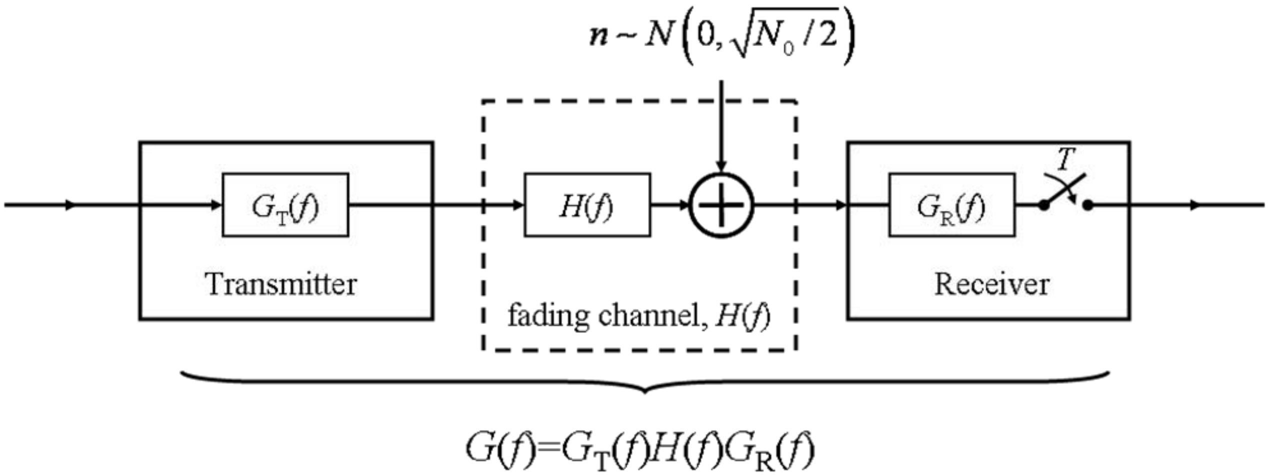
–Multipath Fading, rapid small-scale signal variations (ISI)

–Doppler Spread due to motion of mobile unit



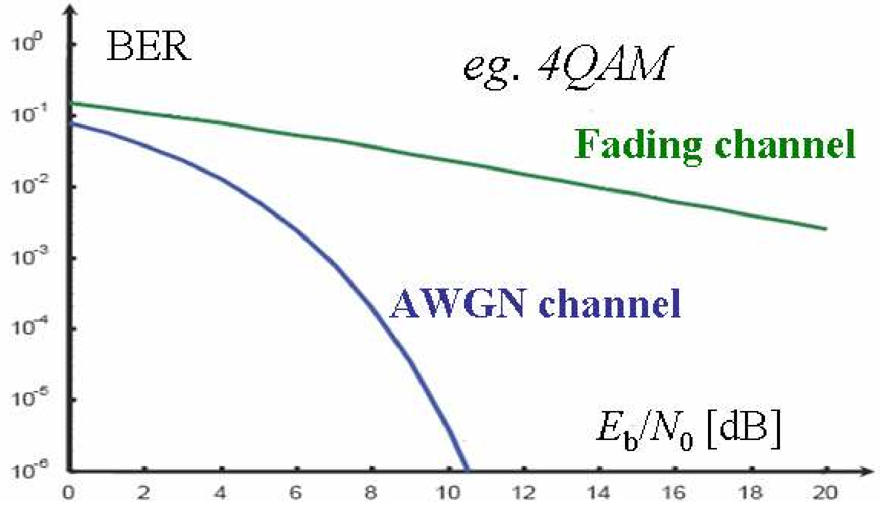
Applications: channel equalization (2)

Due to the distortion of fading channel, the signal propagates in a multipath fashion from the sender to the receiver! This phenomenon causes severe distortion in the transmission characteristics of the channel, i.e. the signals of the past get mixed with present values (Intersymbol Interference) which introduces memory in the IT channel model!



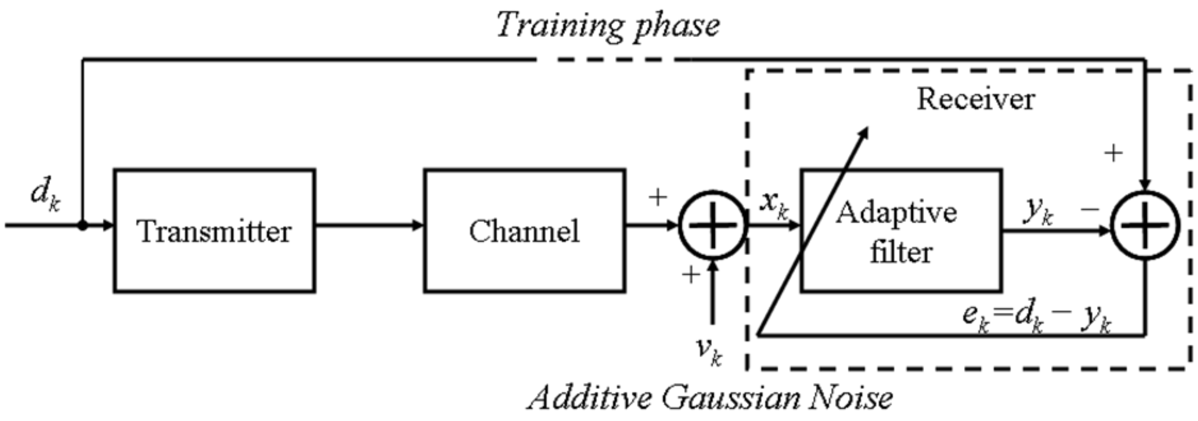
Applications: channel equalization (3)

The channel impairments can lead to significant distortion or attenuation of the received signal (SNR) which degrade Bit Error Rate (BER) of digitally modulated signal.



Applications: channel equalization (4)

- Two techniques are used to improve received signal quality and lower BER:
  - Diversity (expensive and resource consuming): It requires double antennas or double spectrum.
  - Equalization (a more economical approach): it requires only DSP and algorithmic developments.



Applications: channel equalization (5)

- Quality of Service (QoS) of a wireless communication system:

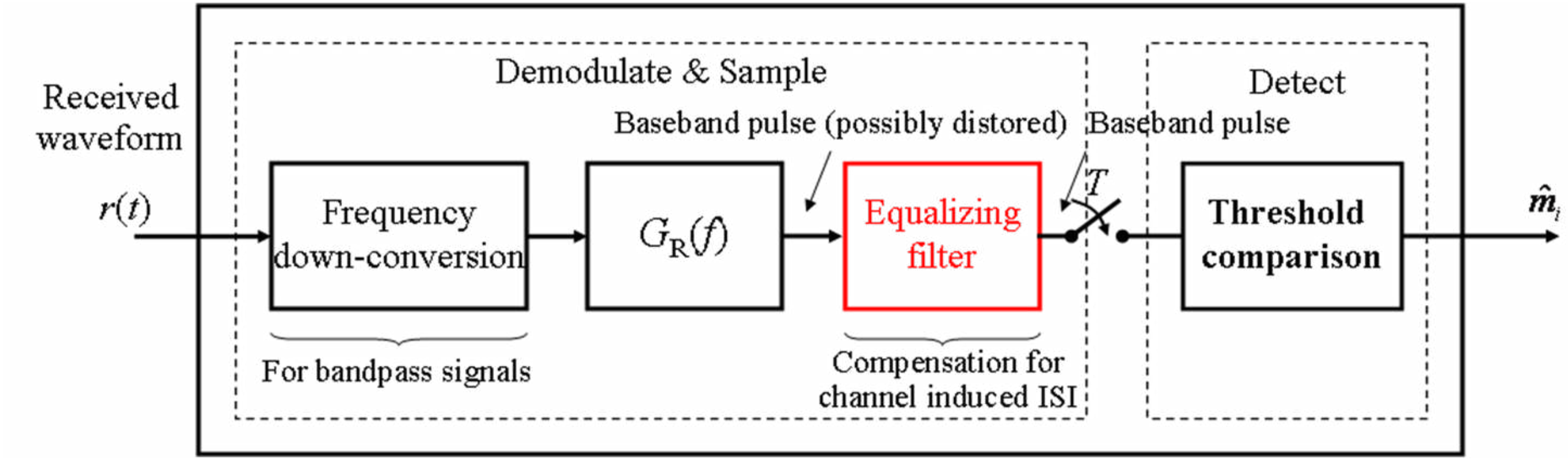
  Spectral efficiency [(bit/sec)/Hz]

  which refers to the maximal data rate that can be transmitted over a given bandwidth in a specific communication system. (e.g. In GSM (1991) system R=0.104Mbit/s per carrier, B=0.2MHz per carrier SE=R/B=0.52 (bit/sec)/Hz, or in an LTE (2009)system SE=16.32 (bit/sec)/Hz, or in 802.11g SE=2.7(bit/sec)/Hz)
- Spectral efficiency is determined by the bit error rate!
- The bit error rate is determined by the channel distortion.
- Thus **adaptive equalization plays a central role**.

Applications: channel equalization (6)

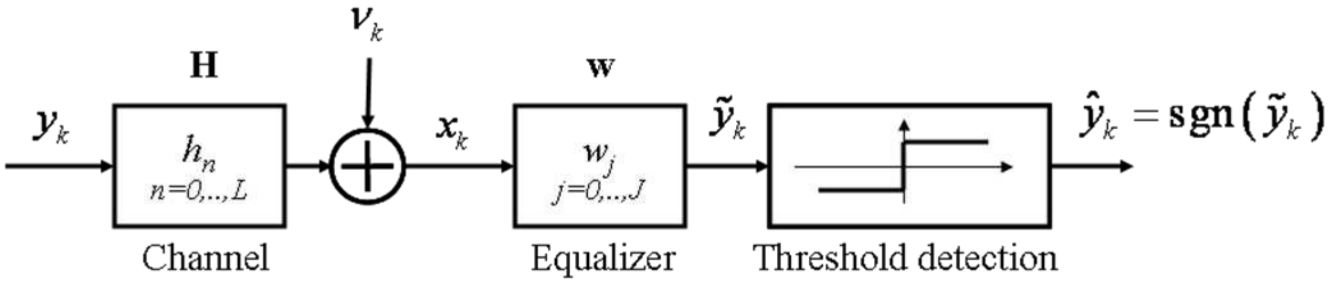
- Equalization algorithms aim to tackle the ISI by implementing linear filters to equalize the channel distortions at the receiver side!
- Challenge: how to develop low complexity adaptive signal processing algorithms, which are:
  - real-time and easily implementable;
  - yield low Bit Error Rate;
  - have learning capabilities to equalize unknown channels only based on input-output samples.

Applications: channel equalization (7)



Applications: channel equalization (8)

- The simplified model and notations:



$$x_k = \sum_{n=0}^{L} h_{k-n} y_n + \nu_k$$

$$\tilde{y}_k = \sum_{j=0}^{J} w_j x_{k-j} = \sum_{j=0}^{J} w_j \left(\sum_{n=0}^{L} h_{k-j-n} y_n + \nu_{k-j} \right) = \sum_{n=0}^{L} \underbrace{\left(\sum_{j=0}^{J} w_j h_{k-j-n} \right)} y_n + \underbrace{\sum_{j=0}^{J} w_j \nu_{k-j}}$$

Equivalent filter

$$\tilde{y}_k = \sum_{n} q_{k-n} y_n + \eta_k =$$

$$E\left\{\eta_k^2\right\} = N_0 \left\|\mathbf{w}\right\|^2$$

ISI

$$= q_0 y_k + \boxed{\sum_{n \neq k} q_{k-n} y_n} + \boxed{\eta_k}$$

$$\eta_k \sim N\left(0, \sqrt{N_0}\left\|\mathbf{w}\right\|\right)$$

Applications: channel equalization (9)

- Signal:

$$y_k : P\left(y_k = +1\right) = P\left(y_k = -1\right) = 0.5$$

- Channel can be represented as a linear filter, which has an impulse response:

$$h_j, j = 0, \ldots, L$$

- Additive Gaussian noise :

$$\nu_k \sim N\left(0, \sqrt{N_0}\right) \quad E\left[\nu_j \nu_i\right] = \delta_{ij} N_0$$

- Adaptive filter with impulse response: $w_j, j = 0, \ldots, J$

- What is optimal **w**?

Applications: channel equalization (10)

- What is optimal **w**?
  - Optimal filter with given channel: **Equalization**
  - Optimal filter with unknown channel: **Adaptive equalization**
  - Optimal filter with unknown channel without learning set: **Blind adaptive equalization**
- Tradeoff between noise and ISI cancellation.

$$\tilde{y}_k = \sum_n q_{k-n} y_n + \eta_k = q_0 y_k + \sum_{n \neq k} q_{k-n} y_n + \eta_k$$

$$\eta_k \sim N\left(0, \sqrt{N_0}\,\|\mathbf{w}\|\right)$$

1. strategy: Zero Forcing (ZF)

Applications: channel equalization (11)

The Zero-Forcing strategy: we assume relatively large signal to noise ratio (the effect of noise can be neglected and focus only on ISI cancellation):

$$\widetilde{y_k} = q_k y_k \qquad q_k = \delta_{k,0} = \begin{cases} 1, \text{ if } k = 0 \\ 0 \text{ else} \end{cases}$$

Typical application: wired communication

The optimal weights:

$$\mathbf{w}_{\text{opt}} : \sum_{j=0}^{J} h_{k-j} w_j = \delta_{k,0}, \; k = 0, \ldots, J + L + 1$$

Applications: channel equalization (12)

Problem: *J+1* free parameters and *J+L+2* equations which leads over determined equation system!

Solution: defining a new goal function

→ Peak Distortion (PD): the maximal value of ISI

$$PD(\mathbf{w}) = \sum_{k=1}^{J+L+1} \left| q_k \right|$$

The optimal weights:

$$\mathbf{w}_{\text{opt}} : \min_{\mathbf{w}} PD(\mathbf{w}) = \min_{\mathbf{w}} PD(\mathbf{w}) = \sum_{k=1}^{J+L+1} \left| \sum_{j=0}^{J} h_{k-j} w_j \right|, \; q_0 = 1$$

Applications: channel equalization (13)

The Peak Distortion :

$$PD(\mathbf{w}) = \sum_{k=1}^{J+L+1} \left(\sum_{j=0}^{J} h_{k-j} w_j \right) \operatorname{sgn} \left(\sum_{j=0}^{J} h_{k-j} w_j \right)$$

Extremal point:

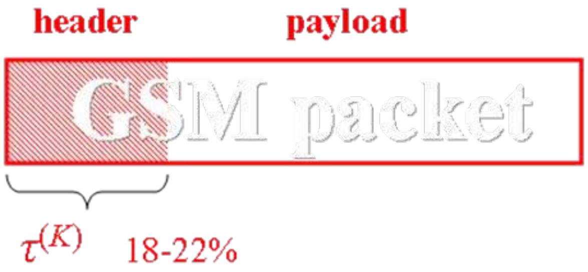
$$\left. \begin{array}{l} \displaystyle\sum_{j=0}^{J} h_{k-j} w_j = 0, \ k = 1,\dots,J \\ \displaystyle\sum_{j=0}^{J} h_{-j} w_j = 1, \ k = 1,\dots,J \end{array} \right\} \rightarrow \sum_{j=0}^{J} h_{k-j} w_j = \delta_{k,0}, \ k = 0,\dots,J$$

$$\boxed{\mathbf{Hw} = \boldsymbol{\delta}, \ \boldsymbol{\delta}^T = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}}$$

Applications: channel equalization (14)

In the case of unknown **H**: adaptive Zero Forcing

Training set:



$$\tau^{(K)} = \left\{ \left(y_k , x_k \right), k = 1,\dots,K \right\}$$

$$w_l(k+1) = w_l(k) - \Delta\left(k\right)\left\{ y_k - \sum_{j=0}^{J} w_j(k) x_{k-j} \right\} y_{k-l}, \quad l = 0,\dots,J$$

The algorithm converges if (Kushner Clark theorem):

$$E\left[\left\{ y_k - \sum_{j=0}^{J} w_j(k) x_{k-j} \right\} y_{k-l} \right] = 0, \quad l = 0,\dots,J$$

Applications: channel equalization (15)

$$E\left[y_k y_{k-l}\right]-\sum_{j=0}^{J} w_j E\left[x_{k-j} y_{k-l}\right]=0$$

$$\sum_{j=0}^{J} w_j E\left[x_{k-j} y_{k-l}\right]=\underbrace{E\left[y_k y_{k-l}\right]}_{\delta_{l,0}}$$

$$E\left[x_{k-j} y_{k-l}\right]=E\left[\left(\sum_n h_{k-j-n} y_n+\nu_{k-j}\right) y_{k-l}\right]=$$

$$=\sum_n h_{k-j-n} \underbrace{E\left[y_n y_{k-l}\right]}_{\delta_{n,k-l}}+\underbrace{E\left[\nu_{k-j} y_{k-l}\right]}_{0}=$$

$$=\sum_n h_{k-j-n} \delta_{n,k-l}=h_{l-j}$$

$$\sum_{j=0}^{J} w_j h_{l-j}=\delta_{l,0}$$

$$\mathbf{H}\mathbf{w}=\boldsymbol{\delta}$$

The algorithm converges to the ZF solution: adaptive ZF!

Applications: channel equalization (16)

Problem: Zero Forcing eliminates ISI, but enhances the effect of additive noise

$$\hat{y}_k = \sum_n q_{k-n} y_n + \boxed{\eta_k}$$

Energy of the noise component:

$$E\left[\eta_k^2\right] = E\left[\left(\sum_{j=0}^{J} w_j v_{k-j}\right)^2\right] = E\left[\sum_{j=0}^{J}\sum_{i=0}^{J} w_j w_i v_{k-j} v_{k-i}\right] =$$

$$= \sum_{j=0}^{J}\sum_{i=0}^{J} w_j w_i \underbrace{E\left[v_{k-j} v_{k-i}\right]}_{N_0 \delta_{i,j}} = \sum_{j=0}^{J} N_0 w_j^2 = N_0 \left\|\mathbf{w}\right\|^2$$

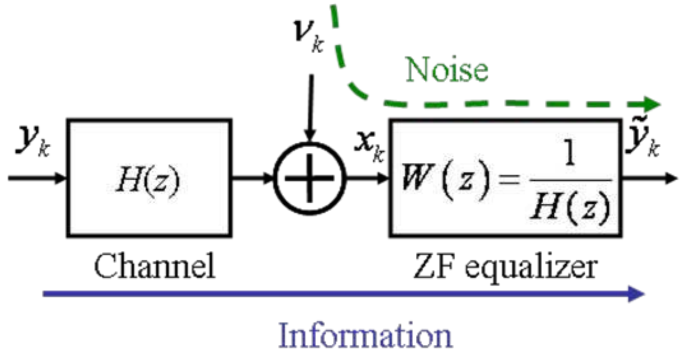
Applications: channel equalization (17)

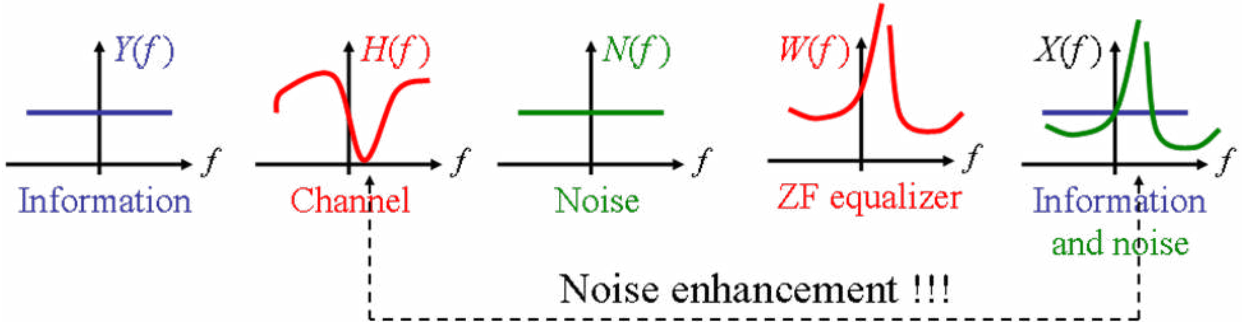
What happens in low SNR domain ?

$$\begin{tikzcd}
 & \left\|\mathbf{Hw}\right\|^2 = \left\|\boldsymbol{\delta}\right\|^2 = 1 \arrow[dl] \arrow[dr] & \\
\left\|\mathbf{H}\right\|^2 \left\|\mathbf{w}\right\|^2 \ge \left\|\mathbf{Hw}\right\|^2 = 1 \arrow[d] & & \left\|\mathbf{w}\right\|^2 = \left\|\mathbf{H}^{-1}\boldsymbol{\delta}\right\|^2 \arrow[d] \\
\left\|\mathbf{w}\right\|^2 \ge \frac{1}{\left\|\mathbf{H}\right\|^2} = \frac{1}{\lambda_{\mathrm{max}}} \arrow[dr] & & \left\|\mathbf{w}\right\|^2 \le \left\|\mathbf{H}^{-1}\right\|^2 \left\|\boldsymbol{\delta}\right\|^2 = \frac{1}{\lambda_{\mathrm{min}}} \arrow[dl] \\
 & \frac{N_0}{\lambda_{\mathrm{max}}} \le E\left[\eta_k^2\right] \le \frac{N_0}{\lambda_{\mathrm{min}}} &
\end{tikzcd}$$

Noise energy can be increase too much in case of bad channel!

Applications: channel equalization (18)





Applications: channel equalization (19)

Minimum Mean Square Error: Adaptive solution of Wiener-filtering!

$$\mathbf{w}_{\text{opt}} : \min_{\mathbf{w}} \mathrm{E}\left[\left(y_k - \hat{y}_k\right)^2\right] \sim \min_{\mathbf{w}} \mathrm{E}\left[\left(y_k - \sum_{j=0}^{J} w_j y_{k-j}\right)^2\right]$$

$$\mathbf{w}_{\text{opt}} : \mathbf{Rw} = \mathbf{r},\ R_{ij} = \mathrm{E}\left[x_{k-i} x_{k-j}\right],\ r_i = \mathrm{E}\left[y_k x_{k-i}\right]$$

Applying the Robins-Monroe algorithm:

$$w_l(k+1) = w_l(k) - \Delta(k)\left\{y_k - \sum_{j=0}^{J} w_j(k) x_{k-j}\right\} x_{k-l},\quad l = 0,\dots,J$$

$$\tau^{(K)} = \left\{\left(y_k, x_k\right), k = 1,\dots,K\right\}$$

Applications: channel equalization (20)

The correlation matrix:

$$R_{ij}=\mathrm{E}\left[x_{k-i}x_{k-j}\right]=\mathrm{E}\left[\left(\sum_n h_{k-i-n}y_n+v_{k-i}\right)\left(\sum_m h_{k-j-m}y_m+v_{k-j}\right)\right]=$$

$$=\mathrm{E}\left[\sum_n\sum_m h_{k-i-n}h_{k-j-m}y_ny_m+v_{k-i}v_{k-j}\right]=$$

$$=\sum_n\sum_m h_{k-i-n}h_{k-j-m}\underbrace{\mathrm{E}\left[y_ny_m\right]}_{\delta_{n,m}}+\underbrace{\mathrm{E}\left[v_{k-i}v_{k-j}\right]}_{N_0\delta_{i,j}}=$$

$$=\sum_n h_{k-i-n}h_{k-j-n}\delta_{n,m}+N_0\delta_{i,j}\rightarrow\mathbf{R}=\mathbf{H}\mathbf{H}^T+N_0\mathbf{I}$$

What about the noise in case of MMSE?

Applications: channel equalization (21)

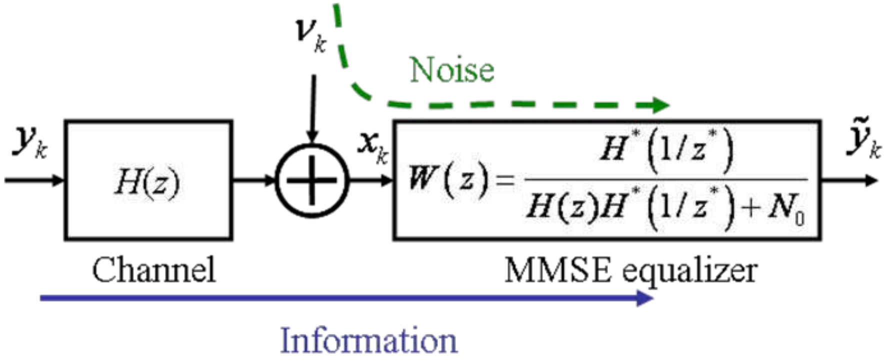
What happens in low SNR domain ?

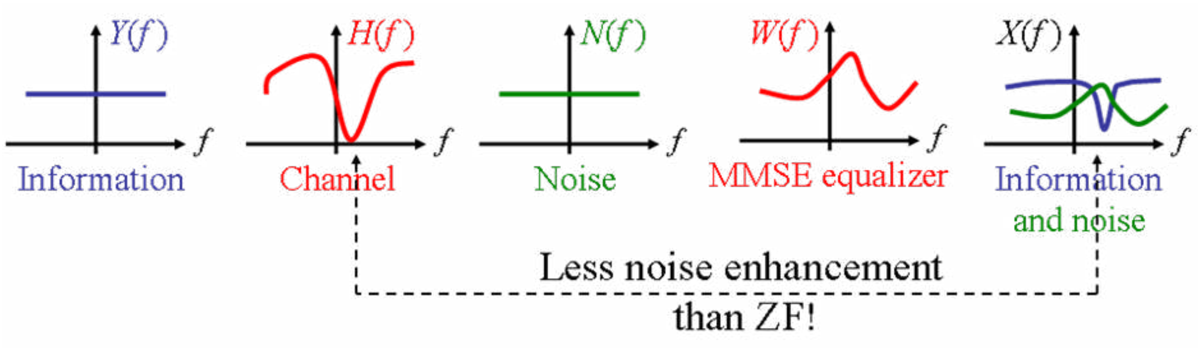
Eigenvectors: $$\mathbf{H} \rightarrow \lambda_i \Rightarrow \mathbf{R} = \mathbf{H}\mathbf{H}^T + N_0\mathbf{I} \rightarrow \lambda_i^2 + N_0$$

Noise energy: $$\frac{N_0}{\lambda_{\max}^2 + N_0} \le E\left[\eta_k^2\right] \le \frac{N_0}{\lambda_{\min}^2 + N_0}$$

The noise can never became infinitely large because of the N_0 component in the denominator !

Applications: channel equalization (22)



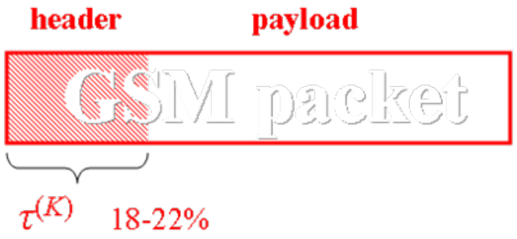


Applications: channel equalization (23)

Adaptive version of MMSE: $\tau^{(K)} = \left\{ \left(y_k , x_k \right), k = 1,\dots, K \right\}$

$$w_l(k+1) = w_l(k) - \Delta\left(k\right) \left\{ y_k - \sum_{j=0}^{J} w_j(k) x_{k-j} \right\} x_{k-l}, \quad l = 0,\dots, J$$

Problem: Channel condition is changing in time, therefore a lot of overhead is needed because of the learning set. E.g.: In GSM system 18% of the packet is this overhead!



Applications: channel equalization (24)

A solution without learning set (Blind signal processing):

$$w_l(k+1) = w_l(k) - \Delta\left(k\right)\left\{\hat{y}_k - \sum_{j=0}^{J} w_j(k)x_{k-j}\right\}x_{k-l}, \quad l = 0,\dots,J$$

Is any guarantee for convergence? If there is,

$$E\left[\left\{\hat{y}_k - \sum_{j=0}^{J} w_j(k)x_{k-j}\right\}x_{k-l}\right] = 0$$

$$E\left[\hat{y}_k x_{k-l}\right] = \sum_{j=0}^{J} w_j(k)E\left[x_{k-j}x_{k-l}\right]$$

Applications: channel equalization (25)

$$E\left[y_k x_{k-l} \middle| \hat{y}_k = y_k\right](1-P_b)+E\left[-y_k x_{k-l} \middle| \hat{y}_k \neq y_k\right]P_b=\sum_{j=0}^{J} R_{lj} w_j$$

$$(1-2P_b)r_l=\sum_{j=0}^{J} R_{lj} w_j$$

$$\boxed{\mathbf{Rw}=(1-2P_b)\mathbf{r}}$$

which is not the original Wiener solution!

Furthermore the probability of bit error is not constant, it depends on the filter:

$$P_b=\Psi(\mathbf{w})$$

Applications: channel equalization (26)

The service provider is interested in the probability of bit error, because it is the real QoS subject which should be minimized:

$$P_b = \Psi(\mathbf{w}) \rightarrow \mathbf{w}_{\text{opt}} = \min_{\mathbf{w}} \Psi(\mathbf{w})$$

What is this $\Psi(\mathbf{w})$ function?

Recall:

$$\underbrace{x_k}_{\text{received distorted signal}} = \sum_{n=0}^{L} \underbrace{h_{k-n}}_{\text{channel distorsion}} \underbrace{y_n}_{\text{sent symbols}} + \underbrace{\nu_k}_{\text{WGN noise}}$$

$$\tilde{y}_k = \sum_{j=0}^{J} w_j x_{k-j} = \sum_{j=0}^{J} w_j \left(\sum_{n=0}^{L} h_{k-j-n} y_n + \nu_{k-j} \right)$$

$$\tilde{y}_k = \sum_{n=0}^{L} \left(\sum_{j=0}^{J} w_j h_{k-j-n} \right) y_n + \sum_{j=0}^{J} w_j \nu_{k-j} = \sum_{n}^{L} q_{k-n} y_n + \eta_k$$

Applications: channel equalization (27)

$$\hat{y}_k = \operatorname{sgn}\left(\tilde{y}_k\right)$$

$$\Psi\left(\mathbf{q}\right) = P\left(\hat{y}_k \neq y_k\right) = P\left(\hat{y}_k = +1 \middle| y_k = -1\right)\frac{1}{2} + P\left(\hat{y}_k = -1 \middle| y_k = +1\right)\frac{1}{2} =$$

$$= P\left(\hat{y}_k > 0 \middle| y_k = -1\right)\frac{1}{2} + P\left(\hat{y}_k < 0 \middle| y_k = +1\right)\frac{1}{2} =$$

$$= \frac{1}{2}\left[P\left(-q_0 + \sum_{n\neq k} q_{k-n} y_n + \eta_k > 0\right) + P\left(q_0 + \sum_{n\neq k} q_{k-n} y_n + \eta_k < 0\right)\right]$$

$\sup\left(\mathbf{q}\right) = L + J + 1 = M$, $\mathbf{y}$ is a random binary vector

Applications: channel equalization (28)

$$P(\mathbf{y}=\mathbf{z})=\frac{1}{2^{L+J}}=\frac{1}{2^{M-1}}$$

$$P_b=\frac{1}{2^{M-1}}\frac{1}{2}\sum_{\mathbf{z}\in\{-1,1\}^{M-1}}\left[P\left(-q_0+\sum_{n\neq k}q_{k-n}z_n+\eta_k>0\right)+P\left(q_0+\sum_{n\neq k}q_{k-n}z_n+\eta_k<0\right)\right]$$

$$\eta_k\sim N\left(0,\sigma_\eta\right),\ \sigma_\eta=N_0\left\|\mathbf{w}\right\|^2$$

$$P_b=\frac{1}{2^{M}}\sum_{\mathbf{z}\in\{-1,1\}^{M-1}}\left[P\left(\eta_k>q_0-\sum_{n\neq k}q_{k-n}z_n\right)+P\left(\eta_k<-q_0-\sum_{n\neq k}q_{k-n}z_n\right)\right]=$$

Applications: channel equalization (29)

$$P_b = \frac{1}{2^M} \sum_{\mathbf{z} \in \{-1,1\}^{M-1}} \left[1 - \Phi\left(\frac{q_0 - \sum_{n \neq k} q_{k-n} z_n}{\sigma_\eta} \right) + \Phi\left(\frac{-q_0 - \sum_{n \neq k} q_{k-n} z_n}{\sigma_\eta} \right) \right]$$

Here Φ(.) denotes the standard normal cdf. and note that

$$\Phi(u) = 1 - \Phi(-u)$$

$$P_b = \frac{1}{2^M} \sum_{\mathbf{z} \in \{-1,1\}^{M-1}} \left[\Phi\left(\frac{-q_0 + \sum_{n \neq k} q_{k-n} z_n}{\sigma_\eta} \right) + \Phi\left(\frac{-q_0 - \sum_{n \neq k} q_{k-n} z_n}{\sigma_\eta} \right) \right]$$

Applications: channel equalization (30)

$$P_b = \frac{1}{2^M} \sum_{\mathbf{z}\in\{-1,1\}^{M-1}} \left[\Phi\left(\frac{-q_0 + \sum_{n\neq k} q_{k-n} z_n}{\sigma_\eta} \right) + \Phi\left(\frac{-q_0 - \sum_{n\neq k} q_{k-n} z_n}{\sigma_\eta} \right) \right] =$$

$$= \frac{1}{2^M} \sum_{\mathbf{z}\in\{-1,1\}^{M-1}} \left[\Phi\left(\frac{-\sum_j w_j h_j + \sum_{n\neq k} \sum_j w_j h_{k-j-n} z_n}{\sqrt{N_0 \left\|\mathbf{w}\right\|^2}} \right) + \right.$$

$$\left. + \Phi\left(\frac{-\sum_j w_j h_j - \sum_{n\neq k} \sum_j w_j h_{k-j-n} z_n}{\sqrt{N_0 \left\|\mathbf{w}\right\|^2}} \right) \right] = \Psi\left(\mathbf{w}\right)$$

Applications: channel equalization (31)

The new goal function:

$$\mathbf{w}_{\text{opt}} = \min_{\mathbf{w}} \Psi(\mathbf{w})$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta \operatorname{grad}_{\mathbf{w}} \Psi(\mathbf{w}(k))$$

Problems:

1. Channel information is needed!
2. Algorithm is too complex: every step needs O($2^{M\text{-}1}$) evaluation!

Applications: channel equalization (32)

Using Monte Carlo methods can be a solution for reducing complexity.

Using upper and lower bounds can be an other solution for the problem of complexity.

Lemma: If $a < b < c$, then

$$\Phi\left(\frac{-c+b}{\sigma}\right)-\Phi\left(\frac{-c+a}{\sigma}\right)>\Phi\left(\frac{-c-a}{\sigma}\right)-\Phi\left(\frac{-c-b}{\sigma}\right)$$

Applications: channel equalization (33)

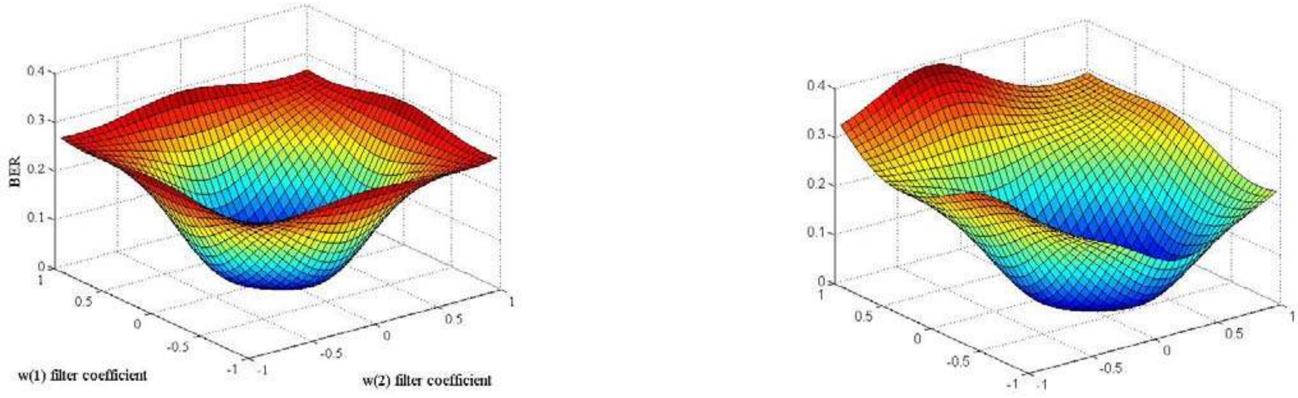
$$F(\mathbf{w}) = \frac{1}{2}\left[\Phi\left(\frac{-\sum_j w_j h_j + \sum_{n\neq k}\sum_j w_j h_{k-j-n} z_n}{\sqrt{N_0 \|\mathbf{w}\|^2}} \right) + \Phi\left(\frac{-\sum_j w_j h_j q_0 + \sum_{n\neq k}\sum_j w_j h_{k-j-n} z_n}{\sqrt{N_0 \|\mathbf{w}\|^2}} \right) \right]$$

$$P_b = \Psi(\mathbf{w}) \leq F(\mathbf{w})$$

The modified goal function: $\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta \underset{\mathbf{w}}{\mathrm{grad}}\, F(\mathbf{w}(k))$

No more complexity problem!

Simulation Results(1)



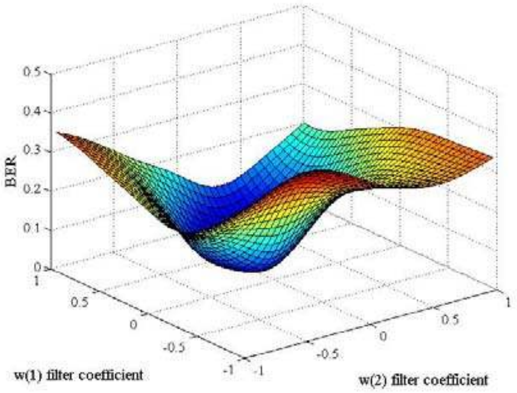
No ISI

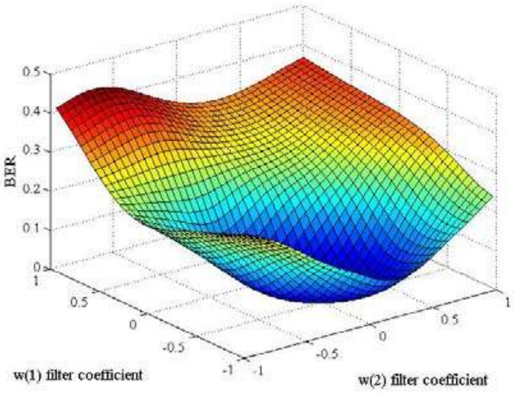
h=(1, 0.3)

$$\mathbf{w}_{\text{opt}}^{(\text{MMSE})} = \begin{bmatrix} 0.9018 & -0.2440 & 0.0660 \end{bmatrix}$$

Surfaces of error probability

Simulation Results(2)





h=(1, -0.5, 0.1)

$$\mathbf{w}_{\text{opt}}^{(\text{MMSE})} = \begin{bmatrix} 0.8895 & 0.3996 & 0.1004 \end{bmatrix}$$

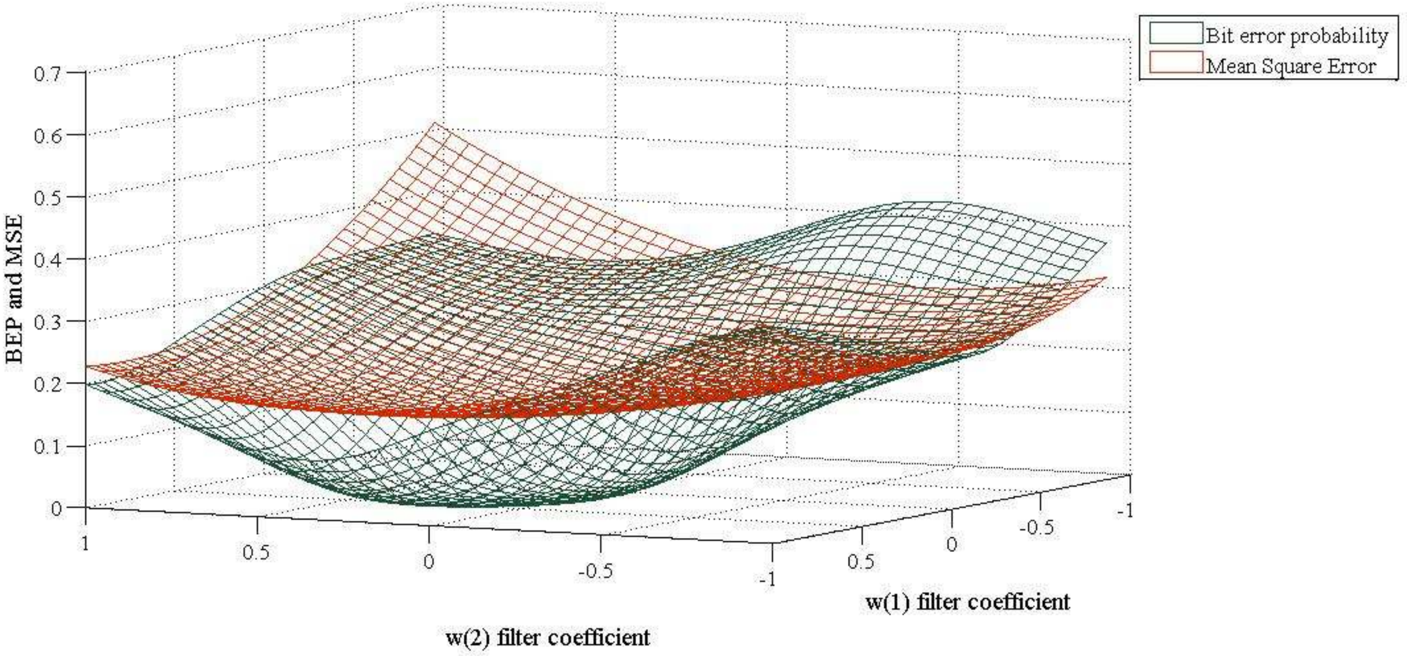
h=(1, 0.5, 0.3)

$$\mathbf{w}_{\text{opt}}^{(\text{MMSE})} = \begin{bmatrix} 0.888 & -0.4047 & -0.0521 \end{bmatrix}$$

Surfaces of error probability

Simulation Results(3)

h=(1, -0.5, 0.1)

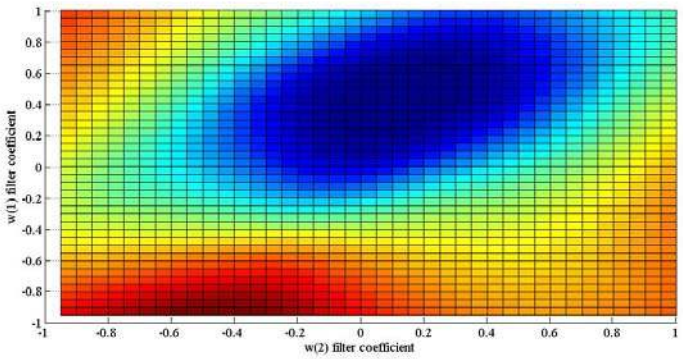


Bit error probability vs. Mean square error

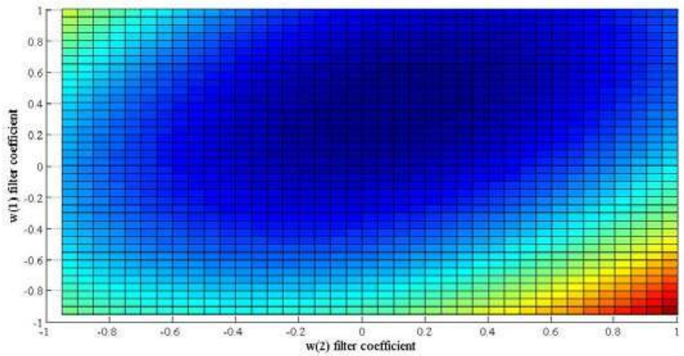
Simulation Results(4)

h=(1, -0.5, 0.1)

Bit error probability



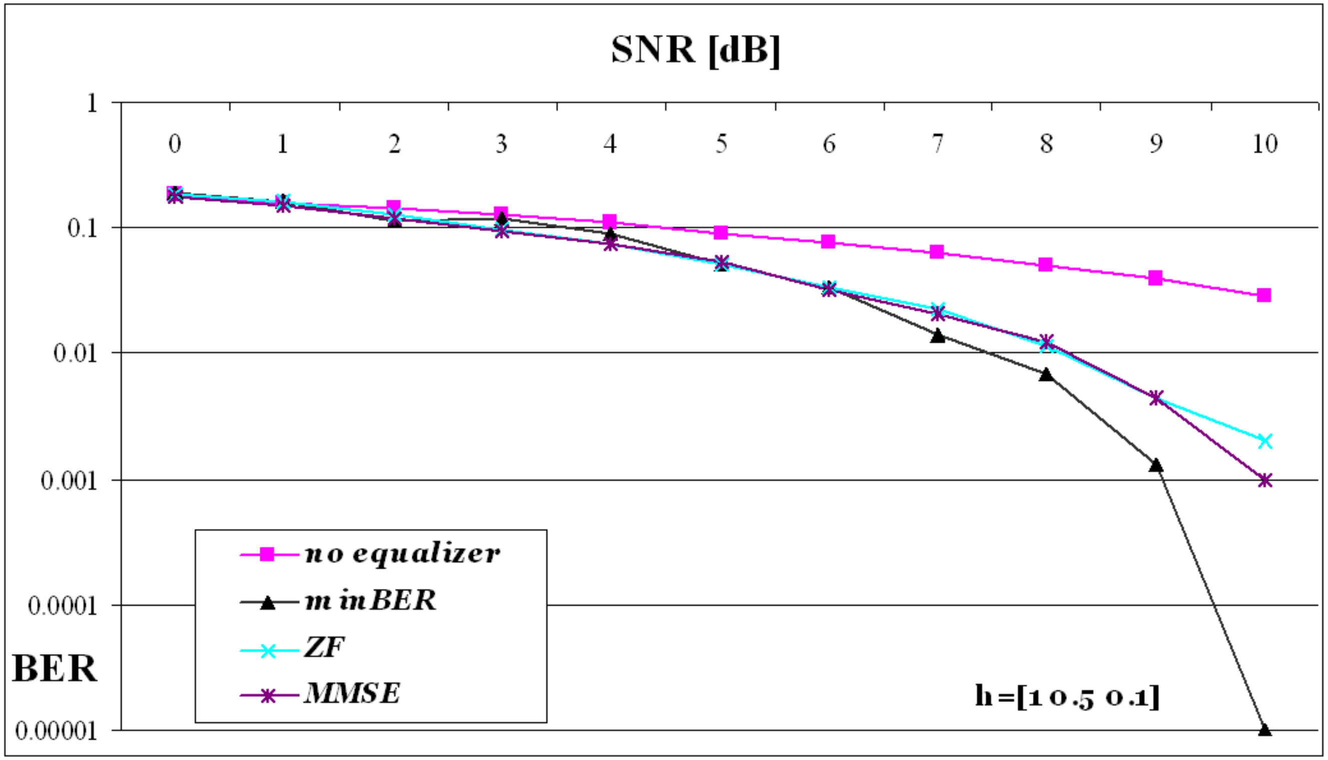
Mean square error



Local minimum points are possible!

Bit error probability vs. Mean square error

Simulation Results(5)



BER vs. SNR

Questions(1)

1. What is Wiener filtering ?
2. What error function the Wiener filter will minimize ?
3. What are the properties of the correlation matrix ?
4. Give two practical examples for the use of Wiener filters ?
5. Summarize the problem of adaptive radio channel equalization! Give the optimal solution!
6. Summarize the problem of first order adaptive-predictive coding! Give the Wiener solution! Show that the solution increases the speed of data transmission!

Questions(2)

7. What is the problem of adaptive-predictive coding in mobile communication systems? Give the block diagram of the solution, and explain the notations! Give the Robinson-Monroe algorithm and explain the formula!
8. What kind of optimality is guaranteed solving the Wiener-Hopf equation? Prove it!
9. What is the Zero Forcing solution in the case of channel equalization? What happens with the noise?

Investing in your future
New Hungary Development Plan

An example of Wiener filtering (1)

Problem:

How can we estimate the correlation matrix **R** based observation? Give an example of the correlation matrix **R** if the dimension is 4! Based on the correlation matrix find the optimal *Δ* parameter which guarantees the fastest convergence!

Solution:

By definition, the correlation matrix of a stochastic process is

$$R_{ij}=\mathrm{E}\{x_i x_j\},$$

which can be estimated using the observations with

$$\tilde{R}_{ij}=\frac{1}{K-\max(i,j)}\sum_{m=0}^{K-\max(i,j)} x_{i+m}x_{j+m}.$$

An example of Wiener filtering (2)

Solution (continued): Let us take the correlation matrix $\mathbf{R}^{(b)}$ from example 1! The *Δ* parameter which guarantees the fastest convergence can be obtained from the eigenvalues of **R**. The eigenvalues of **R** are

$$\lambda_1 = 0.191,\ \lambda_2 = 0.691,\ \lambda_3 = 1.309,\ \ \lambda_4 = 1.809$$

from which the optimal *Δ* is:

$$\Delta_{opt} = \frac{2}{\lambda_{\min} + \lambda_{\max}} = \frac{2}{0.191 + 1.809} = 1.$$

Problems (1)

1. Define R_{ij} element of the correlation matrix!
2. Are any of the following matrices a valid correlation matrix?:

$$\mathbf{R}^{(\mathbf{a})}=\begin{bmatrix}1 & 0.6 & 0.4 & 0.1\\ 0.6 & 1 & 0.2 & 0.4\\ 0.4 & 0.1 & 1 & 0.6\\ 0.2 & 0.4 & 0.6 & 1\end{bmatrix},\mathbf{R}^{(b)}=\begin{bmatrix}1 & 0.5 & 0 & 0\\ 0.5 & 1 & 0.5 & 0\\ 0 & 0.5 & 1 & 0.5\\ 0 & 0 & 0.5 & 1\end{bmatrix}$$

Problems (2)

The channel distortion in a wireless communication system is to be equalized by an FIR filter of third degree. We use the LMS algorithm to set the equalizer coefficients with parameter Δ=1. The learning set is given as follows:

$$\tau^{(5)} = \{(1,0.3);(-1,-0.5);(-1,0.1);(1,1.1);(1,0.9)\}$$

While the initial vector is

$$\mathbf{w}(0) = [1 \quad 0.1 \quad 0.1]^{\mathrm{T}}$$

Give the filter coefficient vector after the first update cycle !

Problems (3)

A random process is to be compressed by a predictor of second degree. The correlation function is given as follows:

$$R(0) = 1 \quad R(1) = 0.5 \quad R(2) = 0.2$$

Calculate the optimal predictor coefficients !

Summary

- Fundamental issues: FIR Wiener-filter, applications of adaptive signal processing: adaptive-predictive coding, channel equalization.
- Problem of signal estimation in the presence of undesired noise or interference can be solved by Wiener filter.
- Adaptivity means real-time re-optimization possibility.
- Theory of Wiener filters can be applied to solve IT problems such as coding or radio channel equalization.

Next lecture: *Introduction to neural processing*