Digital Signal Processing 2

Les 3: Optimale filtering

Prof. dr. ir. Toon van Waterschoot

Faculteit Industriële Ingenieurswetenschappen
ESAT – Departement Elektrotechniek
KU Leuven, Belgium

Digital Signal Processing 2: Vakinhoud

- Les 1: Eindige woordlengte
- Les 2: Lineaire predictie
- Les 3: Optimale filtering
- Les 4: Adaptieve filtering
- Les 5: Detectieproblemen
- Les 6: Spectrale signaalanalyse
- Les 7: Schattingsproblemen 1
- Les 8: Schattingsproblemen 2
- Les 9: Sigma-Deltamodulatie
- Les 10: Transformatiecodering

Les 3: Optimale filtering

- **Introduction**

- **Least-squares and Wiener filter estimation**

  stochastic & deterministic estimation, computational aspects, geometrical interpretation, performance analysis, frequency domain formulation, …

- **Wiener filtering applications**

  noise reduction, time alignment of multi-channel/-sensor signals, channel equalization, …

- **Wiener filter implementation**

  filter order, filter-bank implementation, …

KU LEUVEN

Les 3: Optimale filtering

- **Introduction**
- **Least-squares and Wiener filter estimation**

  S. V. Vaseghi, *Multimedia Signal Processing*
  - Ch. 8, “Least Square Error, Wiener-Kolmogorov Filters”
    - Section 8.1, “LSE Estimation: Wiener-Kolmogorov Filter”
    - Section 8.2, “Block-Data Formulation of the WF”
    - Section 8.3, “Interpretation of WF as Projection in Vector Space”
    - Section 8.4, “Analysis of the Least Mean Square Error Signal”
    - Section 8.5, “Formulation of WFs in the Frequency Domain”
- **Wiener filtering applications**
    - Section 8.6, “Some Applications of Wiener Filters”
- **Wiener filter implementation**
    - Section 8.7, “Implementation of Wiener Filters”

KU LEUVEN

Les 3: Optimale filtering

- **Introduction**

- **Least-squares and Wiener filter estimation**

  stochastic & deterministic estimation, computational aspects, geometrical interpretation, performance analysis, frequency domain formulation, …

- **Wiener filtering applications**

  noise reduction, time alignment of multi-channel/-sensor signals, channel equalization, …

- **Wiener filter implementation**

  filter order, filter-bank implementation, …

KU LEUVEN

Introduction

- **Optimal filters**
  - data-dependent filters
  - designed such as to minimize “difference” between filter output signal and desired or target signal
  - many applications: linear prediction, echo cancellation, signal restoration, channel equalization, radar, system identification
- **Wiener filters**
  - filters for signal prediction or signal/parameter estimation
  - optimal for removing effect of linear distortion (filtering) and/or additive noise from observed data
  - many flavors: FIR/IIR, single-/multi-channel, time-/frequency-domain, fixed/block-adaptive/adaptive

Les 3: Optimale filtering

- **Introduction**

- **Least-squares and Wiener filter estimation**

  stochastic & deterministic estimation, computational aspects, geometrical interpretation, performance analysis, frequency domain formulation, …

- **Wiener filtering applications**

  noise reduction, time alignment of multi-channel/-sensor signals, channel equalization, …

- **Wiener filter implementation**

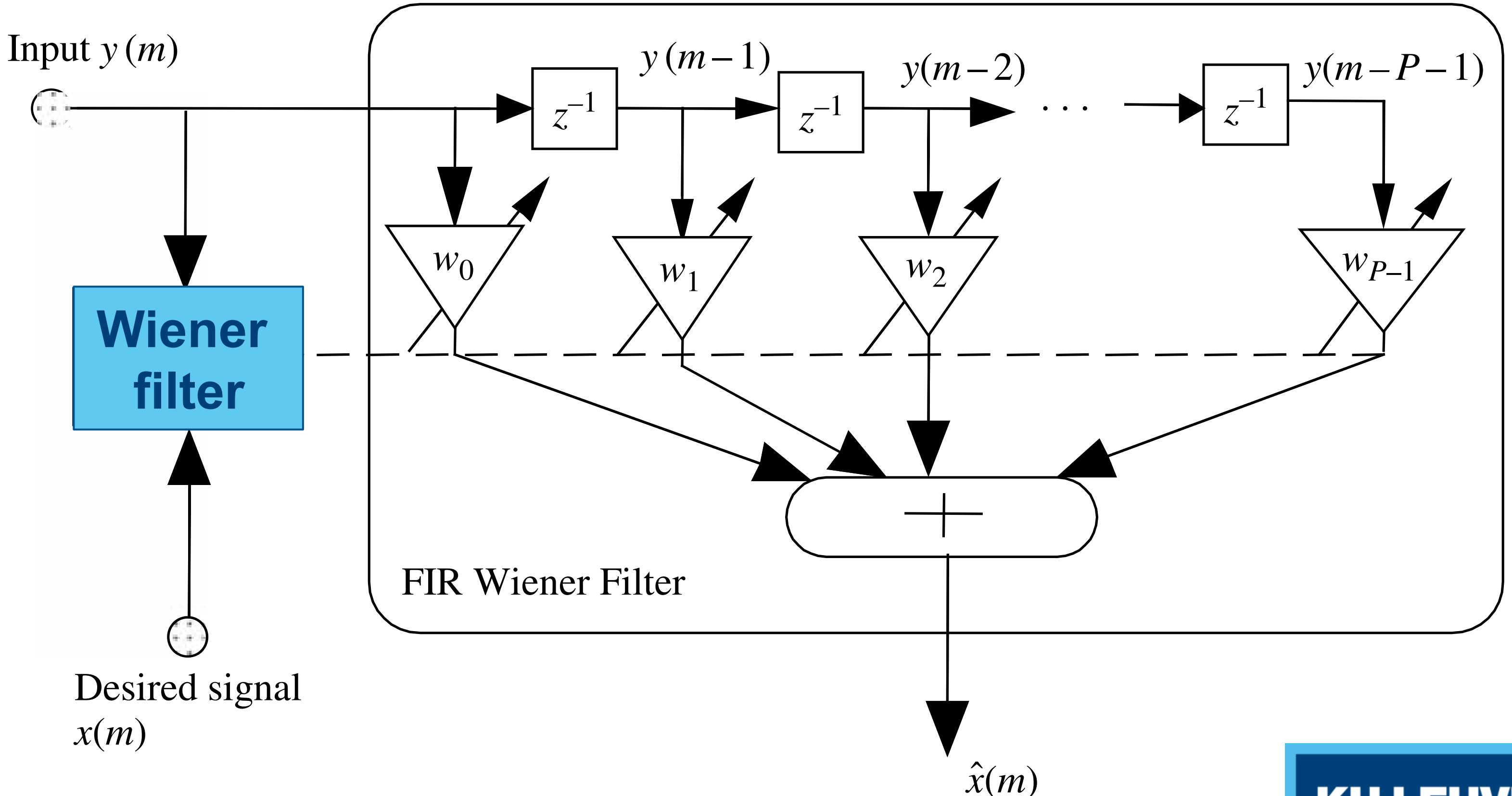
  filter order, filter-bank implementation, …

Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

Stochastic Wiener filter estimation (1)

- **FIR Wiener filter:**
  - signal flow graph



KU LEUVEN

Stochastic Wiener filter estimation (2)

- **FIR Wiener filter:**
  - input signal = noisy or distorted observed data

$$\mathbf{y} = \begin{bmatrix} y(0) & y(1) & \ldots & y(N-1) \end{bmatrix}^T$$

  - desired signal = (unknown) clean data

$$\mathbf{x} = \begin{bmatrix} x(0) & x(1) & \ldots & x(N-1) \end{bmatrix}^T$$

  - Wiener filter coefficients

$$\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \ldots & w_{P-1} \end{bmatrix}^T$$

  - input-output relation

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k y(m-k) = \mathbf{w}^T \mathbf{y}$$

KU LEUVEN

Stochastic Wiener filter estimation (3)

- **Wiener filter error signal:**
  - error signal = desired signal – Wiener filter output signal

$$e(m) = x(m) - \hat{x}(m) = x(m) - \mathbf{w}^T\mathbf{y}$$

  - stacking error signal samples for $m = 0, \ldots, N-1$ yields

$$\begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(N-1) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} - \begin{bmatrix} y(0) & y(-1) & \ldots & y(1-P) \\ y(1) & y(0) & \ldots & y(2-P) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & \ldots & y(N-P) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{P-1} \end{bmatrix}$$

$$\mathbf{e} = \mathbf{x} - \mathbf{Y}\mathbf{w}$$

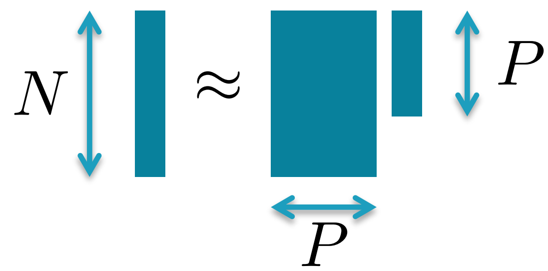
  - initial conditions $y(1-P), \ldots, y(-1)$ are known or assumed zero (cf. Les 2: autocorrelation vs. covariance method)

KU LEUVEN

Stochastic Wiener filter estimation (4)

- **Number of solutions**
  - Wiener filter is optimal filter in sense of minimizing mean squared error signal

$$\mathbf{e} \approx 0 \Rightarrow \mathbf{x} \approx \mathbf{Y}\mathbf{w}$$



  - 3 different cases, depending on no. observations *N* and Wiener filter length *P* (cf. Les 2: linear systems of equations)

$N = P$	**square** system, unique solution with $\mathbf{e} = 0$
$N < P$	**underdetermined** system, ∞ solutions with $\mathbf{e} = 0$
$N > P$	**overdetermined** system, no solutions with $\mathbf{e} = 0$, unique solution with "minimal" $\mathbf{e} \neq 0$

Stochastic Wiener filter estimation (5)

- **Wiener filter estimation:**
  - mean squared error (MSE) criterion

$$
\begin{aligned}
E\{e^2(m)\} &= E\{(x(m) - \mathbf{w}^T\mathbf{y})^2\} \\
&= E\{x^2(m)\} - 2\mathbf{w}^T E\{\mathbf{y}x(m)\} + \mathbf{w}^T E\{\mathbf{y}\mathbf{y}^T\}\mathbf{w} \\
&= r_{xx}(0) - 2\mathbf{w}^T\mathbf{r}_{\mathbf{yx}} + \mathbf{w}^T\mathbf{R}_{\mathbf{yy}}\mathbf{w}
\end{aligned}
$$

  - autocorrelation matrix & cross-correlation vector definition:

$$
\mathbf{R}_{\mathbf{yy}} = \begin{bmatrix} r_{yy}(0) & r_{yy}(1) & \ldots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & \ldots & r_{yy}(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & \ldots & r_{yy}(0) \end{bmatrix} = E\{\mathbf{y}\mathbf{y}^T\}
$$

$$
\mathbf{r}_{\mathbf{yx}} = \begin{bmatrix} r_{yx}(0) & r_{yx}(1) & \ldots & r_{yx}(P-1) \end{bmatrix}^T = E\{\mathbf{y}x(m)\}
$$

KU LEUVEN

Stochastic Wiener filter estimation (6)

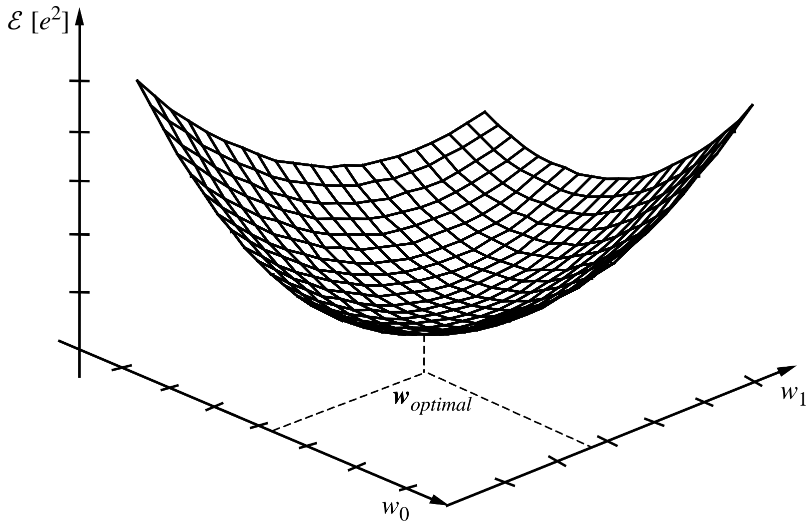
- **Wiener filter estimation:**
  - mean squared error (MSE) criterion

$$E\{e^2(m)\} = r_{xx}(0) - 2\mathbf{w}^T\mathbf{r_{yx}} + \mathbf{w}^T\mathbf{R_{yy}}\mathbf{w}$$

= quadratic function of Wiener filter coefficient vector **w**

  - quadratic function (with full-rank Hessian matrix $\mathbf{R_{yy}}$) is always convex and has unique minimum
  - example:

$$P = 2$$



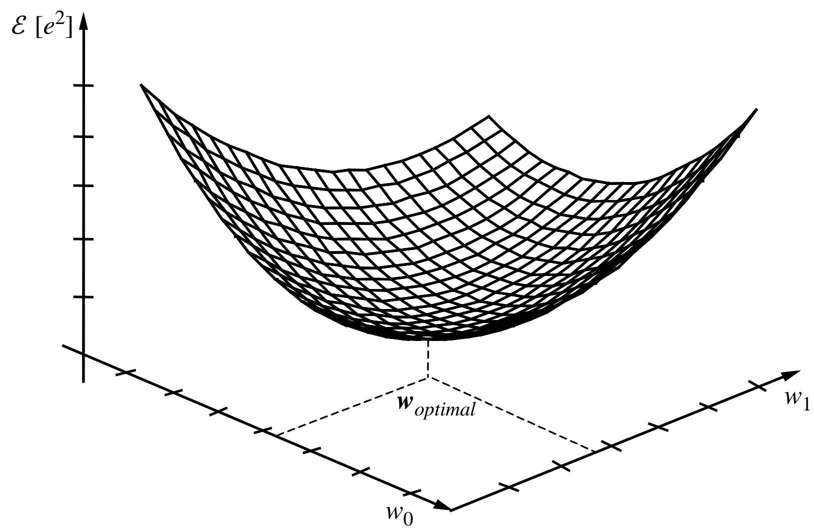
Stochastic Wiener filter estimation (7)

- **Wiener filter estimation:**
  - minimum MSE solution is obtained at point with zero gradient
  - gradient of MSE criterion w.r.t. Wiener filter coefficient vector

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} E\{e^2(m)\} &= \frac{\partial}{\partial \mathbf{w}}\left[r_{xx}(0) - 2\mathbf{w}^T \mathbf{r_{yx}} + \mathbf{w}^T \mathbf{R_{yy}} \mathbf{w}\right] \\ &= -2\mathbf{r_{yx}} + 2\mathbf{R_{yy}}\mathbf{w}\end{aligned}$$

  - example:

$$P = 2$$



Stochastic Wiener filter estimation (8)

- **Wiener filter estimation:**
  - minimum MSE solution is obtained at point with zero gradient

$$\frac{\partial}{\partial \mathbf{w}} E\{e^2(m)\} = 0 \Rightarrow \mathbf{r_{yx}} = \mathbf{R_{yy}} \mathbf{w}$$

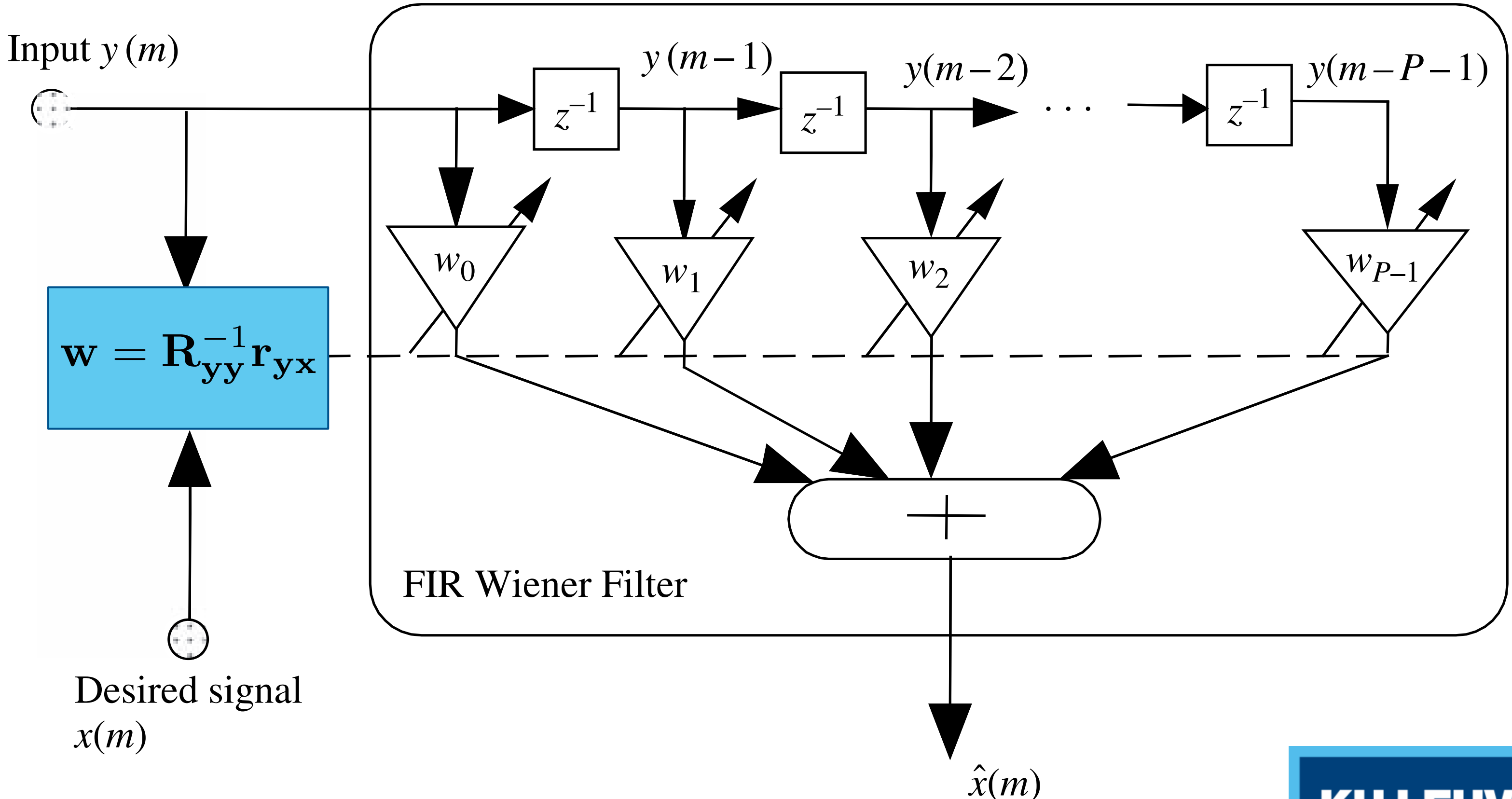
  - minimum MSE Wiener filter estimate:

$$\boxed{\mathbf{w} = \mathbf{R_{yy}^{-1}} \mathbf{r_{yx}}}$$

$$\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{P-1} \end{bmatrix} = \begin{bmatrix} r_{yy}(0) & r_{yy}(1) & \dots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & \dots & r_{yy}(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & \dots & r_{yy}(0) \end{bmatrix}^{-1} \begin{bmatrix} r_{yx}(0) \\ r_{yx}(1) \\ \vdots \\ r_{yx}(P-1) \end{bmatrix}$$

Stochastic Wiener filter estimation (9)

- **FIR Wiener filter:**
  - signal flow graph (revisited)



Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

Deterministic least squares estimation (1)

- **Wiener filter input/output relation**
  - set of *N* linear equations

$$\begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{bmatrix} = \begin{bmatrix} y(0) & y(-1) & \dots & y(1-P) \\ y(1) & y(0) & \dots & y(2-P) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & \dots & y(N-P) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{P-1} \end{bmatrix}$$

$$\hat{\mathbf{x}} = \mathbf{Y}\mathbf{w}$$

  - Wiener filter error signal vector

$$\begin{aligned} \mathbf{e} &= \mathbf{x} - \hat{\mathbf{x}} \\ &= \mathbf{x} - \mathbf{Y}\mathbf{w} \end{aligned}$$

KU LEUVEN

Deterministic least squares estimation (2)

- **Least squares estimation**
  - sum of squared errors criterion

$$\begin{aligned}\sum_{m=0}^{N-1} e^2(m) &= \mathbf{e}^T\mathbf{e} \\ &= (\mathbf{x}-\mathbf{Y}\mathbf{w})^T(\mathbf{x}-\mathbf{Y}\mathbf{w}) \\ &= \mathbf{x}^T\mathbf{x}-\mathbf{x}^T\mathbf{Y}\mathbf{w}-\mathbf{w}^T\mathbf{Y}^T\mathbf{x}+\mathbf{w}^T\mathbf{Y}^T\mathbf{Y}\mathbf{w}\end{aligned}$$

  - difference with MSE criterion: expectation replaced by time averaging
    - mean squared error: $E\{e^2(m)\}$ = **stochastic** criterion
    - sum of squared errors: $\sum_{m=0}^{N-1} e^2(m)$ = **deterministic** criterion

Deterministic least squares estimation (3)

- **Least squares estimation**
  - minimum sum of squared errors is obtained at point with zero gradient

$$\frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \mathbf{w}} = -2\mathbf{Y}^T\mathbf{x} + 2\mathbf{Y}^T\mathbf{Y}\mathbf{w} = 0 \Rightarrow (\mathbf{Y}^T\mathbf{Y})\mathbf{w} = \mathbf{Y}^T\mathbf{x}$$

  - least squares filter estimate:

$$\boxed{\mathbf{w} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}}$$

  - if desired/observed signals are correlation-ergodic processes, least squares estimate converges to Wiener filter estimate

$$\lim_{N\to\infty}\left[\mathbf{w} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}\right] = \mathbf{R}_{\mathbf{yy}}^{-1}\mathbf{r}_{\mathbf{xy}}$$

KU LEUVEN

Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

Computational aspects (1)

- **Calculation of correlation functions**
  - Wiener filter estimate requires autocorrelation matrix $\mathbf{R}_{\mathbf{yy}}$ and cross-correlation vector $\mathbf{r}_{\mathbf{yx}}$
  - correlation functions are obtained by averaging over ensemble of different realizations of desired/observed signals
  - for correlation-ergodic processes, ensemble averaging can be replaced by time averaging so only 1 realization is needed

$$r_{yy}(k) = \frac{1}{N} \sum_{m=0}^{N-1} y(m)y(m+k)$$

$$r_{yx}(k) = \frac{1}{N} \sum_{m=0}^{N-1} y(m)x(m+k)$$

  - choice of *N*: compromise between accuracy and stationarity (cf. Les 2: LP modeling of speech)

KU LEUVEN

Computational aspects (2)

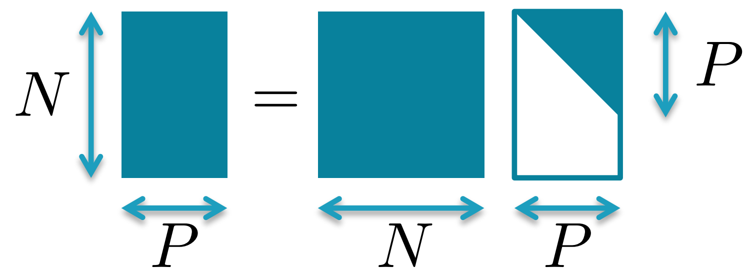
- **Calculation of correlation functions**
  - calculation of cross-correlation vector $\mathbf{r_{yx}}$ is not straightforward if desired signal **x** is unknown
  - two possible solutions:
    - use prior knowledge about **x** to estimate $\mathbf{r_{yx}}$
    - rewrite cross-correlation function in terms of other correlation functions (see later: **Wiener filtering applications**)

KU LEUVEN

Computational aspects (3)

- **Computation of least squares filter estimate**
  - least squares filter estimate: $\mathbf{w} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}$
  - direct matrix inversion has complexity $O(P^3)$
  - QR decomposition of data matrix (Q = orthonormal, R = upper-triangular)

$$\mathbf{Y} = \mathbf{Q}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$



allows to compute least squares filter estimate from a square, triangular system (allowing back-substitution)

$$\mathbf{R}\mathbf{w} = \mathbf{x}_\mathbf{Q}$$

  - QR-based computation of LS estimate has complexity $O(P^2)$ (exploiting Toeplitz structure of data matrix)

Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

KU LEUVEN

Geometrical interpretation (1)

- **Wiener filter input/output relation**
  - input/output relation $\hat{\mathbf{x}} = \mathbf{Y}\mathbf{w}$ can also be written as

$$\begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{bmatrix} = w_0 \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} + w_1 \begin{bmatrix} y(-1) \\ y(0) \\ \vdots \\ y(N-2) \end{bmatrix} + \ldots + w_{P-1} \begin{bmatrix} y(1-P) \\ y(2-P) \\ \vdots \\ y(N-P) \end{bmatrix}$$

$$\hat{\mathbf{x}} = w_0 \mathbf{y_0} + w_1 \mathbf{y_1} + \ldots + w_{P-1} \mathbf{y_{P-1}}$$

  - Wiener filter output signal = linear weighted combination of input signal vectors

    (cf. Les 2: two interpretations of matrix-vector product)

KU LEUVEN

Geometrical interpretation (2)

- **Vector space interpretation**
  - set of P input signal vectors $\{\mathbf{y_0}, \mathbf{y_1}, \ldots, \mathbf{y_{P-1}}\}$ forms P-dimensional subspace of N-dimensional vector space
  - Wiener filter output signal lies in this subspace, since

$$\hat{\mathbf{x}} = w_0\mathbf{y_0} + w_1\mathbf{y_1} + \ldots + w_{P-1}\mathbf{y_{P-1}}$$

$N = P$ **subspace = entire space**, including desired signal

$$\Rightarrow \hat{\mathbf{x}} = \mathbf{x}, \;\; \mathbf{e} = 0$$

$N > P$ **subspace ⊂ entire space**,

output signal is orthogonal projection of desired signal vector onto subspace

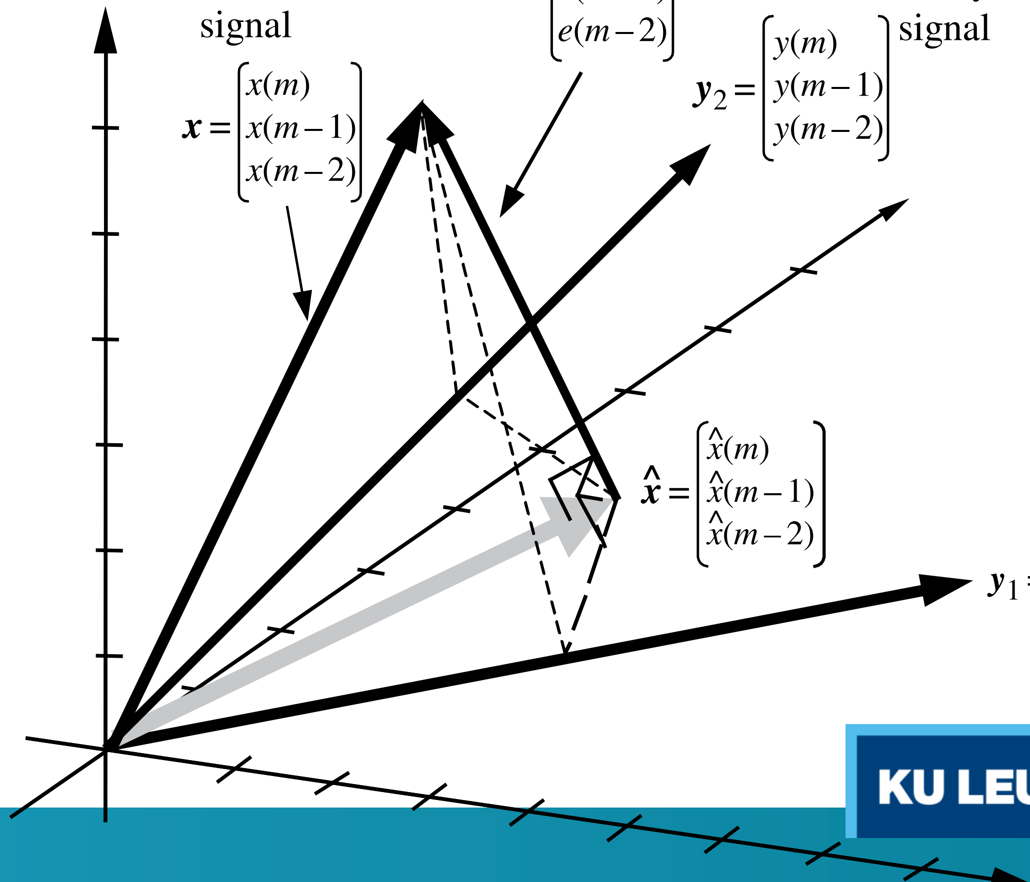
$$\Rightarrow \hat{\mathbf{x}} \neq \mathbf{x}, \;\; \mathbf{e} \neq 0$$

KU LEUVEN

Geometrical interpretation (3)

- **Vector space interpretation**
  - example: $N = 3, P = 2$

$$\begin{bmatrix} \hat{x}(m) \\ \hat{x}(m-1) \\ \hat{x}(m-2) \end{bmatrix} = w_0 \begin{bmatrix} \hat{y}(m) \\ \hat{y}(m-1) \\ \hat{y}(m-2) \end{bmatrix} + w_1 \begin{bmatrix} \hat{y}(m-1) \\ \hat{y}(m-2) \\ \hat{y}(m-3) \end{bmatrix}$$



Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

Performance analysis (1)

- **Variance of Wiener filter estimate**
  - substituting Wiener filter esimate $\mathbf{w} = \mathbf{R}_{\mathbf{yy}}^{-1}\mathbf{r}_{\mathbf{yx}}$ into MSE criterion gives error variance

$$E\{e^2(m)\} = r_{xx}(0) - \mathbf{w}^T\mathbf{r}_{\mathbf{yx}} = r_{xx}(0) - \mathbf{w}^T\mathbf{R}_{\mathbf{yy}}\mathbf{w}$$

  - variance of Wiener filter output signal is

$$E\{\hat{x}^2(m)\} = \mathbf{w}^T\mathbf{R}_{\mathbf{yy}}\mathbf{w}$$

so error variance can be written as

$$E\{e^2(m)\} = E\{x^2(m)\} - E\{\hat{x}^2(m)\}$$

$$\boxed{\sigma_e^2 = \sigma_x^2 - \sigma_{\hat{x}}^2}$$

Performance analysis (2)

- **Variance of Wiener filter estimate**
  - in general, observed data can be decomposed as

$$y(m) = x_c(m) + n(m)$$

    - $x_c(m)$ = part of observation correlated with desired signal $x(m)$
    - $n(m)$ = random noise signal
  - Wiener filter error signal can be decomposed accordingly

$$e(m) = \underbrace{\left(x(m) - \sum_{k=0}^{P-1} w_k x_c(m-k) \right)}_{e_x(m)} \underbrace{- \sum_{k=0}^{P-1} w_k n(m-k)}_{e_n(m)}$$

  - error variance is then $\boxed{\sigma_e^2 = \sigma_{e_x}^2 + \sigma_{e_n}^2}$

Least-squares and Wiener filter estimation

- Stochastic Wiener filter estimation
- Deterministic least squares estimation
- Computational aspects
- Geometrical interpretation
- Performance analysis
- Frequency domain formulation

Frequency domain formulation (1)

- **Frequency domain MSE criterion**
  - frequency domain Wiener filter output and error signal:

$$\hat{X}(f) = W(f)Y(f)$$

$$E(f) = X(f) - \hat{X}(f) = X(f) - W(f)Y(f)$$

  - frequency domain MSE criterion:

$$E\{|E(f)|^2\} = E\left\{(X(f) - W(f)Y(f))^* \, (X(f) - W(f)Y(f))\right\}$$

  - Parseval's theorem: sum of squared errors in time domain = integral of squared error power spectrum

$$\sum_{m=0}^{N-1} e^2(m) = \int_{-f_s/2}^{f_s/2} |E(f)|^2 df$$

KU LEUVEN

Frequency domain formulation (2)

- **Frequency domain Wiener filter estimate**
  - minimum MSE solution is obtained at point with zero gradient

$$\frac{\partial E\{|E(f)|^2\}}{\partial W(f)} = 2W(f)P_{YY}(f) - 2P_{XY}(f) = 0$$

  - power and cross-power spectra:

$$P_{YY}(f) = E\{Y(f)Y^*(f)\}$$
$$P_{XY}(f) = E\{X(f)Y^*(f)\}$$

  - frequency domain Wiener filter estimate:

$$\boxed{W(f) = \frac{P_{XY}(f)}{P_{YY}(f)}}$$

KU LEUVEN

Les 7: Optimale filtering

- **Introduction**

- **Least-squares and Wiener filter estimation**

  stochastic & deterministic estimation, computational aspects, geometrical interpretation, performance analysis, frequency domain formulation, …

- **Wiener filtering applications**

  noise reduction, time alignment of multi-channel/-sensor signals, channel equalization, …

- **Wiener filter implementation**

  filter order, filter-bank implementation, …

KU LEUVEN

Wiener filtering applications

- Application 1: noise reduction
- Application 2: channel equalization
- Application 3: time-alignment of multi-channel/-sensor signals

Application 1: noise reduction (1)

- **Time domain Wiener filter**
  - data model

$$y(m) = x(m) + n(m)$$

  - main assumption: desired signal and noise are uncorrelated
  - time domain Wiener filter:

$$\mathbf{R_{yy}} = \mathbf{R_{xx}} + \mathbf{R_{nn}}$$

$$\mathbf{r_{xy}} = \mathbf{r_{xx}}$$

$$\boxed{\mathbf{w} = (\mathbf{R_{xx}} + \mathbf{R_{nn}})^{-1}\mathbf{r_{xx}}}$$

  - noise correlation matrix is estimated during noise-only periods, which requires **signal activity detection**

KU LEUVEN

Application 1: noise reduction (2)

- **Frequency domain Wiener filter**
  - data model

$$Y(f) = X(f) + N(f)$$

  - main assumption: desired signal and noise are uncorrelated
  - frequency domain Wiener filter:

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)}$$

  - interpretation in terms of signal-to-noise ratio (SNR):

$$\boxed{W(f) = \frac{SNR(f)}{SNR(f) + 1}}$$

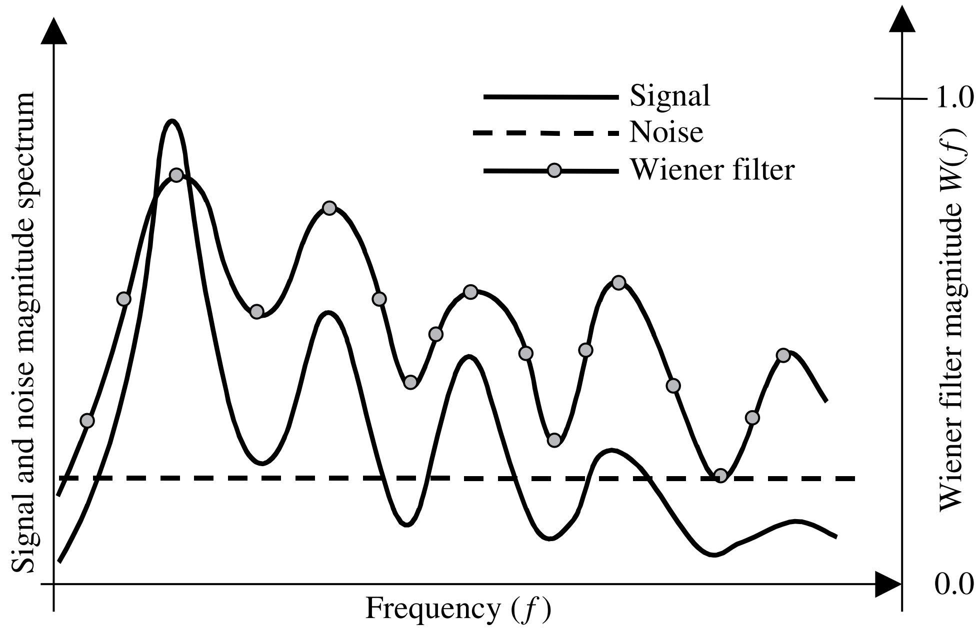
KU LEUVEN

Application 1: noise reduction (3)

- **Frequency domain Wiener filter**

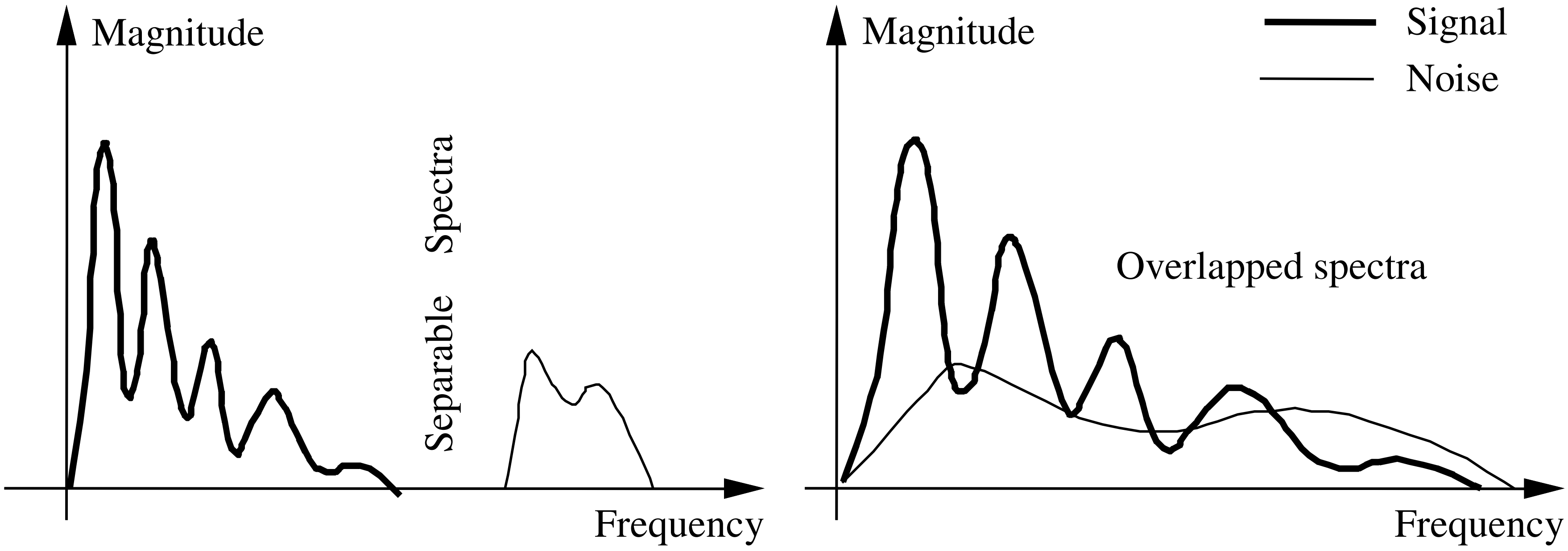
$$W(f) = \frac{SNR(f)}{SNR(f) + 1}$$

  – Wiener filter attenuates each frequency component in proportion to SNR



Application 1: noise reduction (4)

- **Frequency domain Wiener filter**
  – noise can only be removed completely when desired signal and noise spectra are separable



Wiener filtering applications

- Application 1: noise reduction
- Application 2: channel equalization
- Application 3: time-alignment of multi-channel/-sensor signals

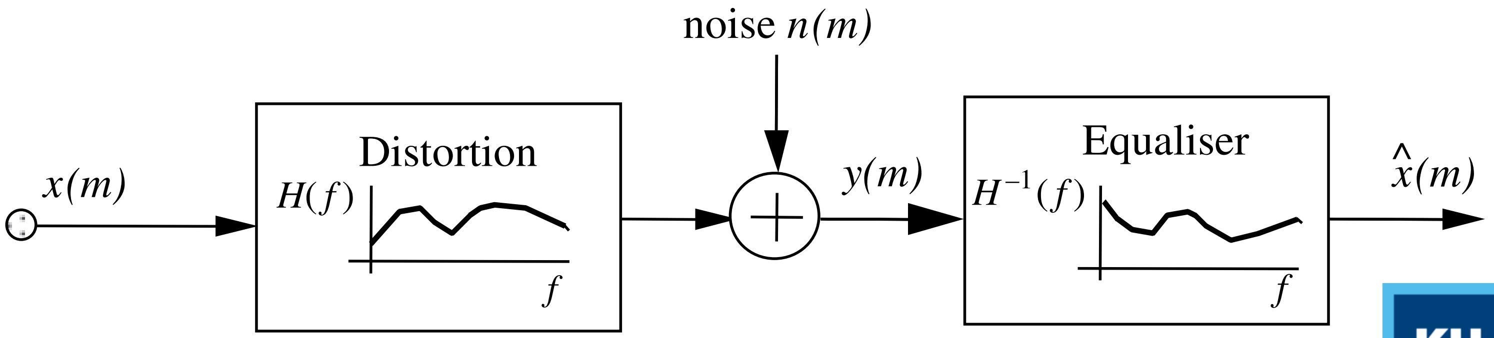
Application 2: channel equalization

- **Frequency domain Wiener filter**
  - data model

$$Y(f) = X(f)H(f) + N(f)$$

  - frequency domain Wiener filter = compromise between channel equalization & noise reduction

$$W(f) = \frac{P_{XX}(f)H^*(f)}{P_{XX}(f)|H(f)|^2 + P_{NN}(f)}$$



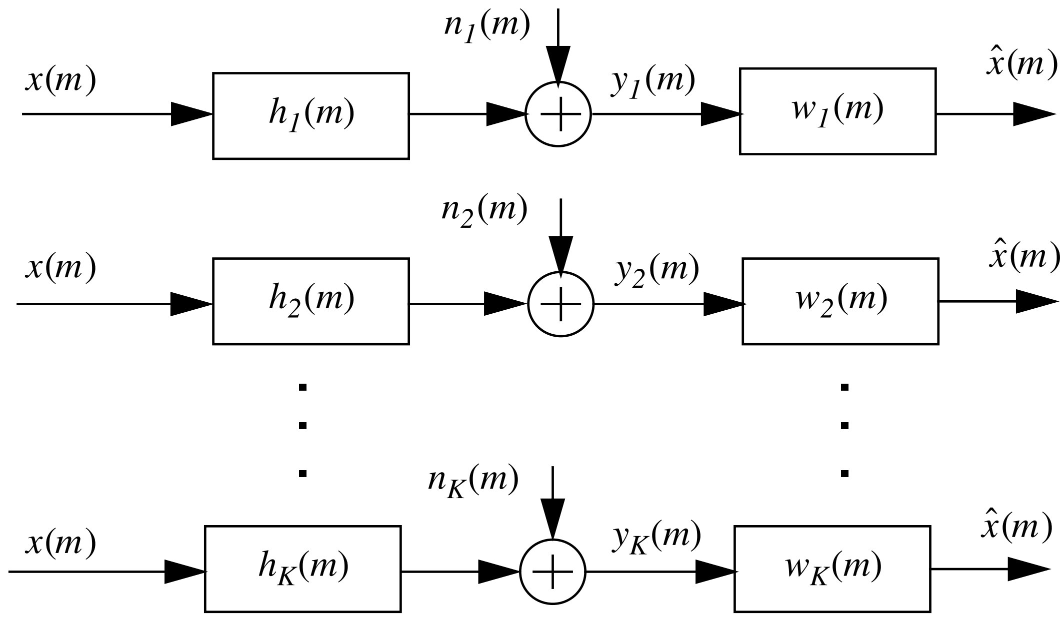
KU LEUVEN

Wiener filtering applications

- Application 1: noise reduction
- Application 2: channel equalization
- Application 3: time-alignment of multi-channel/-sensor signals

Application 3: time-alignment of multi-channel/-sensor signals (1)

- **Multi-channel/-sensor signals:**
  - sensor array = collection of multiple sensors observing same source signal *x* at different positions in space
  - each sensor signal is filtered & noisy version of source signal (linear filter h_k, additive noise n_k)



Application 3: time-alignment of multi-channel/-sensor signals (2)

- **Wiener filter**
  - data model for simple example ($K = 2$, $h_1 = 1$, $h_2 = Az^{-D}$):

$$\begin{aligned} y_1(m) &= x(m) + n_1(m) \\ y_2(m) &= Ax(m-D) + n_2(m) \end{aligned}$$

  - Wiener filter error signal (y_1 = input, y_2 = desired signal):

$$e(m) = y_2(m) - \sum_{k=0}^{P-1} w_k y_1(m)$$

  - time domain Wiener filter: $\mathbf{w} = (\mathbf{R_{xx}} + \mathbf{R_{n_1 n_1}})^{-1} A\mathbf{r_{xx}}(D)$
  - frequeny domain Wiener filter:

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{N_1 N_1}(f)} Ae^{-j\omega D}$$

KU LEUVEN

Les 7: Optimale filtering

- **Introduction**

- **Least-squares and Wiener filter estimation**

  stochastic & deterministic estimation, computational aspects, geometrical interpretation, performance analysis, frequency domain formulation, …

- **Wiener filtering applications**

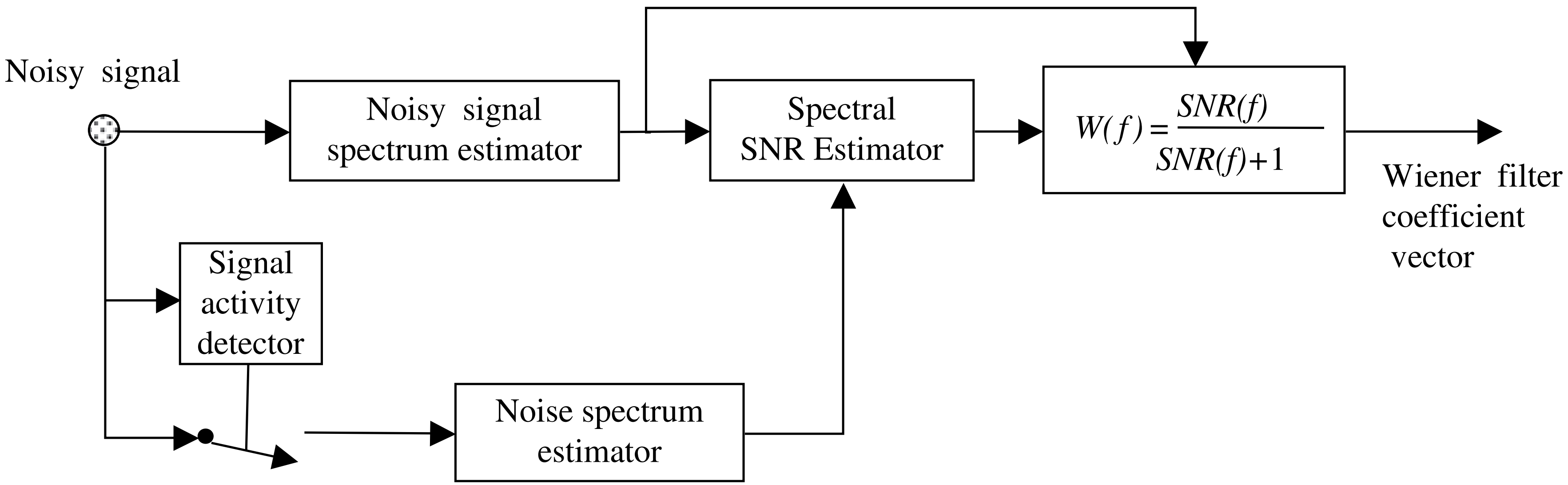
  noise reduction, time alignment of multi-channel/-sensor signals, channel equalization, …

- **Wiener filter implementation**

  filter order, filter-bank implementation, …

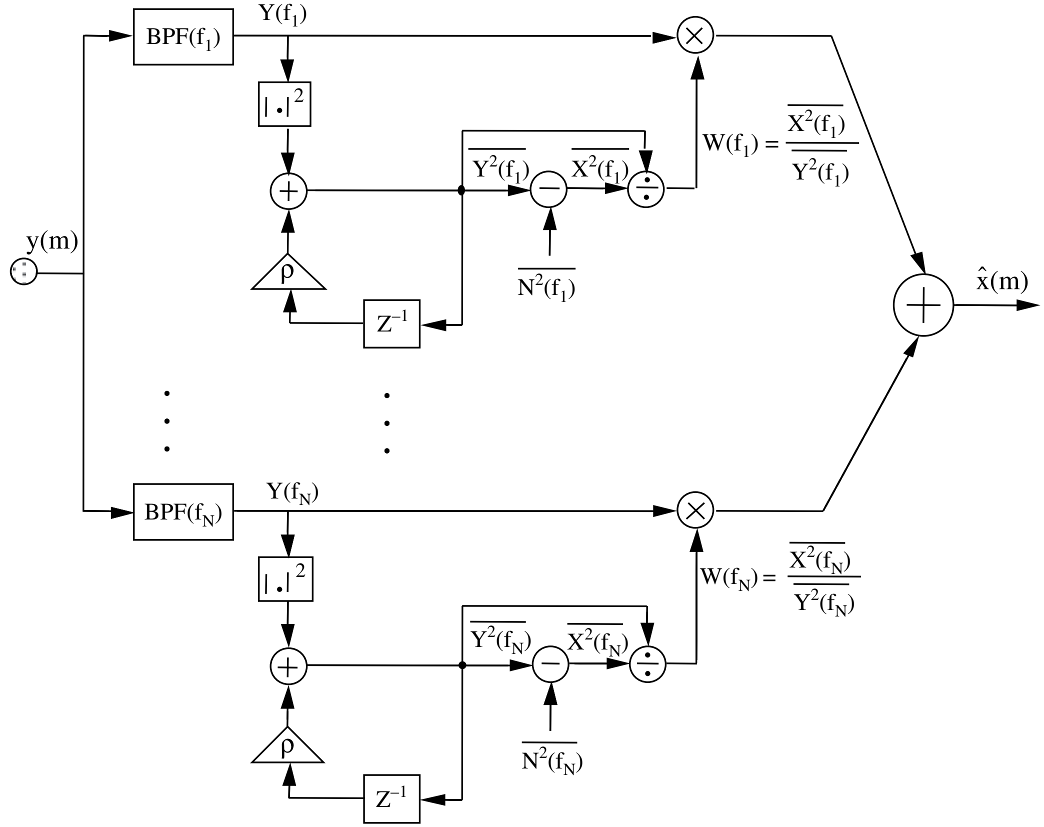
Wiener filter implementation (1)

- **Estimation of noise and noisy signal spectra**
  - use of signal activity detector
  - see Les 5: Detectieproblemen



Wiener filter implementation (2)

- **Filterbank implementation (see DSP-1)**
  - downsampling in subbands leads to complexity reduction



Wiener filter implementation (3)

- **Choice of Wiener filter order**

  – Wiener filter order affects:

  1. ability of filter to model and remove distortion and to reduce noise
  2. computational complexity of filter
  3. numerical stability of Wiener filter solution

  – choice of model order is always trade-off between these three criteria