

# Optimal and Adaptive Filtering

Murat Üney

M.Uney@ed.ac.uk

Institute for Digital Communications (IDCOM)



THE UNIVERSITY *of* EDINBURGH  
School of Engineering



26/06/2017

# Table of Contents

- 1 Optimal Filtering
  - Optimal filter design
  - Application examples
  - Optimal solution: Wiener-Hopf equations
  - Example: Wiener equaliser
- 2 Adaptive filtering
  - Introduction
  - Recursive Least Squares Adaptation
  - Least Mean Square Algorithm
  - Applications
- 3 Optimal signal detection
  - Application examples and optimal hypothesis testing
  - Additive white and coloured noise
- 4 Summary

# Optimal filter design

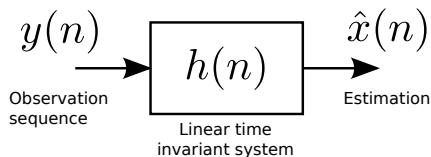


Figure 1: Optimal filtering scenario.

- $y(n)$ : Observation related to a **stationary** signal of interest  $x(n)$ .
- $h(n)$ : The impulse response of an LTI estimator.
- $\hat{x}(n)$ : Estimate of  $x(n)$  given by

$$\hat{x}(n) = h(n) * y(n) = \sum_{i=-\infty}^{\infty} h(i)y(n-i).$$

# Optimal filter design

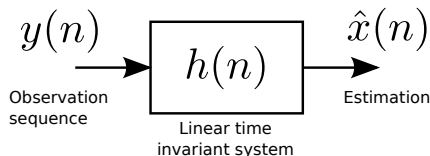


Figure 1: Optimal filtering scenario.

- Find  $h(n)$  with the best error performance:

$$e(n) = x(n) - \hat{x}(n) = x(n) - h(n) * y(n)$$

- The error performance is measured by the mean squared error (MSE)

$$\xi = E \left[ \left( e(n) \right)^2 \right].$$

# Optimal filter design

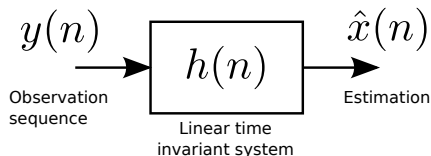


Figure 1: Optimal filtering scenario.

- The MSE is a function of  $h(n)$ , i.e.,

$$\mathbf{h} = [\dots, h(-2), h(-1), h(0), h(1), h(2), \dots]$$

$$\xi(\mathbf{h}) = E \left[ \left( e(n) \right)^2 \right] = E \left[ \left( x(n) - h(n) * y(n) \right)^2 \right].$$

# Optimal filter design

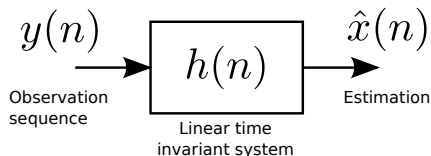


Figure 1: Optimal filtering scenario.

- The MSE is a function of  $h(n)$ , i.e.,

$$\mathbf{h} = [\dots, h(-2), h(-1), h(0), h(1), h(2), \dots]$$

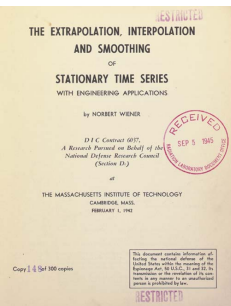
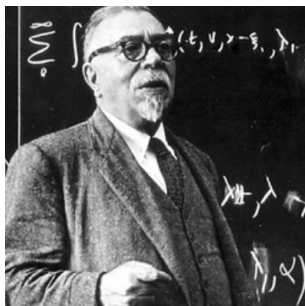
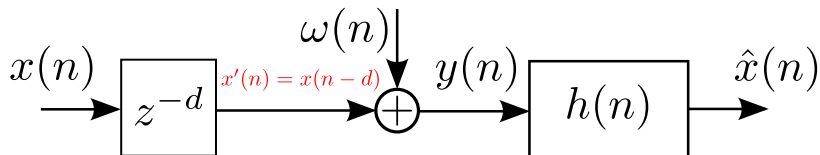
$$\xi(\mathbf{h}) = E \left[ \left( e(n) \right)^2 \right] = E \left[ \left( x(n) - h(n) * y(n) \right)^2 \right].$$

- Thus, optimal filtering problem is

$$\mathbf{h}_{opt} = \arg \min_{\mathbf{h}} \xi(\mathbf{h})$$

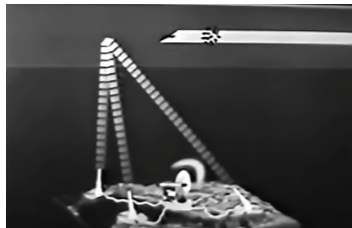
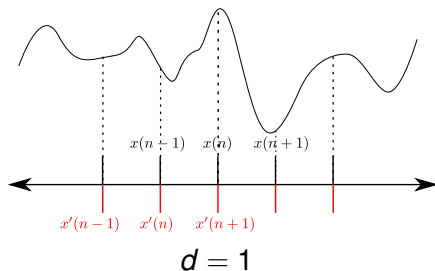
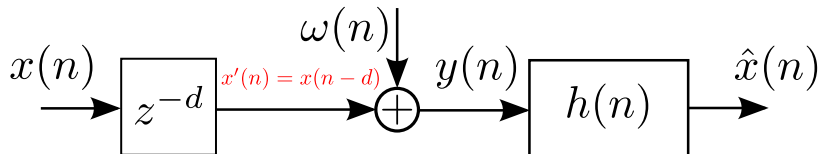
# Application examples

## 1) Prediction, interpolation and smoothing of signals



# Application examples

## 1) Prediction, interpolation and smoothing of signals

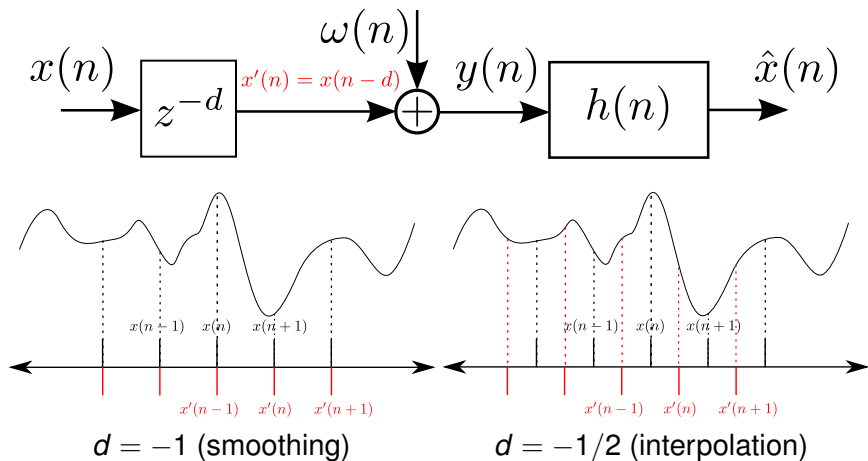


- Prediction for anti-aircraft fire control.



# Application examples

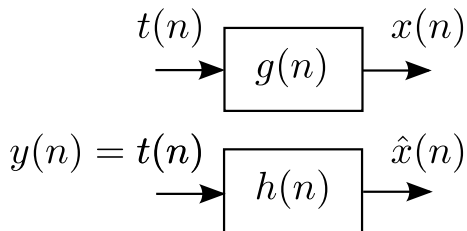
## 1) Prediction, interpolation and smoothing of signals



- Signal denoising applications, estimation of missing data points.

## Application examples

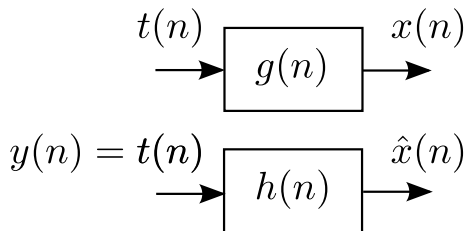
### 2) System identification



**Figure 2:** System identification using a training sequence  $t(n)$  from an ergodic and stationary ensemble.

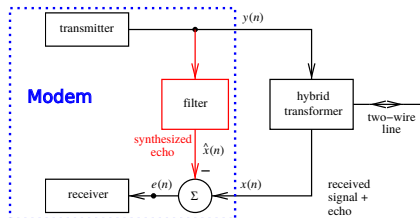
# Application examples

## 2) System identification



**Figure 2:** System identification using a training sequence  $t(n)$  from an ergodic and stationary ensemble.

- Echo cancellation in full duplex data transmission.



## Application examples

### 3) Inverse System identification

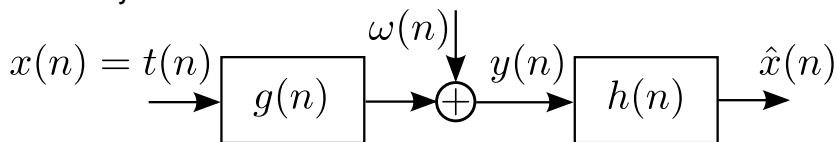


Figure 3: Inverse system identification using  $x(n)$  as a training sequence.

# Application examples

## 3) Inverse System identification

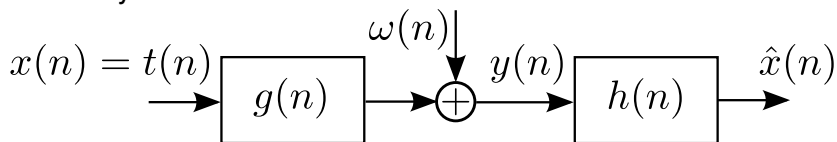
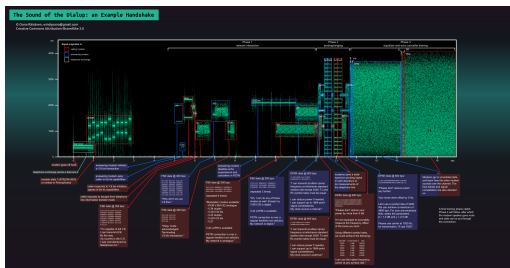


Figure 3: Inverse system identification using  $x(n)$  as a training sequence.

- ▶ Channel equalisation in digital communication systems.



## Optimal solution: Normal equations

- Consider the MSE  $\xi(\mathbf{h}) = E [(e(n))^2]$
- The optimal filter satisfies  $\nabla \xi(\mathbf{h})|_{h_{opt}} = \mathbf{0}$ . Equivalently, for all  $j = \dots, -2, -1, 0, 1, 2, \dots$

$$\begin{aligned}
 \frac{\partial \xi}{\partial h(j)} &= E \left[ 2e(n) \frac{\partial e(n)}{\partial h(j)} \right] \\
 &= E \left[ 2e(n) \frac{\partial (x(n) - \sum_{i=-\infty}^{\infty} h(i)y(n-i))}{\partial h(j)} \right] \\
 &= E \left[ 2e(n) \frac{\partial (-h(j)y(n-j))}{\partial h(j)} \right] \\
 &= -2E [e(n)y(n-j)]
 \end{aligned}$$

## Optimal solution: Normal equations

- Consider the MSE  $\xi(\mathbf{h}) = E [(e(n))^2]$
- The optimal filter satisfies  $\nabla \xi(\mathbf{h})|_{h_{opt}} = \mathbf{0}$ . Equivalently, for all  $j = \dots, -2, -1, 0, 1, 2, \dots$

$$\begin{aligned}
 \frac{\partial \xi}{\partial h(j)} &= E \left[ 2e(n) \frac{\partial e(n)}{\partial h(j)} \right] \\
 &= E \left[ 2e(n) \frac{\partial (x(n) - \sum_{i=-\infty}^{\infty} h(i)y(n-i))}{\partial h(j)} \right] \\
 &= E \left[ 2e(n) \frac{\partial (-h(j)y(n-j))}{\partial h(j)} \right] \\
 &= -2E [e(n)y(n-j)]
 \end{aligned}$$

- Hence, the optimal filter solves the “normal equations”

$$E [e(n)y(n-j)] = 0, j = \dots, -2, -1, 0, 1, 2, \dots$$

## Optimal solution: Wiener-Hopf equations

- The error of  $h_{opt}$  is orthogonal to its observations, i.e., for all  $j \in \mathbb{Z}$

$$E [e_{opt}(n)y(n-j)] = 0$$

which is known as “the principle of orthogonality”.



## Optimal solution: Wiener-Hopf equations

- The error of  $h_{opt}$  is orthogonal to its observations, i.e., for all  $j \in \mathbb{Z}$

$$E [e_{opt}(n)y(n-j)] = 0$$

which is known as “the principle of orthogonality”.

- Furthermore,

$$\begin{aligned} E [e_{opt}(n)y(n-j)] &= E \left[ \left( x(n) - \sum_{i=-\infty}^{\infty} h_{opt}(i)y(n-i) \right) y(n-j) \right] \\ &= E [x(n)y(n-j)] - \sum_{i=-\infty}^{\infty} h_{opt}(i)E [y(n-i)y(n-j)] = 0 \end{aligned}$$

## Optimal solution: Wiener-Hopf equations

- The error of  $h_{opt}$  is orthogonal to its observations, i.e., for all  $j \in \mathbb{Z}$

$$E[e_{opt}(n)y(n-j)] = 0$$

which is known as “the principle of orthogonality”.

- Furthermore,

$$\begin{aligned} E[e_{opt}(n)y(n-j)] &= E\left[\left(x(n) - \sum_{i=-\infty}^{\infty} h_{opt}(i)y(n-i)\right)y(n-j)\right] \\ &= E[x(n)y(n-j)] - \sum_{i=-\infty}^{\infty} h_{opt}(i)E[y(n-i)y(n-j)] = 0 \end{aligned}$$

### Result (Wiener-Hopf equations)

$$\sum_{i=-\infty}^{\infty} h_{opt}(i)r_{yy}(i-j) = r_{xy}(j)$$

# The Wiener filter

- Wiener-Hopf equations can be solved indirectly, in the complex spectral domain:

$$h_{opt}(n) * r_{yy}(n) = r_{xy}(n) \leftrightarrow H_{opt}(z)P_{yy}(z) = P_{xy}(z)$$

# The Wiener filter

- Wiener-Hopf equations can be solved indirectly, in the complex spectral domain:

$$h_{opt}(n) * r_{yy}(n) = r_{xy}(n) \leftrightarrow H_{opt}(z)P_{yy}(z) = P_{xy}(z)$$

## Result (The Wiener filter)

$$H_{opt}(z) = \frac{P_{xy}(z)}{P_{yy}(z)}$$

# The Wiener filter

- Wiener-Hopf equations can be solved indirectly, in the complex spectral domain:

$$h_{opt}(n) * r_{yy}(n) = r_{xy}(n) \leftrightarrow H_{opt}(z)P_{yy}(z) = P_{xy}(z)$$

## Result (The Wiener filter)

$$H_{opt}(z) = \frac{P_{xy}(z)}{P_{yy}(z)}$$

- The optimal filter has an infinite impulse response (IIR), and, is non-causal, in general.

## Causal Wiener filter

- We project the unconstrained solution  $H_{opt}(z)$  onto the set of causal and stable IIR filters by a two step procedure:
- First, factorise  $P_{yy}(z)$  into causal (right sided)  $Q_{yy}(z)$ , and anti-causal (left sided) parts  $Q_{yy}^*(1/z^*)$ , i.e.,  

$$P_{yy}(z) = \sigma_y^2 Q_{yy}(z) Q_{yy}^*(1/z^*).$$
- Select the causal (right sided) part of  $P_{xy}(z)/Q_{yy}^*(1/z^*)$ .

### Result (Causal Wiener filter)

$$H_{opt}^+(z) = \frac{1}{\sigma_y^2 Q_{yy}(z)} \left[ \frac{P_{xy}(z)}{Q_{yy}^*(1/z^*)} \right]_+$$

# FIR Wiener-Hopf equations

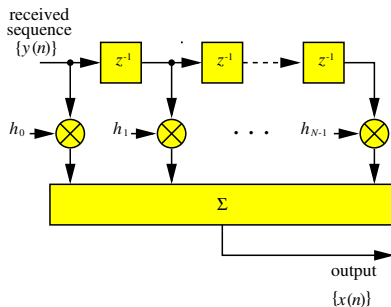


Figure 4: A finite impulse response (FIR) estimator.

- Wiener-Hopf equations for the FIR optimal filter of  $N$  taps:

## Result (FIR Wiener-Hopf equations)

$$\sum_{i=0}^{N-1} h_{opt}(i) r_{yy}(i-j) = r_{xy}(j), \text{ for } j = 0, 1, \dots, N-1.$$

# FIR Wiener Filter

- FIR Wiener-Hopf equations in vector-matrix form.

$$\underbrace{\begin{bmatrix} r_{yy}(0) & r_{yy}(1) & \dots & r_{yy}(N-1) \\ r_{yy}(1) & r_{yy}(0) & \dots & r_{yy}(N-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_{yy}(N-1) & r_{yy}(N-2) & \dots & r_{yy}(0) \end{bmatrix}}_{\triangleq \mathbf{R}_{yy}} \underbrace{\begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix}}_{\triangleq \mathbf{h}_{opt}} = \underbrace{\begin{bmatrix} r_{xy}(0) \\ r_{xy}(1) \\ \vdots \\ r_{xy}(N-1) \end{bmatrix}}_{\triangleq \mathbf{r}_{xy}}$$

$\triangleq \mathbf{R}_{yy}$  : Autocorrelation matrix of  $y(n)$  which is Toeplitz.



## FIR Wiener Filter

- FIR Wiener-Hopf equations in vector-matrix form.

$$\underbrace{\begin{bmatrix} r_{yy}(0) & r_{yy}(1) & \dots & r_{yy}(N-1) \\ r_{yy}(1) & r_{yy}(0) & \dots & r_{yy}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(N-1) & r_{yy}(N-2) & \dots & r_{yy}(0) \end{bmatrix}}_{\triangleq \mathbf{R}_{yy} : \text{Autocorrelation matrix of } y(n) \text{ which is Toeplitz.}} \underbrace{\begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix}}_{\triangleq \mathbf{h}_{opt}} = \underbrace{\begin{bmatrix} r_{xy}(0) \\ r_{xy}(1) \\ \vdots \\ r_{xy}(N-1) \end{bmatrix}}_{\triangleq \mathbf{r}_{xy}}$$

### Result (FIR Wiener filter)

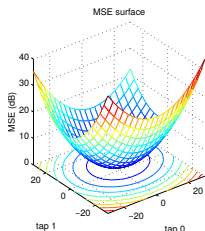
$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy}.$$

# MSE surface

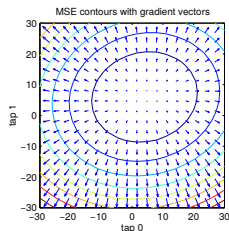
- MSE is a quadratic function of  $\mathbf{h}$

$$\xi(\mathbf{h}) = \mathbf{h}^T \mathbf{R}_{yy} \mathbf{h} - 2\mathbf{h}^T \mathbf{r}_{xy} + E[(x(n))^2]$$

$$\nabla \xi(\mathbf{h}) = 2\mathbf{R}_{yy} \mathbf{h} - 2\mathbf{r}_{xy}$$



(a)



(b)

**Figure 5:** For a 2-tap Wiener filtering example: (a) the MSE surface, (b) gradient vectors.

# Example: Wiener equaliser

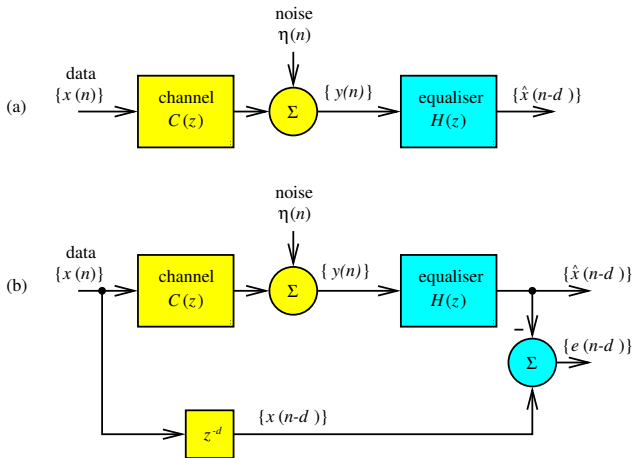


Figure 6: (a) The Wiener equaliser. (b) Alternative formulation.

# Wiener equaliser

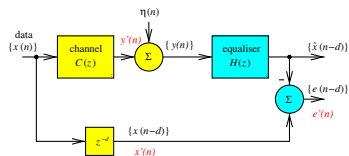


Figure 7: Channel equalisation scenario.

- For notational convenience define:

$$\begin{aligned} x'(n) &= x(n-d) \\ e'(n) &= x(n-d) - \hat{x}(n-d) \end{aligned} \quad (1)$$

- Label the output of the channel filter as  $y'(n)$  where

$$y(n) = y'(n) + \eta(n)$$

# Wiener equaliser

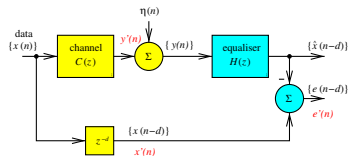


Figure 7: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{x'y} \quad (2)$$

# Wiener equaliser

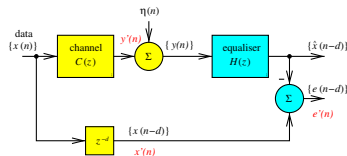


Figure 7: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{x'y} \quad (2)$$

- The  $(i, j)$ th entry in  $\mathbf{R}_{yy}$  is

$$\begin{aligned} r_{yy}(j-i) &= E[y(j)y(i)] \\ &= E[(y'(j) + \eta(j))(y'(i) + \eta(i))] \\ &= r_{y'y'}(j-i) + \sigma_{\eta}^2 \delta(j-i) \\ &\Leftrightarrow P_{yy}(z) = P_{y'y'}(z) + \sigma_{\eta}^2 \end{aligned}$$

# Wiener equaliser

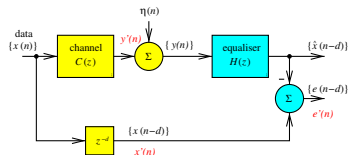


Figure 7: Channel equalisation scenario.

- Remember  $y'(n) = c(n) * x(n)$ 

$$\leftrightarrow r_{y'y'} = c(n) * c(-n) * r_{xx}(n) \leftrightarrow P_{y'y'}(z) = C(z)C(z^{-1})P_{xx}(z)$$
- Consider a white data sequence  $x(n)$ , i.e.,
 
$$r_{xx}(n) = \sigma_x^2 \delta(n) \leftrightarrow P_{xx}(z) = \sigma_x^2.$$
- Then, the complex spectra of the autocorrelation sequence of interest is

$$P_{yy}(z) = P_{y'y'}(z) + \sigma_\eta^2 = C(z)C(z^{-1})\sigma_x^2 + \sigma_\eta^2$$

# Wiener equaliser

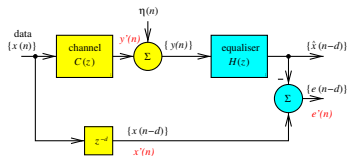


Figure 7: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{x'y} \quad (3)$$



# Wiener equaliser

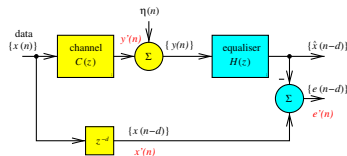


Figure 7: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{x'y} \quad (3)$$

- The ( $j$ )th entry in  $\mathbf{r}_{x'y}$  is

$$\begin{aligned} r_{x'y}(j) &= E [x'(n)y(n-j)] \\ &= E [x(n-d)(y'(n-j) + \eta(n-j))] \\ &= r_{xy'}(j-d) \\ &\Leftrightarrow P_{x'y}(z) = P_{xy'}(z)z^{-d} \end{aligned} \quad (4)$$

# Wiener equaliser

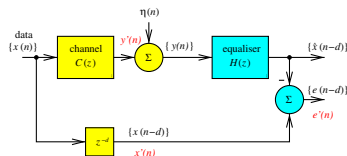


Figure 7: Channel equalisation scenario.

- Remember  $y'(n) = c(n) * x(n)$

$$\leftrightarrow r_{xy'} = c(-n) * r_{xx}(n) \leftrightarrow P_{xy'}(z) = C(z^{-1})P_{xx}(z)$$

- Then, the complex spectra of the cross correlation sequence of interest is

$$P_{x'y}(z) = P_{xy'}(z)z^{-d} = \sigma_x^2 C(z^{-1})z^{-d}$$

## Wiener equaliser

- Suppose that  $\mathbf{c} = [c(0) = 0.5, c(1) = 1]^T \leftrightarrow C(z) = (0.5 + z^{-1})$
- Then,

$$P_{yy}(z) = C(z)C(z^{-1})\sigma_x^2 + \sigma_\eta^2 = (0.5 + z^{-1})(0.5 + z)\sigma_x^2 + \sigma_\eta^2$$

$$P_{x'y}(z) = \sigma_x^2 C(z^{-1})z^{-d} = (0.5z^{-d} + z^{-d+1})\sigma_x^2$$

- Suppose that  $d = 1$ ,  $\sigma_x^2 = 1$ , and,  $\sigma_\eta^2 = 0.1$

$$r_{yy}(0) = 1.35, r_{yy}(1) = 0.5, \text{ and } r_{yy}(2) = 0$$

$$r_{x'y}(0) = 1, r_{x'y}(1) = 0.5, \text{ and } r_{x'y}(2) = 0$$

- The Wiener filter is obtained as

$$\mathbf{h}_{opt} = \left( \begin{bmatrix} 1.35 & 0.5 & 0 \\ 0.5 & 1.35 & 0.5 \\ 0 & 0.5 & 1.35 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.69 \\ 0.13 \\ -0.05 \end{bmatrix}$$

- The MSE is found as  $\xi(\mathbf{h}_{opt}) = \sigma_x^2 - \mathbf{h}_{opt}^T \mathbf{r}_{x'y} = 0.24$ .

# Adaptive filtering - Introduction

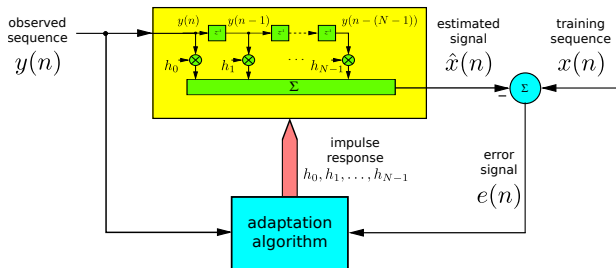


Figure 8: FIR adaptive filtering configuration.

- For notational convenience, define

$$\mathbf{y}(n) \triangleq [y(n), y(n-1), \dots, y(n-N+1)]^T, \quad \mathbf{h}(n) \triangleq [h_0, h_1, \dots, h_{N-1}]^T$$

- The output of the adaptive filter is

$$\hat{x}(n) = \mathbf{h}^T(n) \mathbf{y}(n)$$

- Optimum solution

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy}$$

## Recursive least squares

- Minimise cost function

$$\xi(n) = \sum_{k=0}^n (x(k) - \hat{x}(k))^2 \quad (5)$$

- Solution

$$\mathbf{R}_{yy}(n)\mathbf{h}(n) = \mathbf{r}_{xy}(n)$$

- LS “autocorrelation” matrix

$$\mathbf{R}_{yy}(n) = \sum_{k=0}^n \mathbf{y}(k)\mathbf{y}^T(k)$$

- LS “cross-correlation” vector

$$\mathbf{r}_{xy}(n) = \sum_{k=0}^n \mathbf{y}(k)x(k)$$

## Recursive least squares

- Recursive relationships

$$\mathbf{R}_{yy}(n) = \mathbf{R}_{yy}(n-1) + \mathbf{y}(n)\mathbf{y}^T(n)$$

$$\mathbf{r}_{xy}(n) = \mathbf{r}_{xy}(n-1) + \mathbf{y}(n)x(n)$$

- Substitute for  $\mathbf{r}_{xy}$

$$\mathbf{R}_{yy}(n)\mathbf{h}(n) = \mathbf{R}_{yy}(n-1)\mathbf{h}(n-1) + \mathbf{y}(n)x(n)$$

- Replace  $\mathbf{R}_{yy}(n-1)$

$$\mathbf{R}_{yy}(n)\mathbf{h}(n) = \left( \mathbf{R}_{yy}(n) - \mathbf{y}(n)\mathbf{y}^T(n) \right) \mathbf{h}(n-1) + \mathbf{y}(n)x(n)$$

- Multiple both sides by  $\mathbf{R}_{yy}^{-1}(n)$

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{R}_{yy}^{-1}(n)\mathbf{y}(n)e(n)$$

$$e(n) = x(n) - \mathbf{h}^T(n-1)\mathbf{y}(n)$$

# Recursive least squares

- Recursive relationships

$$\mathbf{R}_{yy}(n) = \mathbf{R}_{yy}(n-1) + \mathbf{y}(n)\mathbf{y}^T(n)$$

- Apply Sherman-Morrison identity

$$\mathbf{R}_{yy}^{-1}(n) = \mathbf{R}_{yy}^{-1}(n-1) - \frac{\mathbf{R}_{yy}^{-1}(n-1)\mathbf{y}(n)\mathbf{y}^T(n)\mathbf{R}_{yy}^{-1}(n-1)}{1 + \mathbf{y}^T(n)\mathbf{R}_{yy}^{-1}(n-1)\mathbf{y}(n)}$$

# Summary

Recursive least squares (RLS) algorithm:

- 1:  $\mathbf{R}_{yy}(0) = \frac{1}{\delta} \mathbf{I}_N$  with small positive  $\delta$  ▷ Initialisation 1
- 2:  $\mathbf{h}(0) = \mathbf{0}$  ▷ Initialisation 2
- 3: **for**  $n = 1, 2, 3, \dots$  **do** ▷ Iterations
- 4:      $\hat{\mathbf{x}}(n) = \mathbf{h}^T(n-1)\mathbf{y}(n)$  ▷ Estimate  $\mathbf{x}(n)$
- 5:      $e(n) = x(n) - \hat{x}(n)$  ▷ Find the error
- 6:      $\mathbf{R}_{yy}^{-1}(n) = \frac{1}{\alpha} \left( \mathbf{R}_{yy}^{-1}(n-1) - \frac{\mathbf{R}_{yy}^{-1}(n-1)\mathbf{y}(n)\mathbf{y}^T(n)\mathbf{R}_{yy}^{-1}(n-1)}{\alpha + \mathbf{y}^T(n)\mathbf{R}_{yy}^{-1}(n-1)\mathbf{y}(n)} \right)$   
▷ Update the inverse of the autocorrelation matrix
- 7:      $\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{R}_{yy}^{-1}(n)\mathbf{y}(n)e(n)$  ▷ Update the filter coefficients
- 8: **end for**



# Stochastic gradient algorithms

- MSE contour - 2-tap example:

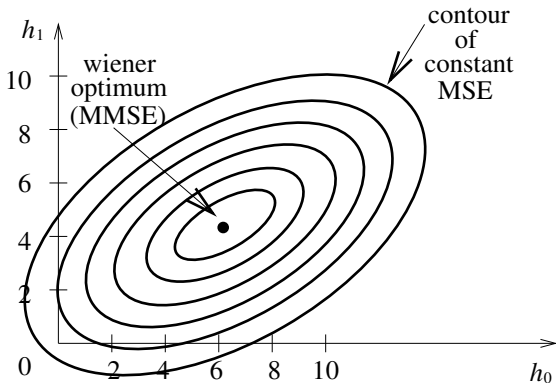
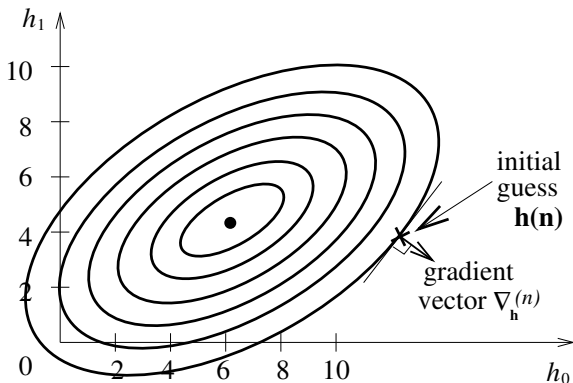


Figure 9: Method of steepest descent.

# Steepest descent

- MSE contour - 2-tap example:

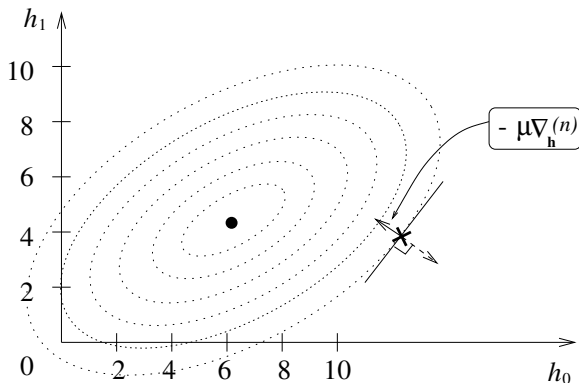


- The gradient vector

$$\nabla_{\mathbf{h}}(n) = \left[ \frac{\partial \xi}{\partial h(0)}, \frac{\partial \xi}{\partial h(1)}, \dots, \frac{\partial \xi}{\partial h(N-1)} \right]^T \bigg|_{\mathbf{h}=\mathbf{h}(n)} = 2\mathbf{R}_{yy}\mathbf{h}(n) - 2\mathbf{r}_{xy}$$

# Steepest descent

- MSE contour - 2-tap example:



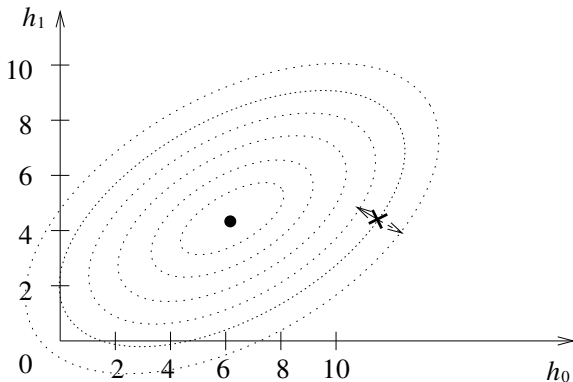
- Update initial guess in the direction of steepest descent:

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu \nabla_{\mathbf{h}}(n)$$

- Step-size  $\mu$ .

# Steepest descent

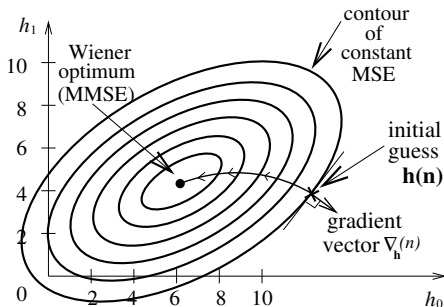
- MSE contour - 2-tap example:



- Gradient at new guess.

## Convergence of steepest descent

- MSE contour - 2-tap example:



$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu \nabla_{\mathbf{h}}(n)$$

$$\nabla_{\mathbf{h}}(n) = \left[ \frac{\partial \xi}{\partial h(0)}, \frac{\partial \xi}{\partial h(1)}, \dots, \frac{\partial \xi}{\partial h(N-1)} \right]^T \bigg|_{\mathbf{h}=\mathbf{h}(n)} = 2\mathbf{R}_{yy}\mathbf{h}(n) - 2\mathbf{r}_{xy}$$

$$0 < \mu < \frac{1}{\lambda_{max}} \quad (6)$$

# Stochastic gradient algorithms

- A time recursion:

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu \hat{\nabla}_{\mathbf{h}}(n)$$

- The exact gradient:

$$\begin{aligned}\nabla_{\mathbf{h}}(n) &= -2E \left[ \mathbf{y}(n)(x(n) - \mathbf{y}(n)^T \mathbf{h}(n)) \right] \\ &= -2E [\mathbf{y}(n)e(n)]\end{aligned}$$

- A simple estimate of the gradient

$$\hat{\nabla}_{\mathbf{h}}(n) = -2\mathbf{y}(n+1)e(n+1)$$

- The error

$$e(n+1) = x(n+1) - \mathbf{h}(n)^T \mathbf{y}(n+1) \quad (7)$$

# The Least-mean-squares (LMS) algorithm:

```

1: h(0) = 0                                ▷ Initialisation
2: for  $n = 1, 2, 3, \dots$  do                ▷ Iterations
3:    $\hat{x}(n) = \mathbf{h}^T(n-1)\mathbf{y}(n)$     ▷ Estimate  $\mathbf{x}(n)$ 
4:    $\mathbf{e}(n) = x(n) - \hat{x}(n)$            ▷ Find the error
5:    $\mathbf{h}(n) = \mathbf{h}(n-1) + 2\mu\mathbf{y}(n)\mathbf{e}(n)$   ▷ Update the filter
   coefficients
6: end for

```

# LMS block diagram

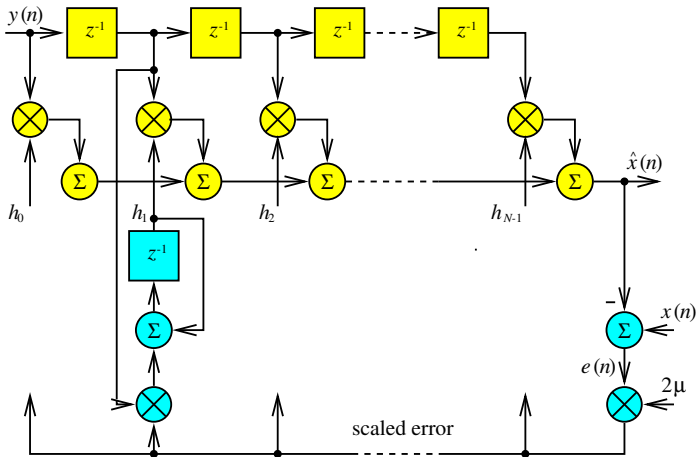
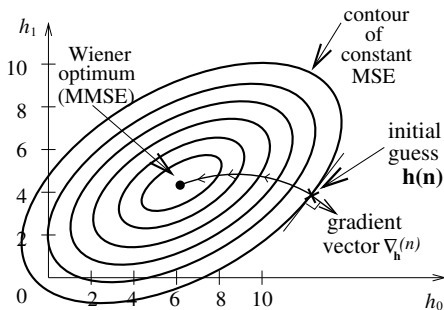


Figure 10: Least mean-square adaptive filtering.



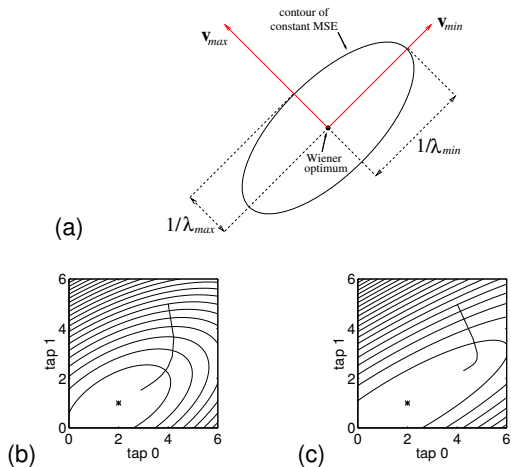
## Convergence of the LMS

- MSE contour - 2-tap example:



- Eigenvalues of  $\mathbf{R}_{yy}$  (in this example,  $\lambda_0$  and  $\lambda_1$ ).
- The largest time constant  $\tau_{max} > \frac{\lambda_{max}}{2\lambda_{min}}$
- Eigenvalue ratio (EVR) is  $\frac{\lambda_{max}}{\lambda_{min}}$
- Practical range for step-size  $0 < \mu < \frac{1}{3N\sigma_y^2}$

# Eigenvalue ratio (EVR)



**Figure 11:** Eigenvectors, eigenvalues and convergence: (a) the relationship between eigenvectors, eigenvalues and the contours of constant MSE; (b) steepest descent for EVR of 2; (c) EVR of 4.

# Comparison of RLS and LMS

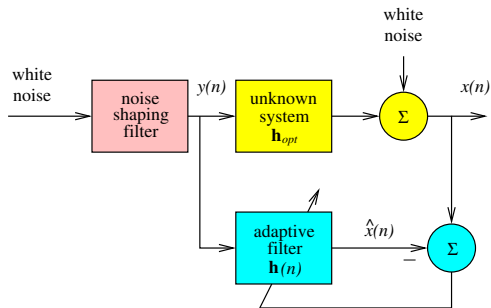
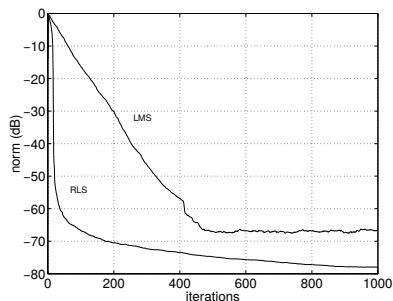


Figure 12: Adaptive system identification configuration.

- Error vector norm

$$\rho(n) = E \left[ (\mathbf{h}(n) - \mathbf{h}_{opt})^T (\mathbf{h}(n) - \mathbf{h}_{opt}) \right]$$

# Comparison: Performance



**Figure 13:** Convergence plots for  $N = 16$  taps adaptive filtering in the system identification configuration:  $EVR = 1$  (i.e., the impulse response of the noise shaping filter is  $\delta(n)$ ).

# Comparison: Performance

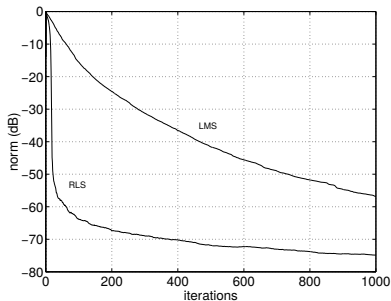
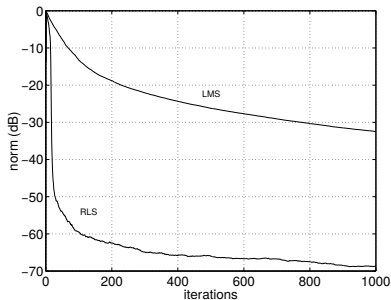


Figure 14: Coverage plots for  $N = 16$  taps adaptive filtering in the system identification configuration:  $EVR = 11$ .

## Comparison: Performance



**Figure 15:** Convergence plots for  $N = 16$  taps adaptive filtering in the system identification configuration: EVR (and, correspondingly the spectral coloration of the input signal) progressively increases to 68.

# Comparison: Complexity

Table: Complexity comparison of  $N$ -point FIR filter algorithms.

<i>Algorithm class</i>	<i>Implementation</i>	<i>Computational load</i>		
		<i>multiplications</i>	<i>adds/subtractions</i>	<i>divisions</i>
RLS	fast Kalman	$10N+1$	$9N+1$	2
SG	LMS	$2N$	$2N$	–
	BLMS (via FFT)	$10\log(N)+8$	$15\log(N)+30$	–

# Applications

- Adaptive filtering algorithms can be used in all application areas of optimal filtering.
- Some examples:
  - ▶ Adaptive line enhancement
  - ▶ Adaptive tone suppression
  - ▶ Echo cancellation
  - ▶ Channel equalisation



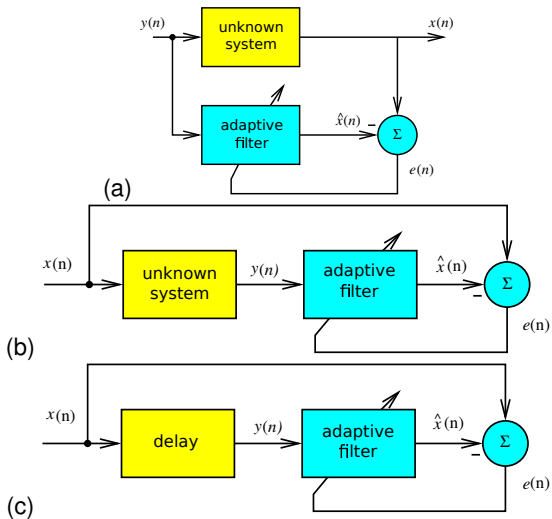


Figure 16: Adaptive filtering configurations: (a) direct system modelling; (b) inverse system modelling; (c) linear prediction.

# Adaptive line enhancement

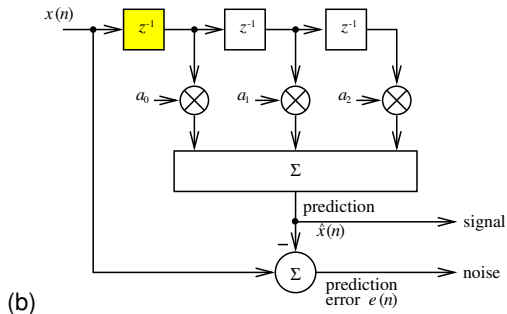
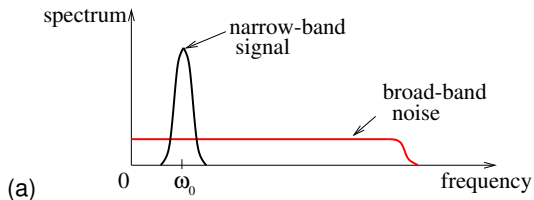
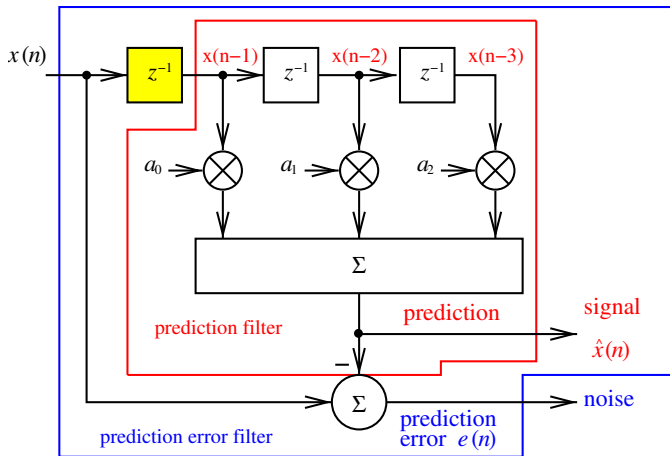


Figure 17: Adaptive line enhancement: (a) signal spectrum; (b) system

# Adaptive predictor



- Prediction filter:  $a_0 + a_1 z^{-1} + a_2 z^{-2}$
- Prediction error filter:  $1 - a_0 z^{-1} - a_1 z^{-2} - a_2 z^{-3}$

# Adaptive tone suppression

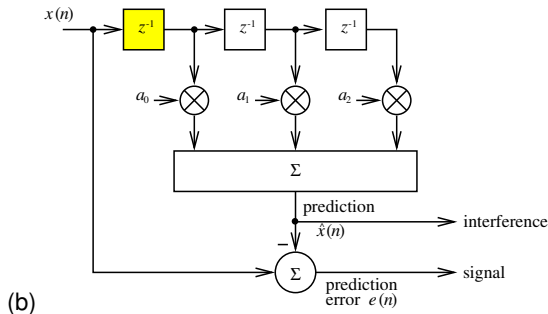
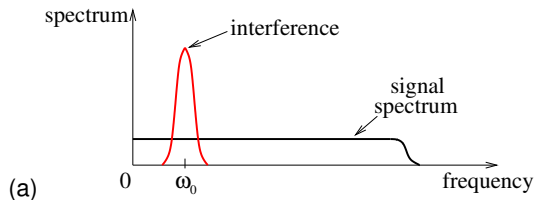


Figure 18: Adaptive tone suppression: (a) signal spectrum; (b) system

# Adaptive noise whitening

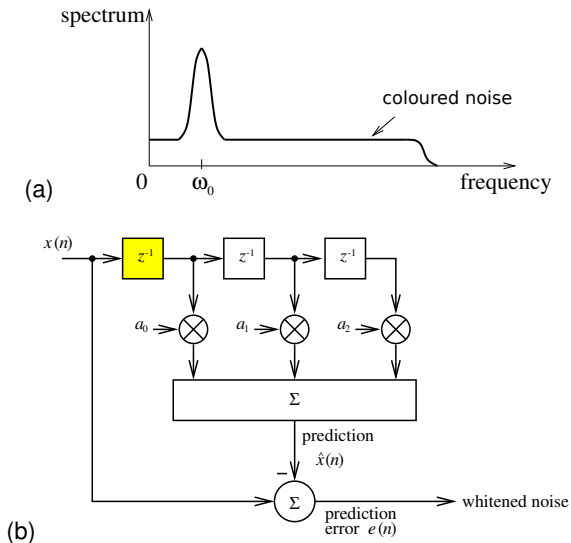
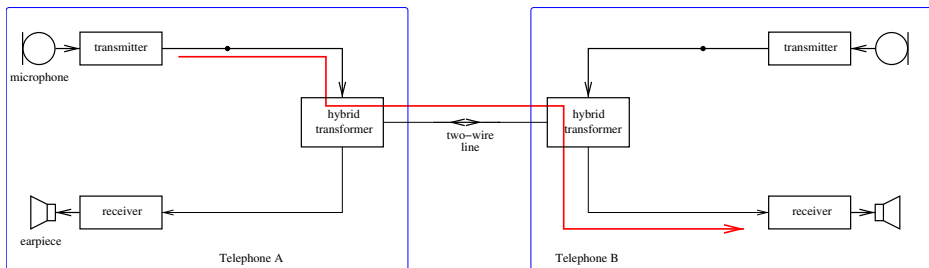


Figure 19: Adaptive noise whitening: (a) input spectrum; (b) system

# Echo cancellation

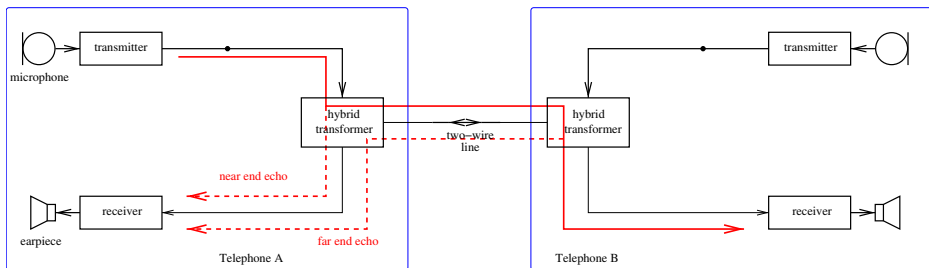
- A typical telephone connection



- Hybrid transformers to route signal paths.

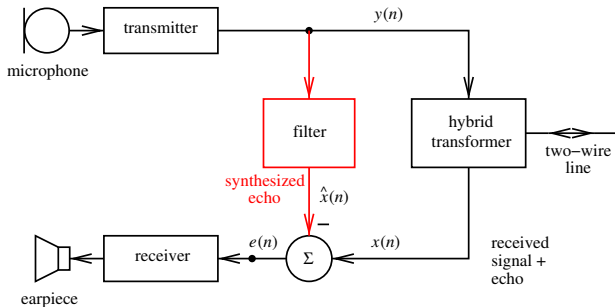
# Echo cancellation (contd)

- Echo paths in a telephone system



- Near and far echo paths.

# Echo cancellation (contd)



- Fixed filter?



# Echo cancellation (contd)

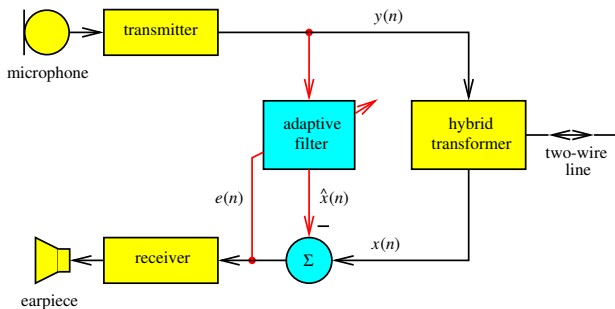


Figure 20: Application of adaptive echo cancellation in a telephone handset.

# Channel equalisation

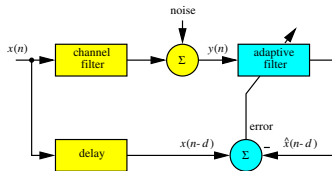


Figure 21: Adaptive equaliser system configuration.

- Simple channel

$$y(n) = \pm h_0 + \text{noise}$$

- Decision circuit

$$\text{if } y(n) \geq 0 \text{ then } x(n) = +1 \text{ else } x(n) = -1$$

- Channel with intersymbol interference (ISI)

$$y(n) = \sum_{i=0}^2 h_i x(n-i) + \text{noise}$$

# Channel equalisation

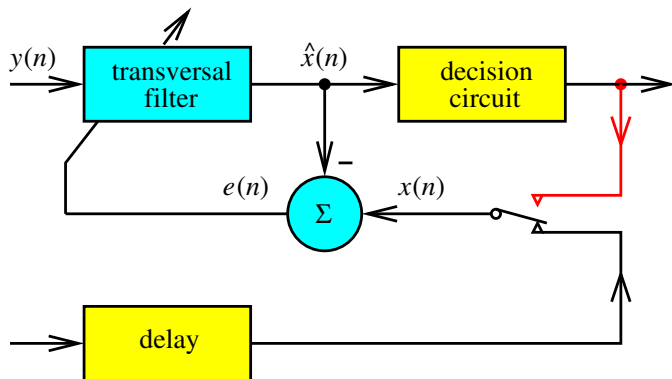
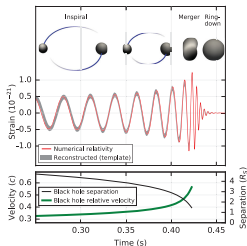
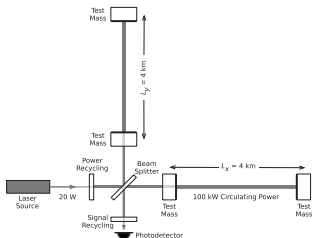
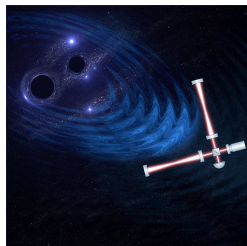


Figure 22: Decision directed equaliser.

# Optimal signal detection - Introduction

# Optimal signal detection - Introduction

- Example: Detection of gravitational waves



**Figure 23:** Gravitational waves from a binary black hole merger (left, Uni. of Birmingham, Gravitational Wave Group), LIGO block diagram (middle), expected signal (right) Abbot et. al., "Observation of gravitational waves from a binary black hole merger", Phys. Rev. Let., Feb. 2016..

## Optimal signal detection

- Signal detection as 2-ary (binary) hypothesis testing:

$$\begin{aligned}H_0 &: y(n) = \eta(n) \\H_1 &: y(n) = x(n) + \eta(n)\end{aligned}\tag{8}$$

- In a sense, decide which of the two possible ensembles  $y(n)$  is generated from.
- Finite length signals, i.e.,

$$n = 0, 1, 2, \dots, N - 1$$

- Vector notation

$$\begin{aligned}H_0 &: \mathbf{y} = \boldsymbol{\eta} \\H_1 &: \mathbf{y} = \mathbf{x} + \boldsymbol{\eta}\end{aligned}$$

# Optimal signal detection

- Example (radar): In active sensing,  $x(t)$  is the probing waveform subject to design.

$$y(n) = a_0 x(n - n_0) + a_1 x(n - n_1) + \dots + \eta(n) \quad (9)$$

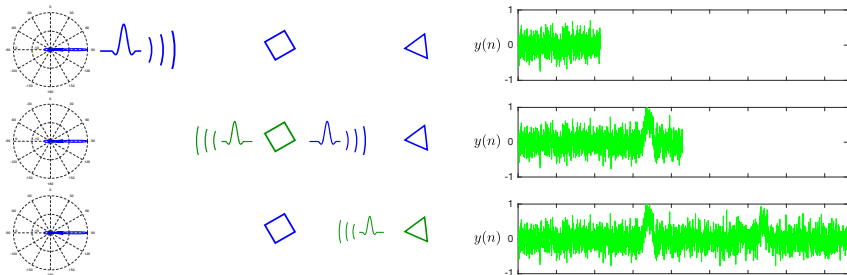


Figure 24: Probing waveform  $x(n)$  and returns from the surveillance region constituting  $y(n)$ .

## Optimal signal detection

- Example (radar): In active sensing,  $x(t)$  is the probing waveform subject to design.

$$y(n) = a_0 x(n - n_0) + a_1 x(n - n_1) + \dots + \eta(n) \quad (9)$$

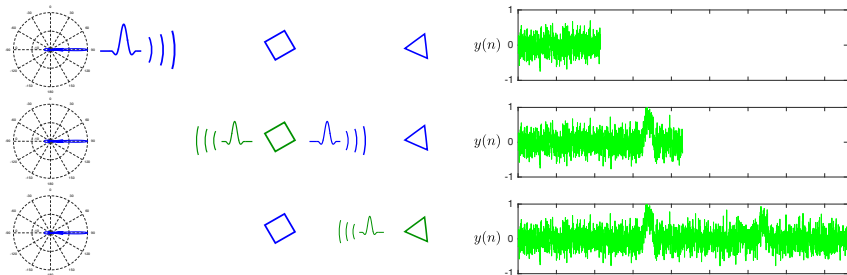


Figure 24: Probing waveform  $x(n)$  and returns from the surveillance region constituting  $y(n)$ .

- A similar problem also arises in digital communications.



## Bayesian hypothesis testing

- Consider a random variable  $\mathbf{H}$  with  $H = \mathbf{H}(\zeta)$  and  $H \in \{H_0, H_1\}$ .
- The measurement  $\bar{\mathbf{y}} = \mathbf{y}(\zeta)$  is a realisation of  $\mathbf{y}$ .
- The measurement model specifies the likelihood  $p_{\mathbf{Y}|H}(\bar{\mathbf{y}}|H)$
- Find the probabilities of  $H = H_1$  and  $H = H_0$  based on the measurement vector  $\bar{\mathbf{y}}$ .
- Decide on the hypothesis with the maximum a posteriori probability:

## Bayesian hypothesis testing

- Consider a random variable  $\mathbf{H}$  with  $H = \mathbf{H}(\zeta)$  and  $H \in \{H_0, H_1\}$ .
- The measurement  $\bar{\mathbf{y}} = \mathbf{y}(\zeta)$  is a realisation of  $\mathbf{y}$ .
- The measurement model specifies the likelihood  $p_{\mathbf{Y}|\mathbf{H}}(\bar{\mathbf{y}}|H)$
- Find the probabilities of  $H = H_1$  and  $H = H_0$  based on the measurement vector  $\bar{\mathbf{y}}$ .
- Decide on the hypothesis with the maximum a posteriori probability:
- Find the *maximum a-posteriori* (MAP) estimate of  $H$ .

$$\hat{H} = \arg \max_{H \in \{H_0, H_1\}} P_{\mathbf{H}|\mathbf{y}}(H|\bar{\mathbf{y}}) \quad (10)$$

where the a posterior probability of  $\mathbf{H}$  is given by

$$P_{\mathbf{H}|\mathbf{y}}(H_i|\bar{\mathbf{y}}) = \frac{p_{\mathbf{Y}|\mathbf{H}}(\bar{\mathbf{y}}|H_i)P_{\mathbf{H}}(H_i)}{p_{\mathbf{Y}|\mathbf{H}}(\bar{\mathbf{y}}|H_0)P_{\mathbf{H}}(H_0) + p_{\mathbf{Y}|\mathbf{H}}(\bar{\mathbf{y}}|H_1)P_{\mathbf{H}}(H_1)} \quad (11)$$

for  $i = 0, 1$ .

# Bayesian hypothesis testing: The likelihood ratio test

- MAP decision rule:

$$\hat{H} = \arg \max_{H \in \{H_0, H_1\}} P(H|\bar{y})$$

- MAP decision as a likelihood ratio test:

$$\begin{aligned} \hat{H}=H_1 & \\ p(H_1|\bar{y}) & \underset{\hat{H}=H_0}{\geq} p(H_0|\bar{y}) \\ p(\bar{y}|H_1)P(H_1) & \underset{H_0}{\underset{H_1}{\geq}} p(\bar{y}|H_0)P(H_0) \\ \frac{p(\bar{y}|H_1)}{p(\bar{y}|H_0)} & \underset{H_0}{\underset{H_1}{\geq}} \frac{P(H_0)}{P(H_1)} \end{aligned}$$

## Bayesian hypothesis testing - AWGN Example

- Example: Detection of deterministic signals in additive white Gaussian noise:

$$H_0 : \mathbf{y} = \boldsymbol{\eta}$$

$$H_1 : \mathbf{y} = \mathbf{x} + \boldsymbol{\eta}$$

where  $\mathbf{x}$  is a known vector,  $\boldsymbol{\eta} \sim \mathcal{N}(\cdot; \mathbf{0}, \sigma^2 \mathbf{I})$ .

- The likelihoods are specified by the noise distribution:

$$\frac{p(\bar{\mathbf{y}}|H_1)}{p(\bar{\mathbf{y}}|H_0)} \underset{H_0}{\overset{H_1}{>}} \frac{P(H_0)}{P(H_1)}$$

$$\frac{\mathcal{N}(\bar{\mathbf{y}}; \mathbf{x}, \sigma^2 \mathbf{I})}{\mathcal{N}(\bar{\mathbf{y}}; \mathbf{0}, \sigma^2 \mathbf{I})} \underset{H_0}{\overset{H_1}{>}} \frac{P(H_0)}{P(H_1)}$$

## Detection of deterministic signals - AWGN (contd)

- The numerator and denominator of the likelihood ratio are

$$\begin{aligned}
 p(\bar{y}|H_1) &= \mathcal{N}(\bar{y} - \mathbf{x}; \mathbf{0}, \sigma^2 \mathbf{I}) \\
 &= \frac{1}{(2\pi\sigma^2)^{N/2}} \prod_{n=0}^{N-1} \exp \left\{ -\frac{(\bar{y}(n) - x(n))^2}{2\sigma^2} \right\} \\
 &= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma^2} \left( \sum_{n=0}^{N-1} (\bar{y}(n) - x(n))^2 \right) \right\} \quad (12)
 \end{aligned}$$

- Similarly

$$\begin{aligned}
 p(\bar{y}|H_0) &= \mathcal{N}(\bar{y}; \mathbf{0}, \sigma^2 \mathbf{I}) \\
 &= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma^2} \left( \sum_{n=0}^{N-1} (\bar{y}(n))^2 \right) \right\} \quad (13)
 \end{aligned}$$

- Therefore

$$\frac{p(\bar{y}|H_1)}{p(\bar{y}|H_0)} = \exp \left\{ \frac{1}{\sigma^2} \left( \sum_{n=0}^{N-1} (\bar{y}(n)x(n) - \frac{1}{2}x(n)^2) \right) \right\} \quad (14)$$

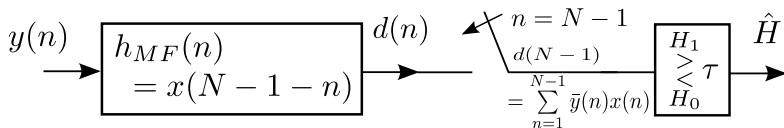
## Detection of deterministic signals - AWGN (contd)

- Take the logarithm of both sides of the likelihood ratio test

$$\log \exp \left\{ \frac{1}{\sigma^2} \left( \sum_{n=0}^{N-1} (\bar{y}(n)x(n) - \frac{1}{2}x(n)^2) \right) \right\} \underset{H_0}{\overset{H_1}{>}} \log \frac{P(H_0)}{P(H_1)}$$

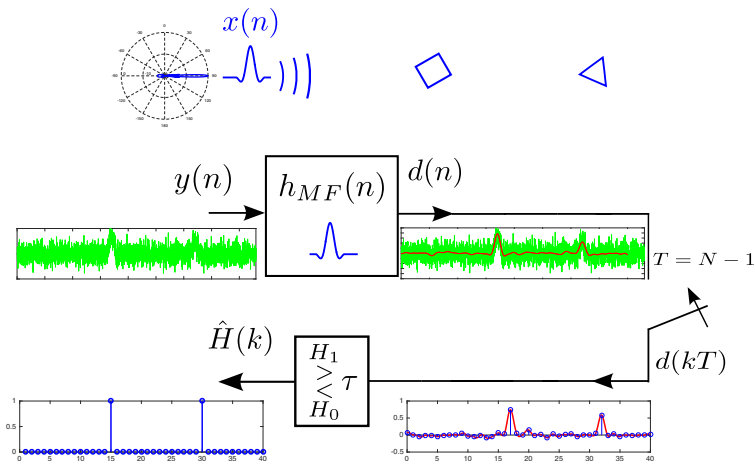
- Now, we have a linear statistical test

$$\sum_{n=0}^{N-1} \bar{y}(n)x(n) \underset{H_0}{\overset{H_1}{>}} \underbrace{\sigma^2 \log \frac{P(H_0)}{P(H_1)} + \frac{1}{2} \sum_{n=0}^{N-1} x(n)^2}_{\triangleq \tau: \text{Decision threshold}}$$



# Detection of deterministic signals - AWGN (contd)

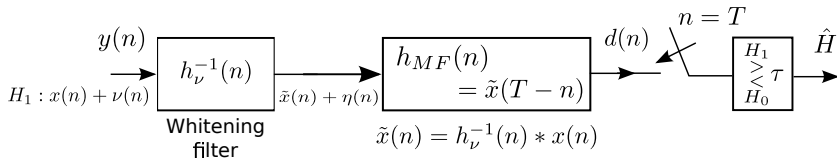
- Matched filtering for optimal detection:



# Detection of deterministic signals under coloured noise

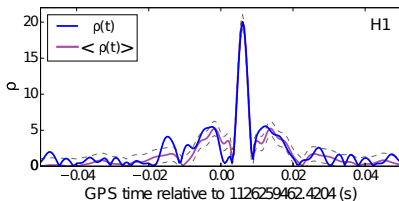
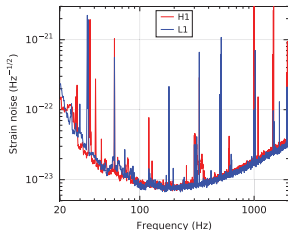
- For the case, the noise sequence  $\rho(n)$  has an auto-correlation function  $r_\nu[l]$  different than  $r_\nu[l] = \sigma_\eta^2 \times \delta[l]$ , and,

$$\begin{aligned}
 H_0 : \mathbf{y} &= \boldsymbol{\nu} \\
 H_1 : \mathbf{y} &= \mathbf{x} + \boldsymbol{\nu}
 \end{aligned}
 \quad \boldsymbol{\nu} \sim \mathcal{N} \left( \mathbf{0}, \mathbf{C}_\nu = \begin{bmatrix} r_\nu(0) & r_\nu(-1) & \dots & r_\nu(-N+1) \\ r_\nu(1) & r_\nu(0) & \dots & r_\nu(-N+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_\nu(N-1) & r_\nu(N-2) & \dots & r_\nu(0) \end{bmatrix} \right)$$





# Detection of deterministic signals - coloured noise ex.



PRL 116, 061102 (2016)

Selected for a Viewpoint in Physics  
PHYSICAL REVIEW LETTERS

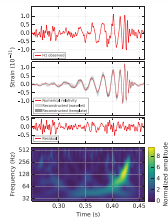
week ending  
12 FEBRUARY 2016

## Observation of Gravitational Waves from a Binary Black Hole Merger

B. P. Abbott *et al.*<sup>\*</sup>  
(LIGO Scientific Collaboration and Virgo Collaboration)  
(Received 21 January 2016; published 11 February 2016)

On September 14, 2015 at 09:50:45 UTC the two detectors of the Laser Interferometer Gravitational-Wave Observatory simultaneously observed a transient gravitational-wave signal. The signal sweeps upwards in frequency from 35 to 250 Hz with a peak gravitational-wave strain of  $1.0 \times 10^{-21}$ . It matches the waveform predicted by general relativity for the inspiral and merger of a pair of black holes and the ringdown of the resulting single black hole. The signal was observed with a matched filter signal-to-noise ratio of 24 and a false alarm rate estimated to be less than 1 event per 203 000 years, equivalent to a significance greater than 5.1 $\sigma$ . The source lies at a luminosity distance of  $410^{+120}_{-180}$  Mpc corresponding to a redshift  $z = 0.09^{+0.01}_{-0.02}$ . In the source frame, the initial black hole masses are  $36^{+5}_{-4} M_{\odot}$  and  $29^{+4}_{-4} M_{\odot}$ , and the final black hole mass is  $62^{+8}_{-7} M_{\odot}$ , with  $3.0^{+0.5}_{-0.4} M_{\odot} c^2$  radiated in gravitational waves. All uncertainties define 90% credible intervals. These observations demonstrate the existence of binary stellar-mass black hole systems. This is the first direct detection of gravitational waves and the first observation of a binary black hole merger.

DOI: 10.1103/PhysRevLett.116.061102



(upper left) Noise (amplitude) spectral density. (upper right) Abstract, Abbot et. al., Phys. Rev. Let., Feb. 2016.. (lower left)

Matched filter outputs: Best MF (blue) and the expected MF (purple). (lower right) Measurement, reconstructed and noise signals

around the detection.

# Summary

- Optimal filtering: Problem statement
- General solution via Wiener-Hopf equations
- FIR Wiener filter
- Adaptive filtering as an online optimal filtering strategy
- Recursive least-squares (RLS) algorithm
- Least mean-square (LMS) algorithm
- Application examples
- Optimal signal detection via matched filtering

## Further reading

- C. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, 1992.
- S. Haykin, *Adaptive Filter Theory*, 5th ed., Prentice-Hall, 2013.
- B. Mulgrew, P. Grant, J. Thompson, *Digital Signal Processing: Concepts and Applications*, 2nd ed., Palgrave Macmillan, 2003.
- D. Manolakis, V. Ingle, S. Kogon, *Statistical and Adaptive Signal Processing*, McGraw Hill, 2000.