Optimal and Adaptive Filtering

Murat Üney

M.Uney@ed.ac.uk

Institute for Digital Communications (IDCOM)

THE UNIVERSITY of EDINBURGH
School of Engineering

20/07/2015

- This presentation aims to provide an introductory tutorial to optimal and adaptive filtering.
- In order to keep a focus on the structures that the problems and their solutions, rather than mathematical details of the theory, we assume real valued random processes. For example, if $x(n)$ is a random process
$$x^*(n) = x(n)$$
throughout the presentation.
- Therefore, complex conjugations are omitted in the mathematical expressions and simplifications in the case of real valued sequences are used, for example, in complex spectra representations.
- This is a living document, and, its latest version can be downloaded from the UDRC Summer School 2015 website.
- Feedback is always welcome.

Table of Contents

- This presentation starts by introducing the problem definition for optimal filtering. Application examples follow this introduction.
- Wiener-Hopf equations are derived which characterise the solution of the problem. Then, the Wiener filter is introduced for both infinite impulse response (IIR) and finite impulse response FIR settings. Wiener channel equalisation is explained with an example.
- Adaptive filtering is introduced as an online and iterative strategy to optimal filtering. We emphasise that this strategy is useful especially when the statistical moments relevant to solving the optimal filtering problem are unknown and should be estimated from the incoming data and a training sequence.
- We derive the recursive least squares (RLS) and the least mean square (LMS) algorithms, and, compare them in an example. We provide system configurations for various applications of adaptive (optimal) filters.
- Finally, we give an overview of known signal detection in noise and relate the “matched filtering” technique to Bayesian hypothesis testing.

Optimal filter design

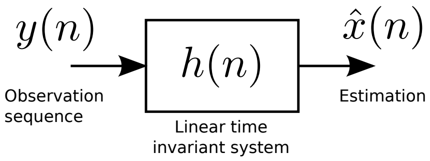


Figure 1: Optimal filtering scenario.

- $y(n)$: Observation related to a signal of interest $x(n)$.
- $h(n)$: The impulse response of an LTI estimator.

- We observe a stationary sequence $y(n)$, and, would like to estimate a desired signal $x(n)$ based on these observations (Fig. 1).
- The estimator we want to use is a linear time invariant (LTI) filter h characterised by its impulse response $h(n)$.
- The output of this estimator is given by the convolution of its input with the impulse response $h(n)$:

$$\hat{x}(n) = h(n) * y(n) = \sum_{i=-\infty}^{\infty} h(i)y(n-i). \tag{1}$$

Optimal filter design

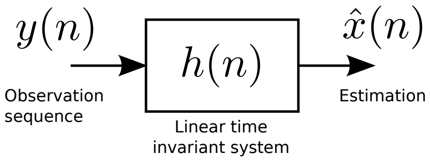


Figure 1: Optimal filtering scenario.

- $y(n)$: Observation related to a signal of interest $x(n)$.
- $h(n)$: The impulse response of an LTI estimator.
- 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)

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

- We would like to find $h(n)$ that would generate an output as close to the desired signal $x(n)$ as possible when driven by the input $y(n)$. Let us define the estimation error by

$$e(n) = x(n) - \hat{x}(n) \tag{2}$$

- $\hat{x}(n)$ is stationary owing to that the estimator is LTI, and, its input $y(n)$ is stationary. Therefore, the error sequence $e(n)$ is also stationary.
- Because $e(n)$ is stationary, it can be characterised by the expectation of its square at any time step n, or, the mean squared error (MSE):

$$\xi \triangleq E\left[e^2(n)\right] = E\left[\left(x(n) - \sum_{i=-\infty}^{\infty} h(i)y(n-i)\right)^2\right]. \tag{3}$$

Optimal filter design

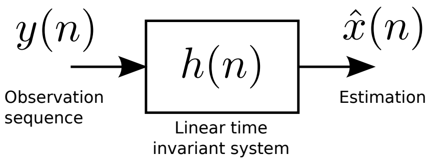


Figure 2: Optimal filtering scenario.

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

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

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

- Note that, the MSE is a function of the estimator impulse response. This point becomes more clear after the error term $e(n)$ is fully expanded to its components.

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

- It is useful to use a vector-matrix notation to cast the filter design problem as an optimisation problem.
- Consider the impulse response as a vector, i.e.,

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

.

Optimal filter design

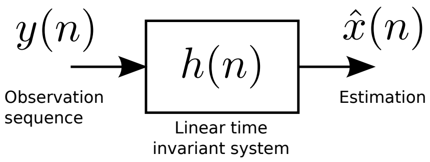


Figure 2: Optimal filtering scenario.

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

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

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

- Thus, optimal filtering problem is

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

- Let $\mathbf{x}$ and $\mathbf{e}$ denote the desired signal vector and the error vector constructed in a similar fashion, respectively. Then, $\mathbf{e} = \mathbf{x} - \mathbf{Y}\mathbf{h}$ expanded as

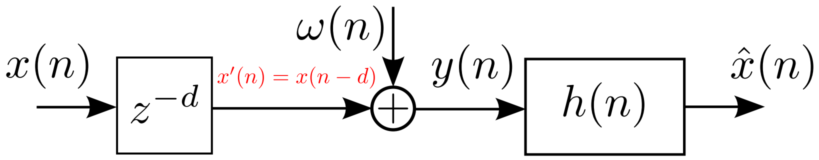
$$\underbrace{\begin{bmatrix} \vdots \\ e(0) \\ e(1) \\ e(2) \\ e(3) \\ \vdots \end{bmatrix}}_{\triangleq \mathbf{e}} = \underbrace{\begin{bmatrix} \vdots \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ \vdots \end{bmatrix}}_{\triangleq \mathbf{x}} - \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & y(0) & y(-1) & y(-2) & y(-3) & \ldots \\ \ldots & y(1) & y(0) & y(-1) & y(-2) & \ldots \\ \ldots & y(2) & y(1) & y(0) & y(-1) & \ldots \\ \ldots & y(3) & y(2) & y(1) & y(0) & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{\triangleq \mathbf{Y}\text{ : Convolution (or, data) matrix of } \mathbf{y} \text{ which is Toeplitz.}} \underbrace{\begin{bmatrix} \vdots \\ h(0) \\ h(1) \\ h(2) \\ h(3) \\ \vdots \end{bmatrix}}_{\triangleq \mathbf{h}} \quad (4)$$

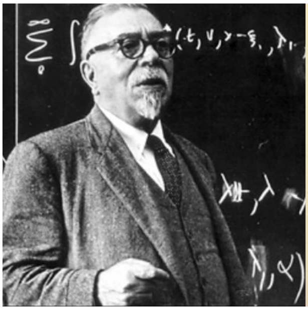
- Optimal filtering problem is the problem of finding $\mathbf{h}$ that leads to the minimum ξ. Equivalently, we want to solve the following optimisation problem:

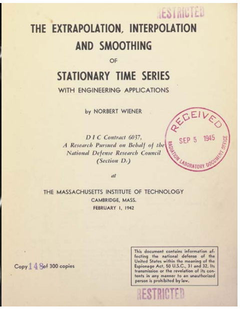
$$\mathbf{h}_{opt} = \arg\min_{\mathbf{h}} \xi(\mathbf{h}). \quad (5)$$

Application examples

1) Prediction, interpolation and smoothing of signals






RESTRICTED

THE EXTRAPOLATION, INTERPOLATION AND SMOOTHING OF STATIONARY TIME SERIES WITH ENGINEERING APPLICATIONS

by NORBERT WIENER

D I C Contract 6037, A Research Pursued on Behalf of the National Defense Research Council (Section D₂)

at

THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASS.
FEBRUARY 1, 1942

RECEIVED SEP 5 1945 RADIATION LABORATORY DOCUMENT OFFICE

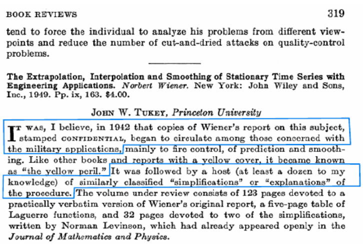
Copy 148 of 300 copies

This document contains information affecting the national defense of the United States within the meaning of the Espionage Act, 50 U.S.C., 31 and 32. Its transmission or the revelation of its contents in any manner to an unauthorized person is prohibited by law.

RESTRICTED

- A mile stone in optimal linear estimation was a World War II-time classified report by Norbert Wiener (1894–1964), a celebrated American mathematician and philosopher. This report was published in 1949 as a monograph.
- The picture is the cover of one of the (300) original copies of this report sold for \$7200 by an auction house.
- A review of this book by J.W. Tukey published in the *Journal of the American Statistical Association* in 1952 mentions that

> *...Wiener's report... was followed by a host (at least a dozen to my knowledge) of similarly classified "simplifications" or "explanations" of the procedure...*


BOOK REVIEWS 319

tend to force the individual to analyze his problems from different viewpoints and reduce the number of cut-and-dried attacks on quality-control problems.

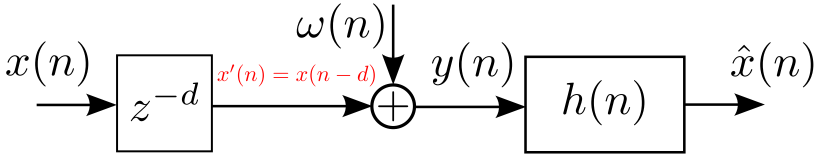
The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications. *Norbert Wiener.* New York: John Wiley and Sons, Inc., 1949. Pp. ix, 163. \$4.00.

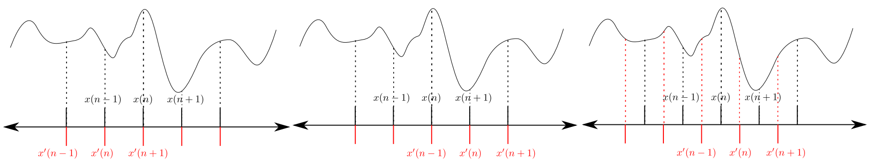
JOHN W. TUKEY, *Princeton University*

IT WAS, I believe, in 1942 that copies of Wiener's report on this subject, stamped CONFIDENTIAL, began to circulate among those concerned with the military applications, mainly to fire control, of prediction and smoothing. Like other books and reports with a yellow cover, it became known as "the yellow peril." It was followed by a host (at least a dozen to my knowledge) of similarly classified "simplifications" or "explanations" of the procedure. The volume under review consists of 123 pages devoted to a practically verbatim version of Wiener's original report, a five-page table of Laguerre functions, and 32 pages devoted to two of the simplifications, written by Norman Levinson, which had already appeared openly in the *Journal of Mathematics and Physics*.

Application examples

1) Prediction, interpolation and smoothing of signals





(a) $d = 1$ (b) $d = -1$ (c) $d = -1/2$

- Linear predictive coding (LPC) in speech processing.

- Here, $y(n)$ is the noisy measurements of a shifted version of the desired signal $x'(n)$. The noise $\omega(n)$ is also stationary.
- If the shift d is a positive integer, then, the optimal filter h_{opt} is the best linear d-step predictor of $x(n)$. For example, if $d = 1$, then, h_{opt} is a one-step predictor.
- If the shift is a negative integer, then, the optimal filter performs smoothing. For example, if $d = -1$, h_{opt} is the best linear one-lag smoother.
- For a rational d, the optimal filter is an interpolator aiming to estimate the (missing) sample between two consecutive data points. For example, for $d = -1/2$, the optimal filter is an interpolator trying to estimate the (missing) sample between $x(n)$ and $x(n-1)$.

Application examples

2) System identification

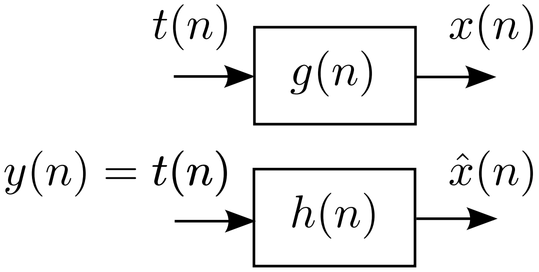


Figure 3: System identification using a training sequence $t(n)$ from an ergodic and stationary ensemble.

- Echo cancellation in full duplex data transmission.

- The optimal filtering framework can be used to solve system identification problems.
- In this example, the system to be identified is $g(n)$. First, a training sequence $t(n)$ is generated to drive the system. $t(n)$ is an instance from an independent and identically distributed (i.i.d) process, e.g., a white noise sequence. Thus, its time averages matches its ensemble averages (first and second order moments).
- The output of the system to this input is used as the desired signal in the optimal filtering problem.
- The optimal filter $h(n)$ that produces an output $\hat{x}(n)$ which is closest to $x(n)$ when driven by $t(n)$ will be the best linear time invariant approximation of $g(n)$.

Application examples

3) Inverse System identification

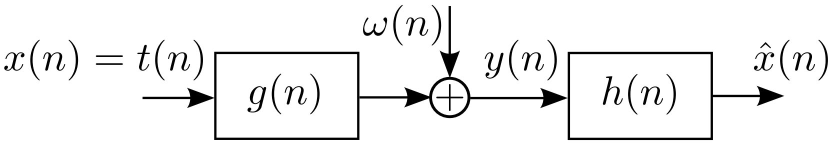


Figure 4: Inverse system identification using $x(n)$ as a training sequence.

- Channel equalisation in digital communication systems.

- The optimal filtering framework can be used to find the inverse of a system.
- In this example, the system to be inverted is $g(n)$ which is a channel that distorts the desired signal $x(n)$ sent by the transmitter side. The receiver side observes $y(n)$, which is a noisy version of the distorted signal.
- We would like to design a filter $h(n)$ which mitigates effects of $g(n)$ and rejects the noise $\omega(n)$ optimally.
- To do that, a training sequence $t(n)$ drives the channel. This sequence is an instance from an ergodic and stationary ensemble, i.e., $t(n)$ is randomly generated such that its time statistics match the ensemble averages of $x(n)$.
- Thus, the receiver side can find the estimation error $e(n)$ corresponding to any **h** by
$$e(n) = x(n) - \hat{x}(n) = x(n) - t(n)$$
- Hence, the optimal filter design problem for finding the best inverse system can be solved at the receiver side.

Optimal solution: Normal equations

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

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

- The optimisation problem for finding the optimal filter has an objective function – the MSE of estimation–, which is quadratic in the unknowns.
- Hence, a unique solution exists which can be characterised by the gradient of the objective - the vector of partial derivatives of the objective with respect to the unknowns. At the optimal point, the gradient equals to the zero vector.
- In the first step, the differentiation is moved into the expectation since expectation is a linear operator. In the following steps, well known rules of differentiation are used.
- Note that, we can evaluate the gradient for any given $\mathbf{h}$, and the error $e(n)$ inside the expectation corresponds to the chosen filter $\mathbf{h}$.

Optimal solution: Normal equations

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

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

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

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

- In order to find the optimal filter, we use its characterisation through its gradient, i.e., the optimal filter solves the set of equations

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

which are known as “the normal equations.”

Optimal solution: Wiener-Hopf equations

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

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

which is known as "the principle of orthogonality".

- Because the optimal filter solves the normal equations, its error $e_{opt}(n)$ satisfies the statistical orthogonality condition with the input variables $y(n-j)$ for $j = \dots, -2, -1, 0, 1, 2, \dots$.
- The geometric interpretion of the normal equations follows the statistical norm of the desired signal expressed in terms of the optimal estimate and the associated error:

$$\begin{aligned}
\langle x(n), x(n) \rangle &\triangleq E\left[(\hat{x}(n))^2\right] \\
&= E\left[(\hat{x}_{opt}(n) + e_{opt}(n))^2\right] \\
&= E\left[(\hat{x}_{opt}(n))^2 + 2\hat{x}_{opt}(n)e_{opt}(n) + (e_{opt}(n))^2\right] \\
&= E\left[(\hat{x}_{opt}(n))^2\right] + E\left[(e_{opt}(n))^2\right] \\
&\quad + 2\sum_{i=-\infty}^{\infty} h_{opt}(i) \underbrace{E\left[e_{opt}(n)y(n-i)\right]}_{=0 \text{ by the principle of orthogonality}} \\
&= E\left[(\hat{x}_{opt}(n))^2\right] + E\left[(e_{opt}(n))^2\right] \\
&= \langle \hat{x}_{opt}(n), \hat{x}_{opt}(n) \rangle + \langle e_{opt}(n), e_{opt}(n) \rangle. \qquad (6)
\end{aligned}$$

- Thus, the optimal estimate and the associated error are orthogonal and follow a Pythagorean relation with the desired signal $x(n)$.

Optimal solution: Wiener-Hopf equations

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

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

which is known as “the principle of orthogonality”.

- Furthermore,

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

- We expand the optimal error term $e_{opt}(n)$ inside the expection.
- After distributing $y(n-j)$ over the summation, and, using the linearity of the expectation operator, we obtain the last line which equals to zero by the principle of orthogonality.

Optimal solution: Wiener-Hopf equations

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

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

which is known as “the principle of orthogonality”.

- Furthermore,

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

Result (Wiener-Hopf equations)

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

- We obtain the Wiener-Hopf equations after carrying the summation term to the right hand side of the equation above and realising that

$$r_{xy}(j) = E\left[x(n)y(n-j)\right]$$

and

$$r_{yy}(i-j) = E\left[y(n-i)y(n-j)\right]$$

- Note that we consider real valued sequences throughout this presentation and omit complex conjugations above.

- When the equalities above are used together with the symmetricity of auto-correlation $r_{yy}(n) = r_{yy}(-n)$, the Weiner-Hopf equations can be written simply as

$$h_{opt}(n) * r_{yy}(n) = r_{xy}(n)$$

where we have the convolution of the optimal filter with the auto-correlation function of the observations on the left hand side, and, the cross-correlation sequence on the right hand side.

The Wiener filter

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

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

- We have not placed any constraints on the optimal filter so as to guarantee an impulse repsonse which can be finitely parameterised.
- Therefore, it is more convenient to consider an indirect characterisation of the optimal impulse response provided by the complex spectral domain (or, the z-transform domain).
- Let us consider the z-transform domain representation of the Wiener-Hopf equations.
- The multiplication of $H_{opt}(z)$ with the power spectral density (PSD) of the input equals to the complex spectra of the cross-correlation sequence.

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 is obtained in the complex spectral domain by the division of the cross-correlation complex spectra with the PSD of the input.
- The impulse response $h_{opt}(n)$ can be found, in principle, using the inverse z-transform:

$$h_{opt}(n) = \frac{1}{2\pi j} \oint_C H_{opt}(z) z^{n-1} \mathrm{d}z. \tag{6}$$

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.

- The region of convergence (ROC) of $H_{opt}(z)$ is not necessarily the outer region of a circle centered at the origin. Correspondingly, $h_{opt}(n)$ is not necessarily a right sided (causal) sequence.
- We assume that the processes we consider are regular, hence, the ROC of $H_{opt}(z)$ contain the unit circle on the z-plane. Correspondingly, $h_{opt}(n)$ is a stable sequence.
- For a process to be regular, its PSD $P(z = e^{j\omega})$ should not have extended regions along ω where it is zero.
- For the case, it can be shown that $P(z)$ can be factorised as

$$P(z) = \sigma^2 Q(z) Q^*(1/z^*)$$

where $Q(z)$ is a minimum-phase (causal) sequence and $Q^*(1/z^*)$ is its anti-causal counterpart.
- The task of identification of $Q(z)$ given $P(z)$ (and σ^2) is referred to as "spectral factorisation".

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)

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

- The “unconstrained” solution of the optimal filtering problem can be projected onto the space of causal IIR filters in a two step procedure.
- First, the causal and anti-causal factors of $P_{yy}(z)$ are identified. This factorisation splits the optimal filter as a cascade of two filters; one causal system followed by a non-causal one.
- The causal filter is then characterised by $H_{opt,1}(z) = \frac{1}{\sigma_y^2 Q_{yy}(z)}$.
- The second system has a non-causal impulse response (both left and right sided) with comlex spectra $H_{opt,2}(z) = \frac{P_{xy}(z)}{Q^*_{yy}(1/z^*)}$.
- Let $h_{opt,2}(n)$ denote the corresponding sequence.
- In order to find the projection of the optimal filter onto the space of causal IIR filters, the second filter is selected as the right sided part of $h_{opt,2}(n)$, i.e., $h'_{opt,2}(n) = h_{opt,2}(n)$, for $n = 0, 1, 2, \ldots$ and $h'_{opt,2}(n) = 0$, otherwise.
- This is often carried out in the spectral domain using partial fraction expansion.

FIR Wiener-Hopf equations

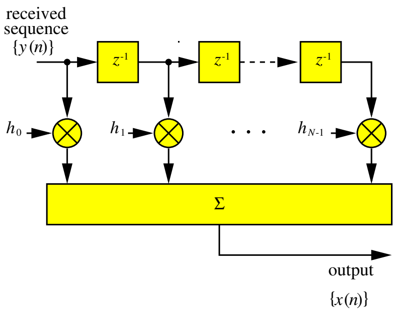


Figure 5: A finite impulse response (FIR) estimator.

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

Result (FIR Wiener-Hopf equations)

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

- Finite impulse response (FIR) filters are stable.
- FIR filters are causal without loss of generality, in that, they can always be cascaded to a delay line z^{-d} to have a causal overall response, where d is the length of the left sided part of the FIR impulse response.
- It is helpful for the designer to restrict the optimisation problem such that the space of LTI systems is constrained to the space of FIR filters as they naturally admit a finite parameterisation – N unknowns for an N-tap FIR filter.

FIR Wiener Filter

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

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

- FIR Wiener-Hopf equations specify a system of N equations in N unknowns.
- In order to solve this system, it is useful to consider the corresponding algebraic form.
- In this presentation, we assume real valued stationary processes. In the case of complex valued stationary processes, $\mathbf{r}_{xy}$ has $r^*_{xy}(l)$ for $l = 0, 1, 2, ..., N-1$ in its fields.
- Similarly $\mathbf{R}_{yy}$ is conjugate transpose symmetric (or, Hermitian symmetric). For example, the first column of $\mathbf{R}_{yy}$ has $r_{yy}(0), r^*_{yy}(1), \dots, r^*_{yy}(N-1)$ in its fields.

FIR Wiener Filter

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

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

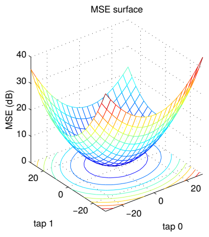
Result (FIR Wiener filter)

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

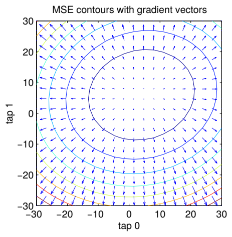
MSE surface

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

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



(a)



(b)

Figure 6: For a 2-tap Wiener filtering example: (a) the MSE surface, (b) gradient vectors.

- The MSE can be found as

$$\xi(\mathbf{h}) = \mathbf{h}^T \mathbf{R}_{yy}\mathbf{h} - 2\mathbf{h}^T \mathbf{r}_{xy} + E\left[(x(n))^2\right] \tag{7}$$

which can be written in the following quadratic form:

$$\begin{aligned}\xi(\mathbf{h}) &= (\mathbf{h} - \mathbf{h}_{opt})^T \mathbf{R}_{yy}(\mathbf{h} - \mathbf{h}_{opt}) + \xi(\mathbf{h}_{opt})\\ \xi(\mathbf{h}_{opt}) &= E\left[(x(n))^2\right] - \mathbf{h}_{opt}^T \mathbf{r}_{xy}\end{aligned} \tag{8}$$

- As $\xi(\mathbf{h})$ is a quadratic function of $\mathbf{h}$, it is smooth (its gradient is defined for all values of $\mathbf{h}$) and has a unique minimum.
- In the case of 2-dimensional $\mathbf{h}$, equal MSE lines are ellipses whose centre is the optimal filter vector and axes are along the eigenvectors of $\mathbf{R}_{yy}$. The major and minor semi-axis lengths are specified by the eigenvalues.

Example: Wiener equaliser

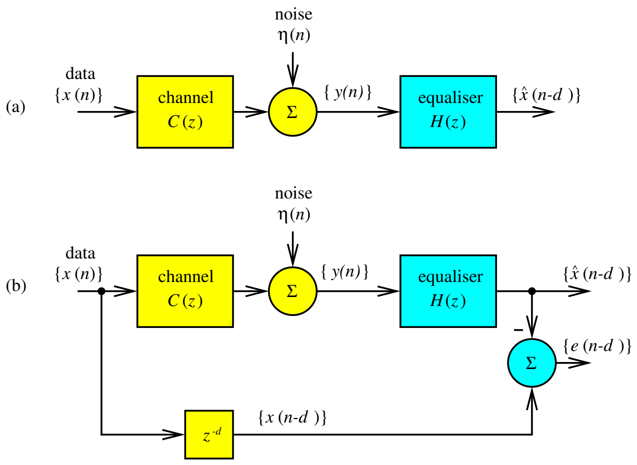


Figure 7: (a) The Wiener equaliser. (b) Alternative formulation.

- In this example, we consider optimal filtering for inverse system identification.
- A white random signal $x(n)$ is transmitted through a communication channel which distorts the signal with the transfer function $C(z)$.
- The receiver front-end receives noisy versions of the distorted signal.
- The goal of the equaliser is to optimally denoise the received signal $y(n)$ and mitigate the effects of distortion.

Wiener equaliser

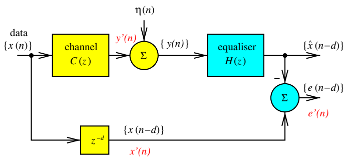


Figure 8: Channel equalisation scenario.

- For notational convenience define:

$$\begin{aligned} x'(n) &= x(n-d) \\ e'(n) &= x(n-d) - \hat{x}(n-d) \end{aligned} \tag{9}$$

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

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

- Let us define the desired signal as a delayed version of $x(n)$. Hence, the estimation error at time n will be $e'(n)$ as defined above.
- The channel output without noise is also a sequence with distinct properties, so, we will label it as $y'(n)$.

Wiener equaliser

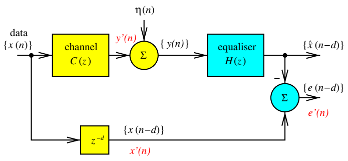


Figure 8: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}{}^{-1}\mathbf{r}_{x'y} \tag{10}$$

- Let us find the fields of the input autocorrelation matrix $\mathbf{R}_{yy}$ and the cross correlation vector $\mathbf{r}_{x'y}$.

Wiener equaliser

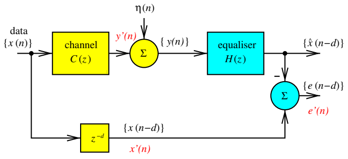


Figure 8: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}{}^{-1}\mathbf{r}_{x'y} \tag{10}$$

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

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

- The input autocorrelation of the equaliser is the sum of the autocorrelation of the channel output and that of the noise sequence.
- Since the noise sequence is white, its autcorrelation is Dirac's delta function weighted by the variance of the noise. The corresponding complex spectra equals to this variance for all z.

Wiener equaliser

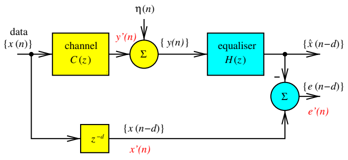


Figure 8: Channel equalisation scenario.

- 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

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

- Let us find the autocorrelation of the channel output in terms of that of the channel input $x(n)$ and the channel transfer function.

Wiener equaliser

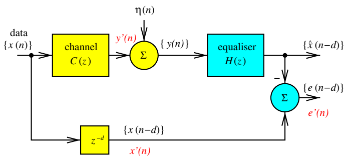


Figure 8: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}{}^{-1}\mathbf{r}_{x'y} \tag{11}$$

- We have found the complex spectra of the sequence that specifies $\mathbf{R}_{yy}$.
- Now, let us consider $\mathbf{r}_{x'y}$.

Wiener equaliser

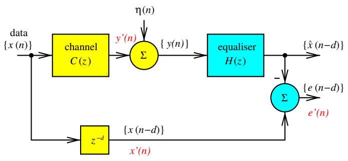


Figure 8: Channel equalisation scenario.

- Wiener filter

$$\mathbf{h}_{opt} = \mathbf{R}_{yy}{}^{-1}\mathbf{r}_{x'y} \tag{11}$$

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

$$\begin{aligned} r_{x'y}(j) &= E\left[x'(n)y(n-j)\right] \\ &= E\left[x(n-d)(y'(n-j)+\eta(n-j))\right] \\ &= r_{xy'}(j-d) \\ &\leftrightarrow P_{x'y}(z) = P_{xy'}(z)z^{-d} \end{aligned} \tag{12}$$

- We have found the complex spectra of the sequence that specifies $\mathbf{r}_{x'y}$ in terms of $P_{xy'}$.
- Next, we specify this cross correlation sequence.

Wiener equaliser

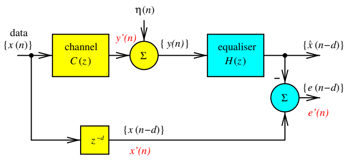


Figure 8: Channel equalisation scenario.

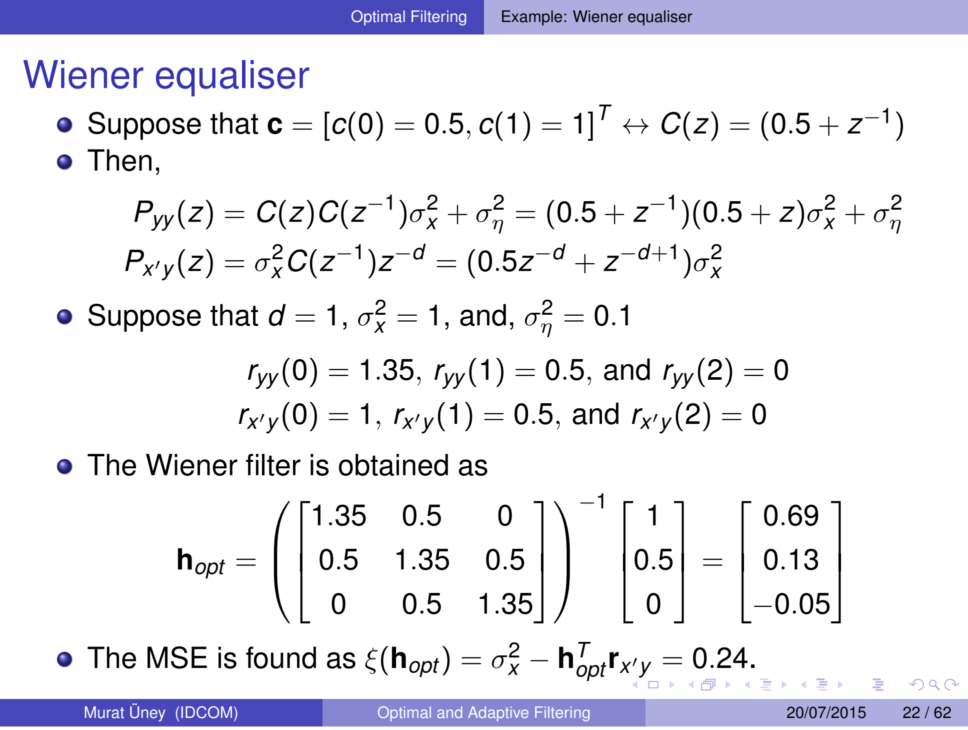
- Remember $y'(n) = c(n) * x(n)$

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

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

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

- We have found the complex spectra of the sequence of concern, in terms of the input auto-correlation and the channel transfer function.



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

- Since we have found all the required quantities that specify Wiener FIR filter in terms of the complex spectra of the input autocorrelation and the transfer function, we can solve the optimal filtering problem for any selection of these functions.
- An example channel response is given in the slide.
- The solution follows trivially from our previous derivations.

Adaptive filtering - Introduction

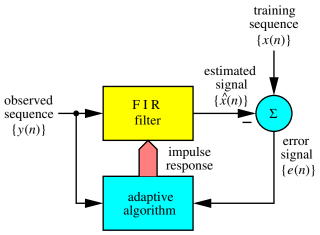


Figure 9: Adaptive filtering configuration.

- For notational convenience, define

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

- The output of the adaptive filter is

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

- Optimum solution

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

- In our previous treatment, the optimal filter design was an offline procedure. The filter works in an open-loop fashion, without any mechanism to compansate for any changes in the second order statistics used for design, during its operation.
- Adaptive filters provide a feedback mechanism to adjust the filter to the actual working conditions. Hence, they are closed-loop systems.
- They can be viewed as strategies to find the optimal filter online, in an iterative fashion, during the operation of the filter.

Recursive least squares

- Minimise cost function

$$\xi(n) = \sum_{k=0}^{n} (x(k) - \hat{x}(k))^2 \tag{13}$$

- 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

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

- In optimal filtering, we considered MSE as the cost function to be minimised.
- Let us choose a cost function which does not involve expectations and can be computed using the desired signal -or, the training sequence- $x(n)$ and its estimates $\hat{x}(n)$.
- The sum of squared error terms over time (13) is such a cost function.
- Let us use the notation $\mathbf{e}(n) = [e(0), e(1), \ldots, e(n)]^T$. It can easily be seen that $\xi(n) = \|\mathbf{e}(n)\|^2 = \mathbf{e}(n)^T\mathbf{e}(n)$.
- The LS “autocorrelation” matrix and “cross-correlation” vector can be found after expanding the error vector in the form given in Eq.(4), and, taking the gradient of this expression with respect to $\mathbf{h}(n)$:

$$\mathbf{e} = \mathbf{x} - \mathbf{Y}\mathbf{h}$$
$$\nabla_{\mathbf{h}}\mathbf{e}^T\mathbf{e} = -2\mathbf{x}^T\mathbf{Y} + 2\mathbf{Y}^T\mathbf{Y}\mathbf{h}$$

- Moreover, if the signals involved are ergodic (i.e., if the time averages and the ensemble averages are the same), then

$$\xi(\mathbf{h}) = E\left[(e(n))^2\right] = \lim_{N\to\infty} \frac{1}{N+1}\sum_{k=0}^{N}(x(k) - \hat{x}(k))^2.$$

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

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

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

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

- Let us consider the recursive relationships that will allow us to derive online update rules for the filter coefficients.

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

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

- The inverse of the autocorrelation matrix at time *n* can further be found in terms of the previous inverse, and the current observation vector.
- Sherman-Morrison identity gives the inverse of a matrix which can be written as the sum of a matrix with a vector outer product.

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 $n = 1, 2, 3, \ldots$ do    ▷ Iterations
4:     $\hat{x}(n) = \mathbf{h}^T(n-1)\mathbf{y}(n)$    ▷ Estimate $x(n)$
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
```

- The steps of the resulting algorithm are given above.

Stochastic gradient algorithms

- MSE contour - 2-tap example:

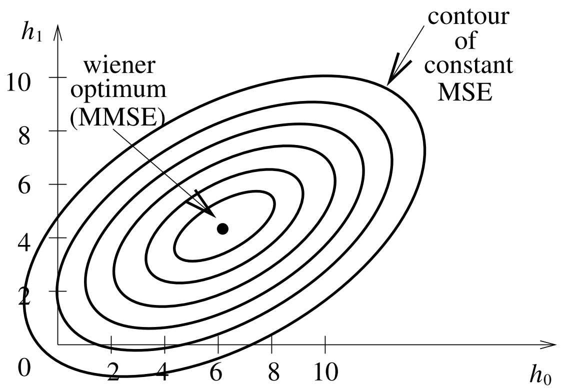
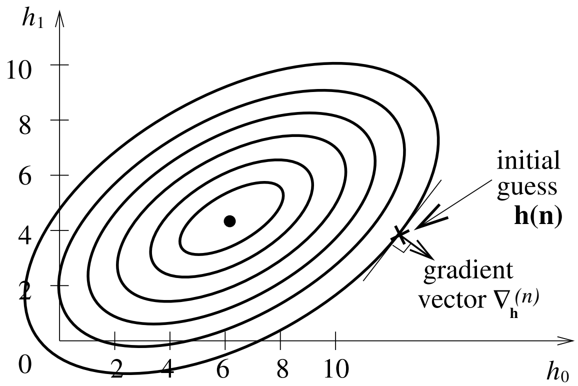


Figure 10: Method of steepest descent.

- Another approach that would iteratively converge to the optimal filter would draw from the descent directions approaches in the optimisation literature.
- A well known procedure - steepest descent - starts with an initial point and moves along the inverse direction of the gradient at that point in order to find the minimiser of a function.

Steepest descent

- MSE contour - 2-tap example:



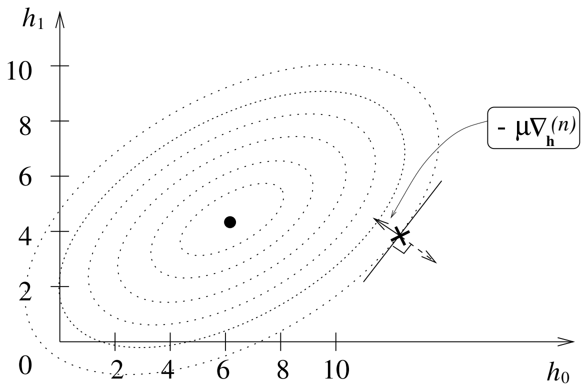
- The gradient vector

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

- Let us assume that we can evaluate the gradient of the MSE for any FIR filter **h**.
- In this case, we can find the optimal filter coefficients, nevertheless, let us try to consider the steepest descent procedure in order to start with an initial filter and iteratively converge to the optimal one.
- In this respect, let us consider *n* as the iteration counter –not as the time index – for now.

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 μ.

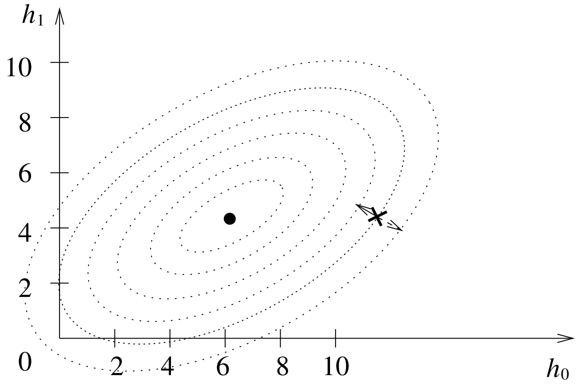
- At each iteration n, we move along the inverse direction of the MSE gradient.
- One way to perform this move is to make a line search along that direction using, for example, the golden section search, in order to solve a 1-D optimisation problem.

$$\lambda^* = \arg\min_{\lambda} \xi(\mathbf{h}(n) - \lambda\nabla_{\mathbf{h}}(n))$$

- However, this would bring an additional computational cost. Instead, let us select a fixed step size μ and use it as the optimal step size $\lambda^* = \mu$. We will show that, under certain conditions, this selection still results with convergence to the optimal point.

Steepest descent

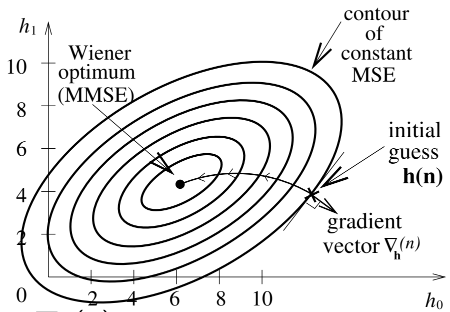
- MSE contour - 2-tap example:



- Gradient at new guess.

Convergence of steepest descent

- MSE contour - 2-tap example:



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

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

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{14}$$

- Because the MSE is a quadratic function of the filter coefficients, equal MSE contours are multidimensional elliptic structures (for example, for the 3-dimensional case, ellipsoids). The eigenvectors of the autocorrelation matrix specify the principle axes, and the eigenvalues specify how stretched these surfaces are.
- With a fixed step size μ, the distance from the optimal point decreases, provided that μ is smaller than the inverse of the largest eigenvalue.

Stochastic gradient algorithms

- A time recursion:

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

- The exact gradient:

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

- A simple estimate of the gradient

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

- The error

$$e(n+1) = x(n+1) - \mathbf{h}(n)^T\mathbf{y}(n+1) \qquad (15)$$

- Stochastic gradient algorithms replace the gradient in gradient descent procedures with a “noisy” estimate.
- The simplest guess of the gradient would ignore the expectation and use the instantenous values of the variables involved.

The Least-mean-squares (LMS) algorithm:

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

- The resulting algorithm is known as the least mean square (LMS) algorithm.
- The steps of the algorithm are as given.

LMS block diagram

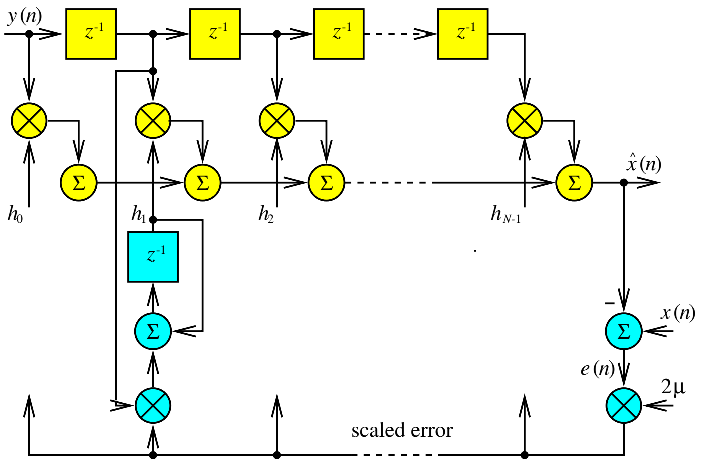
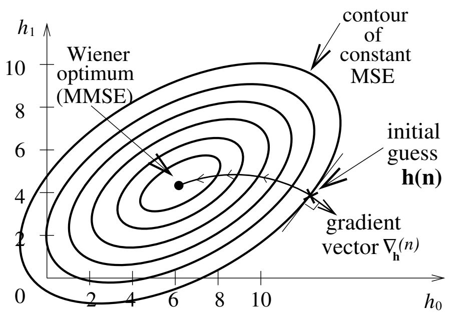


Figure 11: Least mean-square adaptive filtering.

- LMS algorithm admits a computational structure convenient for hardware implementations.

Convergence of the LMS

- MSE contour - 2-tap example:



- Eigenvalues of $\mathbf{R}_{yy}$ (in this example, λ_0 and λ_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 $0 < \mu < \frac{1}{3N\sigma_y^2}$

- Because the gradient in the steepest descent is replaced with its noisy estimate in the LMS algorithm, the study of its convergence behaviour slightly different from that for the steepest descent algorithm.

Eigenvalue ratio (EVR)

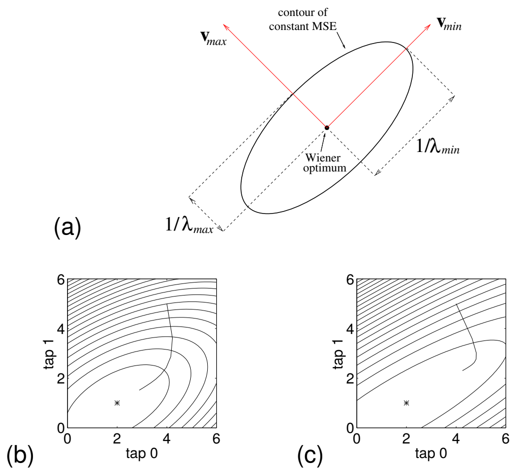


Figure 12: Eigenvectors, eigenvalues and convergence: (a) the relationship between eigenvectors, eigenvalues and the contours of constant MSE; (b) steepest descent for EVR of 2; (c) EVR of 4.

Comparison of RLS and LMS

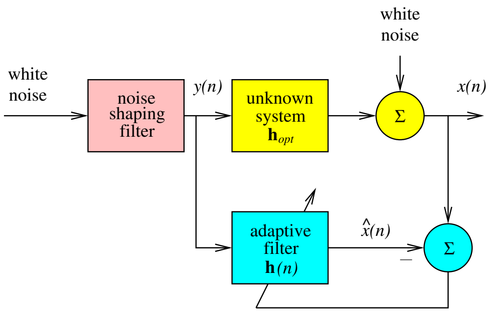


Figure 13: Adaptive system identification configuration.

- Error vector norm

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

- In this example, we would like to identify the unknown system. Its impulse response is, hence, the optimal solution.
- The noise shaping filter allows us to change the EVR by colorating the white noise at its input.
- It is often than not the case that we can only have noisy measurements from the system to be identified. The additive white noise in the block diagram is used to model this aspect.

Comparison: Performance

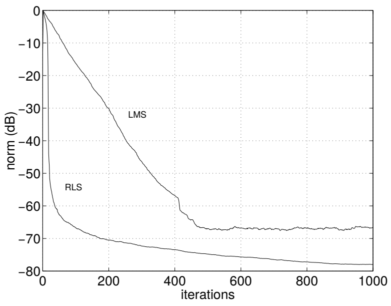


Figure 14: Covergence plots for $N = 16$ taps adaptive filtering in the system identification configuration: EVR = 1 (i.e., the impulse response of the noise shaping filter is $\delta(n)$).

Comparison: Performance

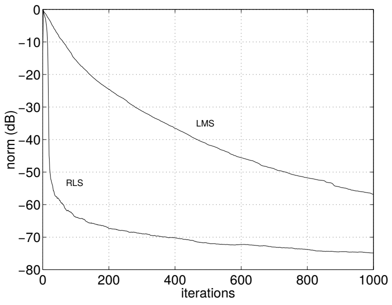


Figure 15: Covergence plots for $N = 16$ taps adaptive filtering in the system identification configuration: EVR = 11.

Comparison: Performance

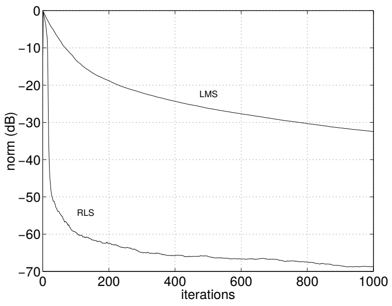


Figure 16: Covergence plots for $N = 16$ taps adaptive filtering in the system identification configuration: EVR (and, correspondingly the spectral coloration of the input signal) progressively increases to 68.

Comparison: Complexity

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

Algorithm class	*Implementation*	*Computational load*		
		multiplications	*adds/subtractions*	*divisions*
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

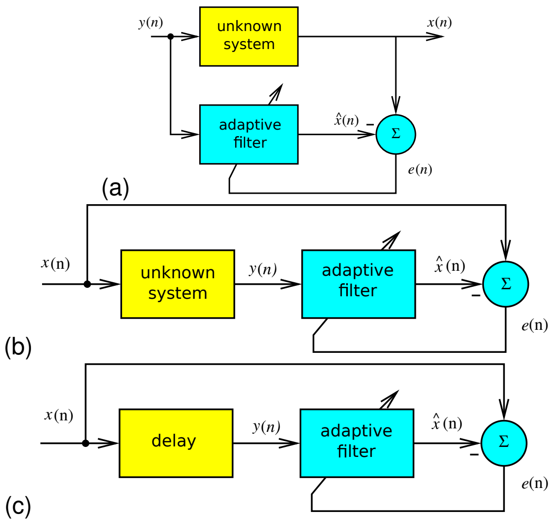
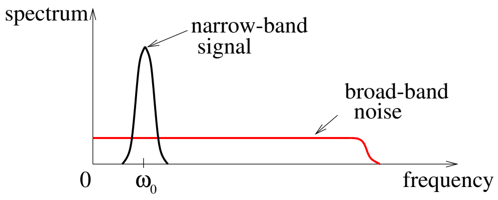


Figure 17: Adaptive filtering configurations: (a) direct system modelling; (b) inverse system modelling; (c) linear prediction.

Adaptive line enhancement

(a)



(b)

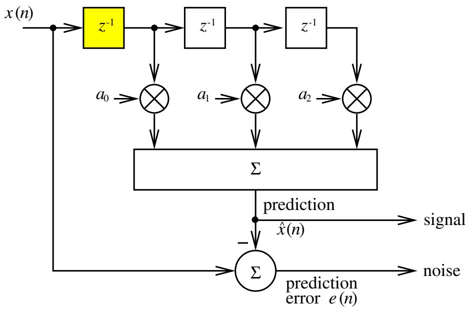
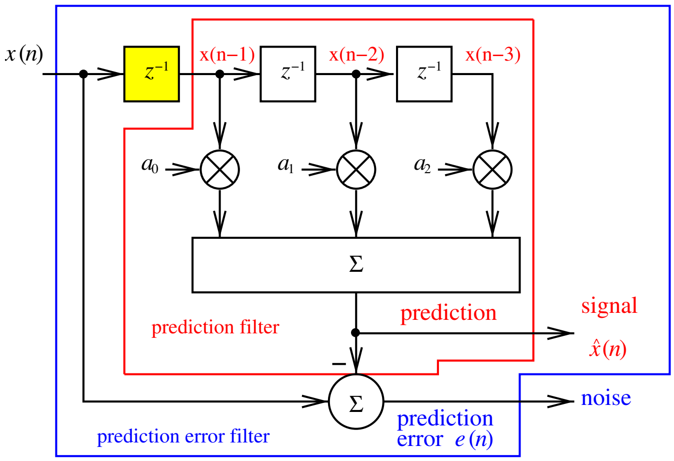


Figure 18: Adaptive line enhacement: (a) signal spectrum; (b) system

Adaptive predictor



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

Adaptive tone suppression

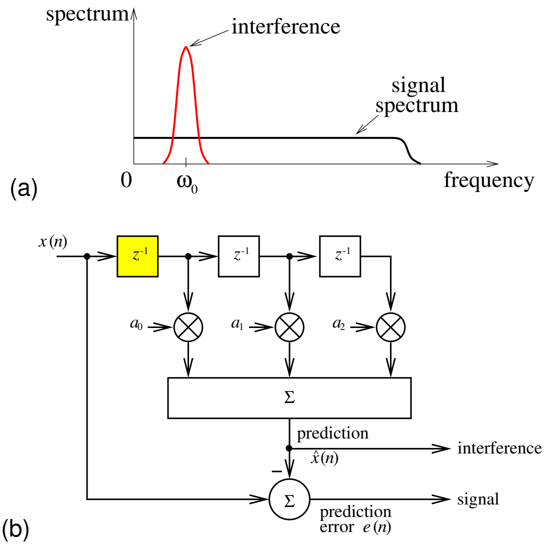
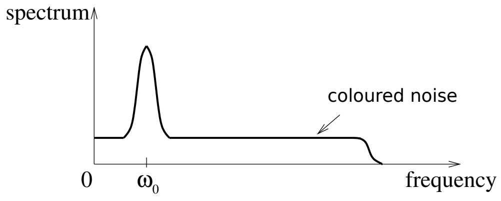


Figure 19: Adaptive tone suppression: (a) signal spectrum; (b) system

Adaptive noise whitening

(a)



(b)

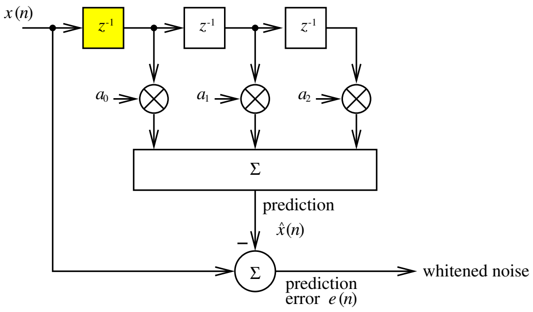
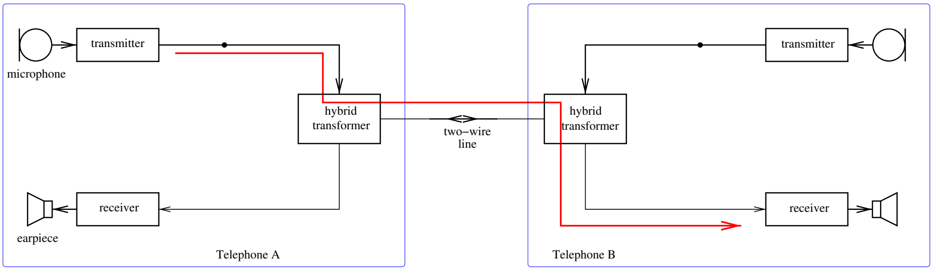


Figure 20: Adaptive noise whitening: (a) input spectrum; (b) system

Echo cancellation

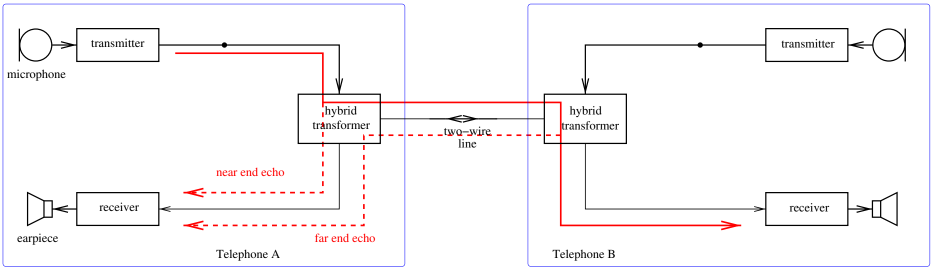
- A typical telephone connection



- Hybrid transformers to route signal paths.

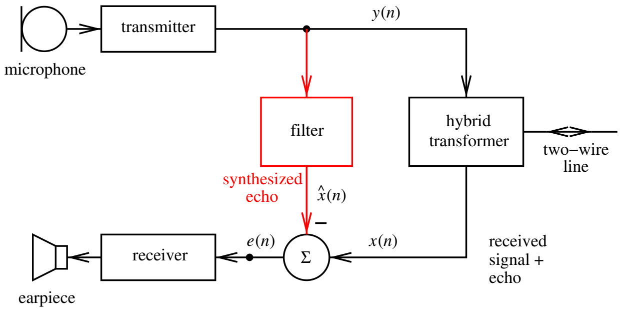
Echo cancellation (contd)

- Echo paths in a telephone system



- Near and far echo paths.

Echo cancellation (contd)



- Fixed filter?

Echo cancellation (contd)

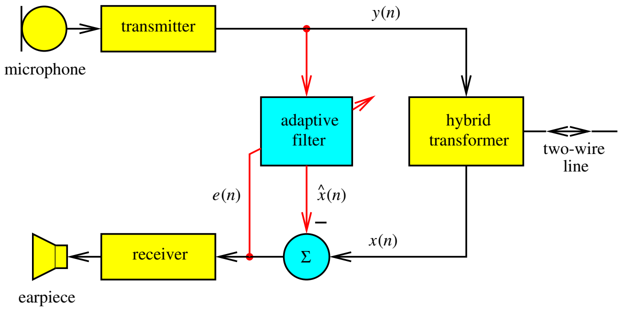


Figure 21: Application of adaptive echo cancellation in a telephone handset.

Channel equalisation

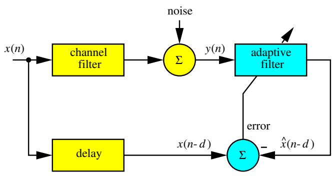


Figure 22: Adaptive equaliser system configuration.

- 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

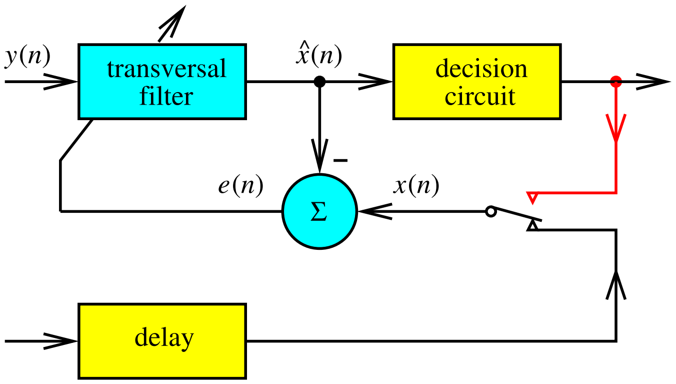


Figure 23: Decision directed equaliser.

Optimal signal detection

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

$$\begin{aligned} H_0 &: y(n) = \eta(n) \\ H_1 &: y(n) = x(n) + \eta(n) \end{aligned} \tag{16}$$

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

$$n = 0, 1, 2, ..., N - 1$$

- Vector notation

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

- Our discussion so far, has been on optimal estimation of signals.
- We did not explicitly use probability distributions over signals - instead, we used second order characterisations of ensembles, i.e., auto-correlation and cross-correlation sequences.
- The sense of optimality has been the expection of the squared estimation error.
- If the variable we would like to estimate takes values from a countable and finite set, the problem setting is referred to as a hypothesis testing problem. For M possible values of the variable, we have a M-ary hypothesis testing problem. A binary hypothesis testing problem in which case $M = 2$ is referred to as a detection problem.
- Let us consider the detection problem in (16). In a sense, we are asked, of two possible ensembles, which one generated the observations $y(n)$.
- This problem arises in digital communications, where the signal $x(n)$ encodes data. In radar and sonar applications, $x(n)$ is the signal we expect to see at the receiver front-end if there exists a reflector within the region where the signal is collected from.

Bayesian hypothesis testing

- Having observed $\mathbf{y} = \bar{y}$, find the probabilities of H_1 and H_0 and decide on the hypothesis with the maximum value.
- Equivalently,
  i) consider a random variable $H \in \{H_0, H_1\}$ and find its posteriori distribution:

$$P(H = H_i|\bar{y}) = \frac{p(\bar{y}|H = H_i)p(H = H_i)}{p(\bar{y}|H = H_0)P(H = H_0) + p(\bar{y}|H = H_1)P(H = H_1)}$$

  for $i = 0, 1$.
  ii) Find the *maximum a-posteriori* (MAP) estimate of H.

$$\hat{H} = \arg\max_{H} p(H|\bar{y})$$

Detection of deterministic signals - white Gaussian noise

- $\mathbf{x}$ is a known vector, $\boldsymbol{\eta} \sim \mathcal{N}(.; \mathbf{0}, \sigma^2 \mathbf{I})$.
- MAP decision as a likelihood ratio test:

$$p(H_1|\bar{y}) \underset{H_0}{\overset{H_1}{\gtrless}} p(H_0|\bar{y})$$

$$p(\bar{y}|H_1)P(H_1) \underset{H_0}{\overset{H_1}{\gtrless}} p(\bar{y}|H_0)P(H_0)$$

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

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

Detection of deterministic signals - AWGN (contd)

- The numerator and denominator of the likelihood ratio are

$$\begin{aligned} p(\bar{y}|H_1) &= \mathcal{N}(\bar{y} - \mathbf{x}; \mathbf{0}, \sigma^2\mathbf{I}) \\ &= \frac{1}{(2\pi\sigma^2)^{N/2}} \prod_{n=0}^{N-1} \exp\left\{ -\frac{(\bar{y}(n) - x(n))^2}{2\sigma^2} \right\} \\ &= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{n=0}^{N-1} (\bar{y}(n) - x(n))^2 \right) \right\} \end{aligned} \quad (17)$$

- Similarly

$$\begin{aligned} p(\bar{y}|H_0) &= \mathcal{N}(\bar{y}; \mathbf{0}, \sigma^2\mathbf{I}) \\ &= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{n=0}^{N-1} (\bar{y}(n))^2 \right) \right\} \end{aligned} \quad (18)$$

- Therefore

$$\frac{p(\bar{y}|H_1)}{p(\bar{y}|H_0)} = \exp\left\{ \frac{1}{\sigma^2} \left(\sum_{n=0}^{N-1} (\bar{y}(n)x(n) - \frac{1}{2}x(n)^2) \right) \right\} \quad (19)$$

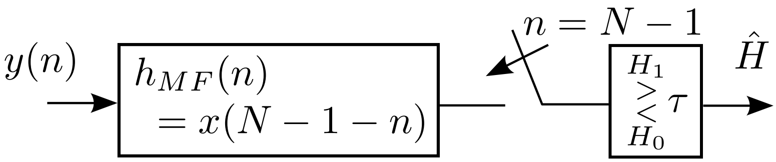
Detection of deterministic signals - AWGN (contd)

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

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

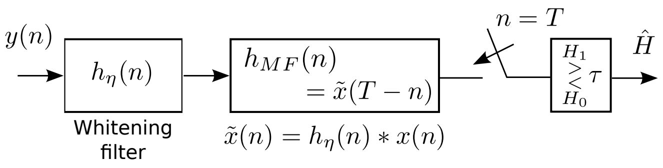
- Now, we have a linear statistical test

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



Detection of deterministic signals under coloured noise

- For the case, $\boldsymbol{\eta} \sim \mathcal{N}(.; \mathbf{0}, \mathbf{C}_{\eta})$.



$y(n)$
$h_\eta(n)$
Whitening filter
$h_{MF}(n) = \tilde{x}(T-n)$
$\tilde{x}(n) = h_\eta(n) * x(n)$
$n = T$
$H_1 \gtrless \tau$ H_0
$\hat{H}$

- The whitening filter can be designed in the optimal filtering framework or an adaptive algorithm can be used.
- The results on optimal detection under white Gaussian noise holds for the signal $x(n) * h_\eta(n)$.

- In the case that the noise is non-white, i.e., its autocorrelation sequence is not a scaled version of Dirac's delta function, the previous results do not apply immediately.
- Nevertheless, the linearity and commutativity of LTI systems can be used to show that, we can still use the results for detection under white noise if we use a "whitening filter" as a pre-processing stage.
- Note that, in this case, the signal to be detected will be the convolution of the whitening filter with the original signal $x(n)$.
- Design of a whitening filter is an optimal filtering problem and was discussed in this presentation in both offline and adaptive settings.
- Thus, the solution to this problem draws from all the methods we have presented throughout.

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.