DSP-CIS

Part-III : Optimal & Adaptive Filters

Chapter-7 : Optimal Filters - Wiener Filters

Marc Moonen
Dept. E.E./ESAT-STADIUS, KU Leuven
marc.moonen@kuleuven.be
www.esat.kuleuven.be/stadius/

Part-III : Optimal & Adaptive Filters

Chapter-7 **Optimal Filters - Wiener Filters**
- Introduction : General Set-Up & Applications
- Wiener Filters

Chapter-8 **Adaptive Filters - LMS & RLS**
- Least Means Squares (LMS) Algorithm
- Recursive Least Squares (RLS) Algorithm

Chapter-9 **Square Root & Fast RLS Algorithms**
- Square Root Algorithms
- Fast Algorithms

Chapter-10 **Kalman Filters**
- Introduction – Least Squares Parameter Estimation
- Standard Kalman Filter
- Square Root Kalman Filter

Introduction : General Set-Up

1. 'Classical' Filter Design

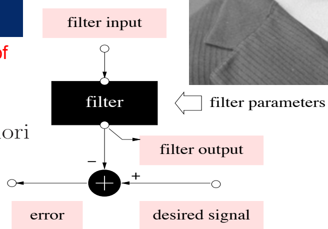
lowpass/bandpass/notch filters/...

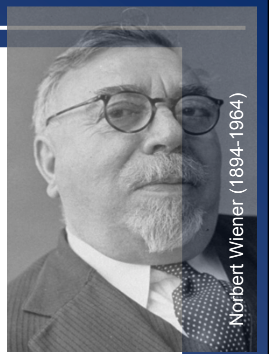
See Part-II

2. 'Optimal' Filter Design

- signals are viewed as realizations of *stochastic processes* (H249-HB78)
- filter optimisation/design in a *statistical sense* based on a priori *statistical information*

→ *Wiener filters*

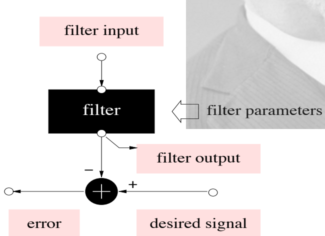


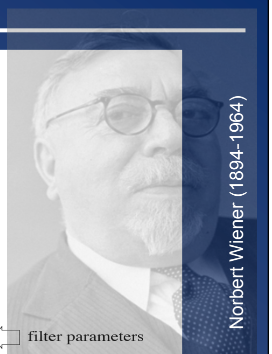


Introduction : General Set-Up

Prototype optimal filtering set-up :

Design filter such that for a given (i.e. 'statistical info available') input signal, filter output signal is 'optimally close' (to be defined) to a given 'desired output signal'.





Introduction : General Set-Up

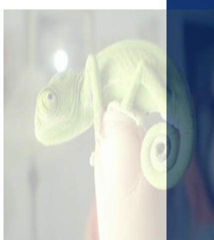
when a priori statistical information is not available :

3. 'Adaptive' Filters

- self-designing
- adaptation algorithm to monitor environment
- properties of adaptive filters :

  convergence/tracking
  numerical stability/accuracy/robustness
  computational complexity
  hardware implementation

Introduction : General Set-Up

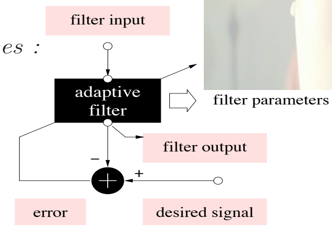


Prototype adaptive filtering set-up :

Basic operation involves 2 processes :

1. *filtering process*
2. *adaptation process*

   adjusting filter parameters to
   (time-varying) environment
   adaptation is steered by error signal

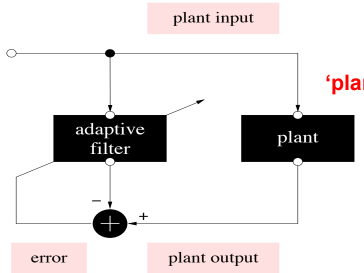


- Depending on the application, either the filter parameters, the filter output or the error signal is of interest

Introduction : Applications

Introduction : Applications

system identification/modeling



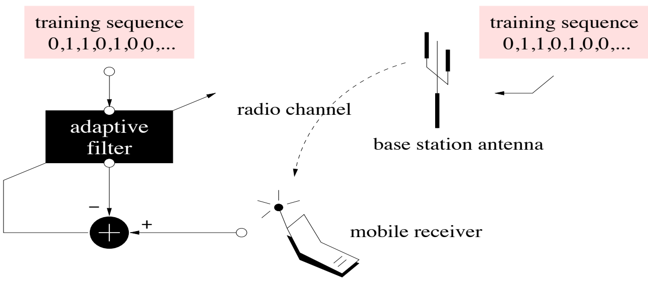
'plant' can be any system

Optimal/adaptive filter to provide mathematical model
for input/output-behavior of the 'plant'

Introduction : Applications

Introduction : Applications

example : channel identification

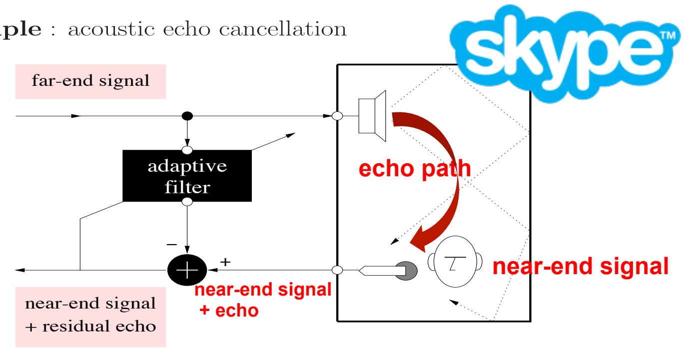


Optimal/adaptive filter to provide mathematical model
for signal propagation in a radio channel, from transmitter to receiver

Introduction : Applications

Introduction : Applications

example : acoustic echo cancellation

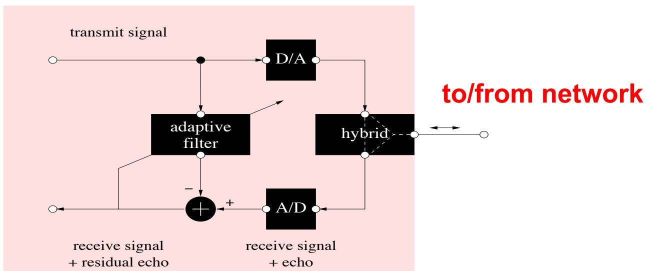


Optimal/adaptive filter to provide mathematical model
for signal propagation in acoustic channel, from loudspeaker to microphone

Introduction : Applications

Introduction : Applications

example : echo cancellation in full-duplex modems

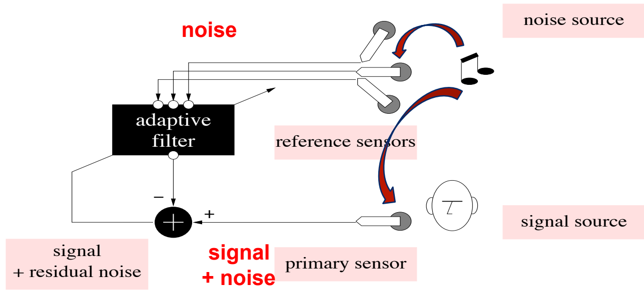


'Hybrid' is never ideally matched to line impedance, hence generates echo of transmitted signal into received signal
Optimal/adaptive filter to model echo path, from transmitter into receiver

Introduction : Applications

Introduction : Applications

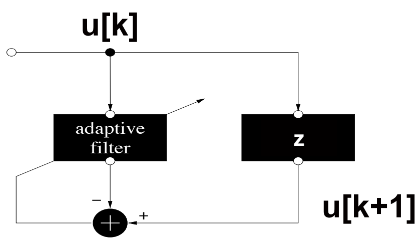
example : acoustic noise cancellation



Introduction : Applications

Introduction : Applications

Example: Linear Prediction

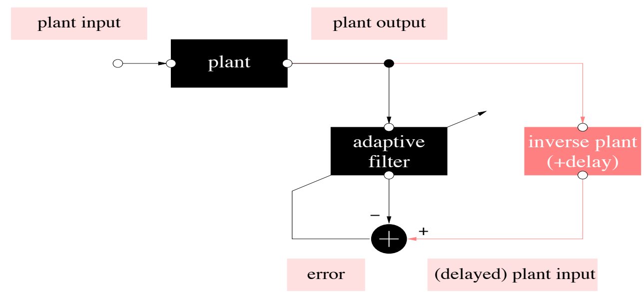


Optimal/adaptive filter to provide prediction model, predicting next sample u[k+1] from previous samples u[k], u[k-1], ⋯,u[k-L]
Used in speech codecs, etc⋯

Introduction : Applications

Introduction : Applications

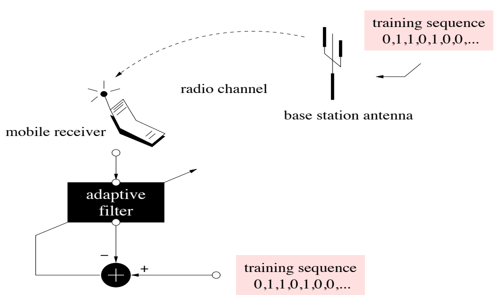
Inverse modeling



Introduction : Applications

Introduction : Applications

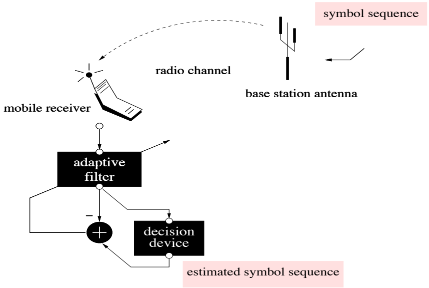
example : channel equalization (training mode)



Introduction : Applications

Introduction : Applications

example : channel equalization (decision-directed mode)



Optimal Filtering : Wiener Filters

Optimal filtering/ Wiener filters

Prototype optimal filter revisited

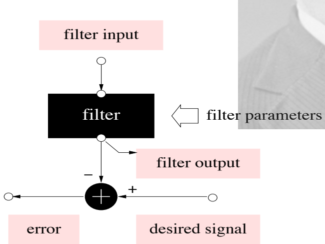
Have to decide on 2 things..

1 *filter structure ?*

→ FIR filters
(=pragmatic choice)

2 *cost function ?*

→ quadratic cost function
(=pragmatic choice)



Optimal Filtering : Wiener Filters

1 **Will use**

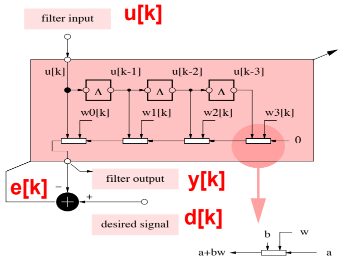
FIR filters (=tapped-delay line filter/'transversal' filter)

$$y_k = \sum_{l=0}^{L} w_l . u_{k-l} = \mathbf{w}^T . \mathbf{u}_k = \mathbf{u}_k^T . \mathbf{w}$$

where

$$\mathbf{w}^T = \begin{bmatrix} w_0 & w_0 & \ldots & w_L \end{bmatrix}$$

$$\mathbf{u}_k^T = \begin{bmatrix} u_k & u_{k-1} & \ldots & u_{k-L} \end{bmatrix}$$

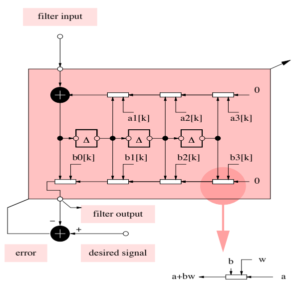


PS: Shorthand notation u_k=u[k], y_k=y[k], d_k=d[k], e_k=e[k],
Filter coefficients ('weights') are w_l (replacing b_l of previous chapters)
For adaptive filters w_l also have a time index $w_l[k]$

Optimal Filtering : Wiener Filters

Note : generalization to *IIR (infinite impulse response)* is non-trivial

- convergence problems
- stability problems

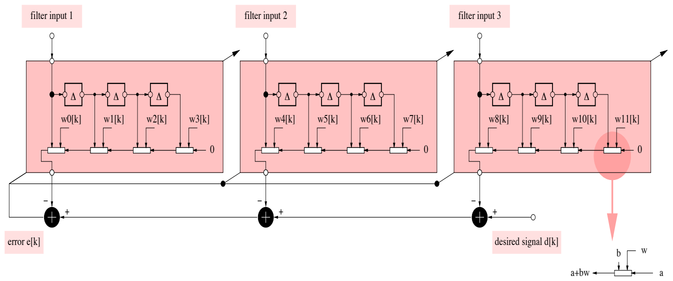


Note : generalization to *non-linear filters* not treated here

Optimal Filtering : Wiener Filters

PS: Can generalize FIR filter to ‘multi-channel FIR filter’

example: see page 11



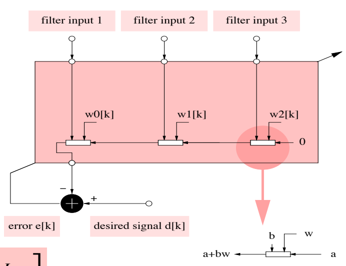
Optimal Filtering : Wiener Filters

PS: Special case of ‘multi-channel FIR filter’ is ‘linear combiner’

$$y_k = \mathbf{u}_k^T \mathbf{w}$$

where

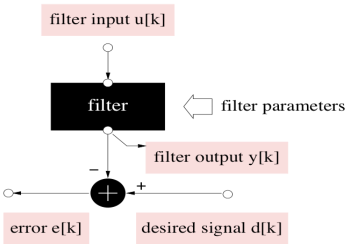
$$\mathbf{u}_k^T = \begin{bmatrix} u_k^0 & u_k^1 & \dots & u_k^L \end{bmatrix}$$



FIR filter may then also be viewed as special case of ‘linear combiner’
where input signals are delayed versions of each other

Optimal Filtering : Wiener Filters

2 **Will use**



Quadratic cost function :

minimum mean-square error (MMSE) criterion

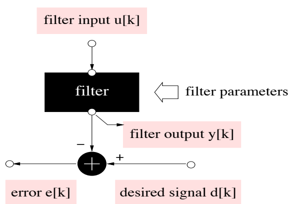
= minimize

$$J_{MSE}(\mathbf{w}) = \mathrm{E}\left\{e_k^2\right\} = \mathrm{E}\left\{\left(d_k - y_k\right)^2\right\} = \mathrm{E}\left\{\left(d_k - \mathbf{u}_k^T\mathbf{w}\right)^2\right\}$$

$\mathcal{E}\{x\}$ is 'expected value' (mean) of x

Optimal Filtering : Wiener Filters

MMSE cost function can be expanded as…



$$J_{MSE}(\mathbf{w}) = \mathrm{E}\left\{e_k^2\right\}$$

$$= \mathrm{E}\left\{\left(d_k - \mathbf{u}_k^T\mathbf{w}\right)^2\right\}$$

$$= \mathcal{E}\{d_k^2\} + \mathbf{w}^T \underbrace{\mathcal{E}\{\mathbf{u}_k\mathbf{u}_k^T\}}_{\bar{\mathbb{X}}_{uu}} \mathbf{w} - 2\mathbf{w}^T \underbrace{\mathcal{E}\{\mathbf{u}_k d_k\}}_{\bar{\mathbb{X}}_{du}} .$$

$\bar{\mathbb{X}}_{uu}$ = *correlation matrix* $\bar{\mathbb{X}}_{du}$ = *cross-correlation vector*

Optimal Filtering : Wiener Filters

Correlation matrix has a special structure…

for a stationary discrete-time stochastic process $\{u_k\}$:

autocorrelation coefficients : $\bar{x}_{uu}(\delta) = \mathcal{E}\{u_k \cdot u_{k-\delta}\}$

correlation matrix :

with $\mathbf{u}_k^T = \begin{bmatrix} u_k & u_{k-1} & \ldots & u_{k-L} \end{bmatrix}$

$$\bar{\aleph}_{uu} = \mathrm{E}\{\mathbf{u}_k.\mathbf{u}_k^T\} = \begin{bmatrix} \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \bar{x}_{uu}(2) & \cdots & \bar{x}_{uu}(L) \\ \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \cdots & \bar{x}_{uu}(L-1) \\ \bar{x}_{uu}(2) & \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \cdots & \bar{x}_{uu}(L-2) \\ \vdots & \vdots & \vdots & & \vdots \\ \bar{x}_{uu}(L) & \bar{x}_{uu}(L-1) & \bar{x}_{uu}(L-2) & \cdots & \bar{x}_{uu}(0) \end{bmatrix}$$

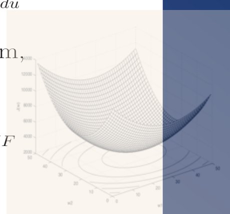
i.e. symmetric & Toeplitz & non-negative definite

Optimal Filtering : Wiener Filters

MMSE cost function can be expanded as…(continued)

$$J_{MSE}(\mathbf{w}) = \mathcal{E}\{d_k^2\} + \mathbf{w}^T \underbrace{\mathcal{E}\{\mathbf{u}_k\mathbf{u}_k^T\}}_{\bar{\mathbb{X}}_{uu}} \mathbf{w} - 2\mathbf{w}^T \underbrace{\mathcal{E}\{\mathbf{u}_k d_k\}}_{\bar{\mathbb{X}}_{du}}.$$

cost function is convex, with a (mostly) unique minimum, obtained by setting the gradient equal to zero:



$$0 = [\frac{\partial J_{MSE}(\mathbf{w})}{\partial \mathbf{w}}]_{\mathbf{w}=\mathbf{w}_{WF}} = [2\bar{\mathbb{X}}_{uu}\mathbf{w} - 2\bar{\mathbb{X}}_{du}]_{\mathbf{w}=\mathbf{w}_{WF}}$$

Wiener-Hopf equations :

$\bar{\mathbb{X}}_{uu} \cdot \mathbf{w}_{WF} = \bar{\mathbb{X}}_{du}$ $\rightarrow$ $\mathbf{w}_{WF} = \bar{\mathbb{X}}_{uu}^{-1}\bar{\mathbb{X}}_{du}$ …simple enough!

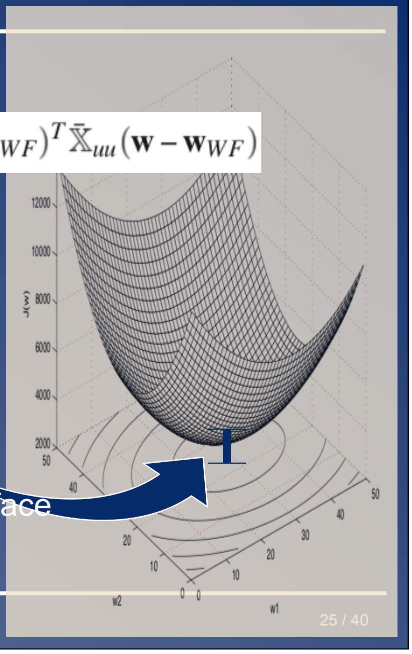
This is the 'Wiener Filter' solution

Optimal Filtering : Wiener Filters

PS: Can easily verify that

$$J_{MSE}(\mathbf{w}) = J_{MSE}(\mathbf{w}_{WF}) + (\mathbf{w}-\mathbf{w}_{WF})^T \bar{\mathbb{X}}_{uu}(\mathbf{w}-\mathbf{w}_{WF})$$

Example L=1



Bowl-shaped error performance surface
where X_{uu} defines shape of the bowl

Optimal Filtering : Wiener Filters

PS: Can easily verify that

$$[\mathcal{E}\{\mathbf{u}_k \cdot e_k\}]_{\mathbf{w}=\mathbf{w}_{WF}} = \mathcal{E}\{\mathbf{u}_k \cdot d_k\} - \mathcal{E}\{\mathbf{u}_k \mathbf{u}_k^T\}\mathbf{w}_{WF} = \bar{\mathbb{X}}_{du} - \bar{\mathbb{X}}_{uu}\mathbf{w}_{WF} = 0.$$

This is referred to as the '**orthogonality principle**'
i.e. the error signal for the optimal filter is orthogonal to the input signals used for the estimation

As a corollary, the error signal signal is also orthogonal to the optimal filter output

$$[\mathcal{E}\{y_k \cdot e_k\}]_{\mathbf{w}=\mathbf{w}_{WF}} = \mathcal{E}\{\mathbf{w}_{WF}^T \mathbf{u}_k \cdot e_k\} = 0.$$

Optimal Filtering : Wiener Filters

How do we solve the Wiener–Hopf equations?

solving linear systems ($L+1$ linear equations in $L+1$ unknowns)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \mathbf{w}_{WF} = \begin{bmatrix} 3 \\ 7 \end{bmatrix} \quad \rightarrow \quad \mathbf{w}_{WF} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

requires $O(L^3)$ arithmetic operations

requires $O(L^2)$ arithmetic operations if $\bar{\mathbb{X}}_{uu}$ is Toeplitz

- Schur algorithm
- Levinson-Durbin algorithm

= used intensively in applications, e.g. in speech codecs, etc.
details omitted (see next slides)

Appendix

Skip this slide

Appendix : The Levinson-Durbin algorithm

- *forward linear prediction problem* : $d_k = -u_{k+1}$

$$\overbrace{\begin{bmatrix} \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \bar{x}_{uu}(2) & \dots & \bar{x}_{uu}(N-1) \\ \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \dots & \bar{x}_{uu}(N-2) \\ \bar{x}_{uu}(2) & \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \dots & \bar{x}_{uu}(N-3) \\ & \ddots & \ddots & & \\ \bar{x}_{uu}(N-1) & \bar{x}_{uu}(N-2) & \bar{x}_{uu}(N-3) & \dots & \bar{x}_{uu}(0) \end{bmatrix}}^{\bar{\mathbb{X}}_{uu}} \cdot \mathbf{w}_{WF} = \overbrace{\begin{bmatrix} -\bar{x}_{uu}(1) \\ -\bar{x}_{uu}(2) \\ -\bar{x}_{uu}(3) \\ \vdots \\ -\bar{x}_{uu}(N) \end{bmatrix}}^{\bar{\mathbb{X}}_{du}} .$$

- *backward linear prediction problem* : $d_k = -u_{k-N}$

$$\overbrace{\begin{bmatrix} \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \bar{x}_{uu}(2) & \dots & \bar{x}_{uu}(N-1) \\ \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \dots & \bar{x}_{uu}(N-2) \\ \bar{x}_{uu}(2) & \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \dots & \bar{x}_{uu}(N-3) \\ & \ddots & \ddots & & \\ \bar{x}_{uu}(N-1) & \bar{x}_{uu}(N-2) & \bar{x}_{uu}(N-3) & \dots & \bar{x}_{uu}(0) \end{bmatrix}}^{\bar{\mathbb{X}}_{uu}} \cdot \mathbf{b}_{WF} = \overbrace{\begin{bmatrix} -\bar{x}_{uu}(N) \\ -\bar{x}_{uu}(N-1) \\ -\bar{x}_{uu}(N-2) \\ \vdots \\ -\bar{x}_{uu}(1) \end{bmatrix}}^{\bar{\mathbb{X}}_{du}} .$$

Appendix

41

Appendix : The Levinson-Durbin algorithm

- It is readily verified that if

$$\mathbf{a}_{WF} = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_{N-1} \end{bmatrix}^T$$

then

$$\mathbf{b}_{WF} = \begin{bmatrix} a_{N-1} & a_{N-2} & a_{N-3} & \ldots & a_0 \end{bmatrix}^T$$

Skip this slide

Appendix

42

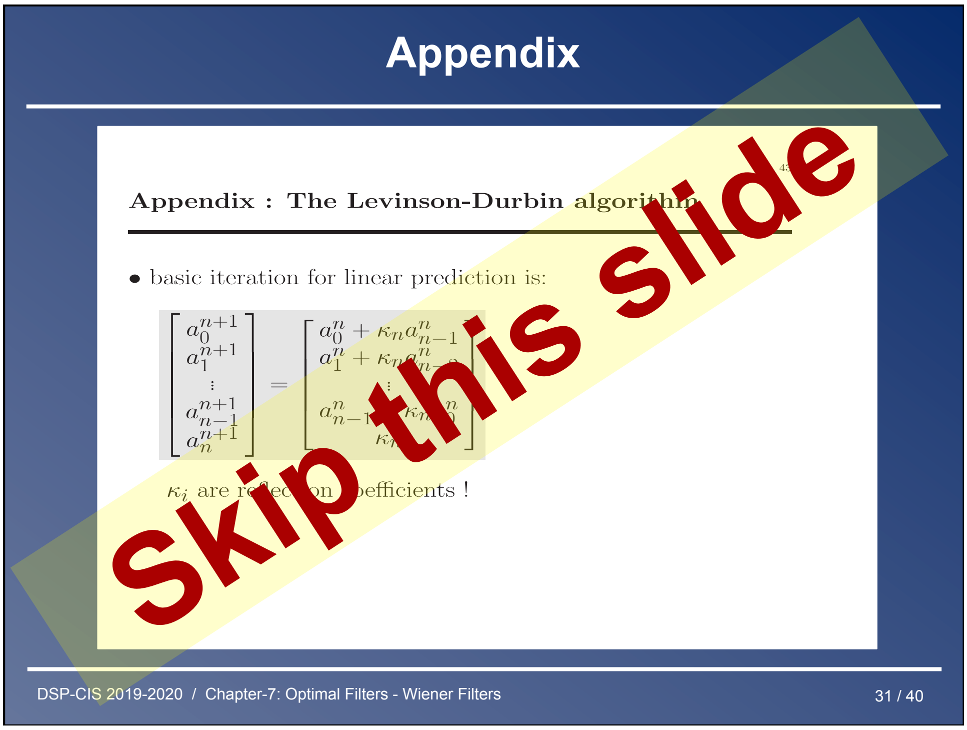
Appendix : The Levinson-Durbin algorithm

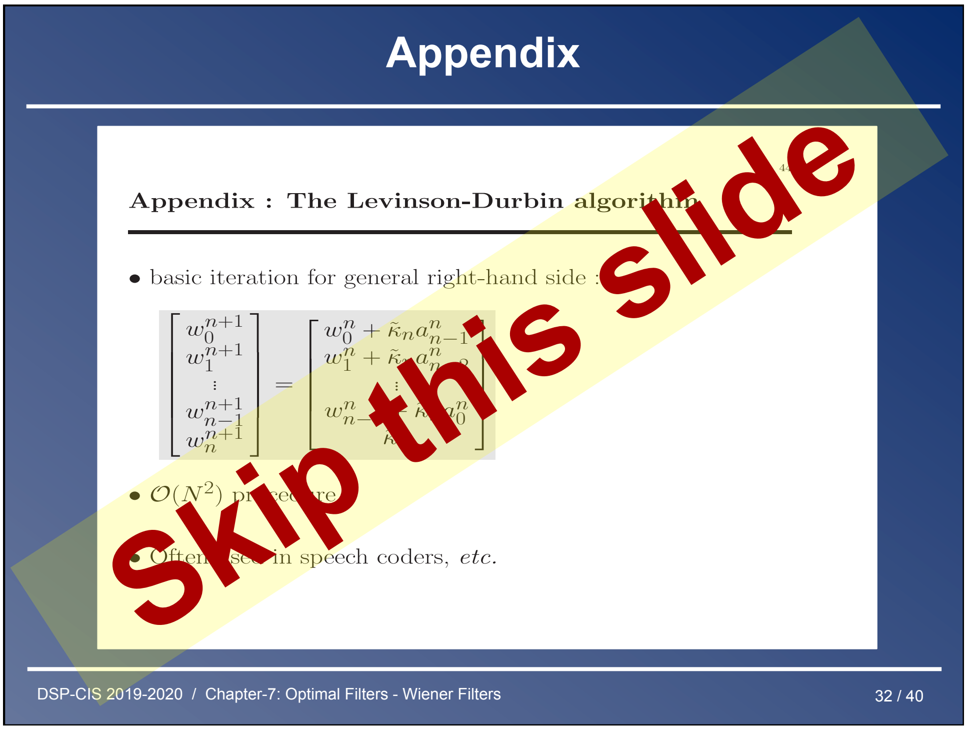
n-th order problem is

$$\begin{bmatrix} \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \ldots & \bar{x}_{uu}(n-1) \\ \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \ldots & \bar{x}_{uu}(n-2) \\ & \ddots & \ddots & \\ \bar{x}_{uu}(n-1) & \bar{x}_{uu}(n-2) & \ldots & \bar{x}_{uu}(0) \end{bmatrix} \cdot \underbrace{\begin{bmatrix} a_0^n \\ a_1^n \\ \vdots \\ a_{n-1}^n \end{bmatrix}}_{\mathbf{a}_{WF}^n} = \begin{bmatrix} -\bar{x}_{uu}(1) \\ -\bar{x}_{uu}(2) \\ \vdots \\ -\bar{x}_{uu}(n) \end{bmatrix}$$

Levinson-Durbin algorithm :
compute $n+1$-st order prediction coefficients
from n-th order coefficients
in $O(N)$ arithmetic steps

Skip this slide


Appendix
Appendix : The Levinson-Durbin algorithm
• basic iteration for linear prediction is:
κ_i are reflection coefficients !
Skip this slide
DSP-CIS 2019-2020 / Chapter-7: Optimal Filters - Wiener Filters
31 / 40


Appendix
Appendix : The Levinson-Durbin algorithm
• basic iteration for general right-hand side :
• $\mathcal{O}(N^2)$ procedure
in speech coders, etc.
Skip this slide
DSP-CIS 2019-2020 / Chapter-7: Optimal Filters - Wiener Filters
32 / 40

Optimal Filtering : Wiener Filters

PS: 'Unrealizable Wiener Filter'

In the case of FIR filters, it is interesting to investigate the effect of the chosen filter length L on the overall outcome. We have assumed from the start that the Wiener filter is a causal, finite-length FIR filter, and therefore realizable in real-time. Let us now remove the real-time realizability condition and assume that the filter is

$$\overset{\infty}{W}_{WF}(z) = \sum_{k=-\infty}^{\infty} \overset{\infty}{w}_k \cdot z^{-k}.$$

This is referred to as the **unrealizable Wiener filter**, which is indicated here by means of the ∞-superscript. The corresponding Wiener-Hopf equations are

$$\sum_{m=-\infty}^{\infty} \bar{x}_{uu}(k-m)w_m = \bar{x}_{du}(k) \quad k = \ldots,-2,-1,0,1,2,\ldots$$

Optimal Filtering : Wiener Filters

PS: 'Unrealizable Wiener Filter' (continued)

... which may be viewed as an 'infinitely large' set of linear equations

$$\begin{bmatrix} \ddots & & & & \\ & \bar{x}_{uu}(0) & \bar{x}_{uu}(1) & \ldots & \bar{x}_{uu}(N) \\ & \bar{x}_{uu}(1) & \bar{x}_{uu}(0) & \ldots & \bar{x}_{uu}(N-1) \\ & \vdots & \vdots & & \vdots \\ & \bar{x}_{uu}(N) & \bar{x}_{uu}(N-1) & \ldots & \bar{x}_{uu}(0) \\ & & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ w_{-1} \\ w_0 \\ \vdots \\ w_{N-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \bar{x}_{du}(-1) \\ \bar{x}_{du}(0) \\ \vdots \\ \bar{x}_{du}(N-1) \\ \vdots \end{bmatrix}$$

Note that the N-th order equations may actually be derived by picking the relevant (shaded) submatrices/subvectors in the above equation.

Optimal Filtering : Wiener Filters

PS: 'Unrealizable Wiener Filter' (continued)

The left-hand side infinite matrix-vector product may be viewed as a convolution of $\{\bar{x}_{uu}(k)\}$ with $\{w_k\}$. By taking the z-transform, one obtains

$$\mathbb{X}_{uu}(z)\cdot \overset{\infty}{W}_{WF}(z) = \mathbb{X}_{du}(z)$$

where

$$\mathbb{X}_{uu}(z) = \sum_{\delta=-\infty}^{\infty} \bar{x}_{uu}(\delta) z^{-\delta}$$

is the *power spectral density*, or *power spectrum*, of the process u_k, and

$$\mathbb{X}_{du}(z) = \sum_{\delta=-\infty}^{\infty} \bar{x}_{du}(\delta) z^{-\delta}$$

is the *cross-power spectrum* of d_k and u_k. So finally, we obtain a simple expression for the unrealizable Wiener filter

$$\overset{\infty}{W}_{WF}(z) = \frac{\mathbb{X}_{du}(z)}{\mathbb{X}_{uu}(z)}.$$

Compare to WF solution on p24

Optimal Filtering : Wiener Filters

PS: 'Unrealizable Wiener Filter' (continued)

Unrealizable WF provides lower bound on attainable MSE

='<u>irreducible error</u>' = the part of d_k that no WF can ever remove

$$J_{MSE}(\overset{\infty}{\mathbf{w}}_{WF}) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\mathbb{X}_{dd}(e^{j\omega}) - \frac{\mathbb{X}_{du}(e^{j\omega})}{\mathbb{X}_{uu}(e^{j\omega})}\mathbb{X}_{du}(-e^{j\omega})\right]d\omega$$

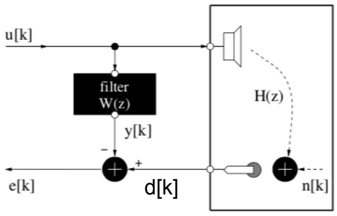
For L-th order filter, then MSE is

$$J_{MSE}(\mathbf{w}) = J_{MSE}(\overset{\infty}{\mathbf{w}}_{WF}) + \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbb{X}_{uu}(e^{j\omega})\left|\overset{\infty}{W}_{WF}(e^{j\omega}) - W(e^{j\omega})\right|^2 d\omega$$

= irreducible error + least squares error when unrealizable WF is approximated by causal L-th order filter, with input power spectrum included as a weighting function (proofs omitted)

Optimal Filtering : Wiener Filters

Example 1



Independence assumption :

$$\bar{x}_{nu}(\delta) = \mathcal{E}\{n_k \cdot u_{k-\delta}\} = 0 \quad \rightarrow \quad \bar{\mathbb{X}}_{nu}(z) = \sum_{\delta=-\infty}^{\infty} \bar{x}_{nu}(\delta) z^{-\delta} = 0$$

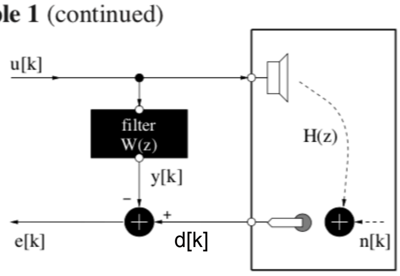
Unrealizable Wiener filter :

$$\overset{\infty}{W}_{WF}(z) = \frac{\bar{\mathbb{X}}_{du}(z)}{\bar{\mathbb{X}}_{uu}(z)} = \frac{H(z)\bar{\mathbb{X}}_{uu}(z) + \bar{\mathbb{X}}_{nu}(z)}{\bar{\mathbb{X}}_{uu}(z)} = H(z)$$

PS: Realizable when H(z) is FIR (and causal)

Optimal Filtering : Wiener Filters

Example 1 (continued)

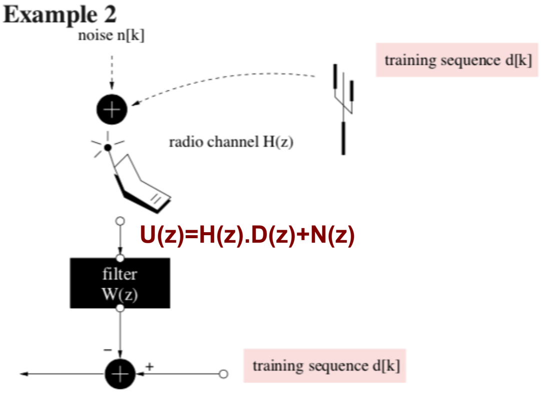


Irreducible error :

$$J_{MSE}(\overset{\infty}{\mathbf{w}}_{opt}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} [\bar{\mathbb{X}}_{dd}(e^{j\omega}) - H(e^{j\omega})\bar{\mathbb{X}}_{du}(-e^{j\omega})]d\omega = \ldots = \frac{1}{2\pi}\int_{-\pi}^{\pi} \bar{\mathbb{X}}_{nn}(e^{j\omega})d\omega = \mathcal{E}\{n_k^2\}$$

Optimal Filtering : Wiener Filters

Example 2



Independence assumption :

$$\bar{x}_{dn}(\delta) = \mathcal{E}\{d_k \cdot n_{k-\delta}\} = 0 \quad \rightarrow \quad \bar{\mathbb{X}}_{dn}(z) = \sum_{\delta=-\infty}^{\infty} \bar{x}_{dn}(\delta) z^{-\delta} = 0$$

Optimal Filtering : Wiener Filters

Example 2 (continued)

Unrealizable Wiener filter :

$$\overset{\infty}{W}_{WF}(z) = \frac{H(z^{-1})\bar{\mathbb{X}}_{dd}(z)}{\bar{\mathbb{X}}_{nn}(z) + H(z)H(z^{-1})\bar{\mathbb{X}}_{dd}(z)}$$

$$\bar{\mathbb{X}}_{nn}(z) = 0 \quad \rightarrow \quad \overset{\infty}{W}_{WF}(z) = \frac{1}{H(z)}$$

$$\bar{\mathbb{X}}_{nn}(z) \rightarrow \infty \quad \rightarrow \quad \overset{\infty}{W}_{WF}(z) \rightarrow 0$$

Irreducible error :

$$J(\overset{\infty}{\mathbf{w}}_{WF}) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\frac{\bar{\mathbb{X}}_{nn}(e^{j\omega})\bar{\mathbb{X}}_{dd}(e^{j\omega})}{\bar{\mathbb{X}}_{nn}(e^{j\omega}) + H(e^{j\omega})H(-e^{j\omega})\bar{\mathbb{X}}_{dd}(e^{j\omega})}\right]d\omega$$