Advanced Digital Signal Processing

ECE 551

PowerPoint Notes

Professor Andrew Singer

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 1

Professor Andrew Singer
Department of Electrical and Computer Engineering

Class Administrative

Prof. Andrew C. Singer

118 CSL (Building nearest Beckman)

217 244-9263

Office Hours: M 2-3 p.m. in 114 CSRL

Email acsinger@uiuc.edu

Course web page
http://online.engr.uiuc.edu/webcourses/ece451/

Introduction Topics

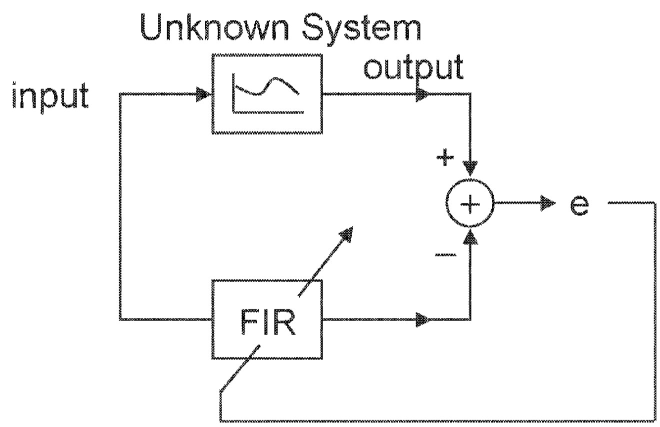
- Signal theory, DT system theory ~5 hrs
  Structures ~2 hrs
  Sampling, Quantization, OSAD ~6 hrs
  IIR / FIR filter design ~5 hrs
- DFT, FFT, fast convolution, spectral analysis ~4 hrs
  Linear prediction (AR modeling/lattice filters)~3 hrs
- Adaptive Filtering, LMS, RLS ~7 hrs
- Multirate Signal Processing, Filter banks ~5 hrs
- Special topics

Review

- Review of S&S and undergraduate DSP
- "DT Proc of CT signals"
- Finite word length effects
- Filter design, optimality results "anything is optimal if you pick the right criteria"

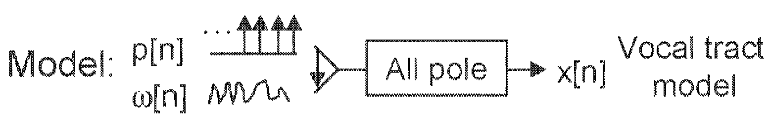
  L^2, L^∞ mostly

Modeling/ System Identification

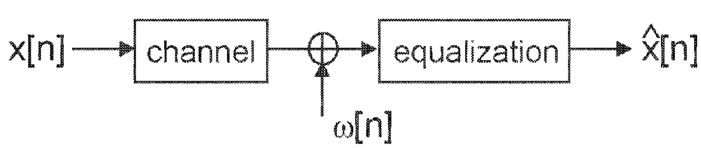


Applications

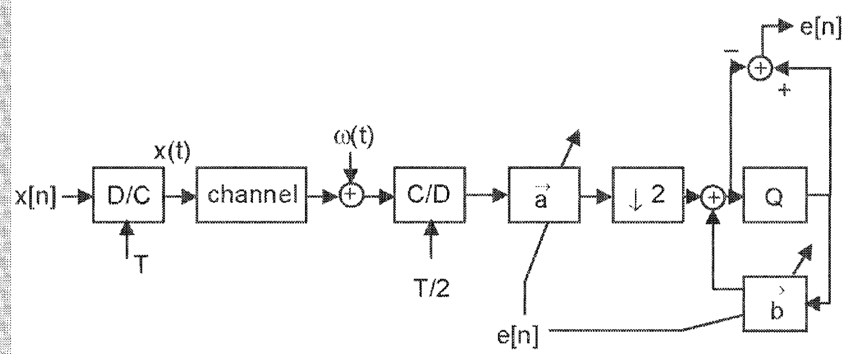
Speech modeling/coding/recognition



- deconvolution, equalization



Applications, cont'd



"Decision Feedback Equalizer"

Linear Prediction

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n]$$

Fast algorithms/
Structures

-Speech (all pole model)
  - spectral envelope
  - formants

-Parametric Spectrum Estimation

-EEG analysis/EKG

-Geophysics

-Dow Jones Industrial Averages
  - trends
  - cycles

Adaptive Filtering

"System ID" or "Modeling" unknown or time-varying systems. Coefficients adapt to signal statistics for "closed loop" signal processing

- → Adaptive equalizers/ communications
- → Beam forming/sonar/radar
- → Array processing

Adaptive Equalization – Decision feedback equalizers Lucky (1965)

Adaptive Filtering, cont'd

Underwater Acoustic Equalization

(was) 2 bits/sec to 2 bits/minute!

(now) now up to 10-20kbps up to 60 nautical miles

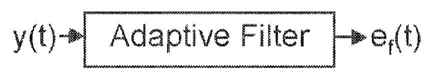
(was) ~ 2,400 bps modems over phone lines

(now) ~ 56kbps with adaptive equalization
~ 52Mbps using VDSL modem.

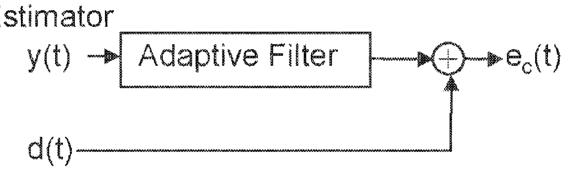
3 Classes of Adaptive Filters

1) Estimator

2) Filter (usually predictor)

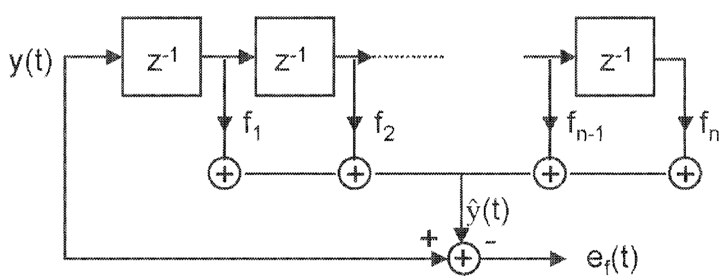


3) Joint Process Estimator



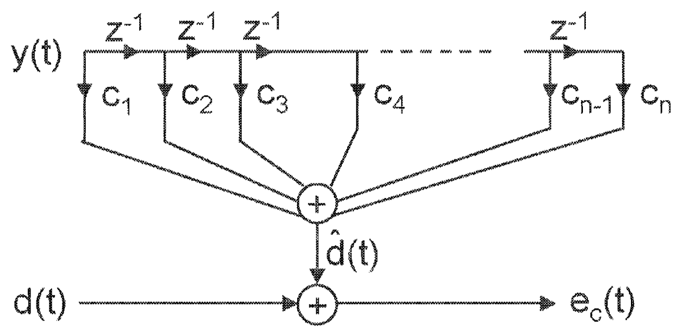
3 Classes of Filters, cont'd

Predictor structure:



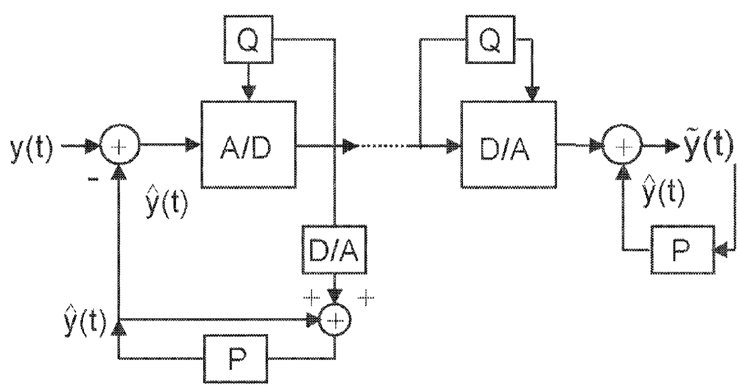
3 Classes of Filters, cont'd

Joint process estimator:

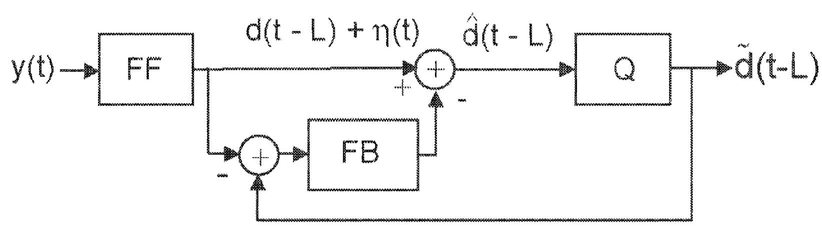


3 Classes of Filters, cont'd

Waveform coding of speech:

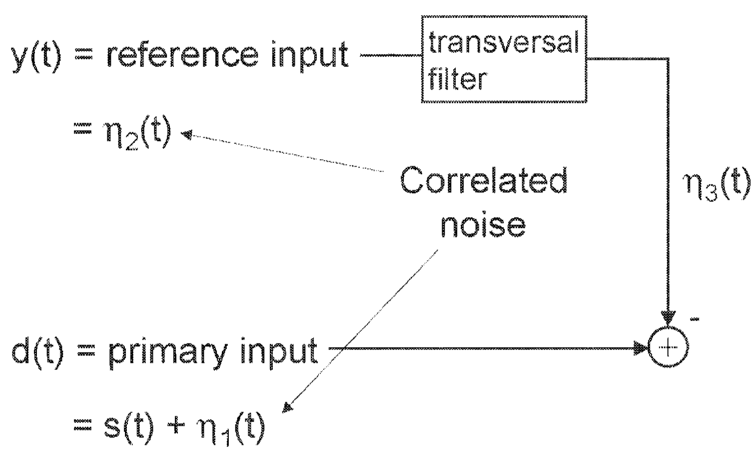


3 Classes of Filters, cont'd

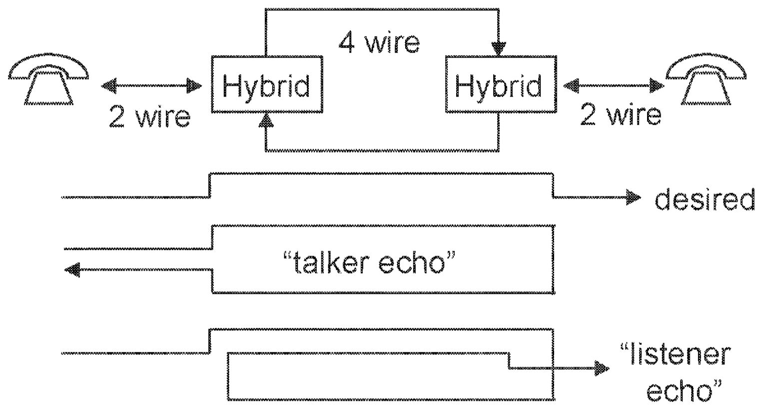


Decision Feedback Equalizer as a predictor

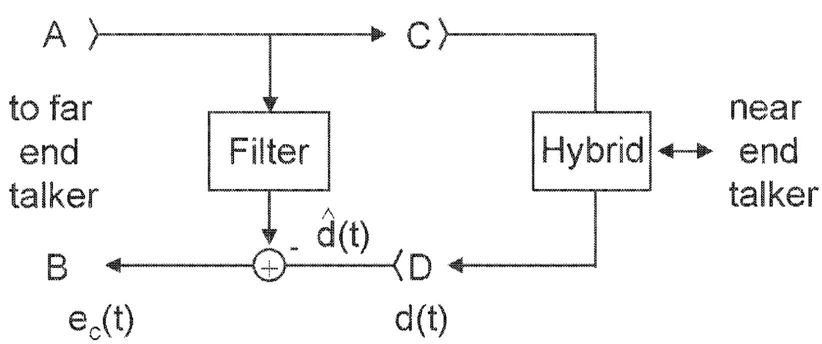
Adaptive Noise Cancelling



Echo Cancellation

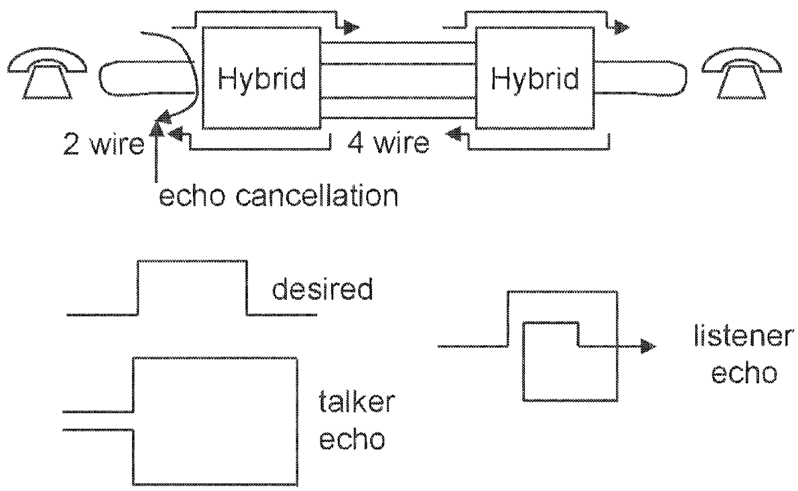


Echo Cancellation, cont'd

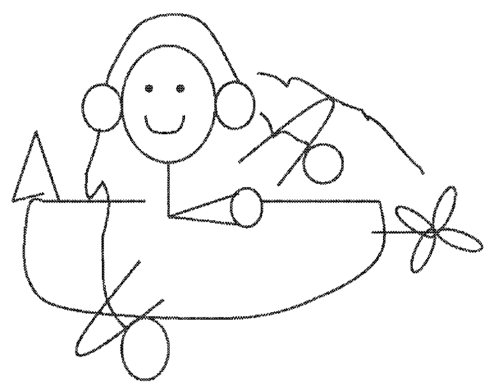


Echo canceller for 1 direction of transmission

Adaptive Filtering Examples

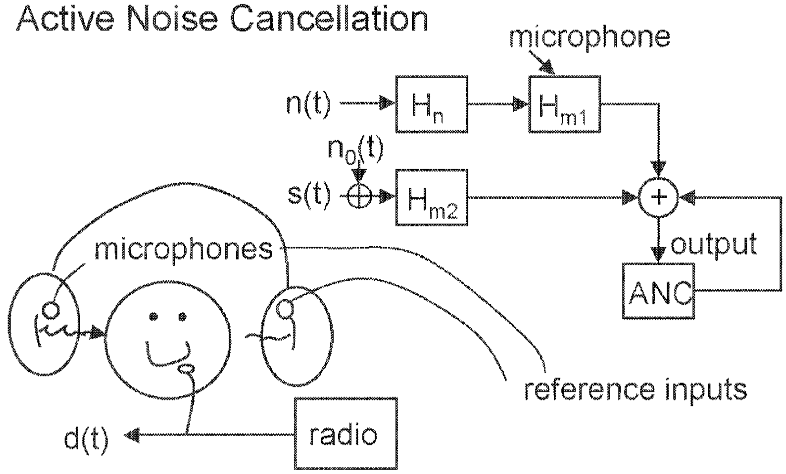


Examples, cont'd



Examples, cont'd

Active Noise Cancellation

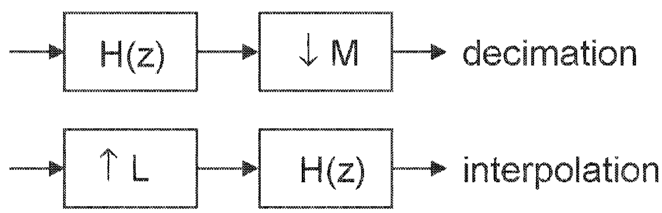


Algorithms

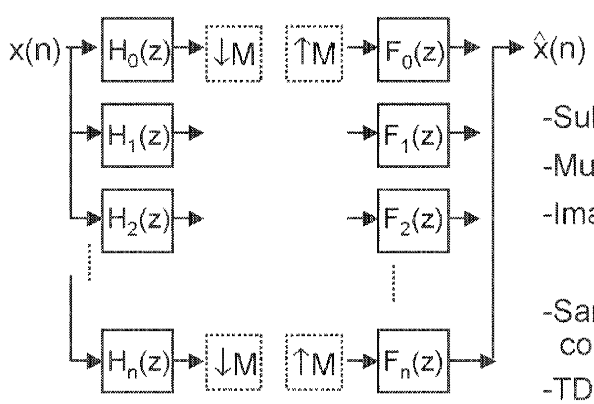
- Stochestic Gradient, LMS (Least Mean Squares) Woodrow/Stearns ~ 1960
- Stability/ performance
- Advanced Algorithms (RLS, Lattices, etc…)

Multirate DSP

- Communication with fractionally-spaced equalizer
- “tools” for multirate signal processing:



Analysis/Synthesis Filterbank



-Sub band coding

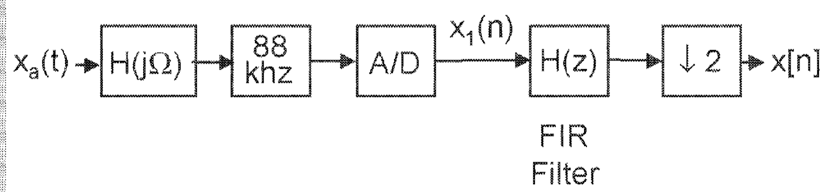
-Multirate filters

-Image coding

-Sample rate conversion

-TDM/FDM

Digital Audio



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 2

Professor Andrew Singer
Department of Electrical and Computer Engineering

Notation

A continuous-time signal defined on the real line will be denoted $x(t)$, or $x_c(t)$.

A discrete-time signal defined on the integers will be denoted $x[n]$, or $x_d[n]$.

Multidimensional extensions will be clear: e.g. $x(t, s)$, $x[m, n]$, or $x(t, n]$.

Classes of Signals

(L^2) Finite energy: $\int_{-\infty}^{\infty} |x(t)|^2 \, dt < \infty$

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$$

(L^1) Absolutely Integrable/Summable:

$$\int_{-\infty}^{\infty} |x(t)| \, dt < \infty$$

$$\sum_{n=-\infty}^{\infty} |x(t)| < \infty$$

Classes of Signals, cont'd

(L^∞) Bounded Signals:

$$\sup_t |x(t)| < \infty$$

$$\sup_n |x[n]| < \infty$$

Signals in L^P:

$$\|x(t)\|_p \triangleq \left(\int_{-\infty}^{\infty} |x(t)|^P \, dt \right)^{1/P} < \infty$$

$$\|x[n]\|_p \triangleq \left(\sum_{n=-\infty}^{\infty} |x[n]|^P \right)^{1/P} < \infty$$

Classes of Signals, cont'd

Finite Duration: $x(t) : t \in T$

$x[n] : n_1 \leq n \leq n_2$

} only defined on an interval

- sometimes useful to think of them as either periodically repeated (FS/DFS) or as zero outside the interval (FT/DTFT).

Examples

$$x(t) = t^{-\alpha},\ t \in [0, 1],\ \frac{1}{2} < \alpha < 1$$

$$\|x(t)\|_1 = \int_0^1 \left|t^{-\alpha}\right| dt$$

$$= \frac{1}{1-\alpha}\left(\lim_{t \to 1} t^{1-\alpha} - \lim_{t \to 0} t^{1-\alpha}\right) = \frac{1}{1-\alpha}$$

Examples, cont'd

$$\|x(t)\|_2^2 = \int_0^1 \left|t^{-\alpha}\right|^2 dt$$

$$= \frac{1}{1-2\alpha}\left(\lim_{t\to 1} t^{1-2\alpha} - \lim_{t\to 0} t^{1-2\alpha}\right)$$

$$= \frac{1}{1-2\alpha}(1-\infty)$$

$x(t) \in L^1$, but not $\in L^2$.

Examples, cont'd

$$h_{lpf}[n] = \frac{\sin w_c n}{\pi n},\ -\infty < n < \infty,\ 0 < w_c < \pi$$

$$\left\|h_{lpf}[n]\right\|_1 = \sum_{n=-\infty}^{\infty}\left|\frac{\sin w_c n}{\pi n}\right|$$

for

$$m_k \in \left[\frac{2\pi k}{w_c} + \frac{\pi}{4w_c}, \frac{2\pi k}{w_c} + \frac{3\pi}{4w_c}\right], > \sum_{k=1}^{\infty}\left|\frac{\sin w_c m_k}{\pi m_k}\right|$$

$$> \sum_{k=1}^{\infty}\left(\frac{w_c}{2\pi(k+1)}\right)\left(\frac{\sqrt{2}/2}{\pi}\right)$$

$$> \lim_{k\to\infty} C\ln(k) = \infty$$

Examples, cont'd

$$\left\| h_{\ell pf}[n] \right\|_2^2 = \sum_{n=-\infty}^{\infty} \left| \frac{\sin w_c n}{\pi n} \right|^2$$

$$\leq \frac{w_c}{\pi} + 2\sum_{n=1}^{\infty} \left| \frac{1}{\pi n} \right|^2$$

$$< \frac{w_c}{\pi} + \frac{2}{\pi^2}\frac{\pi^2}{6} = \frac{w_c}{\pi} + \frac{1}{3}$$

(or use Parseval's Thm $\frac{w_c}{\pi}$)

$h_{\ell pf}[n] \in L^2, \notin L^1$

Linear Time-Invariant (LTI) Systems

$x[n] \longrightarrow$ LTI $\longrightarrow y[n] \quad \Leftrightarrow \quad y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

impulse response (→ $h[n-k]$)

Linear Constant Coefficient Diff. Eq

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{m} b_k x[n-k]$$

- satisfying LCCDE $\not\Rightarrow$ LTI
- need additional constraints

$x_0[n] \leftrightarrow y_0[n]$ such that LCCDE is satisfied.

$y[n] = y_0[n] + y_h[n]$ also satisfies LCCDE

$$\hookrightarrow \sum_{k=0}^{N} a_k y_k[n-k] = 0$$

LCCDE, cont'd

Need auxiliary information

- "Initial Rest Conditions"
- "Final Rest Conditions"

Boundary conditions necessary to uniquely determine y[n].

Is "LTI" enough to determine y[n] for any x[n]?

NO! May need stability, causality, etc. to make it unique!

Example

$$y[n]-\frac{1}{2}y[n-1]=x[n],\ x_0[n]=k\delta[n]$$

$$y_0[n]=k\left(\frac{1}{2}\right)^n u[n]+y_h[n]$$

$$y_h[n]=A\left(\frac{1}{2}\right)^n$$

Let A = 1. Is this system Linear?

No. Check superposition

Example, cont'd

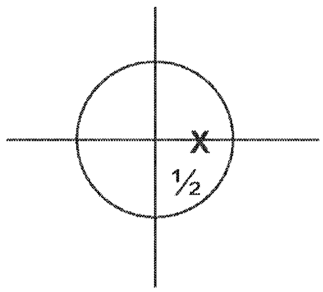
Unique, LTI $\leftrightarrow$ System function well defined

$$H(z)=\sum_{n=-\infty}^{\infty}h[n]z^{-n},\ \text{Region of convergence}$$

$$Y(z)-\frac{1}{2}Y(z)z^{-1}=X(z)$$

$$H(z)=\frac{1}{1-\frac{1}{2}z^{-1}}$$

Example, cont'd



$|z| > \frac{1}{2}$ right sided

$|z| < \frac{1}{2}$ left sided

$$h[n] = -\left(\frac{1}{2}\right)^n u[-n-1]$$

ECE 451 - Lecture 2 14 of 33

Example, cont'd

$$y[n] = K\left(\frac{1}{2}\right)^n u[n] + A\left(\frac{1}{2}\right)^n$$

$A = 0 \quad \Rightarrow h[n] = \left(\frac{1}{2}\right)^n u[n]$ causal, stable

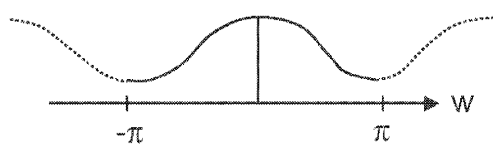
$A = -K \Rightarrow h[n] = -\left(\frac{1}{2}\right)^n u[-n-1]$ Non-causal not stable

ECE 451 - Lecture 2 15 of 33

Example, cont'd

$$H\left(e^{jw}\right) = H(z)\Big|_{z=e^{jw}} = \frac{1}{1-\frac{1}{2}e^{-jw}}$$

$$\left|H(e^{jw})\right|^2 = \frac{1}{\frac{5}{4} - \cos w}$$



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 3

Professor Andrew Singer
Department of Electrical and Computer Engineering

Discrete-Time Fourier Transform

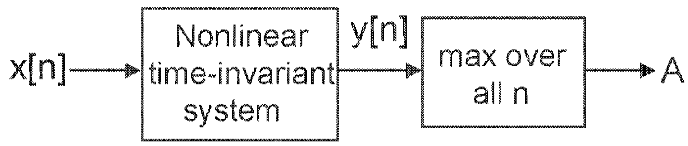
Analysis Equation

$$X(e^{jw}) \triangleq \sum_{n=-\infty}^{\infty} x[n]e^{-jwn}$$

periodic with period 2π. Why?

<u>Ans</u>. All frequencies are uniquely represented by an interval of length 2π.

DTFT Periodic?



$e^{jwn} \longrightarrow$ $\longrightarrow$ define $A(e^{jw})$

Is $A(e^{jw})$ periodic with period 2π?
with any period?

DTFT Periodic?, cont'd

Yes!

$$e^{jw_0 n} = e^{j(w_0 + k2\pi)n}$$

$\rightarrow$ same input must yield same output or system is not well-defined.

Eigenfunction Property of LTI Systems

For input $x[n] = e^{jwn}$:

Eigenfunction Property: $y[n] = \sum_{k=-\infty}^{\infty} h[k]e^{jw(n-k)}$

$$= e^{jwn}\underbrace{\left(\sum_{k=-\infty}^{\infty} h[k]e^{-jwk}\right)}$$

$x[n] = e^{jwn} \xrightarrow{\text{LTI}} y[n] = e^{jwn} \quad H(e^{jw})$

eigenfunction — eigenvalue

$H(e^{jw})$ is the "spectrum" of eigenvalues

DTFT Synthesis Equation

$$x[n] \triangleq \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{jw})e^{jwn}dw$$

- $x[n]$ is composed of a superposition of complex exponentials $e^{j\omega n}$
- Each frequency ω is weighted $X(e^{j\omega n})d\omega$
- Consider, e.g., how to construct $x[n]= e^{j\omega n}$

DTFT Synthesis Equation, cont'd

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{m=-\infty}^{\infty} x[m]e^{-jwm}\right)e^{jwn}dw = \hat{x}[n]$$

If the sum converges uniformly for all w, we can exchange $\int$ and $\sum$

$$\hat{x}[n] = \sum_{m=-\infty}^{\infty} x[m]\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{jw(n-m)}dw\right)$$

DTFT Synthesis Equation, cont'd

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{jw(n-m)}dw = \frac{\sin\pi(n-m)}{\pi(n-m)} = \delta[n-m]$$

$\Rightarrow \hat{x}[n] = x[n]$ (IDTFT (DTFT(x[n])) $= x[n]$)

Sufficient Convergence of $X(e^{jw})$

$$\left|X(e^{jw})\right| = \left|\sum_{n=-\infty}^{\infty} x[n]e^{-jwn}\right|$$

$$\leq \sum_{n=-\infty}^{\infty} \left|x[n]\right|\left|e^{-jwn}\right|$$

$$\leq \sum_{n=-\infty}^{\infty} \left|x[n]\right| < \infty$$

If x[n] is in L^1 $\Rightarrow$ $X(e^{jw})$ exists.

$\Rightarrow$ $X(e^{jw})$ converges uniformly to a continuous function

Not Necessary

$$\sum_{n=-\infty}^{\infty} \left|x[n]\right|^2 < \infty, \; x[n] \in L^2$$

$$\Rightarrow \text{for} \quad X_m(e^{jw}) \triangleq \sum_{n=-m}^{m} x[n]e^{-jwn}$$

$$\Rightarrow \quad \lim_{m\to\infty} \int_{-\pi}^{\pi} \left|X(e^{jw}) - X_m(e^{jw})\right|^2 dw = 0$$

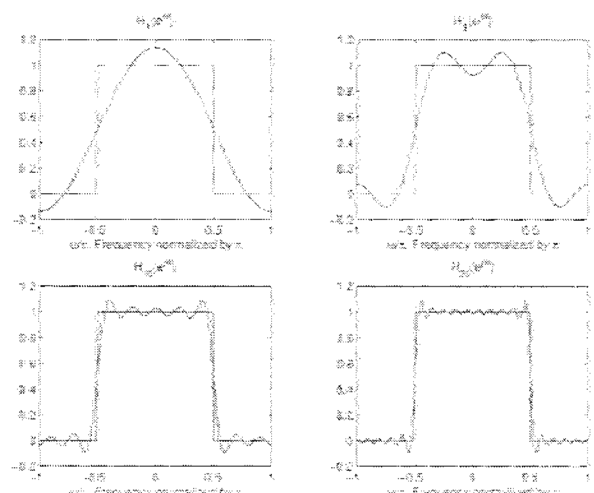
Example

$$H_{lp}(e^{jw}) = \begin{cases} 1, & |w| < w_c \\ 0, & w_c < |w| \le \pi \end{cases}$$

$$h_{lp}[n] = \frac{\sin w_c n}{\pi n} \qquad \in L^2 \quad \notin L^1$$

$$\lim_{m\to\infty} \int_{-\pi}^{\pi} \left| H_{lp}(e^{jw}) - H_m(e^{jw}) \right|^2 dw = 0$$

$H_m(e^{jw})$ for m=1,3,10,20



Example, cont'd

The maximum overshoot remains constant

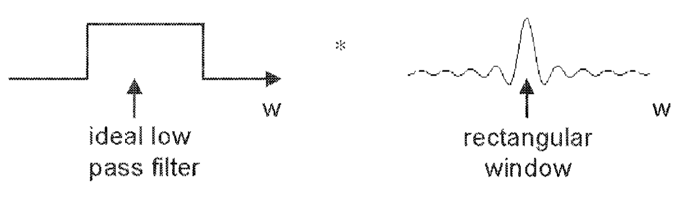
What is the maximum L^1 error? (trick question!)

L^2 error $\rightarrow 0$

L^∞ error $\rightarrow$ constant

Example

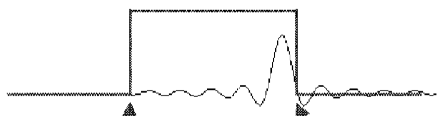
Truncation of DTFT = Convolution in frequency domain



$$\frac{\sin(w(2m+1)/2)}{\sin(w/2)}$$

Example, cont'd

maximum overshoot



$$= \frac{1}{2\pi}\int_{-\pi}^{\frac{2\pi}{2m+1}} \frac{\sin(w(2m+1)/2)}{\sin(w/2)}\,dw$$

$\cong$ 1.0895 ~ 9% overshoot

Symmetry Properties of DTFT

$x^*[n]$	$\leftrightarrow$	$X^*(e^{-jw})$	
$x^*[-n]$	$\leftrightarrow$	$X^*(e^{jw})$	
$Re\{x[n]\}$	$\leftrightarrow$	$X_e(e^{jw})$	conj. symm. part
$Im\{x[n]\}$	$\leftrightarrow$	$X_o(e^{jw})$	conj. antisymm. aart
$x_e[n]$	$\leftrightarrow$	$X_R(e^{jw})$	
$x_o[n]$	$\leftrightarrow$	$X_I(e^{jw})$	

$x_e[n] = ½ (x[n] + x^*[-n])$, $X_e(e^{jw}) = ½ (X(e^{jw}) + X^*(e^{-jw}))$

DTFT Theorems

$$a\,x[n] + b\,y[n] \quad \leftrightarrow \quad a\,X(e^{jw}) + b\,Y(e^{jw})$$

$$x[n-n_0] \quad \leftrightarrow \quad e^{-jwn_0} X(e^{jw})$$

$$e^{jw_0 n} x[n] \quad \leftrightarrow \quad X(e^{j(w-w_0)})$$

$$x[-n] \quad \leftrightarrow \quad X(e^{-jw})$$

$$n\,x[n] \quad \leftrightarrow \quad j\frac{d}{dw} X(e^{jw})$$

DTFT Theorems, cont'd

$$x[n] * y[n] \quad \leftrightarrow \quad X(e^{jw})Y(e^{jw})$$

$$x[n]y[n] \quad \leftrightarrow \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(w-\theta)})d\theta$$

$$\|x[n]\|_2^2 = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|X(e^{jw})\right|^2 dw = \frac{1}{2\pi}\left\|X(e^{jw})\right\|_2^2$$

$$\sum_{n=-\infty}^{\infty} x[n]y^*[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{jw})Y^*(e^{jw})dw$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 4

Professor Andrew Singer
Department of Electrical and Computer Engineering

Magnitude and Phase

$$Y(z) = H(z)\,X(z)\Big|_{z=e^{j\omega}} \rightarrow \text{ if stable } \rightarrow \; Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

system function — frequency response

$$\left|Y(e^{j\omega})\right| = \left|H(e^{j\omega})\right|\left|X(e^{j\omega})\right|$$

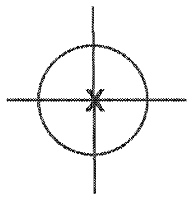
"magnitude distortion", "magnitude response"

$$\measuredangle Y(e^{j\omega}) = \measuredangle H(e^{j\omega}) + \measuredangle X(e^{j\omega})$$

phase distortion

Linear Phase

$h[n] = \delta[n-1] \leftrightarrow H(z) = z^{-1}, H(e^{j\omega}) = e^{-j\omega}$



$|H(e^{j\omega})| = 1, \measuredangle \underline{H(e^{j\omega}) = -}\omega, |\omega| < \pi$

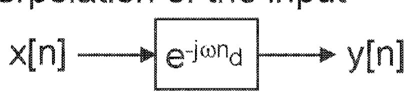
ideal delay

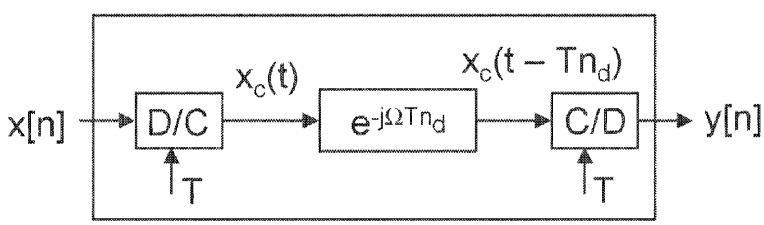
$H(e^{j\omega}) = e^{-j\omega n_d} \rightarrow$ for n_d = integer = delay by n_d

? $n_d \neq$ integer

Non-integer Delay

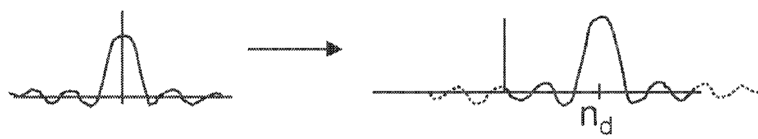
"Samples of a delayed version of the BL interpolation of the input"





Causal Implementations

How can we implement $\approx H(e^{j\omega}) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \text{else} \end{cases}$?



noncausal — causal

Effect of Delay

$$\Rightarrow \quad H(e^{j\omega}) = \begin{cases} e^{-j\omega n_d} & , |\omega| < \omega_c \\ 0 & , \text{else} \end{cases}$$

$$|H(e^{j\omega})| = \begin{cases} 1 \, , |\omega| < \omega_c \\ 0 \, , \text{else} \end{cases}$$

$$\measuredangle H(e^{j\omega}) = \begin{cases} -\omega n_d, |\omega| < \omega_c \\ 0 \, , \text{else} \end{cases}$$

Linear Phase / Group Delay

Linear Phase = Delay (for integer n_d)

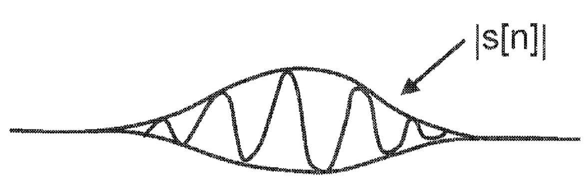
= SODVOBLOI (for non-integer)

<u>Group delay</u> is a measure of linearity of phase.

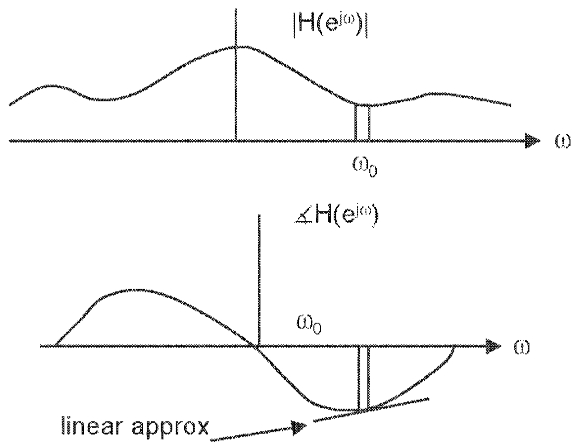
Group Delay (cont.)

$x[n] \longrightarrow$ [$H(e^{j\omega})$] $\longrightarrow y[n]$

$x[n] = s[n] \cos(\omega_0 n)$ narrowband input



Group Delay (cont.)



Group Delay (cont.)

$$H(e^{j\omega}) \approx \left|H(e^{j\omega_0})\right| e^{\underbrace{j\measuredangle H(e^{j\omega_0}) + \frac{d}{dw}\measuredangle H(e^{j\omega})\big|_{\omega=\omega_0}(\omega-\omega_0)}_{-\phi_0 \,-\, wn_d}}$$

$$y[n] \approx s[n-n_d]A(\omega_0)\cos(\omega_0 - \phi_0 - \omega_0 n_d)$$

↑ envelope delayed by group delay

Continuous Phase

Need $\measuredangle H(e^{j\omega})$ to be a continuous function of ω

Write $\arg[H(e^{j\omega})] \triangleq$ continuous phase of $H(e^{j\omega})$

$$\tau(\omega) = \text{grd}[H(e^{j\omega})] = \frac{-d}{dw}\arg[H(e^{j\omega})]$$

$$= \frac{-d}{dw}\text{Arg}[H(e^{j\omega})]$$

↑ ignoring impulses

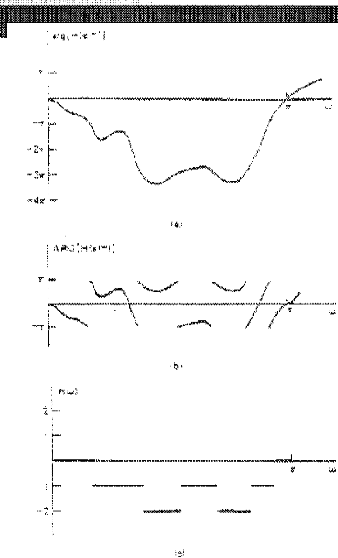
Computing Group Delay

$\arg[H(e^{j\omega})] = \text{Arg}[H(e^{j\omega})] + 2\pi\, r(\omega)$

unwrapped phase | wrapped phase | integer to compensate for 2π jumps

To compute $\tau(\omega) = \frac{-d}{dw}\underbrace{\measuredangle H(e^{j\omega})}$

use numerical differentiation



Computing Group Delay (cont.)

-or- use properties of DTFT:

$$h[n] \leftrightarrow H(e^{j\omega})$$

$$nh[n] \leftrightarrow j\frac{d}{dw}H(e^{j\omega})$$

Computing Group Delay with DTFT

$$\frac{\mathrm{DTFT}\{nh[n]\}}{\mathrm{DTFT}\{h[n]\}} = \frac{jA'(\omega)e^{j\measuredangle H(e^{j\omega})} + A(\omega)\tau(\omega)e^{j\measuredangle H(e^{j\omega})}}{A(\omega)e^{j\measuredangle H(e^{j\omega})}}$$

$$= j\frac{A'(\omega)}{A(\omega)} + \tau(\omega)$$

$$\mathrm{Re}\left\{\frac{\mathrm{DTFT}\{nh[n]\}}{\mathrm{DTFT}\{h[n]\}}\right\} = \tau(\omega)$$

Examples

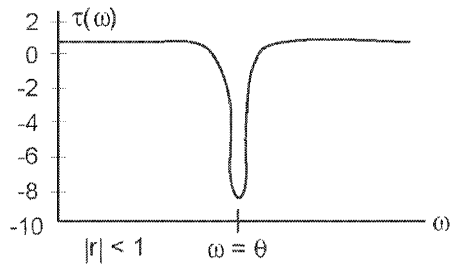
$$H(e^{j\omega}) = \begin{cases} e^{-j\omega\alpha} \ , \ |\omega| < \omega_c \\ 0 \ , \ \text{else} \end{cases}$$

$$\tau(\omega) = \alpha \ , \ |\omega| < \omega_c$$

Examples, cont'd

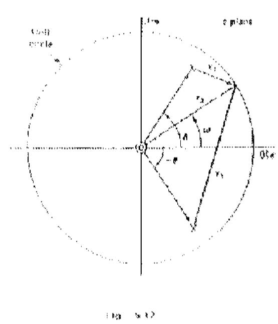
One zero $H(e^{j\omega}) = (1 - re^{j\theta} e^{-j\omega})$

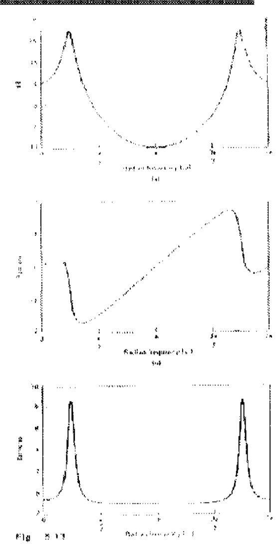
$$\tau(\omega) = \frac{-d}{dw} \tan^{-1}\left(\frac{\mathrm{Im}(1 - re^{j\theta}e^{-j\omega})}{\mathrm{Re}(1 - re^{j\theta}e^{-j\omega})}\right)$$

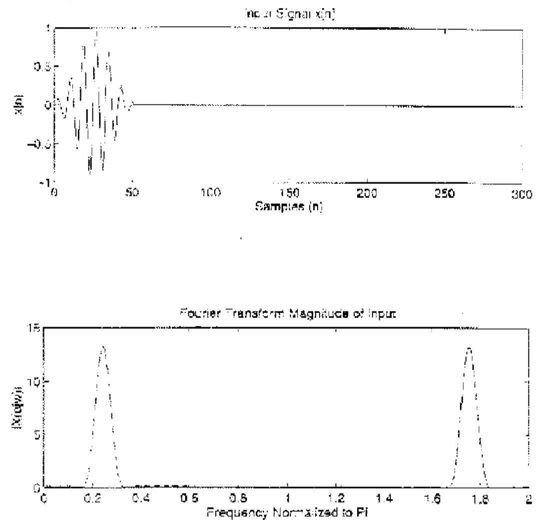


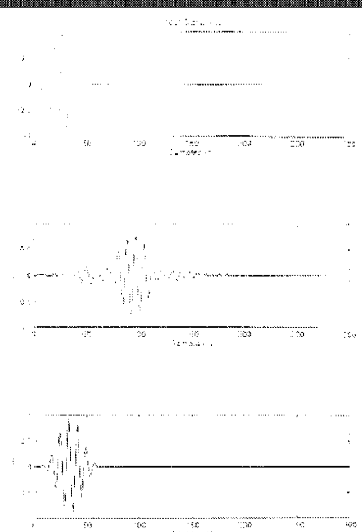
$$= \frac{r^2 - r\cos(\omega - \theta)}{\left|1 - re^{j\theta}e^{-j\omega}\right|^2}$$

What does sign ± imply?

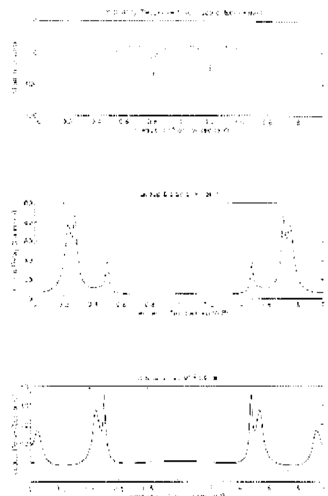


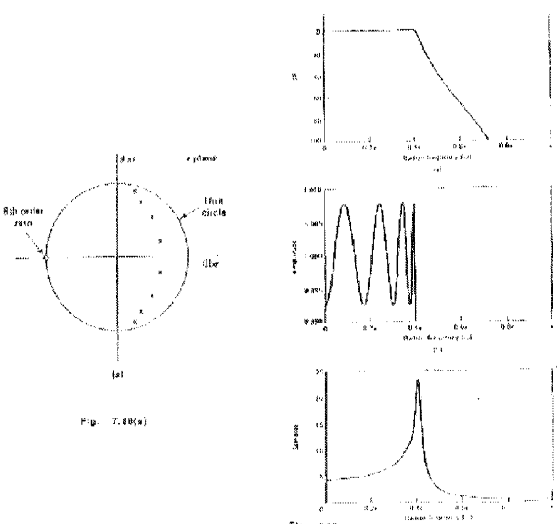


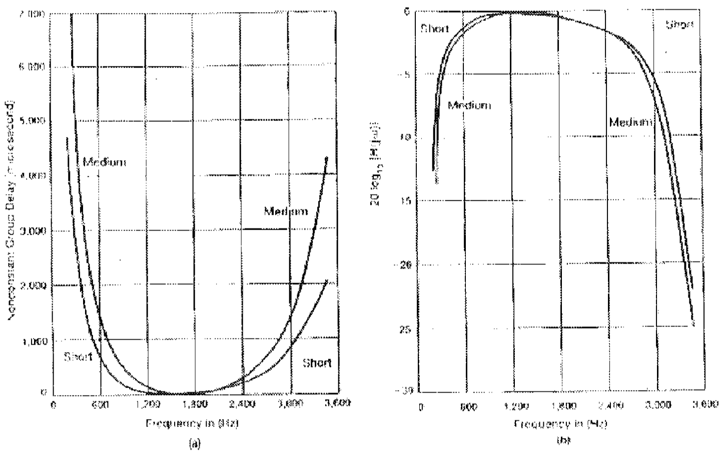




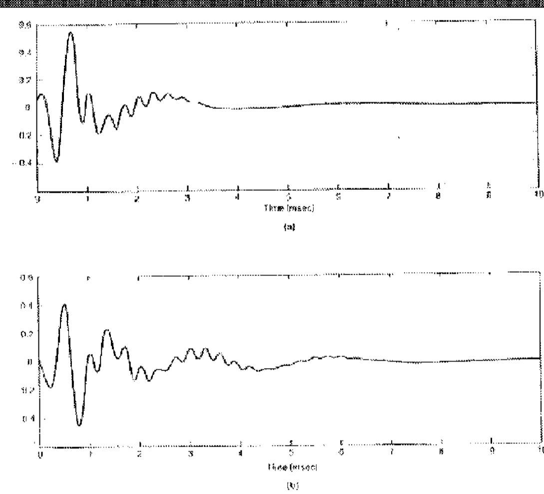
© 2001 Andrew Singer, University of Illinois at Urbana-Champaign, All Rights Reserved







"Signals & Systems"
Oppenheim & Willsky
Fig. 6.6



"Signals & Systems"
Oppenheim & Willsky
Fig. 6.7

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

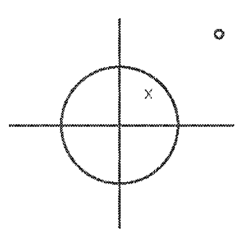
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 5

Professor Andrew Singer
Department of Electrical and Computer Engineering

All Pass Systems



zero @ $\frac{1}{a_k^*}$

$$H_{ap}(z) = \prod_{k=1}^{N} \frac{z^{-1} - a_k^*}{1 - a_k z^{-1}}$$

pole @ a_k

(causal, stable)?

$$\left|H(e^{j\omega})\right| = 1 \quad \forall\omega$$

All Pass Systems, cont'd

Causal all pass has $\tau(\omega) \geq 0 \ \ \forall\ \omega$

$$\tau(\omega) = \frac{1-r^2}{\left|1-re^{j\theta}e^{-j\omega}\right|^2},\ |r|<1 \quad (\text{why?})$$

$$\Rightarrow \arg[H_{ap}(e^{j\omega})] \leq 0 \qquad 0 \leq \omega < \pi$$

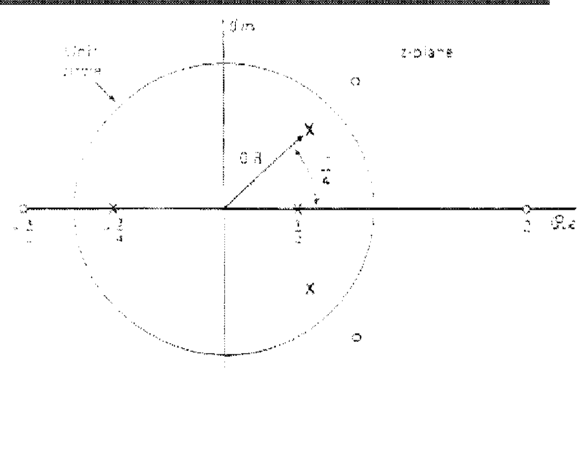


Fig. 5.18

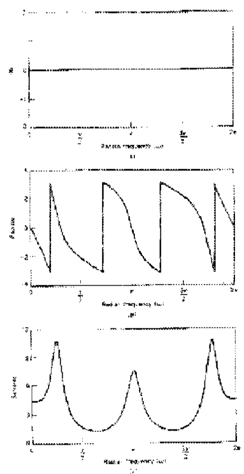


Figure 5.21

Minimum Phase Systems

H(z) causal, stable, $H^{-1}(z)$ causal, stable

poles inside u.c.
no pole @ ∞

zeros inside u.c.
no zero @ ∞

If $H(e^{j\omega})$ is m.p. ⇒ given $|H(e^{j\omega})|$, phase is unique.

Minimum Phase Systems, cont'd

it is the "minimum" of all phases for $|H(e^{j\omega})|$

How to find?:

$$\left|H(e^{j\omega})\right|^2 = H(e^{j\omega})H^*(e^{j\omega}) = \underbrace{H(z)H^*\left(\frac{1}{z^*}\right)\Big|_{z=e^{j\omega}}}$$

"spectral factorization"

$$\frac{(z-a)}{(z+b)} \cdot \frac{\left(\frac{1}{z}-a^*\right)}{\left(\frac{1}{z}-b^*\right)}$$

Example

Choose poles and zeros inside u.c.

$$\left|H(e^{j\omega})\right|^2 = \frac{\frac{5}{4} - \cos(\omega)}{\frac{17}{16} + \frac{1}{2}\cos(\omega)}$$

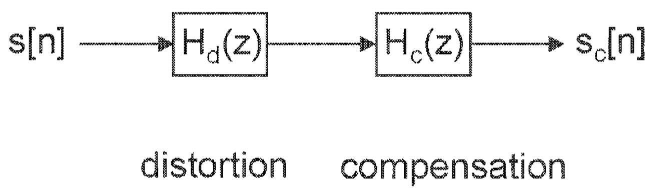
$$= \frac{\left(1-\frac{1}{2}e^{-j\omega}\right)\left(1-\frac{1}{2}e^{j\omega}\right)}{\left(1+\frac{1}{4}e^{-j\omega}\right)\left(1+\frac{1}{4}e^{j\omega}\right)}$$

$$H_{mp}(e^{j\omega}) = \frac{\left(1-\frac{1}{2}e^{-j\omega}\right)}{\left(1+\frac{1}{4}e^{-j\omega}\right)}$$

Example, cont'd

$\forall$ Rational H(z), $H(z) = H_{mp}(z)\, H_{ap}(z)$

↑ reflects zeros outside u.c. to match H(z)



Example, cont'd

If $H_d(z) = H_{dmp}(z)\, H_{dap}(z)$

$$\Rightarrow H_c(z) = \frac{1}{H_{dmp}(z)}$$

$$H_{eff}(z) = H_{dap}(z)$$

Properties of M.P. Systems

For systems with same $|H(e^{j\omega})|$,

- "minimum phase lag"

impose additional constraint

$$H(e^{j0}) = \sum_{n=-\infty}^{\infty} h[n] > 0$$

$$\arg[H(e^{j\omega})] = \arg[H_{mp}(e^{j\omega})] + \arg[H_{ap}(e^{j\omega})]$$

↑ negative $0 \le \omega \le \pi$

$H_{mp}(e^{j\omega})$ has smallest "phase lag" - $\measuredangle H(e^{j\omega})$

Properties of M.P. Systems, cont'd

"minimum group delay"

$$\text{grd}[H(e^{j\omega})] = \text{grd}[H_{mp}(e^{j\omega})] + \underline{\text{grd}[H_{ap}(e^{j\omega})]}$$

$\ge 0 \ \forall\omega$

$H_{mp}(e^{j\omega})$ has smallest group delay

Properties of M.P. Systems, cont'd

- "minimum energy delay"

Note $\forall$ $H(e^{j\omega})$ with same DTFT $|\cdot|$

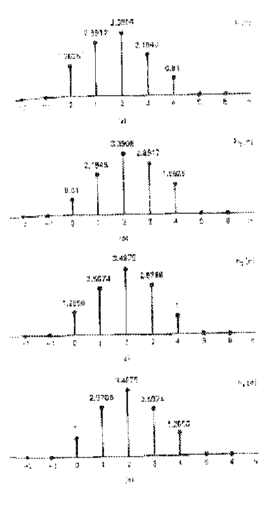
$$\sum_{n=-\infty}^{\infty} |h[n]|^2 = \sum_{n=-\infty}^{\infty} |h_{mp}[n]|^2$$

$$E[n] \triangleq \sum_{m=0}^{n} |h[m]|^2$$

$$\Rightarrow E_{mp}[n] \geq E[n] \quad \forall n$$

"front loaded" imp. resp.

Filter design, realizability, stability, communication channels, causal wiener filtering



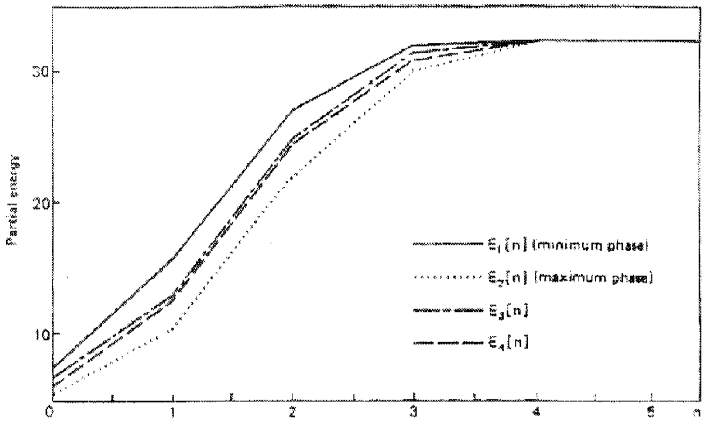


Figure 5.29

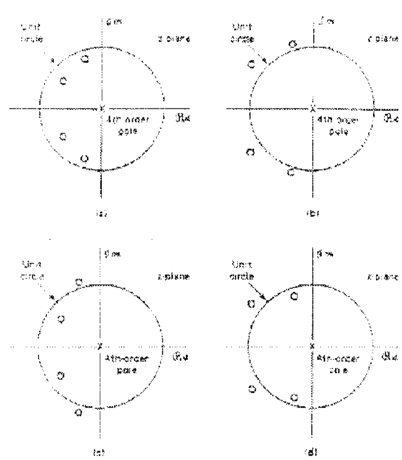


Figure 5.27

Linear Phase

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{-j\omega\alpha}$$

↑ does it have a symmetric imp. resp.?

No. $h[n] = \dfrac{\sin \omega_c (n-\alpha)}{\pi(n-\alpha)}$

$$H(e^{j\omega}) = \begin{cases} e^{-j\omega\alpha}, & |\omega| < \omega_c \\ 0, & \text{else} \end{cases}$$

Generalized Linear Phase

$$H(e^{j\omega}) = A(e^{j\omega}) e^{-j\alpha\omega + j\beta}$$

real, bipolar real, constants

$\tau(\omega) = \alpha,$

$\arg[H(e^{j\omega})] = \beta - \alpha\omega \quad 0 < \omega < \pi$

Generalized Linear Phase, cont'd

$$h[n]=\begin{cases} h[m-n], & 0\le n\le m \\ 0\ , & \text{else}\end{cases} \Rightarrow H(e^{j\omega})=A_e(e^{j\omega})e^{-j\omega m/2}$$

$$h[n]=\begin{cases} -h[m-n], & 0\le n\le m \\ 0\ , & \text{else}\end{cases} \Rightarrow H(e^{j\omega})=A_0(e^{j\omega})e^{-j\omega m/2+j\pi/2}$$

- Sufficient conditions for generalized linear phase with a causal system.
- Necessary?

$A_e(e^{j\omega})=A_e^*(e^{-j\omega})$ conj. even

$A_0(e^{j\omega})=-A_0^*(e^{-j\omega})$ conj. odd

NASC Conditions

$$h_a(t)=\sum_{n=-\infty}^{\infty} h[n]\frac{\sin\pi(t-n)}{\pi(t-n)}$$

is symmetric about some point t = d.

Question: does there exist a causal system which has generalized linear phase but no symmetry?

Equivalent Question

Does a continuous function exist s.t.

$h_a(t)$ is bandlimited to $(-\pi, \pi)$

$h_a(t) = 0, \; t = -1, -2, -3, \ldots$

$h_a(t) = 0, \; t = 1 + r, 2 + r, \ldots$

$h_a(t) = c, \; t = 0$?

Equivalent Question, cont'd

Ans. Yes! $S_r(t) = \dfrac{\Gamma(1+r)}{\Gamma(1+t)\Gamma(1+r-t)}$

Gamma Function:

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad \Gamma(n) = (n-1)!$$

$$\rightarrow S_r(t) \in L^\infty, \in L^2 \text{ for } r > -\frac{1}{2}, \in L^1 \text{ for } r \geq -1$$

$$\rightarrow S_r[n] \in L^\infty \text{ for } r \geq -1, \in L^2 \text{ for } r > -\frac{1}{2}, \in L^1 \text{ for } r \geq 0$$

Equivalent Question, cont'd

$\rightarrow S_0(t) = \dfrac{\sin \pi t}{\pi t}$

$\rightarrow -1 < r < 0 \quad \tau(\omega) = {}^{-r}\!/_{2} \Rightarrow$ time advance!
but h[n] is causal!

$\rightarrow S_r(z) = (1 + z^{-1})^r$

FIR Linear Phase System

Type I: $h[n] = h[m-n], \quad 0 \le n \le m, \quad m$ even,

$$\alpha = \frac{m}{2}, \ \beta = 0, \pi$$

Type II: $h[n] = h[m-n], \ 0 \le n \le m, \ m$ odd,

$$\alpha = \frac{m}{2}, \ \beta = 0, \ \pi$$

FIR Linear Phase System, cont'd

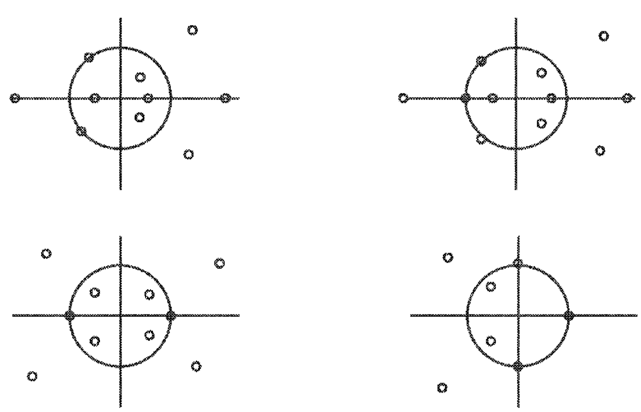
Type III: $h[n] = -h[m-n],\ 0 \le n \le m,\ m$ even,

$$\alpha = \frac{m}{2},\ \beta = \frac{\pi}{2},\ \frac{3\pi}{2}$$

Type IV: $h[n] = -h[m-n],\ 0 \le n \le m,\ m$ odd,

$$\alpha = \frac{m}{2},\ \beta = \frac{\pi}{2},\ \frac{3\pi}{2}$$

FIR Linear Phase System, cont'd



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

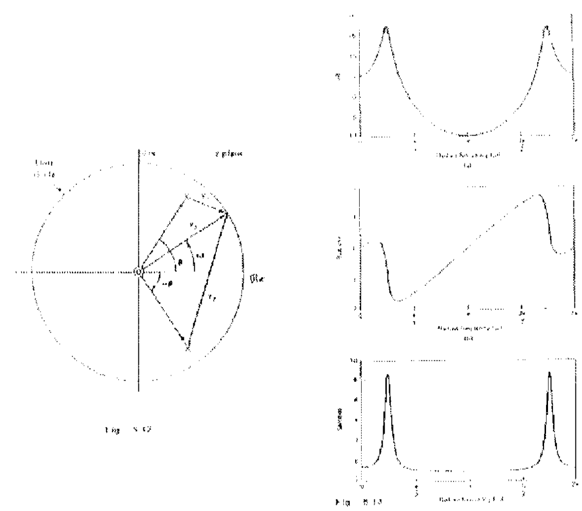
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

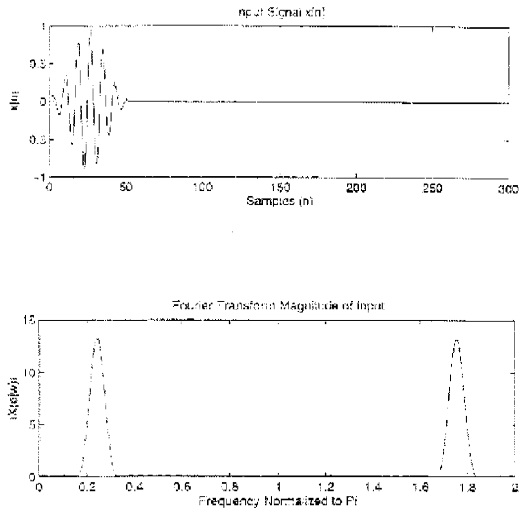
Lecture 6

Professor Andrew Singer
Department of Electrical and
Computer Engineering

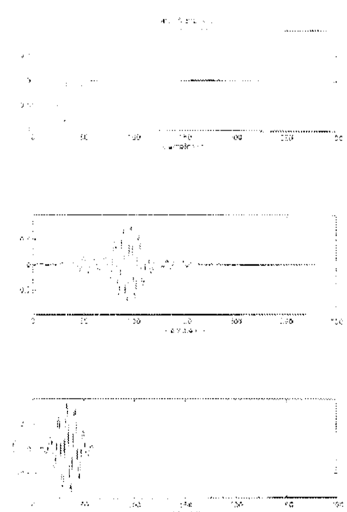
Magnitude, Phase & Group delay



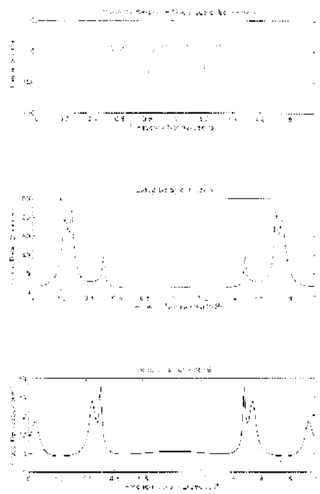
Narrowband pulse $x[n]=s[n]\cos(w_0 n)$



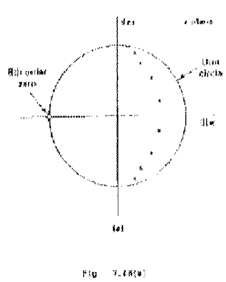
Input and two possible outputs

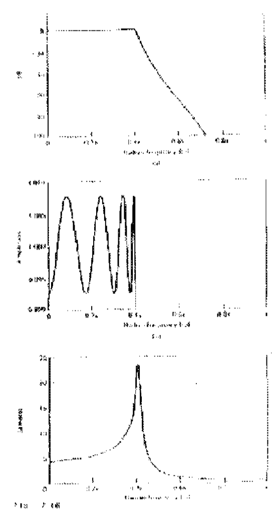


A Tale of 2 Group delays…

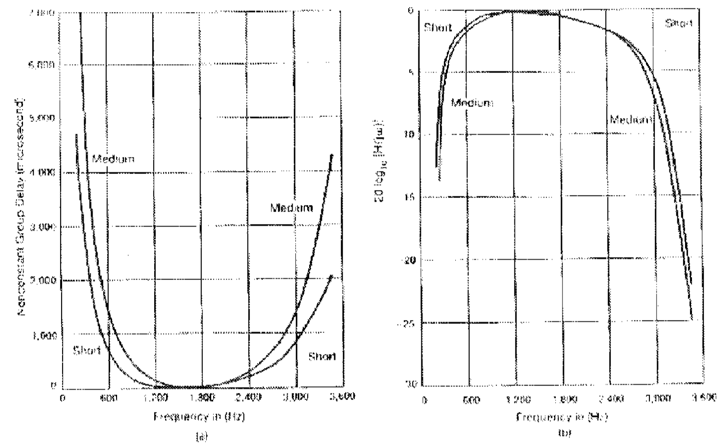


Elliptic Filter



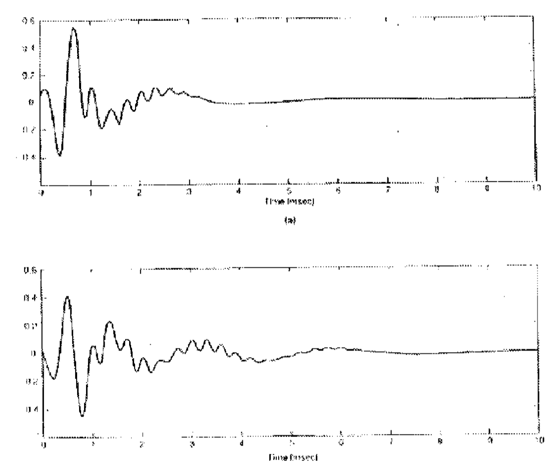


Telephone Channel



"Signals & Systems"
Oppenheim & Willsky
Fig. 6.6

Telephone Channel Groupdelay



"Signals & Systems"
Oppenheim & Willsky
Fig. 6.7

Properties of M.P. Systems

For systems with same $|H(e^{jw})|$,

- "minimum phase lag"

  impose additional constraint

$$H(e^{j0}) = \sum_{n=-\infty}^{\infty} h[n] > 0$$

$$\arg[H(e^{jw})] = \arg[H_{mp}(e^{jw})] + \arg[H_{ap}(e^{jw})]$$

negative $0 \le w \le \pi$

$H_{mp}(e^{jw})$ has smallest "phase lag" - $\measuredangle H(e^{jw})$

Properties of M.P. Systems, cont'd

- "minimum group delay"

$$\text{grd}[H(e^{jw})] = \text{grd}[H_{mp}(e^{jw})] + \underline{\text{grd}[H_{ap}(e^{jw})]}$$

$\ge 0 \ \forall w$

$H_{mp}(e^{jw})$ has smallest group delay

Properties of M.P. Systems, cont'd

- "minimum energy delay"

Note $\forall$ $H(e^{jw})$ with same DTFT $|\cdot|$

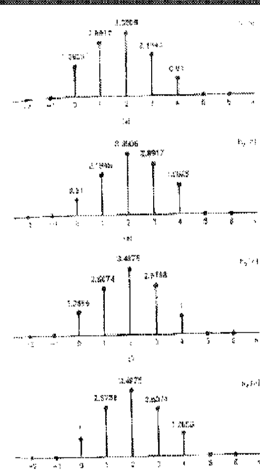
$$\sum_{n=-\infty}^{\infty} |h[n]|^2 = \sum_{n=-\infty}^{\infty} |h_{mp}[n]|^2$$

$$E[n] \triangleq \sum_{m=0}^{n} |h[m]|^2$$

$$\Rightarrow E_{mp}[n] \geq E[n] \quad \forall n$$

"front loaded" imp. resp.

Filter design, realizability, stability, communication channels, causal wiener filtering



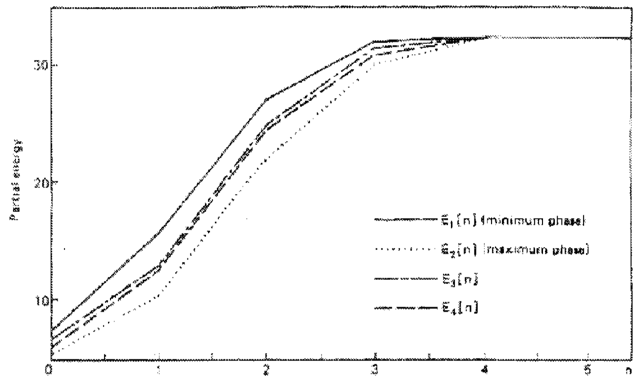


Figure 5.29

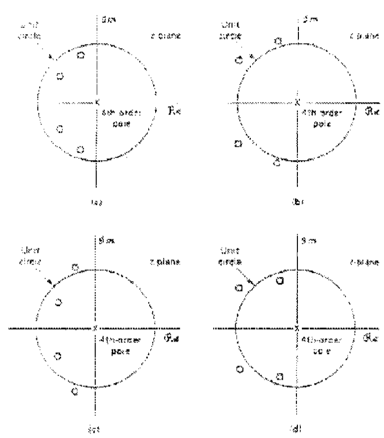


Figure 5.27

Linear Phase

$$H(e^{jw}) = |H(e^{jw})| e^{-jw\alpha}$$

↑ does it have a symmetric imp. resp.?

No. $h[n] = \dfrac{\sin w_c(n-\alpha)}{\pi(n-\alpha)}$

$$H(e^{jw}) = \begin{cases} e^{-jw\alpha}, & |w| < w_c \\ 0, & \text{else} \end{cases}$$

Generalized Linear Phase

$$H(e^{jw}) = A(e^{jw}) e^{-j\alpha w + j\beta}$$

$A(e^{jw})$: real, bipolar; α, β: real, constants

$\tau(w) = \alpha,$

$\arg[H(e^{jw})] = \beta - \alpha w \quad 0 < w < \pi$

Generalized Linear Phase, cont'd

$$h[n] = \begin{cases} h[m-n], & 0 \le n \le m \\ 0, & \text{else} \end{cases} \Rightarrow H(e^{jw}) = A_e(e^{jw})e^{-jwm/2}$$

$$h[n] = \begin{cases} -h[m-n], & 0 \le n \le m \\ 0, & \text{else} \end{cases} \Rightarrow H(e^{jw}) = A_0(e^{jw})e^{-jwm/2 + j\pi/2}$$

- Sufficient conditions for generalized linear phase with a causal system.
- Necessary?

$A_e(e^{jw}) = A_e^*(e^{-jw})$ conj. even

$A_0(e^{jw}) = -A_0^*(e^{-jw})$ conj. odd

NASC Conditions

$$h_a(t) = \sum_{n=-\infty}^{\infty} h[n]\frac{\sin \pi(t-n)}{\pi(t-n)}$$

is symmetric about some point t = d.

Question: does there exist a causal system which has generalized linear phase but no symmetry?

Equivalent Question

Does a continuous function exist s.t.

1) $h_a(t)$ is bandlimited to $(-\pi, \pi)$
2) $h_a(t) = 0, \ t = -1, -2, -3, \ldots$
3) $h_a(t) = 0, \ t = 1 + r, 2 + r, \ldots$
4) $h_a(t) = c, \ t = 0$?

Equivalent Question, cont'd

Ans. Yes! $S_r(t) = \dfrac{\Gamma(1+r)}{\Gamma(1+t)\Gamma(1+r-t)}$

Gamma Function:

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad \Gamma(n) = (n-1)!$$

$$\rightarrow S_r(t) \in L^\infty, \in L^2 \text{ for } r > -\frac{1}{2}, \in L^1 \text{ for } r \geq -1$$

$$\rightarrow S_r[n] \in L^\infty \text{ for } r \geq -1, \in L^2 \text{ for } r > -\frac{1}{2}, \in L^1 \text{ for } r \geq 0$$

Equivalent Question, cont'd

$$\rightarrow S_0(t) = \frac{\sin \pi t}{\pi t}$$

$$\rightarrow -1 < r < 0 \quad \tau(w) = -r/2 \quad \Rightarrow$$ time advance!
but h[n] is causal!

$$\rightarrow S_r(z) = (1 + z^{-1})^r$$

FIR Linear Phase System

Type I: $h[n] = h[m-n], \quad 0 \le n \le m, \quad m$ even,

$$\alpha = \frac{m}{2}, \ \beta = 0, \pi$$

Type II: $h[n] = h[m-n], \ 0 \le n \le m, \ m$ odd,

$$\alpha = \frac{m}{2}, \ \beta = 0, \ \pi$$

FIR Linear Phase System, cont'd

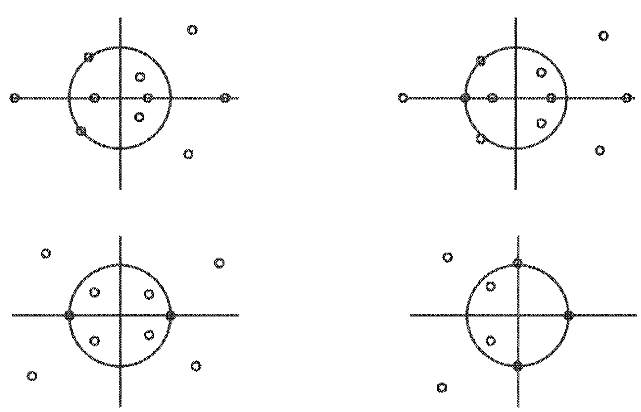
Type III: $h[n] = -h[m-n],\ 0 \le n \le m$, m even,

$$\alpha = \frac{m}{2},\ \beta = \frac{\pi}{2},\ \frac{3\pi}{2}$$

Type IV: $h[n] = -h[m-n],\ 0 \le n \le m$, m odd,

$$\alpha = \frac{m}{2},\ \beta = \frac{\pi}{2},\ \frac{3\pi}{2}$$

FIR Linear Phase System, cont'd



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 7

Professor Andrew Singer
Department of Electrical and Computer Engineering

Structures for DT Systems

System Function: $H(z) = \dfrac{b_0 + b_1 z^{-1}}{1 - a z^{-1}}$

Impulse Response: $h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1]$

(causal)

$\rightarrow$ IIR, cannot implement via convolution.

Difference Equation Representation

Difference equation:

$$y[n] - ay[n-1] = b_0 x[n] + b_1 x[n-1]$$

$$y[n] = ay[n-1] + b_0 x[n] + b_1 x[n-1]$$

- Recursive algorithm for implementation, assuming the system is causal.
- We require: initial conditions (e.g., initial rest)

Backwards Difference Equation

What about:

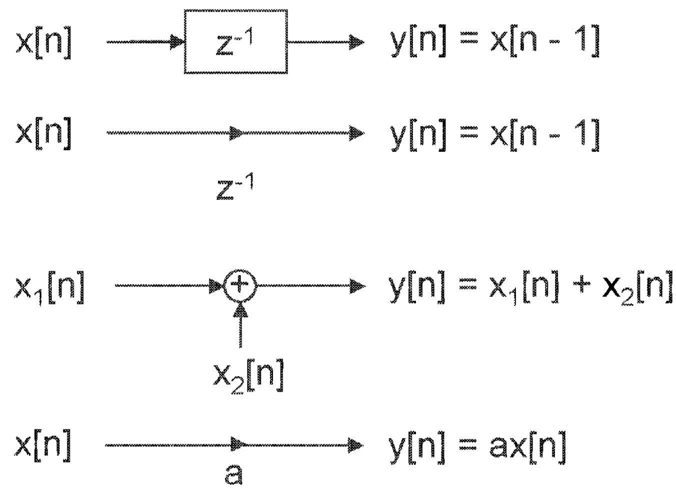
$$y[n-1] = \frac{1}{a}(y[n] + b_0 x[n] + b_1 x[n-1]) \quad ?$$

Recursive implementation for noncausal system.

We will discuss implementation of causal DT systems described by LCCDE's.

Non-causal DT systems require splitting into causal and anti-causal subsections.

Delay Adder Gain Models



Difference Equation Notation

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k \; x[n-k]$$

$$\rightarrow y[n] = \frac{1}{a_0}\left\{-\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k \, x[n-k]\right\}$$

"direct form implementation"

Assume $a_0 = 1$, rewrite LCCDE's now such that

Canonical Notation

$$y[n] = \sum_{k=1}^{N} a_k \, y[n-k] + \sum_{k=0}^{M} b_k \, x[n-k]$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

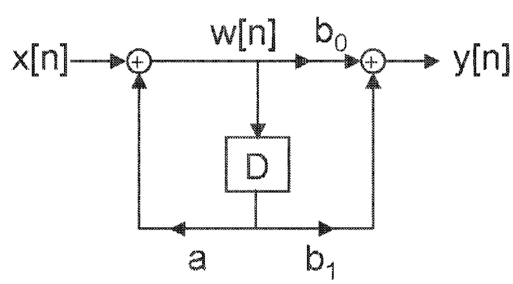
- Numerator ← feed forward, zeros
- Denominator ← poles, feedback

Signal Flow Graph

DAG models express linear equations: we can write flowgraphs using z-domain variables.

Why?

Consider:



Why not do it all in the time domain?

$w[n] = x[n] + a\, w[n-1]$

$y[n] = b_0\, w[n] + b_1\, w[n-1]$

$y[n] = b_0(x[n] + aw[n-1]) + b_1 w[n-1]$

$w[n-1] = x[n-1] + aw[n-2]$

$y[n] = b_0(x[n] + a(x[n-1] + aw[n-2]) + b_1(x[n-1] + aw[n-2])$

$y[n-1] = b_0 w[n-1] + b_1 w[n-2]$

$= b_0(x[n-1] + aw[n-2]) + b_1 w[n-2]$

$= b_0 x[n-1] + (b_0 a + b_1)\, w[n-2]$

$= b_0 x[n-1] + (b_0 a + b_1)(w[n-1] - x[n-1])\frac{1}{a}$

$= b_0 x[n-1] + (b_0 a + b_1)\left(\frac{1}{a}\right)\left((y[n] - b_0 x[n])\left(\frac{1}{b_0 a + b_1}\right) - x[n-1]\right)$

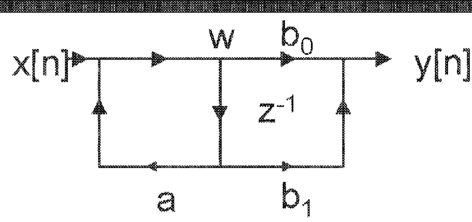
$y[n-1] = b_0 x[n-1] + \frac{1}{a} y[n] - \frac{b_0}{a} x[n] - \frac{1}{a} x[n-1](b_0 a + b_1)$

$ay[n-1] = y[n] - b_0 x[n] - b_1 x[n-1]$

$y[n] = ay[n-1] + b_0 x[n] + b_1 x[n-1]$

All of this work, just to get y[n] in terms of x[n]!

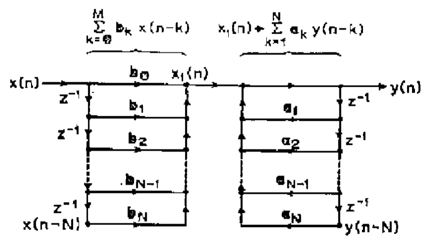
Now using the z-domain



$w(z) = x(z) + az^{-1}w(z) \Rightarrow w(z)(1-az^{-1}) = x(z)$

$y(z) = b_0 w(z) + b_1 z^{-1} w(z) \Rightarrow y(z) = (b_0 + b_1 z^{-1})w(z)$

$$\therefore \frac{y(z)}{x(z)} = \frac{(b_0 + b_1 z^{-1})}{(1 - az^{-1})}$$

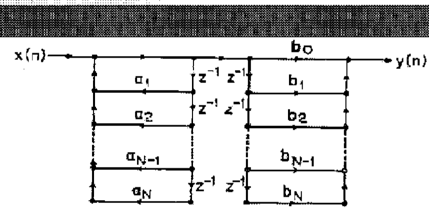


Direct Form I

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

$$y(n) = \sum_{k=0}^{M} b_k\, x(n-k) + \sum_{k=1}^{N} a_k\, y(n-k)$$

$$H(z) = \left[\sum_{k=0}^{M} b_n z^{-k}\right]\left[\frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}}\right]$$

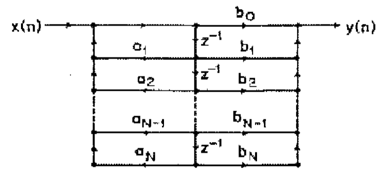


$$y(n) = \sum_{k=0}^{M} b_k\, x(n-k) + \sum_{k=1}^{N} a_k\, y(n-k)$$

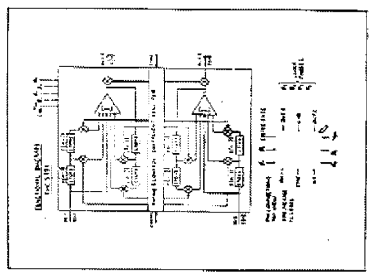
$$H(z) = \left[\frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}}\right]\left[\sum_{k=0}^{M} b_k z^{-k}\right]$$

$$y_1(n) = x(n) + \sum_{k=1}^{N} a_k\, y_1(n-k)$$

$$y(n) = \sum_{k=0}^{M} b_k\, y_1(n-k)$$



Direct Form II

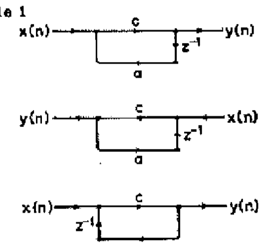


TRANSPOSITION THEOREM

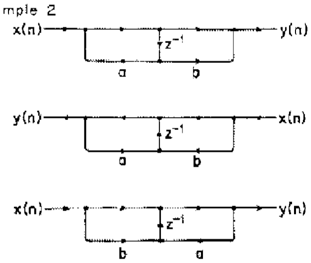
1. REVERSE DIRECTION OF ALL BRANCHES
2. INTERCHANGE INPUT AND OUTPUT

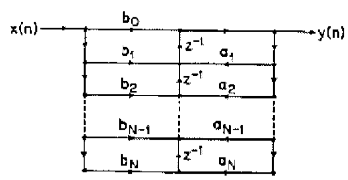
TRANSFER FUNCTION REMAINS THE SAME

Example 1



Example 2



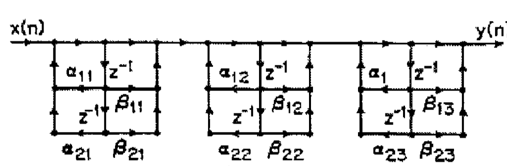


Cascade Form

Cascade Structure

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

$$= A \prod \frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}}$$

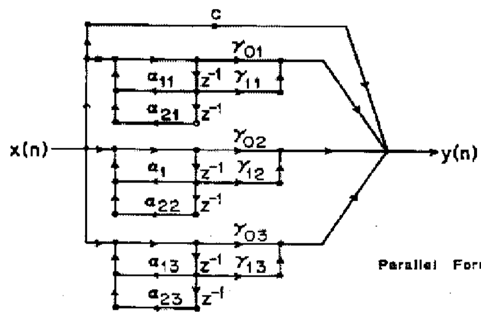


Parallel Form

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

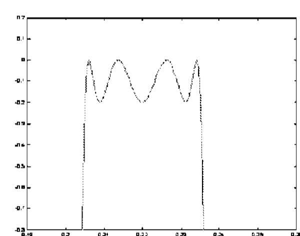
$$= \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})} + \sum_{k=0}^{M-N} c_k z^{-k}$$

$$= \sum_{k=1}^{\frac{N+1}{2}} \frac{(\gamma_{0k} + \gamma_{1k} z^{-1})}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}} + \sum_{k=0}^{M-N} c_k z^{-k}$$

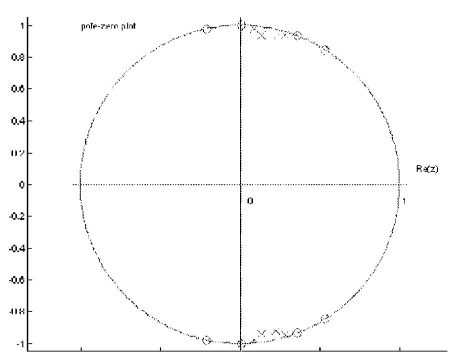


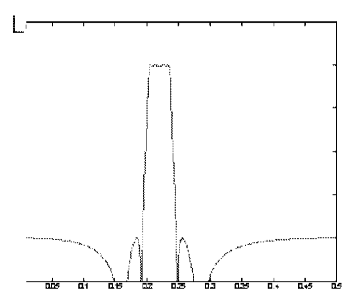
8th order Elliptic Filter

Passband log-magnitude



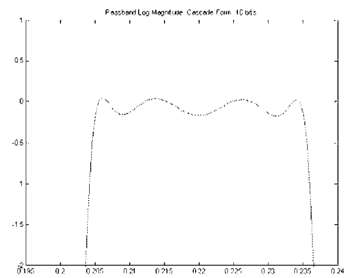
Pole-zero plot



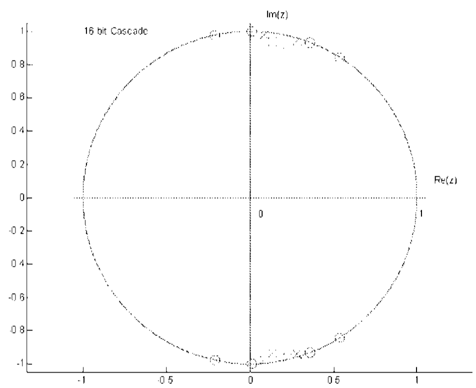


16 Bit Cascade Form

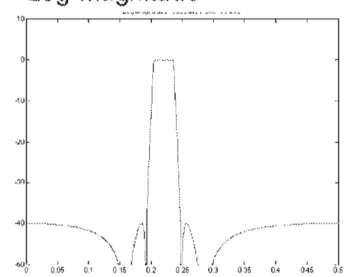
Passband log-magnitude



Pole-zero plot

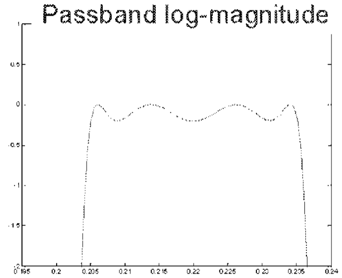


Log-magnitude

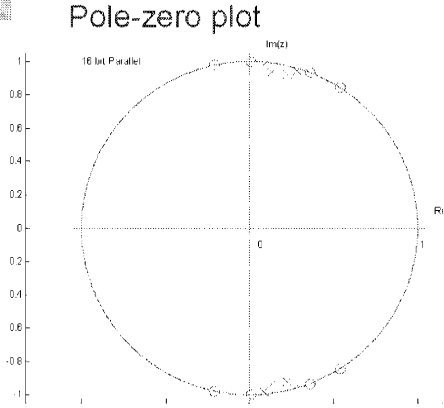


16 Bit Parallel Form

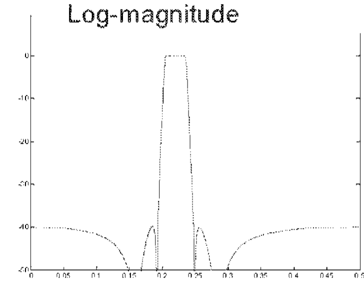
Passband log-magnitude



Pole-zero plot

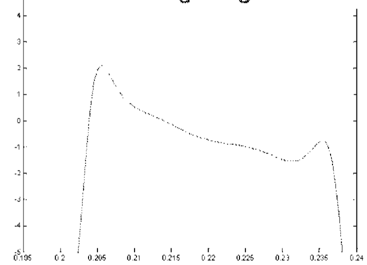


Log-magnitude

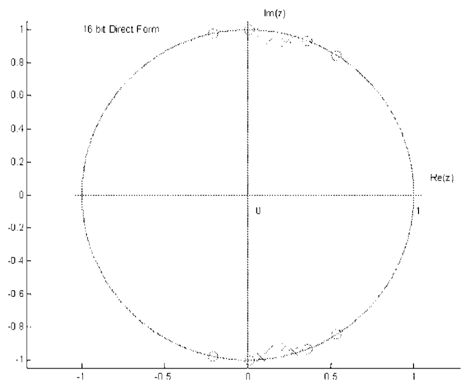


16 Bit Direct Form

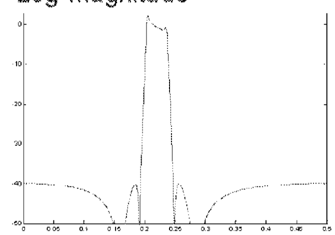
Passband log-magnitude



Pole-zero plot

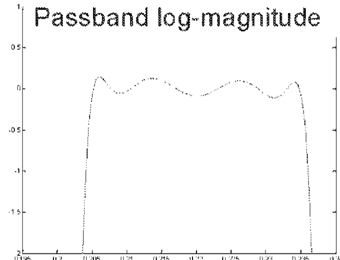


Log-magnitude

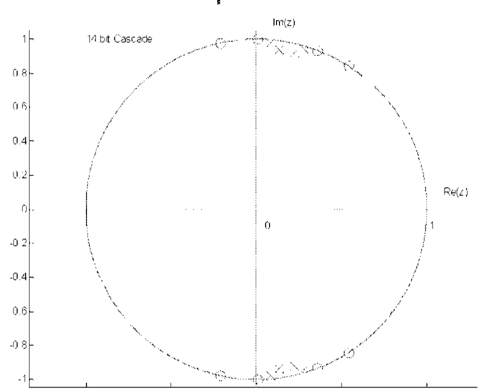


14 Bit Cascade Form

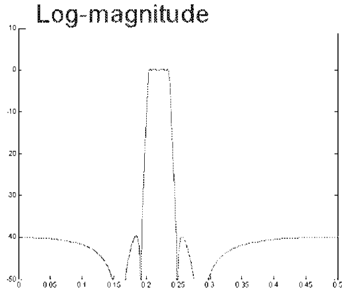
Passband log-magnitude

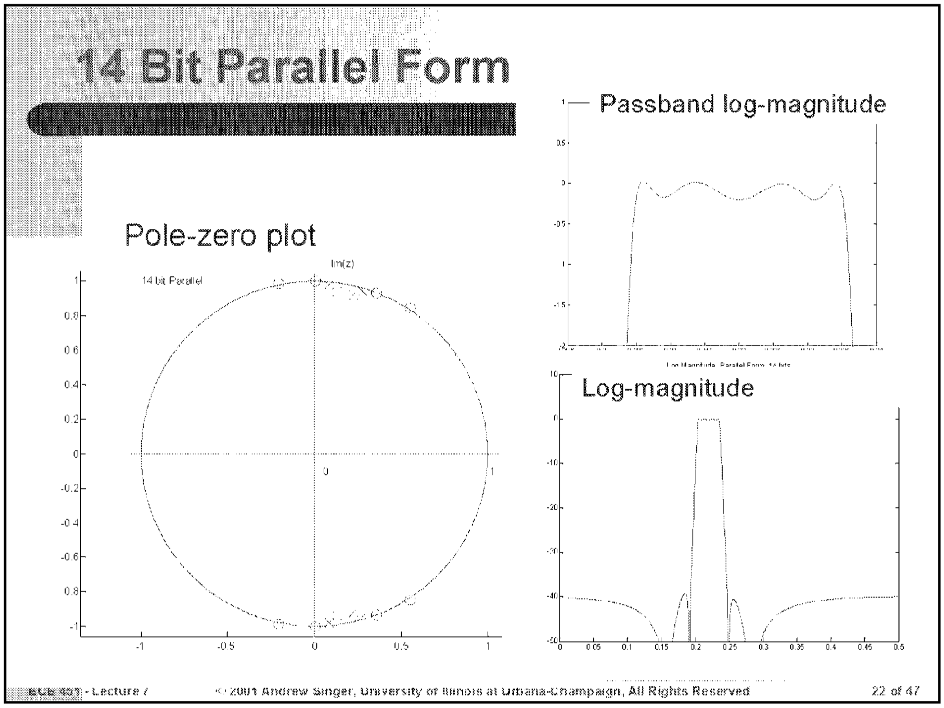


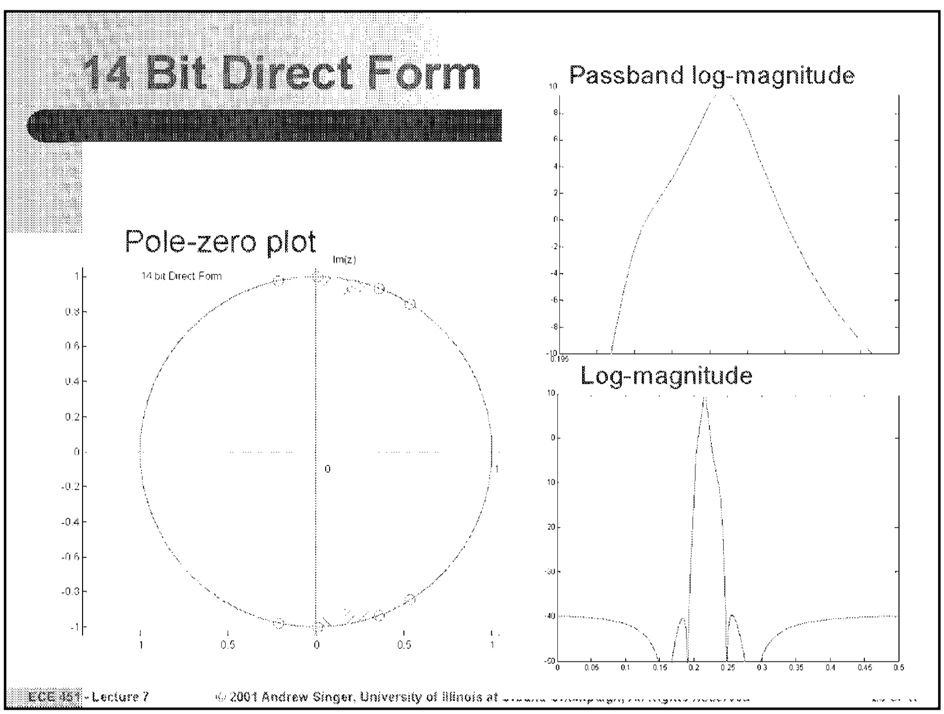
Pole-zero plot



Log-magnitude

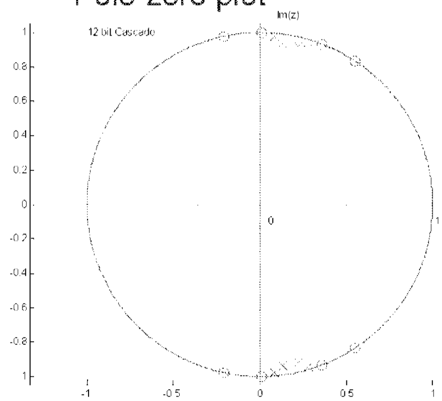



14 Bit Parallel Form
Passband log-magnitude
Pole-zero plot
14 bit Parallel
Im(z)
Log-magnitude
ECE 451 - Lecture 7
© 2001 Andrew Singer, University of Illinois at Urbana-Champaign, All Rights Reserved
22 of 47

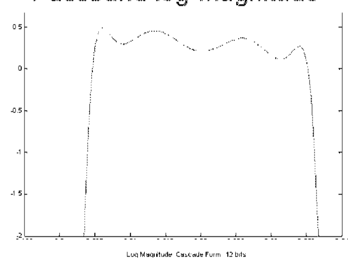

14 Bit Direct Form
Passband log-magnitude
Pole-zero plot
14 bit Direct Form
Im(z)
Log-magnitude
ECE 451 - Lecture 7
© 2001 Andrew Singer, University of Illinois at

12 Bit Cascade Form

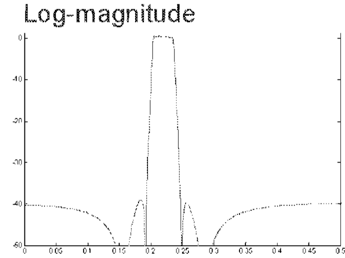
Pole-zero plot



Passband log-magnitude

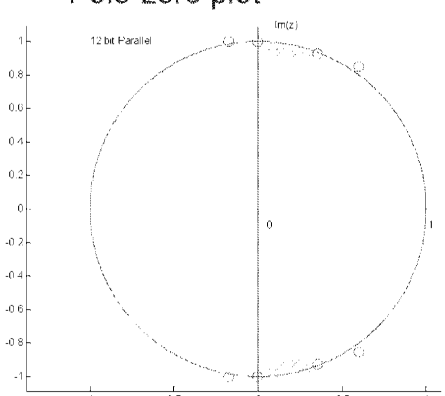


Log-magnitude

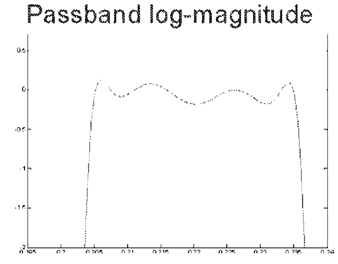


12 Bit Parallel Form

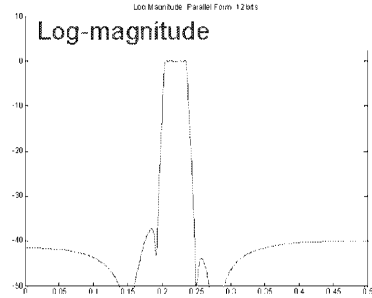
Pole-zero plot



Passband log-magnitude

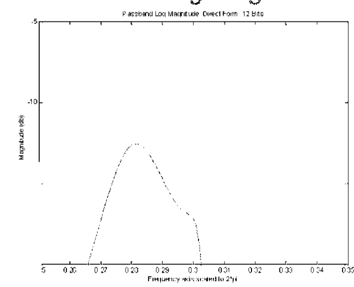


Log-magnitude

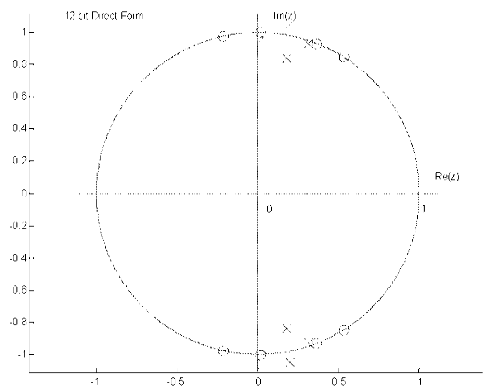


12 Bit Direct Form

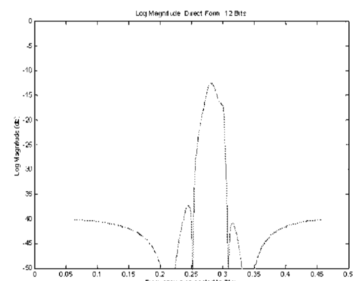
Passband log-magnitude



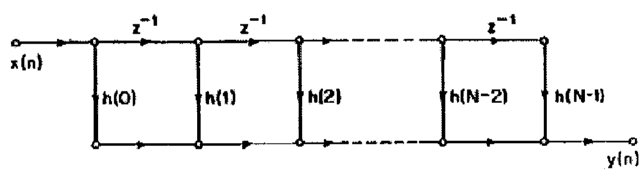
Pole-zero plot



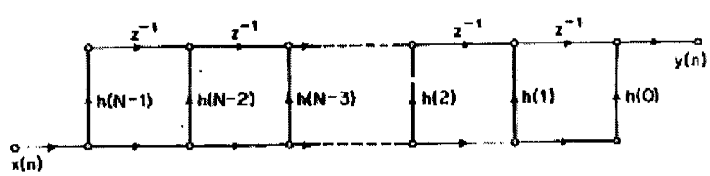
Log-magnitude



FIR Direct Form

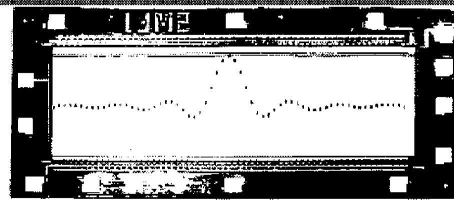


Direct-form realization of an FIR system.

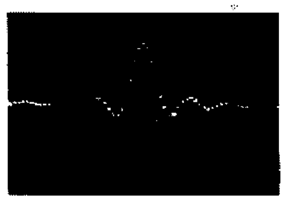


Transposition of the network

Lowpass Filter



(A) Photomicrograph of Low Pass Transversal Filter



(B) Observed Impulse Response

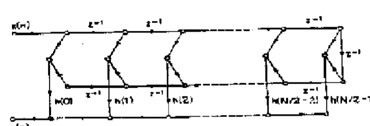
Symmetric FIR

$$h(n) = h(N-1-n)$$
$$h(0) = h(N-1)$$
$$h(1) = h(N-2)$$

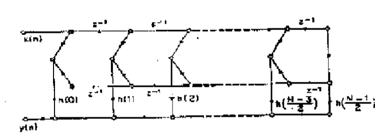
etc.

$$H(z) = \sum_{n=0}^{N-1} h(n)\, z^{-n}$$
$$= h(0)\left[1 + z^{-N+1}\right] + h(1)\left[z^{-1} + z^{-(N-2)}\right] + \ldots$$

(Last term slightly different depending on whether N is even or odd.)



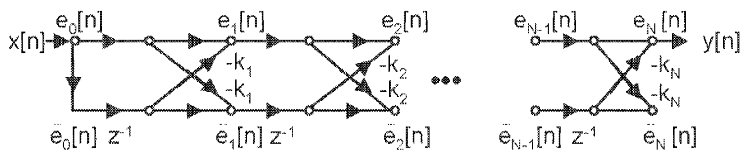
Direct-form realization for an FIR system of even order with linear phase.



Direct-form realization for an FIR system of odd order with linear phase.

Lattice Structures

- Adaptive filtering
- All pole modeling, speech analysis
- Linear prediction, equalization



FIR Lattice System

1) $e_0[n] = \tilde{e}_0[n] = x[n]$

$$e_i[n] = e_{i-1}[n] - k_i\tilde{e}_{i-1}[n-1], \qquad i = 1, \ldots, N$$

$$\tilde{e}_i[n] = -k_i e_{i-1}[n] + \tilde{e}_{i-1}[n-1], \qquad i = 1, \ldots, N$$

$$e_N[n] = y[n]$$

FIR Lattice System, cont'd

Note:
- System is FIR, Nth order
- $h[0] = 1$, $h[N] = -k_N$
- k_i's called "k-coefficients" or "reflection coefficients"
- System function

$$H(z) = 1 - \sum_{k=1}^{N} a_k z^{-k} \triangleq A_N(z)$$

FIR Lattice System, cont'd

Define:

$$A_i(z) = \frac{E_i(z)}{E_0(z)} = 1 - \sum_{k=1}^{i} a_k^{(i)} z^{-k}$$

$$\tilde{A}_i(z) = \frac{\tilde{E}_i(z)}{\tilde{E}_0(z)} = 1 - \sum_{k=1}^{i} \tilde{a}_k^{(i)} z^{-k}$$

i-th transfer function

FIR Lattice System, cont'd

(1) In z-domain:

$$E_0(z) = \tilde{E}_0(z) = x(z)$$

$$E_i(z) = E_{i-1}(z) - k_i z^{-1}\tilde{E}_{i-1}(z)$$

$$\tilde{E}_i(z) = -k_i E_{i-1}(z) + z^{-1}\tilde{E}_{i-1}(z)$$

$$Y(z) = E_N(z)$$

FIR Lattice System, cont'd

2)

$$\left(\begin{aligned} &A_i(z) = A_{i-1}(z) - k_i z^{-i} A_{i-1}(z^{-1}) \\ &\tilde{A}_i(z) = z^{-i} A_i(z^{-1}) \end{aligned} \right) \text{Recursion}$$

FIR Lattice System, cont'd

⇒ Can be proven by induction:

- $A_0(z) = \tilde{A}_0(z) = 1$

- $A_1(z) = E_1(z)/E_0(z) = \dfrac{x(z) - k_1 z^{-1} x(z)}{x(z)}$

$$= (1 - k_1 z^{-1}) = A_0(z) - k_1 z^{-1} A_0(z^{-1})$$

FIR Lattice System, cont'd

- $A_i(z) = E_i(z)/E_0(z)$

$$= [E_{i-1}(z) - k_i z^{-1} \tilde{E}_{i-1}(z)]/x(z)$$

$$= A_{i-1}(z) - k_i z^{-1} (\tilde{E}_{i-1}(z)/x(z))$$

$$= A_{i-1}(z) - k_i z^{-1} \tilde{A}_{i-1}(z)$$

$$= A_{i-1}(z) - k_i z^{-1} z^{-i+1} A_{i-1}(z^{-1})$$

FIR Lattice System, cont'd

Recursion for $\tilde{A}_n$:

$$\tilde{A}_1(z) = z^{-1}A_1(z^{-1}) = z^{-1}(1 - k_1 z)$$

$$= z^{-1} - k_1$$

$$\tilde{A}_i(z) = \tilde{E}_i(z) / E_0(z)$$

$$= [-k_i E_{i-1}(z) + z^{-1}\tilde{E}_{i-1}(z)] / x(z)$$

FIR Lattice System, cont'd

$$= -k_i A_{i-1}(z) + z^{-1}\tilde{A}_{i-1}(z)$$

$$= -k_i A_{i-1}(z) + z^{-i}A_{i-1}(z^{-1})$$

$$= z^{-i}(A_{i-1}(z^{-1}) - k_i z^i A_{i-1}(z))$$

$$= z^{-i}A_i(z^{-1})$$

FIR Lattice System, cont'd

Write $A_i(z) = 1 - \sum_{k=1}^{i} a_k^{(i)} z^{-k}$ in (2) and

match terms of order k:

$$\Rightarrow 1 - \sum_{k=1}^{i} a_k^{(i)} z^{-k} = 1 - \sum_{k=1}^{i-1} a_k^{(i-1)} z^{-k} - k_i z^{-i}(1 - \sum_{k=1}^{i-1} a_k^{(i-1)} z^{k})$$

$$\sum_{k=1}^{i} a_k^{(i)} z^{-k} = \sum_{k=1}^{i-1} a_k^{(i-1)} z^{-k} + k_i z^{-i} - k_i \sum_{k=1}^{i-1} a_k^{(i-1)} z^{k-i}$$

FIR Lattice System, cont'd

3)

$$\Rightarrow \quad a_i^{(i)} = k_i$$

$$\Rightarrow \quad a_k^{(i)} = a_k^{(i-1)} - k_i a_{i-k}^{(i-1)}$$

i = 1, …, N

FIR Lattice System, cont'd

- To recover a_k's in $H(z) = 1 - \sum_{k=1}^{N} a_k z^{-k}$,need $a_k^{(N)}$, k = 1, …, N, since H(z) = $A_N(z)$
- Begin with $a_1^1 = k_1$, then repeat 3) for i = 1, … N to obtain $a_k^{(i)}$ $\forall i = 1, \ldots, N$

$\Rightarrow$ This converts from k-coefficients to the a_k's in H(z)

To convert back to k-coefficients from a_k's:

FIR Lattice System, cont'd

4)

$$k_N = a_N^{(N)}$$

$$k_i = a_i^{(i)} \qquad i = N, \ldots, 1$$

$$a_m^{(i-1)} = (a_m^{(i)} + k_i a_{i-m}^{(i)}) / (1 - k_i^2)$$

To obtain k_i's from $a_k\text{'s} = a_k^{(N)}\text{'s}$, repeat (4)

for i = N, N – 1, …, 1

All Pole (IIR) Lattice

5) $e_N[n] = x[n]$

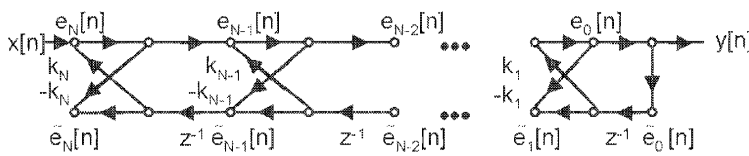
$$e_{i-1}[n] = e_i[n] + k_i \tilde{e}_{i-1}[n-1]$$

$$\tilde{e}_i[n] = -k_i e_{i-1}[n] + \tilde{e}_{i-1}[n-1]$$

$$y[n] = e_0[n] = \tilde{e}_0[n]$$

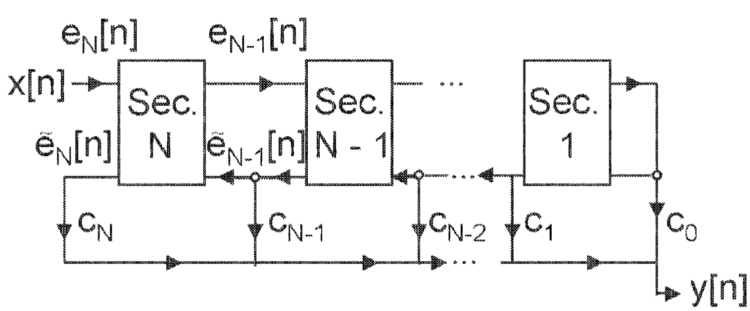
inverse system

$$H(z) = \frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$



Pole-Zero Lattice

Generalize to poles and zeros:



Pole-Zero Lattice, cont'd

6)

$$b_m = c_m - \sum_{i=m+1}^{N} c_i a_{i-m}^{(i)}$$

$$H(z) = \frac{B(z)}{A(z)}, \quad B(z) = \sum_{m=0}^{N} b_m z^{-m}$$

- echo cancellation, pole-zero modeling adaptive equalization, etc…

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

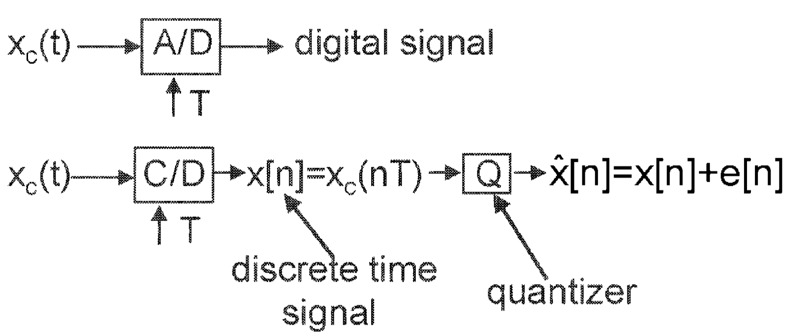
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 8

Professor Andrew Singer
Department of Electrical and
Computer Engineering

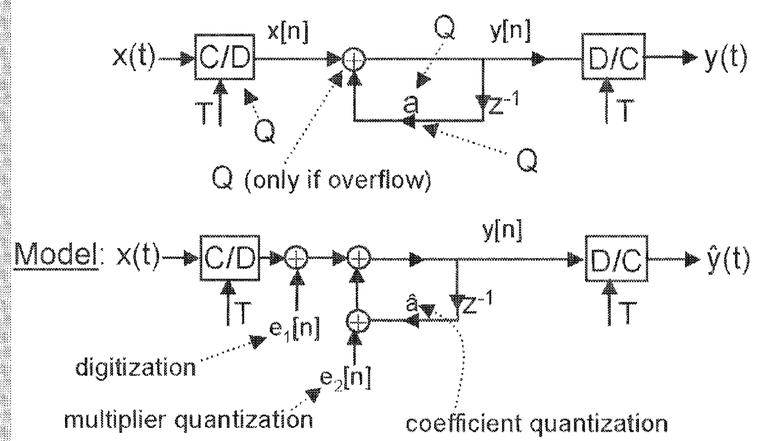
Sampling and Quantization



$$\hat{x}[n]=Q(x[n])$$

Quantization in an LTI System

Where does quantization affect a system?



Two Categories of Quantization Effects

- Coefficient quantization:
  - Frequency response changes due to finite precision
  - "static" effect of quantization
- Arithmetic with round-off in LTI systems
  - Adds quantization "noise" into computations
  - "dynamic" effect of quantization

Coefficient Quantization

For $H(z) = \dfrac{B(z)}{A(z)}$, look at the poles: $A(z) = 1-\sum_{k=1}^{N} a_k z^{-k}$

We want $\dfrac{dz_i}{da_k} \Rightarrow \left.\dfrac{dA(z)}{dz_i}\right|_{z=z_i} \dfrac{dz_i}{da_k} = \left.\dfrac{dA(z)}{da_k}\right|_{z=z_i}$

$$\frac{dz_i}{da_k} = \frac{\left.\dfrac{dA(z)}{da_k}\right|_{z=z_i}}{\left.\dfrac{dA(z)}{dz_i}\right|_{z=z_i}} = \frac{z_i^{N-k}}{\prod_{k=1,k\neq j}^{N}(z_j - z_k)}$$

$$\Delta z_i \approx \sum_{k=1}^{N} \frac{dz_i}{da_k}\Delta a_k$$

→ Worse for closely spaced poles/zeros!

→ Worse for more poles/zeros!

Implications:

- Direct form implementation is bad for N>2
- Cascade or parallel subsections (second order) are much better for coefficient quantization of poles
- Parallel structure gets the zeros from all sections, but is still better than direct form
- Normalized lattice structure is more robust, but also requires about twice as much computation
- We can play with the flow graph to improve quantization for all-pass, complex conjugate pairs, etc.

For FIR Filters:

- Linear phase can be preserved
- Zeros are typically spread out, so direct form can be ok

Fixed Point Arithmetic

- 2's complement → most common
- Sign magnitude, Offset binary, others

In general assume:

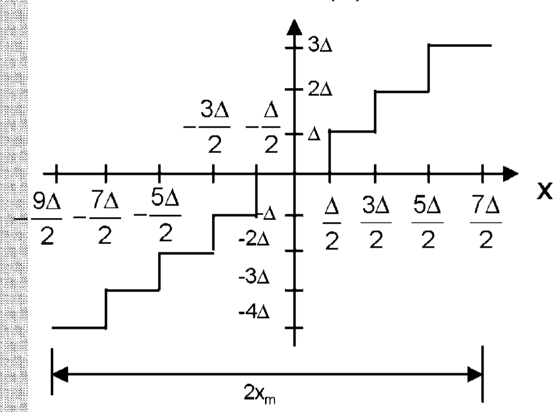
$$|x[n]| < x_m$$

$$\hat{x}[n] = x_m\underbrace{(-a_0 2^0 + a_1 2^{-1} + \ldots + a_B 2^{-B})}_{\text{(B + 1) bit representation}}$$

$$\hat{x}[n] = x_m(-b_0 + \sum_{i=1}^{B} b_i 2^{-i})$$

2's Complement

$$\hat{x} = Q(x)$$



2's complement	
0.11	¾
0.10	½
0.01	¼
0.00	0
1.11	-¼
1.10	-½
1.01	-¾
1.00	-1

Quantization Error Signal e[n]

Error range is $\pm\frac{\Delta}{2}$, $\Delta = \frac{2x_m}{2^{B+1}} = \frac{x_m}{2^B}$ = lsb step size

$$\frac{-x_m}{2}2^{-B} \le e \le \frac{x_m}{2}2^{-B}$$ → (for rounding)

$$\frac{-\Delta}{2} \le e[n] \le \frac{\Delta}{2}$$ → As long as $|x[n]| < x_m$

$$-x_m - \frac{\Delta}{2} \le x[n] \le x_m - \frac{\Delta}{2}$$

$$\hat{x}[n] = x[n] + e[n]$$

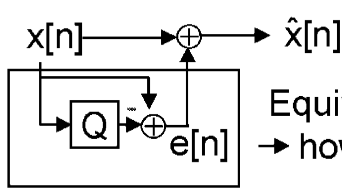
Analysis of Quantization Errors

Q(·) is nonlinear and hard to analyze

Simplified model: $x[n] \rightarrow \boxed{Q} \rightarrow \hat{x}[n]$

$$\Downarrow$$

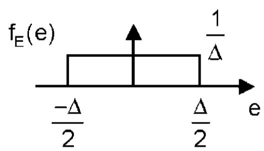
linear → $x[n] \rightarrow \oplus \rightarrow \hat{x}[n]$, with $e[n]$ entering $\oplus$ from below



Equivalent if e[n] is correct.
→ how can we model e[n]?

"Leap of Faith" Simplifying Assumptions

- e[n] is a sample function of wss random process
- e[n] is uncorrelated with x[n]
- e[n] is white, i.e. uncorrelated samples
- pdf (uniform pdf)



(can also assume Gaussian PDF, but want
$\sigma^2 << 2^{-B}$ $\sigma^2 \alpha \Delta^2$)

Round-off Noise In Digital Filters

$$E\{e[n]\} = m_e = 0, \quad \sigma_e^2 = \frac{X_m^2 2^{-2B}}{12}$$

Autocorrelation Function

$$R_{ee}[m] = E\{e[n]e[n+m]\}$$

$$= \sigma_e^2 \delta[m] + m_e^2 = \sigma_e^2 \delta[m]$$

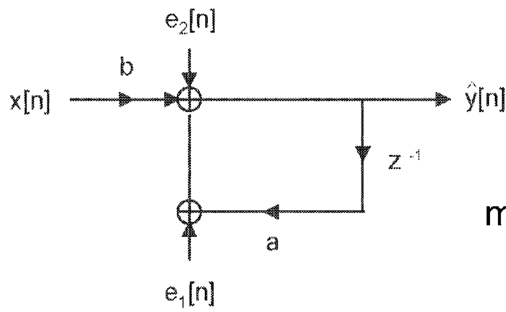
Power Spectrum

$$S_{ee}(e^{jw}) = DTFT(R_{ee}[m])$$

$$= \sigma_e^2 \qquad \text{"white noise"}$$

First-order IIR Example

First-order IIR (let $x_m = 1$)

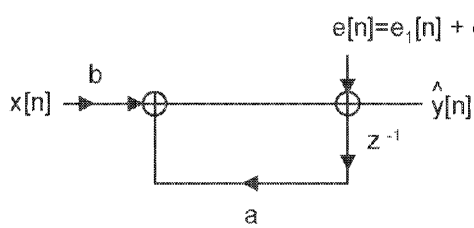


$$m_{e_1} = m_{e_2} = 0$$

$$\sigma_{e_1}^2 = \sigma_{e_2}^2 = \frac{2^{-2B}}{12}$$

Linear Quantization Noise Model

Equivalent to:



$$m_e[n] = 0$$

$$\sigma_e^2 = \frac{2(2^{-2B})}{12}$$

$$H(z) = \frac{b}{1 - az^{-1}} \quad H_e(z) = \frac{1}{1 - az^{-1}}$$

$$h_e[n] = a^n u[n]$$

Response due to the noise

$$\hat{y}[n] = y[n] + f[n]$$

$$f[n] = a\ f[n-1] + e[n]$$

$$m_f = E\{f[n]\} = E\left\{\sum_{k=-\infty}^{\infty} h_e[k]e[n-k]\right\} =$$

$$= m_e \sum_{k=-\infty}^{\infty} h_e[k] = m_e H_e(e^{j0})$$

$$= 0$$

Second moment of the noise response

$$\sigma_f^2 = E\left\{(f[n]-m_f)^2\right\} = E\left\{f[n]^2\right\}$$

$$= E\left\{\sum_{k=-\infty}^{\infty} h_e[k]e[n-k] \sum_{\ell=-\infty}^{\infty} h_e[\ell]e[n-\ell]\right\}$$

$$= \sum_{k=-\infty}^{\infty}\sum_{\ell=-\infty}^{\infty} h_e[k]h_e[\ell]E\left\{e[n-k]e[n-\ell]\right\}$$

$$= \sum_{k=-\infty}^{\infty}\sum_{\ell=-\infty}^{\infty} h_e[k]h_e[\ell]\delta[k-\ell]\sigma_e^2$$

Second moment, cont'd

$$\sigma_f^2 = \sigma_e^2 \sum_{\ell=-\infty}^{\infty} |h_e[\ell]|^2 = \sigma_e^2 \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_e(e^{jw})|^2 \, dw$$

$$= \frac{2\, 2^{-2B}}{12} \sum_{k=0}^{\infty} a^{2k}$$

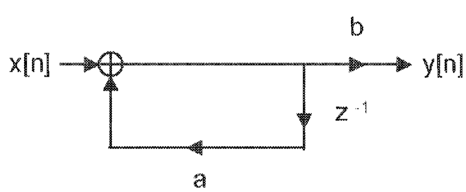
$$\sigma_f^2 = \frac{2\, 2^{-2B}}{12} \left(\frac{1}{1-a^2} \right), \qquad m_f = 0$$

$$\hat{y}[n] = y[n] + f[n]$$

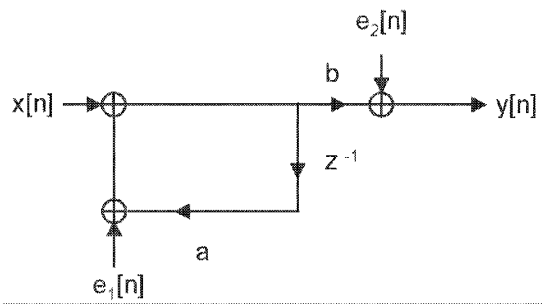
$$a \to 1 \qquad \sigma_f^2 \to \infty$$

Example with Pole First

What about?



Linear Noise Model



$$m_f = 0, \quad \sigma_f^2 = \underbrace{\frac{2^{-2B}}{12}\left(\frac{b^2}{1-a^2}\right)}_{\text{pole}} + \underbrace{\frac{2^{-2B}}{12}}_{\text{zero}}$$

In General

DFII: $$\sigma_f^2 = \frac{N\,2^{-2B}}{12}\sum_{k=-\infty}^{\infty}|h[k]|^2 + (M+1)\frac{2^{-2B}}{12}$$

DFI: $$\sigma_f^2 = (M+N+1)\frac{2^{-2B}}{12}\sum_{k=-\infty}^{\infty}|h[k]|^2$$

What About Overflow?

Let $x_m = 1$ Require $|y[n]| \le 1$

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

$$|y[n]| = \left|\sum_{k=-\infty}^{\infty} h[k]x[n-k]\right| \le \sum_{k=-\infty}^{\infty} |h[k]||x[n-k]|$$

$$\le X_{max} \sum_{k=-\infty}^{\infty} |h[k]|$$

X_{max}: max |x[n]|

not x_m

What About Overflow?, cont'd

$$\text{for } |y[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{\sum_{k=-\infty}^{\infty} |h[k]|}$$

Largest value of |x[n]| → X_{max}

$X_{max} \le (1 - a)$ for 1-pole example

conservative in general

To Compute the SQNR

Assume x[n] is a WSS IID random process

For 1-pole example with $-(1-a) \le x[n] \le (1-a)$

$$m_x = 0$$

$$\sigma_x^2 = \frac{\left((1-a) - -(1-a)\right)^2}{12} = \frac{(1-a)^2}{3}$$

Signal and Noise Powers

$$e[n] \rightarrow \sigma_f^2 = \frac{2\,2^{-2B}}{12}\left(\frac{1}{1-a^2}\right)$$

$$x[n] \rightarrow \sigma_y^2 = \frac{(1-a)^2}{3}\left(\frac{1}{1-a^2}\right)$$

SQNRin and SQNRout

$$SQNR_{out} = \frac{\sigma_y^2}{\sigma_f^2} = \frac{\frac{(1-a)^2}{3}}{\frac{2\,2^{-2B}}{12}} = 2(1-a)^2 2^{2B}$$

$$SQNR_{in} = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\frac{(1-a)^2}{3}}{\frac{2^{-2B}}{12}}$$

Quantization noise (σ_e^2)

$$= 4 \cdot 2^{2B}(1-a)^2 = 2\,SQNR_{out}$$

Different Scaling Method

Now let $x[n] = X_{max}\cos = (w_0 n)$

$y[n] = X_{max} H(e^{jw_0})\cos(w_0 n + \phi)$

$|y[n]| \le X_{max}\,|H(e^{jw_0})|$

for $|y[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{|H(e^{jw_0})|}\ \forall w_0$

Different Scaling Method, cont'd

1-pole example:

$$X_{max} \le \sqrt{1 + a^2 - 2a\cos w}\,, \forall w$$

$$\le \sqrt{1 + a^2 + 2a}$$

$$X_{max} \le \sqrt{(1+a)^2} = (1+a) \quad \leftarrow \text{Less conservative}$$

Energy Scaling

$$X_{max} \le \frac{1}{\sqrt{\sum_k |h[k]|^2}}$$

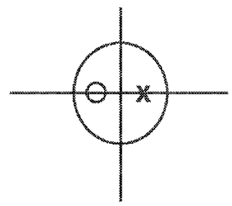
$$\le \frac{1}{\sqrt{(1-a^2)^{-1}}} = \sqrt{1-a^2} = \sqrt{(1+a)(1-a)}$$

$$\max_w |H(e^{jw})| \le \sqrt{\sum |h[n]|^2} \le \sum |h[n]|$$

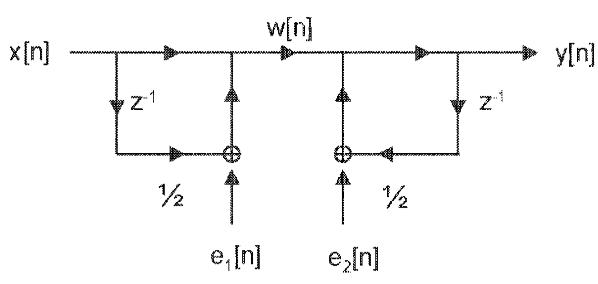
Least Conservative ⟶ Most Conservative

Another Example

$$H(z) = \frac{1+\frac{1}{2}z^{-1}}{1-\frac{1}{2}z^{-1}}$$



DFI



DFI, cont'd

$$He_1(z) = \frac{1}{1-\frac{1}{2}z^{-1}}, \quad He_2(z) = \frac{1}{1-\frac{1}{2}z^{-1}}$$

$$(X_m = 1) \quad \sigma_f^2 = \frac{2\ 2^{-2B}}{12} \frac{1}{1-\left(\frac{1}{2}\right)^2} = \frac{2}{9} 2^{-2B}$$

DFI, cont'd

Now scale x[n] so |w[n]| ≤ 1 & |y[n]| ≤ 1

$$|w[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{1+\frac{1}{2}} = 2/3$$

$$|y[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{\left(\sum |h[n]|\right)} \le 1/3$$

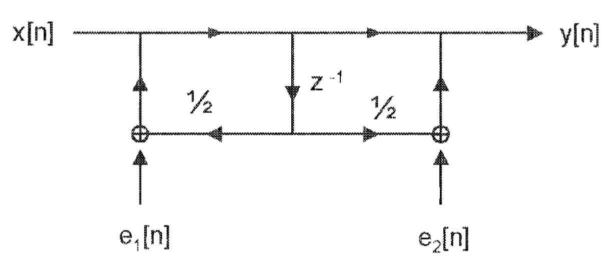
DFI, cont'd

$$\Rightarrow \sigma_x^2 = \frac{\left(2 \cdot \frac{1}{3}\right)^2}{12} = 1/27$$

$$\Rightarrow \sigma_y^2 = \left(\tfrac{1}{27}\right) \sum_{n=-\infty}^{\infty} \left|h[n]\right|^2 = \left(\tfrac{7}{3}\right)\left(\tfrac{1}{27}\right)$$

$$\text{SNR}_{out} = \frac{\sigma_y^2}{\sigma_f^2} = \frac{7}{18} 2^{2B}$$

DFII



DFII, cont'd

$X_{max} \le 1/3$ (why?)

$$\Rightarrow \sigma_y^2 = \left(\frac{1}{27}\right)\left(\frac{7}{3}\right)$$

$$\sigma_f^2 = \left(\frac{7}{3}\right)\frac{2^{-2B}}{12} + (1)\frac{2^{-2B}}{12}$$

$\sum_n (h_1[n])^2$; ?

DFII, cont'd

$$= \frac{5}{18} 2^{-2B}$$

$$\Rightarrow SNR_{out} = \frac{14}{45} 2^{2B} \text{ vs. } \frac{7}{18} 2^{2B}$$

$$\text{DFI} \quad \frac{35}{90} 2^{2B} > \frac{28}{90} 2^{2B} \quad \text{DFII}$$

Q: parallel form ? …

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 9

Professor Andrew Singer
Department of Electrical and Computer Engineering

What About Overflow?

Let $x_m = 1$ Require $|y[n]| \le 1$

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

$$|y[n]| = \left|\sum_{k=-\infty}^{\infty} h[k]x[n-k]\right| \le \sum_{k=-\infty}^{\infty} |h[k]||x[n-k]|$$

$$\le X_{max} \sum_{k=-\infty}^{\infty} |h[k]|$$

max $|x[n]|$
not x_m

What About Overflow?, cont'd

for $|y[n]| \le 1 \;\Rightarrow\; X_{max} \le \dfrac{1}{\sum_{k=-\infty}^{\infty} |h[k]|}$

Largest value of |x[n]|

$X_{max} \le (1 - a)$ for 1-pole example

<u>conservative</u> in general

To Compute the SQNR

Assume x[n] is a WSS IID random process

For 1-pole example with $-(1-a) \le x[n] \le (1-a)$

$$m_x = 0$$

$$\sigma_x^2 = \frac{\left((1-a) - -(1-a)\right)^2}{12} = \frac{(1-a)^2}{3}$$

Signal and Noise Powers

$$e[n] \rightarrow \sigma_f^2 = \frac{2\,2^{-2B}}{12}\left(\frac{1}{1-a^2}\right)$$

$$x[n] \rightarrow \sigma_y^2 = \frac{(1-a)^2}{3}\left(\frac{1}{1-a^2}\right)$$

SQNRin and SQNRout

$$SQNR_{out} = \frac{\sigma_y^2}{\sigma_f^2} = \frac{\frac{(1-a)^2}{3}}{\frac{2\,2^{-2B}}{12}} = 2(1-a)^2 2^{2B}$$

$$SQNR_{in} = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\frac{(1-a)^2}{3}}{\frac{2^{-2B}}{12}}$$

Quantization noise → σ_e^2

$$= 4 \cdot 2^{2B}(1-a)^2 = 2\,SQNR_{out}$$

Different Scaling Method

Now let $x[n] = X_{max} \cos = (w_0 n)$

$$y[n] = X_{max} H(e^{jw_0}) \cos(w_0 n + \phi)$$

$$|y[n]| \leq X_{max} |H(e^{jw_0})|$$

for $|y[n]| \leq 1 \Rightarrow X_{max} \leq \dfrac{1}{|H(e^{jw_0})|} \; \forall w_0$

Different Scaling Method, cont'd

1-pole example:

$$X_{max} \leq \sqrt{1 + a^2 - 2a\cos w}, \forall w$$

$$\leq \sqrt{1 + a^2 + 2a}$$

$$X_{max} \leq \sqrt{(1+a)^2} = (1+a) \quad \leftarrow \text{Less conservative}$$

Energy Scaling

$$X_{max} \le \frac{1}{\sqrt{\sum_k |h[k]|^2}}$$

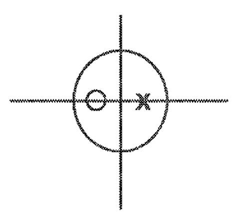
$$\le \frac{1}{\sqrt{(1-a^2)^{-1}}} = \sqrt{1-a^2} = \sqrt{(1+a)(1-a)}$$

$$\max_w |H(e^{jw})| \le \sqrt{\sum |h[n]|^2} \le \sum |h[n]|$$

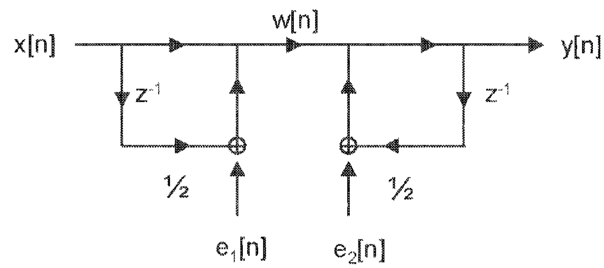
Least Conservative ⟶ Most Conservative

Another Example

$$H(z) = \frac{1+\frac{1}{2}z^{-1}}{1-\frac{1}{2}z^{-1}}$$



DFI



DFI, cont'd

$$He_1(z) = \frac{1}{1-\frac{1}{2}z^{-1}}, \quad He_2(z) = \frac{1}{1-\frac{1}{2}z^{-1}}$$

$$(X_m = 1) \quad \sigma_f^2 = \frac{2\,2^{-2B}}{12} \quad \frac{1}{1-\left(\frac{1}{2}\right)^2} = \frac{2}{9}2^{-2B}$$

DFI, cont'd

Now scale x[n] so |w[n]| ≤ 1 & |y[n]| ≤ 1

$$|w[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{1+\frac{1}{2}} = 2/3$$

$$|y[n]| \le 1 \Rightarrow X_{max} \le \frac{1}{\left(\sum |h[n]|\right)} \le 1/3$$

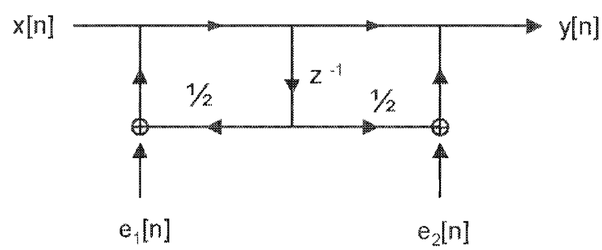
DFI, cont'd

$$\Rightarrow \sigma_x^2 = \frac{\left(2 \cdot \frac{1}{3}\right)^2}{12} = 1/27$$

$$\Rightarrow \sigma_y^2 = \left(\tfrac{1}{27}\right) \sum_{n=-\infty}^{\infty} |h[n]|^2 = \left(\tfrac{7}{3}\right)\left(\tfrac{1}{27}\right)$$

$$SNR_{out} = \frac{\sigma_y^2}{\sigma_f^2} = \frac{7}{18} 2^{2B}$$

DFII



DFII, cont'd

$X_{max} \leq 1/3$ (why?)

$$\Rightarrow \sigma_y^2 = \left(\frac{1}{27}\right)\left(\frac{7}{3}\right)$$

$$\sigma_f^2 = \left(\frac{7}{3}\right)\frac{2^{-2B}}{12} + (1)\frac{2^{-2B}}{12}$$

$\left(\frac{7}{3}\right)$ ↖ $\sum_n (h_1[n])^2$

(1) ↑ ?

DFII, cont'd

$$= \frac{5}{18} 2^{-2B}$$

$$\Rightarrow SNR_{out} = \frac{14}{45} 2^{2B} \text{ vs. } \frac{7}{18} 2^{2B}$$

$$\text{DFI} \quad \frac{35}{90} 2^{2B} > \frac{28}{90} 2^{2B} \quad \text{DFII}$$

Q: parallel form ? ...

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 10

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Practical Digital Communication

- Practical communication channels do not permit baseband communications

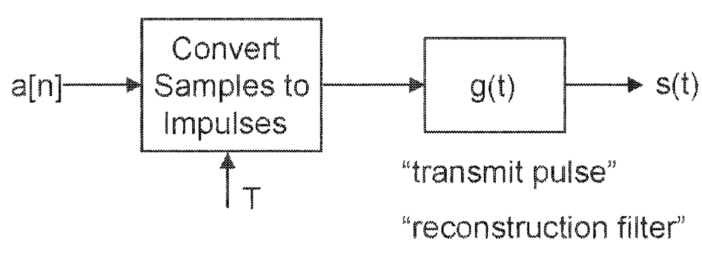
  For example:
  - telephone channel 300 – 3300 Hz
  - ADSL: 25 KHz-1.1 MHz
  - VDSL: 1.6 MHz – 10 MHz
  - ISM Band (wireless) 902 – 928 M Hz
  - Underwater Acoustic 1.5 – 3.5 k Hz

Passband Communications

- Practical digital communications typically modulate a baseband communication into the transmission band.
- Baseband "Pulse Amplitude Modulation" PAM for digital communication

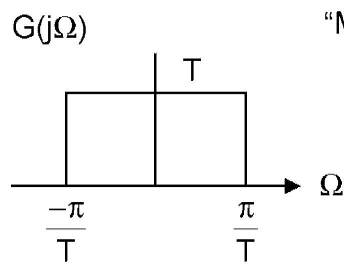
$$s(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT)$$

Pulse Amplitude Modulation (PAM)



Transmit Pulse Shaping

If g(t) is an ideal low pass filter:



"Minimum Bandwidth Pulse"

Then this is just an ideal C/D converter!

Symbol Transmission Rate

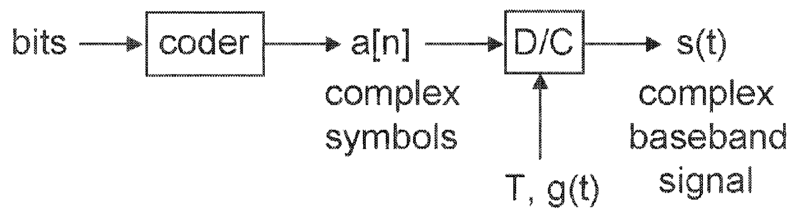
VDSL Band ~ 10 MHz:

$$\frac{2\pi}{T} = 2\pi \cdot 10\text{MHz} = 2\pi fs$$

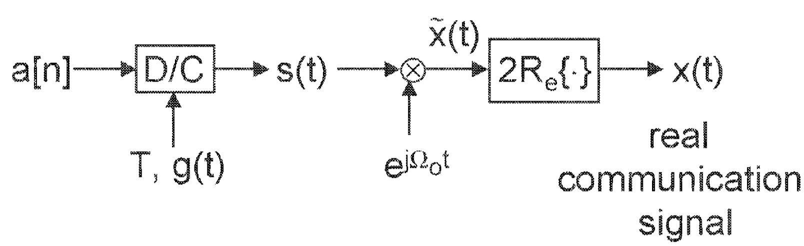
$fs = 10\text{M}$ samples/sec
"symbols/sec"

Underwater Modem ~ 2 kHz: 2 Ksymbols/sec

Baseband PAM



Passband PAM



Three interpretations of PAM

1. $x(t) = R_e\left\{e^{j\Omega_0 t}\sum_{m=-\infty}^{\infty} a[m]g(t-mT)\right\}$

$$x(t) = R_e\left\{(S_R(t) + jS_I(t))(\cos\Omega_0 t + j\sin\Omega_0 t)\right\}$$

$$= S_R(t)\cos\Omega_0 t - S_I(t)\sin\Omega_0 t$$

Real part of complex passband PAM

Three interpretations of PAM

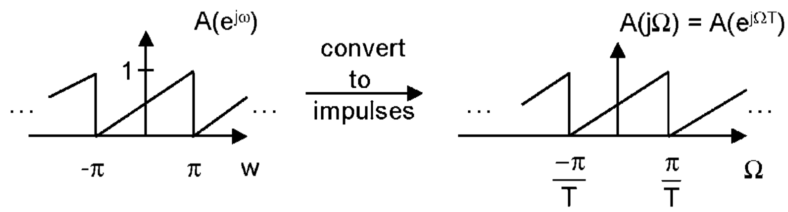
2. $= \cos\Omega_0 t \sum_{m=-\infty}^{\infty} a_R[m]g(t-mT) -$

$$\sin\Omega_0 t \sum_{m=-\infty}^{\infty} a_I[m]g(t-mT)$$

In phase and quadrature modulation

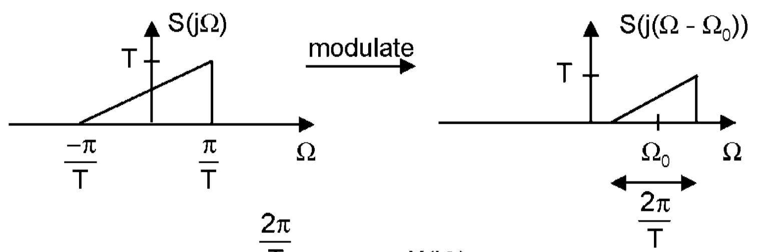
3. $= \underbrace{|a[m]|}_{AM}\cos(\Omega_0 t + \underbrace{\measuredangle a[m]}_{PM})g(t-mT)$ $\Rightarrow$ psk if $|a[n]| = 1$

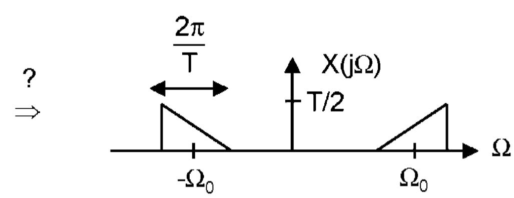
PAM in the Frequency Domain



Then apply pulse shaping filter…

Making things real

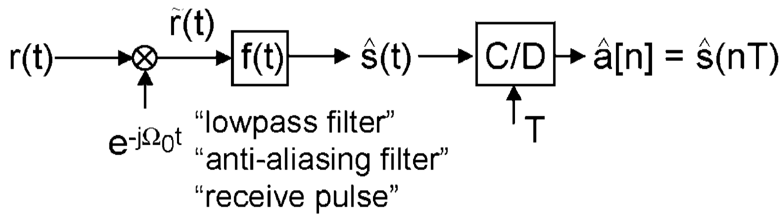




$$\frac{1}{2}\left(S(j\Omega)+S^{*}(-j\Omega)\right)=S_e(j\Omega)\leftrightarrow R_e\{S(t)\}$$

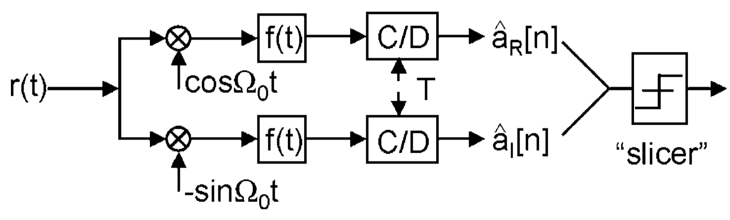
Passband PAM Receivers

For now, assume no channel, r(t) = x(t)

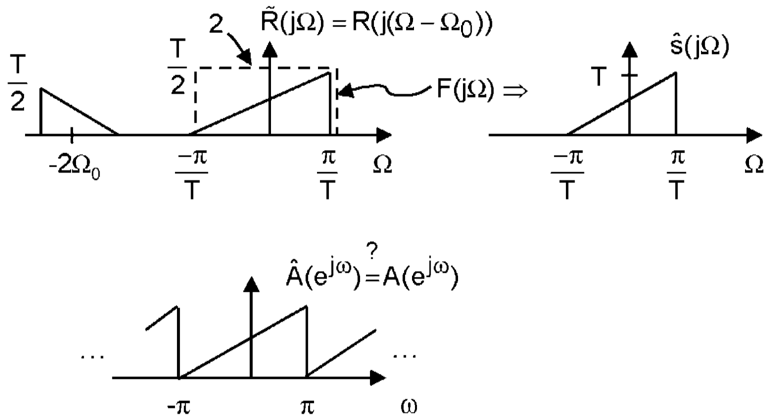


$$\tilde{r}(t) = r(t)\left(\cos(\Omega_0 t) - j\sin(\Omega_0 t)\right)$$

In-phase and quadrature (I&Q)



Passband PAM Receivers, cont'd



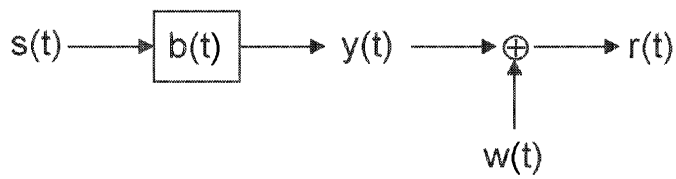
Realistic Models:

- Channel not ideal
- g(t), f(t), not ideal lowpass filters,"MBW" pulses
- C/D & D/C not ideal → quantization, filtering

Bandlimited channel model

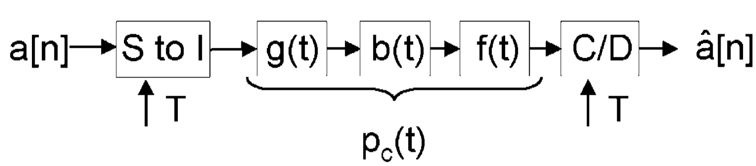
Channel Model:

- Dispersive (bandlimited) channel
- Additive noise

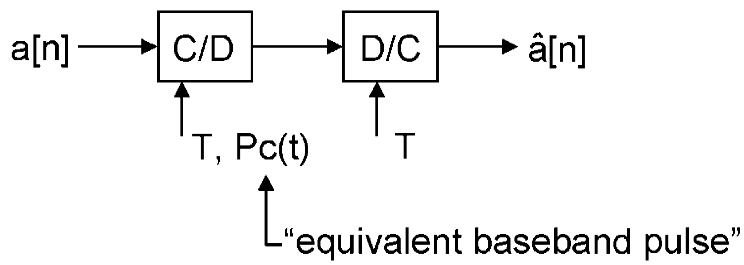


Equivalent Baseband Channel Model

$$\hat{S}(j\Omega) = \underbrace{\left[G(j\Omega)B(j(\Omega+\Omega_0))F(j\Omega)\right]}_{P_c(j\Omega)\ \leftrightarrow\ p_c(t)} A(j\Omega)$$



Equivalent Baseband Pulse



$\rightarrow$ want $\hat{a}[n] = a[n]$

Remember Sampling?

What about aliasing?

$$\hat{A}(e^{j\omega}) = \left(\frac{1}{T} \sum_{k=-\infty}^{\infty} Pc\left(j\Omega - \frac{2\pi k}{T} \right) A\left(j\Omega - \frac{2\pi k}{T} \right) \right) \Bigg|_{\Omega = \frac{\omega}{T}}$$

$$= A(e^{j\omega}) \underbrace{\frac{1}{T} \sum_{k=-\infty}^{\infty} Pc\left(j\frac{\omega}{T} - \frac{2\pi k}{T} \right)}_{\text{must be a constant!}}$$

periodic!

must be a constant!

Zero ISI, Nyquist Condition

⇒ "Nyquist criterion" or "zero forcing" condition

$$\hat{a}[n] = \sum_{m=-\infty}^{\infty} a[m]p[n-m] \qquad p[n] = p_c(nT)$$

$$\hat{a}[n] = a[n]p_c(0) + \sum_{\substack{k=-\infty \\ k\neq n}}^{\infty} a[k]p_c(nT-kT)$$

desired symbol — Intersymbol Interference

Zero ISI

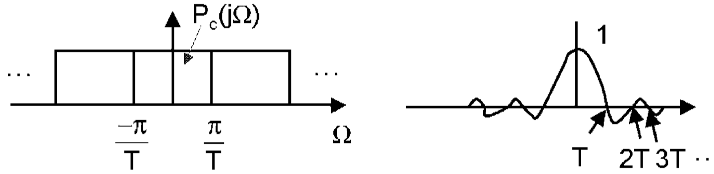
Want no ISI → $p(kT) = \delta[k]$ (1)

Design $p_c(t)$ for no ISI:

$$\text{FT of (1)} \Rightarrow \frac{1}{T}\sum_{k=-\infty}^{\infty} P\left(j\Omega - \frac{2\pi k}{T}\right) = 1$$

Minimum BW Pulse (ideal LPF)

$$P_c(j\Omega) = \begin{cases} 1, & |\Omega| < \pi/T \\ 0, & \text{else} \end{cases} \quad \longleftrightarrow \quad P_c(t) = \frac{\sin(\pi t/T)}{(\pi t/T)}$$

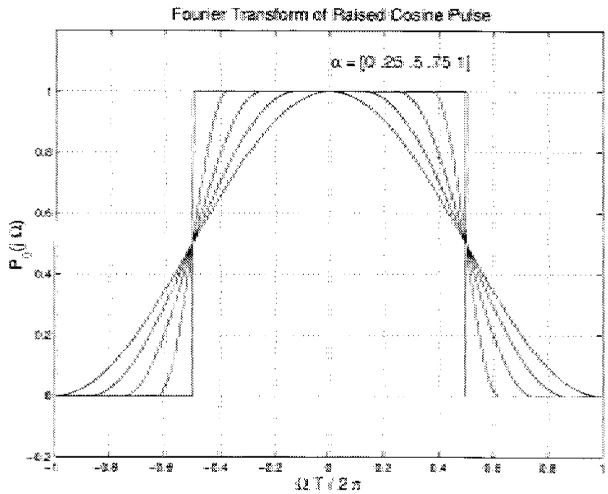


Raised Cosine Pulse

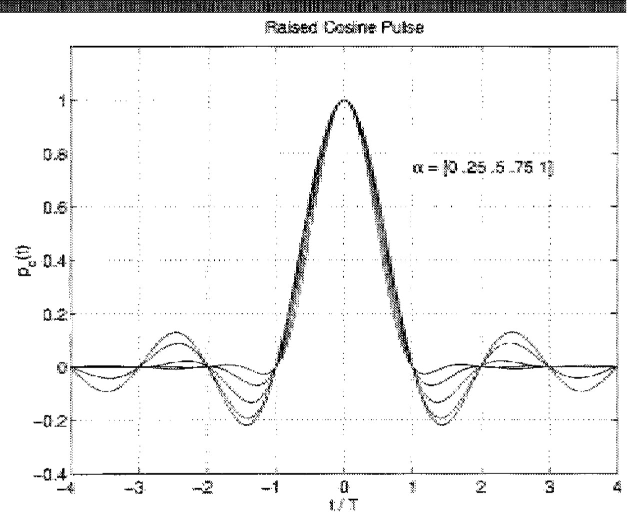
$$P_c(j\Omega) = \begin{cases} T, & 0 \le |\Omega| \le (1-\alpha)\pi/T \\ \frac{T}{2}\left[1 - \sin\left(\frac{T}{2\alpha}[|\Omega| - \pi/T]\right)\right], & (1-\alpha)\frac{\pi}{T} \le |\Omega| \le (1+\alpha)\frac{\pi}{T} \\ 0, & |\Omega| > (1+\alpha)\pi/T \end{cases}$$

$$p_c(t) = \left[\frac{\sin(\pi t/T)}{(\pi t/T)}\right]\left[\frac{\cos(\alpha\pi t/T)}{1-(2\alpha t/T)^2}\right]$$

Raised Cosine Pulse, $P_c(j\Omega)$



Raised Cosine Pulse, $p_c(t)$



Channel Equalization

$$\hat{S}(j\Omega) = \underbrace{(G(j\Omega)B(j(\Omega+\Omega_0))F(j\Omega))}_{P_c(j\Omega)}A(j\Omega)$$

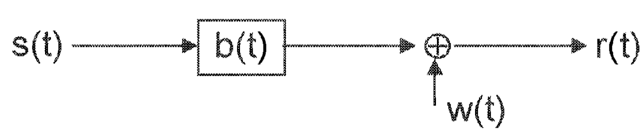
→ design $P_c(j\Omega)$ first, then choose:

$$F(j\Omega) = \begin{cases} \dfrac{P_c(j\Omega)}{G(j\Omega)B(j(\Omega+\Omega_0))}, & G(j\Omega)B(j(\Omega+\Omega_0)) \neq 0 \\ 0, & G(j\Omega)B(j(\Omega+\Omega_0)) = 0 \end{cases}$$

→ For $P_c(j\Omega)=1$: "Zero Forcing Equalizer"
noise enhancement

Additive Noise in the Channel

Now, add noise back in:

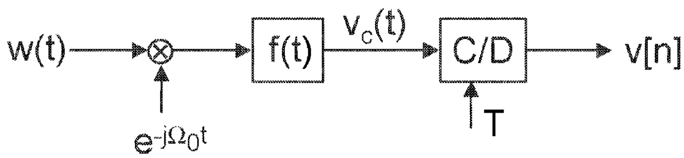


w(t) white, wss, $S_{ww}(j\Omega) = N_o$

$R_{ww}(\tau) = N_o\,\delta(\tau)$

$= E(w(t)w(t+\tau))$

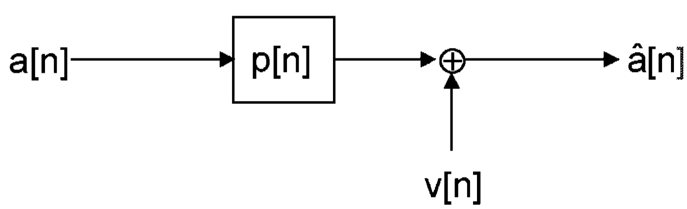
Noise Statistical Properties



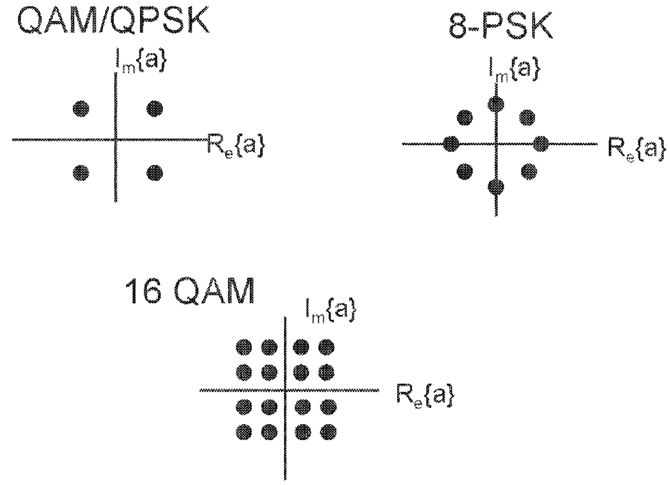
$$s_{v_c v_c}(j\Omega) = N_o \left|F(j\Omega)\right|^2$$

$$\therefore s_{vv}(e^{j\omega}) = \frac{N_o}{T} \sum_{k=-\infty}^{\infty} \left| F\left(j\frac{\omega}{T} - j\frac{2\pi k}{T} \right) \right|^2$$

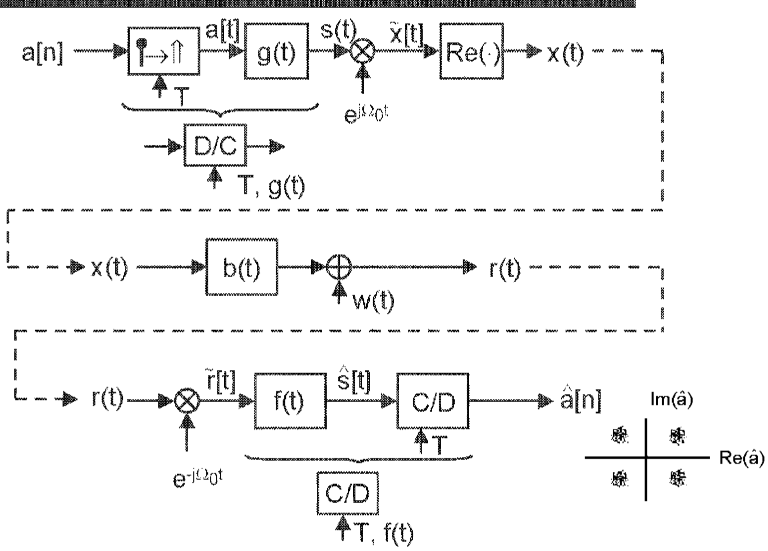
Equivalent DT Baseband Model

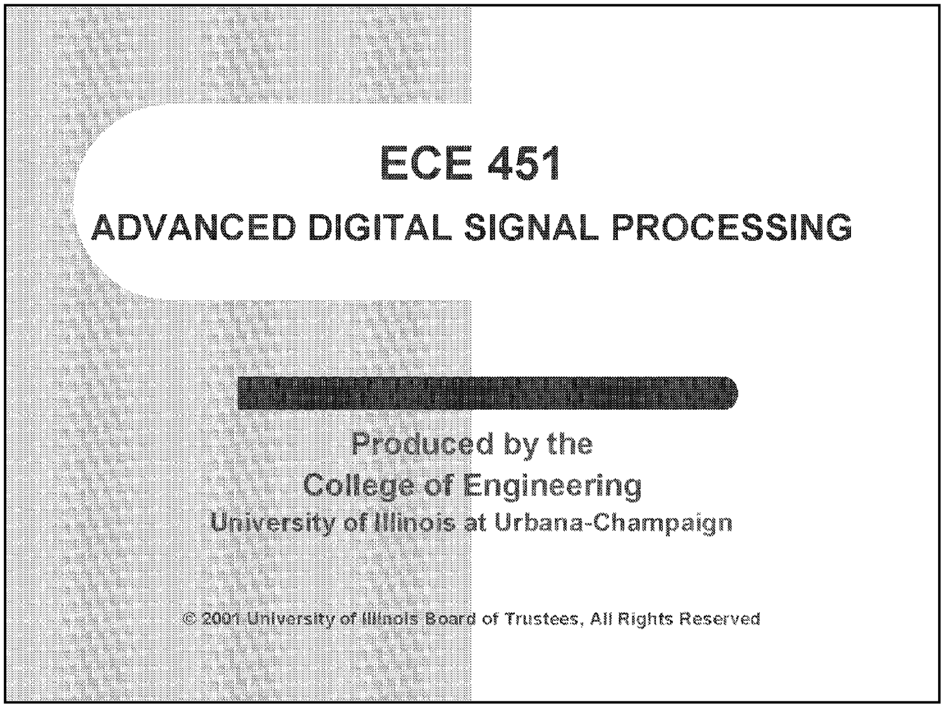


Signal Constellations a[n]



Big Picture




ECE 451
ADVANCED DIGITAL SIGNAL PROCESSING
Produced by the
College of Engineering
University of Illinois at Urbana-Champaign
© 2001 University of Illinois Board of Trustees, All Rights Reserved

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

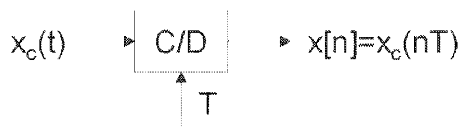
Lecture 11

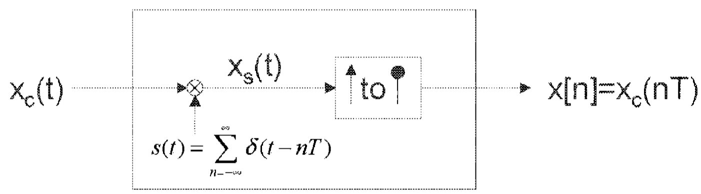
Professor Andrew Singer
Department of Electrical and Computer Engineering

Over-sampling A/D Converters

- We discussed a model for quantization noise and used it to model/analyze digital filter structures.
- Trade-offs between quantization noise and word-length or computational complexity can be performed using these models.
- We can also trade sampling rate with quantization noise, by using over-sampling to decrease quantization noise, or decrease bit-length (word-length) at a fixed SQNR.

Ideal C/D Converter



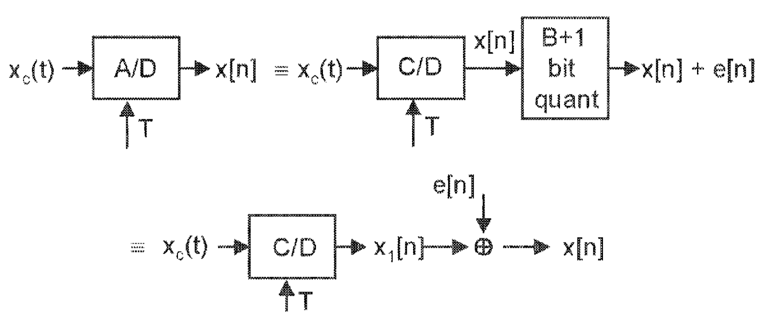


$$X_s(j\Omega)=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_c\left(j\left(\Omega-\frac{2\pi k}{T}\right)\right)$$

$$X(e^{j\omega})=X_s(j\Omega)\big|_{\Omega=\frac{\omega}{T}}=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{T}-\frac{2\pi k}{T}\right)\right)$$

Quantization Model

Our quantization model:



e[n] white, stationary, uniformly distributed, uncorrelated with x[n]

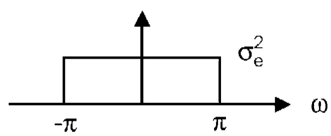
Random Processes, Noise Model

$$E\{e[n]e[n+m]\} = R_{ee}[m] = \sigma_e^2 \delta[m]$$

$$\sigma_e^2 = \frac{2^{-2B}}{12} X_m^2$$

$$S_{ee}(e^{j\omega}) = \sigma_e^2, \quad |\omega| < \pi$$

$$R_{ee}[0] = E\{e[n]^2\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{ee}(e^{j\omega}) d\omega$$



Sampling Random Processes

Will also assume $x_c(t)$ is a WSS random process with autocorrelation and power spectrum:

$R_{x_c x_c}(\tau)$, $S_{x_c x_c}(j\Omega) = F\,(R_{x_c x_c}(\tau))$

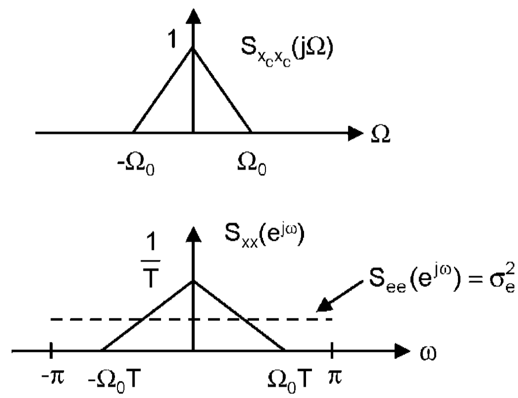
$R_{x_c x_c}(\tau) = E\{x_c(t)x_c(t + \tau)\}$

$R_{xx}[m] = E\{x[n]x[n + m]\} = E\{x_c(nT)x_c(nT + mT)\}$

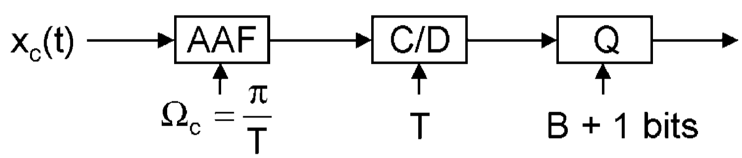
$= R_{x_c x_c}(mT)$

Effects of Sampling on Spectrum

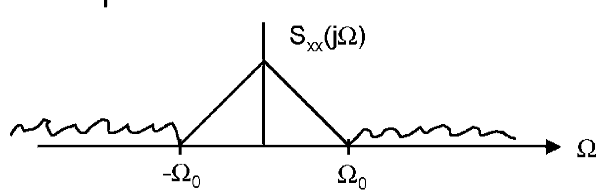
$$R_{xx}[0] = R_{x_c x_c}(0), \quad E(x_c^2(t)) = E(x^2[n])$$



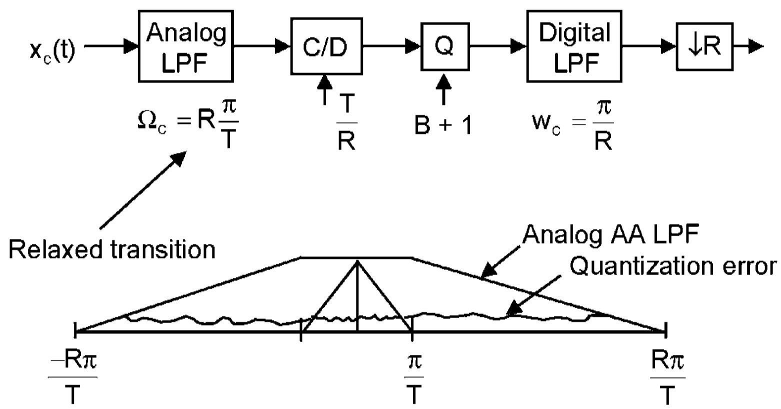
Nyquist Rate Converter



Need $\Omega_c = \frac{\pi}{T}$, sharp cutoff



R-times Oversampling Converter



Oversampling A/D, $T' = T_{nyq}/R$

$$R = \text{oversampling ratio:}\quad \frac{f_s}{f_{nyquist}} = \frac{(1/T')}{(\Omega_0/\pi)}$$

$$\text{or,}\quad \Omega_0 T_{nyq} = \frac{\pi}{R} \qquad\qquad R = \frac{\pi}{\Omega_0 T_{nyq}}$$

Quantization noise power in band =

$$\frac{1}{2\pi}\int_{-\Omega_0 T_{nyq}}^{\Omega_0 T_{nyq}} \sigma_e^2 d\omega = \frac{2\Omega_0 T_{nyq}\sigma_e^2}{2\pi} = \frac{\sigma_e^2}{R}$$

Effective Number of Bits (ENOB)

Signal power in band =

$$P_x = P_{xc} = E(x[n]^2) = E(x_c(t)^2)$$

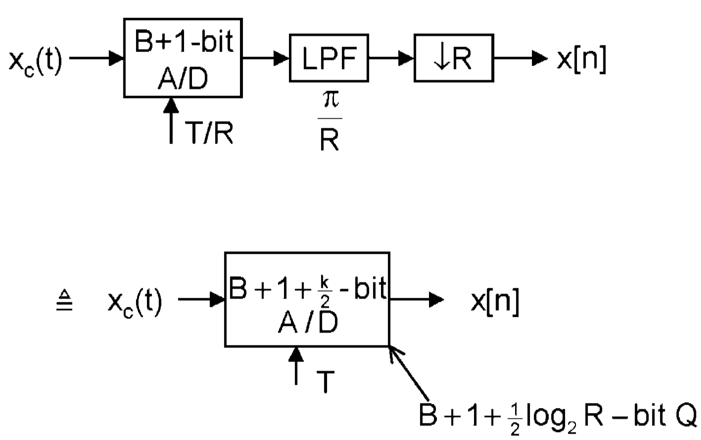
$$\text{SQNR} = \frac{P_x R}{\sigma_e^2} = R\left(\frac{P_{xc}}{x_m^2}\right) 12 \cdot 2^{2B}$$

$$\text{write } R = 2^k \Rightarrow \left(\frac{P_{xc}}{x_m^2}\right) 12 \cdot 2^{2(B+\frac{1}{2}k)}$$

$$\boxed{\Delta B = \tfrac{1}{2}k = \tfrac{1}{2}\log_2 R}$$

Effective # of bits
"ENOB"

Oversampling A/D Converter

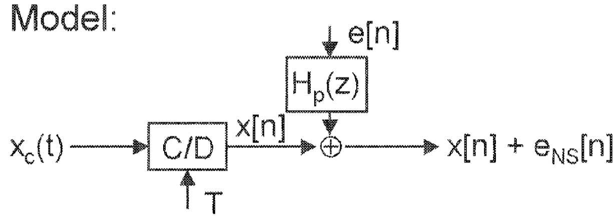


Is there a better way?

- This assumes no quantization noise is added by the LPF.
- Reasonable if FIR, implemented with 2B+1 bits until after the accumulation.
- Need factor of 4x oversampling to gain 1 bit!
- Can get a better tradeoff through "Noise shaping"
- Idea → put more Q noise energy outside band of interest.

Noise Shaping

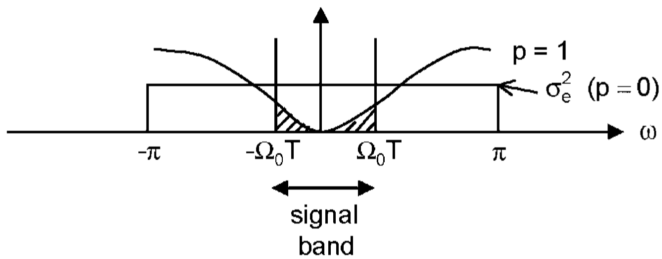
Model:



P^{th}-order noise shaping loop "shapes" the noise spectrum using $H_p(z)$.

Shaping Filter $H_p(z)$

$$\left|H_p(e^{j\omega})\right|^2 = [2\sin(\omega/2)]^{2P}$$



Noise Power

In band Q-noise power =

$$\frac{1}{2\pi}\int_{-\Omega_0 T}^{\Omega_0 T} \sigma_e^2\left(2\sin\left(\tfrac{\omega}{2}\right)\right)^{2P} d\omega$$

For R large, $\sin(\omega/2) \approx \omega/2$

$$\Rightarrow \frac{1}{2\pi}\int_{-\Omega_0 T}^{\Omega_0 T} \sigma_e^2(\omega)^{2P} d\omega = \frac{1}{2\pi}\int_{-\pi/R}^{\pi/R} \sigma_e^2\omega^{2P} d\omega =$$

$$= \frac{\pi^{2P}\sigma_e^2}{(2p+1)R^{2p+1}}$$

ENOB for p-th-order Noise Shaping

$$SQNR = \frac{12 \cdot 2^{2b}}{(\pi)^{2p}}\left(\frac{P_{xc}}{X_m^2}\right)(2p+1)R^{(2p+1)}$$

$$R = 2^k$$

$$\Rightarrow \frac{12 \cdot P_{xc}}{\pi^{2p} X_m^2}(2p+1)2^{2\left(B+(p+\frac{1}{2})k\right)}$$

$$\triangle B = \left(p+\tfrac{1}{2}\right)k = \left(p+\tfrac{1}{2}\right)\log_2 R$$

Why is a 1-bit A/D better than a...

CD systems use R = 256, B+1 = 1,

1 – bit A/D converter

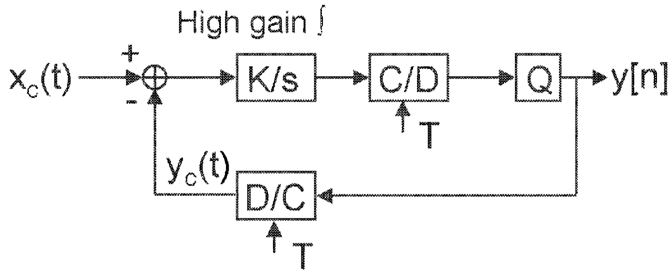
8p + 4 extra bits or 8p + 5 equivalent bits

p = 2 yields 21 effective bits!

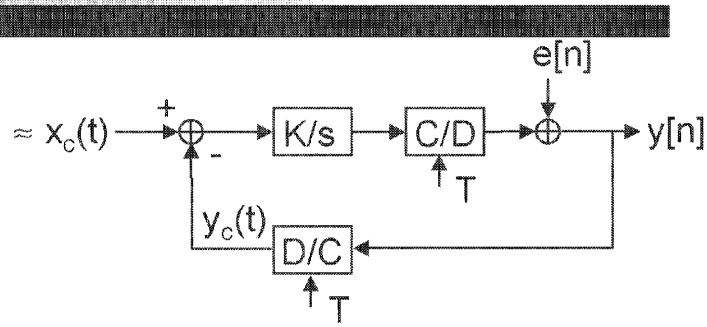
Implementations

Σ - Δ (Δ - Σ) A/D converters

Analog Σ - Δ converter (p = 1)



CT Model



$$H_x(s) = \frac{Y_c(s)}{X_c(s)} = \frac{K/s}{1+\frac{K}{s}} = \frac{K}{S+K} \underset{k\to\infty}{\approx} 1$$

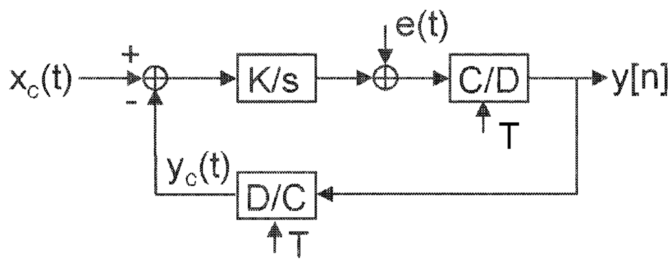
$$|x_c(j\Omega)| = 0, |\Omega| > \Omega_0$$

CT Model, cont'd

$k >> 1$, $x_c(t)$ B.L., $y[n] = y_x[n] + y_e[n]$

$y_x[n] \approx x_c(nT)$

Redraw: define $e(nT) = e[n]$



Contribution from the Noise

$$S_{ee}(j\Omega) = \begin{cases} \sigma_e^2, & |\Omega| < \pi/T \\ 0, & \text{else} \end{cases}$$

$$H_e(s) = \frac{1}{1+\frac{K}{s}} = \frac{s}{s+K}$$

Noise Spectrum

For k >> 1, $x_c(t)$ B.L. : $H_e(s) \approx s / K$

$$S_{y_e y_e}(e^{j\omega}) = \left| \frac{j\Omega}{K} \right|^2 \Bigg|_{\Omega=\frac{\omega}{T}} S_{ee}(e^{j\omega})$$

$$= \frac{\omega^2}{K^2 T^2} S_{ee}(e^{j\omega}), \ |\omega| < \pi$$

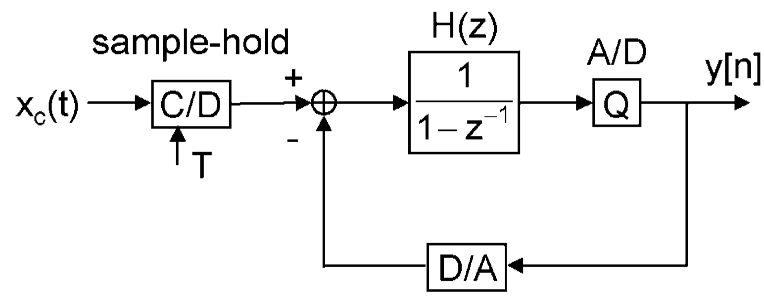
ENOB

$$\frac{1}{2\pi} \int_{-\pi/R}^{\pi/R} \sigma_e^2 \frac{\omega^2}{K^2 T^2} d\omega = \frac{\pi^2 \sigma_e^2}{K^2 T^2 3R^3}$$

$$R^3 = 2^{3\log_2 R} = 2^{2\left(\frac{3}{2}\right)\log_2 R}$$

$$\boxed{\Delta B = \tfrac{3}{2}\log_2 R}$$

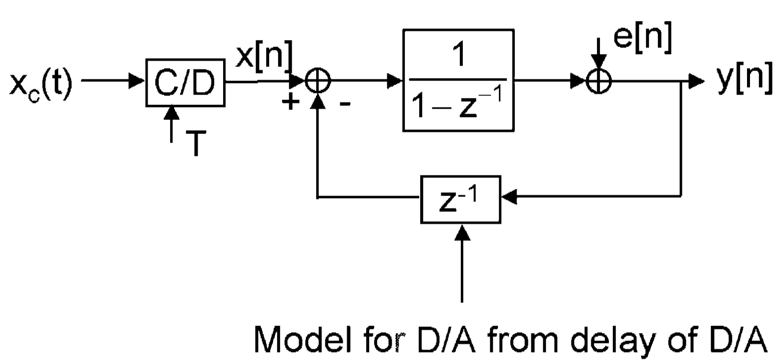
Sampled Data Σ - Δ Converter



H(z) is an <u>accumulator</u>, replacing analog integrator.

Accumulation done with D-T analog samples.

Model for DT Σ - Δ Converter



Signal and Noise System Functions

$$H_x(z) = \frac{\frac{1}{(1-z^{-1})}}{1+\frac{z^{-1}}{(1-z^{-1})}} = 1$$

$$H_e(z) = \frac{1}{1+\frac{z^{-1}}{(1-z^{-1})}} = (1-z^{-1})$$

Signal untouched, Noise "shaped"

$$\left|H_e(e^{j\omega})\right|^2 = \left|1-e^{-j\omega}\right|^2 = \left(2\sin(\tfrac{\omega}{2})\right)^2, \quad p=1$$

$$\boxed{\text{SNR Gain} \cong \frac{3R^3}{\pi^2}} \quad = 9.03k - 5.17 \ \text{(dB)}$$

$$\cong 6.02\left(\left(1+\tfrac{1}{2}\right)k\right) \ \text{(dB)}$$

ENOB

Since SNR α 6.02 B+1 $\rightarrow$ 6.02(B+1 + 3/2 k)

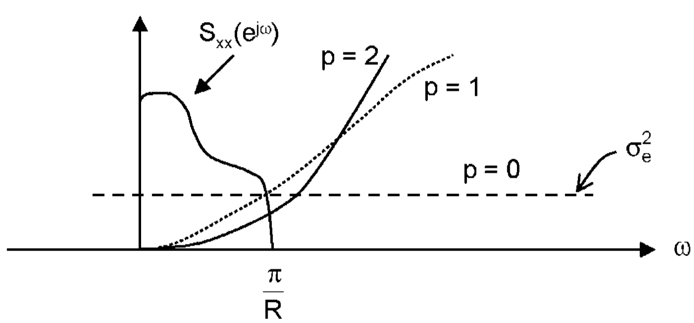
$\uparrow \Delta B$

enob

$$(p = 2) \Rightarrow \text{SNR Gain} \cong \frac{5R^5}{\pi^4} \qquad \text{(homework)}$$

$$\approx \tfrac{5}{2}k = \tfrac{5}{2}\log_2 R \text{ bits}$$

Noise Shaping for p=0,1,2



1 Bit or Δ - Σ OSADC's

- Inose and Yasuda 1976 "delta-sigma modulation"
- noise shaping began in multi-bit form in 1950's
- N.S. in A/D converters 1970's

ECE 451 - Lecture 11 30 of 36

1 Bit or Δ - Σ OSADC's, cont'd

- One-bit A/D's typically have R>>1 (256 typical)
- Easier to make "good" 1-bit internal A/D and D/A elements than multi-bit elements.
- Unlimited potential SNR
- Simpler decimator design
- Analysis & simulation very different

ECE 451 - Lecture 11 31 of 36

1 Bit or $\Delta - \sum$ OSADC's, cont'd

- Additive noise model for quantizer does not hold for N = 1

  highly correlated samples

  - limit cycles produce tones
    - (p = 1 typically uses a dither)
    - (p > 1 typically okay)

1 Bit or $\Delta - \sum$ OSADC's, cont'd

- Very difficult to analyze theoretically

  SNR overestimated by linear models

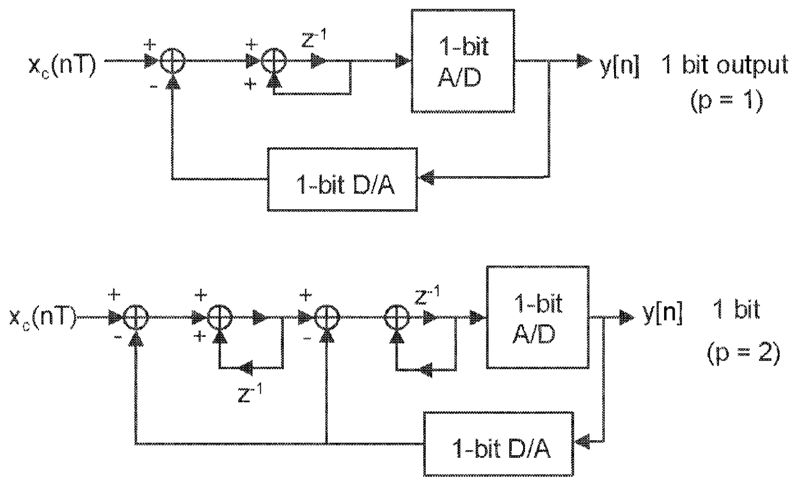
  e.g. p = 1, R = 128 $\rightarrow$ 100 dB vs. 86 dB

  $y[n + 1] = f\{h[n] * [x[n] - y[n]]\}$

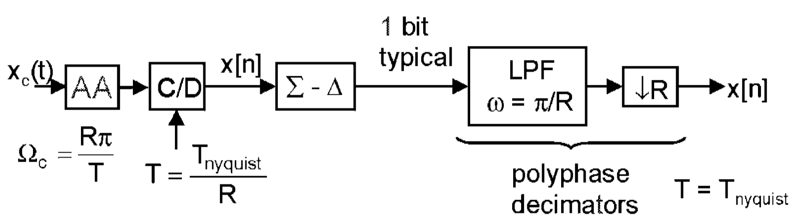
  signum function (arrow pointing to f)   active research topic

  Nonlinear difference equation

1 Bit or Δ - Σ OSADC's, cont'd



1 Bit or Δ - Σ OSADC's, cont'd



$x[n] = x_c(nT_{nyquist})$

LPF typically adds quantization noise

1 Bit or Δ - $\sum$ OSADC's, cont'd

- Note that quantization of x[n] is achieved partly in analog and partly in digital domains.
- Also, anti-aliasing filtering is done partly in analog and partly digital.

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

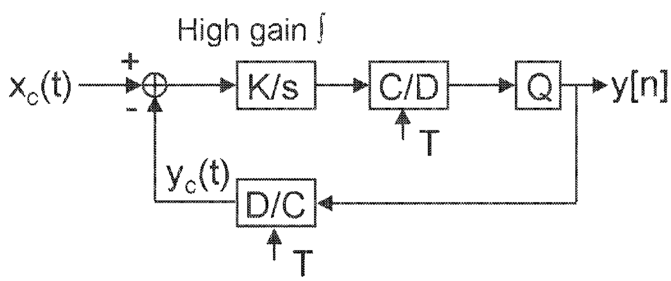
Lecture 12

Professor Andrew Singer
Department of Electrical and
Computer Engineering

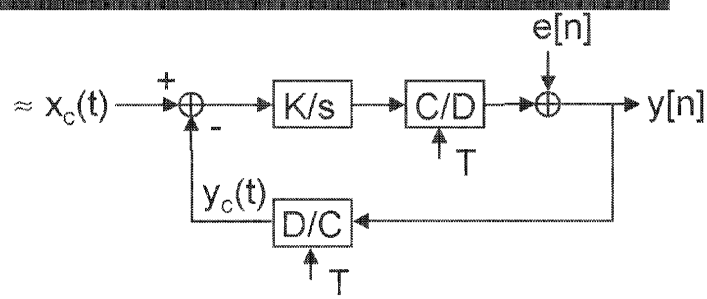
Implementations

Σ - Δ (Δ - Σ) A/D converters

Analog Σ - Δ converter (p = 1)



CT Model



$$H_x(s) = \frac{Y_c(s)}{X_c(s)} = \frac{K/s}{1+\frac{K}{s}} = \frac{K}{S+K} \underset{k\to\infty}{\approx} 1$$

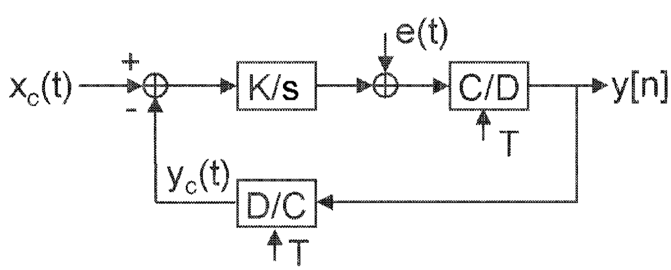
$$|x_c(j\Omega)| = 0, |\Omega| > \Omega_0$$

CT Model, cont'd

$k >> 1$, $x_c(t)$ B.L., $\quad y[n] = y_x[n] + y_e[n]$

$y_x[n] \approx x_c(nT)$

Redraw: define e(nT) = e[n]



Contribution from the Noise

$$S_{ee}(j\Omega) = \begin{cases} \sigma_e^2, & |\Omega| < \pi/T \\ 0, & \text{else} \end{cases}$$

$$H_e(s) = \frac{1}{1+\frac{K}{s}} = \frac{s}{s+K}$$

Noise Spectrum

For k >> 1, $x_c(t)$ B.L. : $H_e(s) \approx s / K$

$$S_{y_e y_e}(e^{j\omega}) = \left|\frac{j\Omega}{K}\right|^2_{\Omega=\frac{\omega}{T}} S_{ee}(e^{j\omega})$$

$$= \frac{\omega^2}{K^2T^2} S_{ee}(e^{j\omega}),\ |\omega| < \pi$$

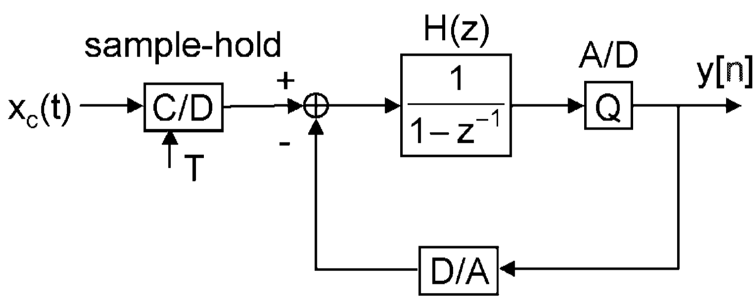
ENOB

$$\frac{1}{2\pi}\int_{-\pi/R}^{\pi/R}\sigma_e^2\frac{\omega^2}{K^2T^2}d\omega=\frac{\pi^2\sigma_e^2}{K^2T^23R^3}$$

$$R^3=2^{3\log_2 R}=2^{2\left(\frac{3}{2}\right)\log_2 R}$$

$$\boxed{\Delta B=\tfrac{3}{2}\log_2 R}$$

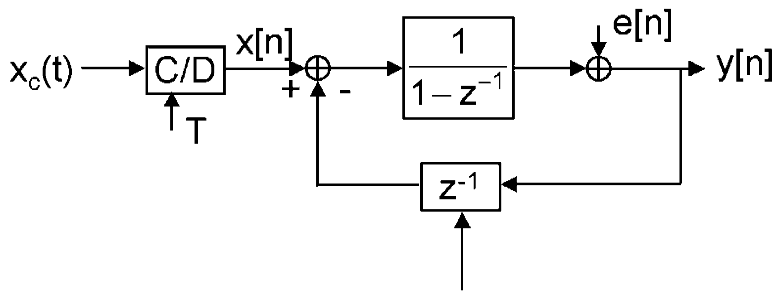
Sampled Data Σ - Δ Converter



H(z) is an accumulator, replacing analog integrator.

Accumulation done with D-T analog samples.

Model for DT Σ - Δ Converter



Model for D/A from delay of D/A

Signal and Noise System Functions

$$H_x(z) = \frac{\dfrac{1}{(1-z^{-1})}}{1+\dfrac{z^{-1}}{(1-z^{-1})}} = 1$$

$$H_e(z) = \frac{1}{1+\dfrac{z^{-1}}{(1-z^{-1})}} = (1-z^{-1})$$

Signal untouched, Noise "shaped"

$$\left|H_e(e^{j\omega})\right|^2 = \left|1 - e^{-j\omega}\right|^2 = \left(2\sin(\tfrac{\omega}{2})\right)^2, \quad p = 1$$

$$\boxed{\text{SNR Gain} \cong \frac{3R^3}{\pi^2}} \quad = 9.03k - 5.17 \ \text{(dB)}$$

$$\cong 6.02\left(\left(1 + \tfrac{1}{2}\right)k\right) \ \text{(dB)}$$

ENOB

Since SNR α 6.02 B+1 $\rightarrow$ 6.02(B+1 + 3/2 k)

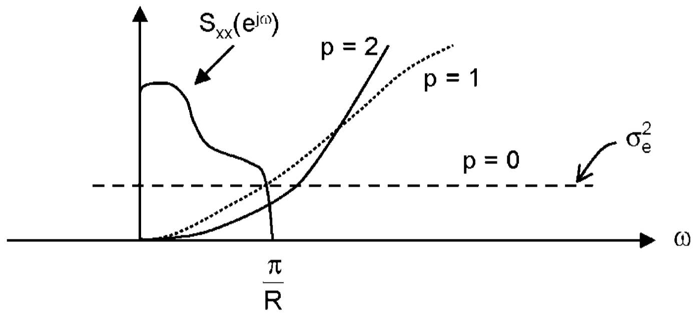
↑ ΔB

enob

$$(p = 2) \Rightarrow \text{SNR Gain} \cong \frac{5R^5}{\pi^4} \qquad \text{(homework)}$$

$$\approx \tfrac{5}{2}k = \tfrac{5}{2}\log_2 R \ \text{bits}$$

Noise Shaping for p=0,1,2



1 Bit or Δ - ∑ OSADC's

- Inose and Yasuda 1976 "delta-sigma modulation"
- noise shaping began in multi-bit form in 1950's
- N.S. in A/D converters 1970's

1 Bit or Δ - Σ OSADC's, cont'd

- One-bit A/D's typically have R>>1 (256 typical)
- Easier to make "good" 1-bit internal A/D and D/A elements than multi-bit elements.
- Unlimited potential SNR
- Simpler decimator design
- Analysis & simulation very different

1 Bit or Δ - Σ OSADC's, cont'd

- Additive noise model for quantizer does not hold for N = 1

  highly correlated samples

  - limit cycles produce tones
    (p = 1 typically uses a dither)
    (p > 1 typically okay)

1 Bit or Δ - ∑ OSADC's, cont'd

- Very difficult to analyze theoretically

  SNR overestimated by linear models

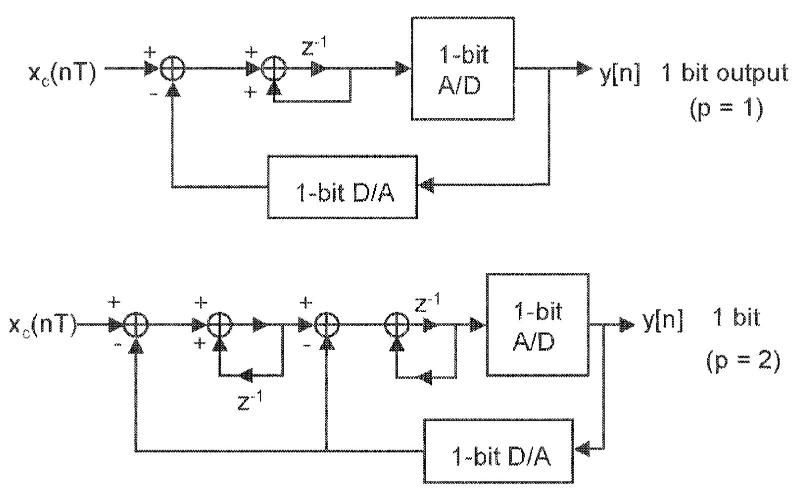
  e.g. $p = 1$, $R = 128 \rightarrow 100$ dB vs. 86 dB

  $y[n + 1] = f\{h[n] * [x[n] - y[n]]\}$

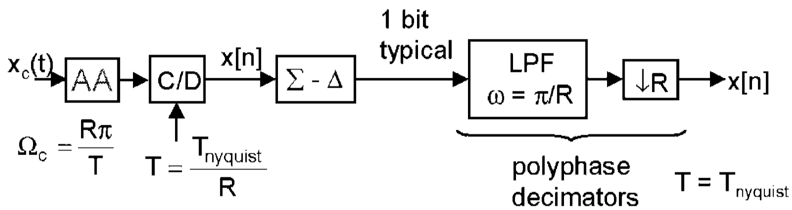
  signum function   active research topic

  Nonlinear difference equation

1 Bit or Δ - ∑ OSADC's, cont'd



1 Bit or Δ - ∑ OSADC's, cont'd



$x[n] = x_c(nT_{nyquist})$

LPF typically adds quantization noise

1 Bit or Δ - ∑ OSADC's, cont'd

- Note that quantization of x[n] is achieved partly in analog and partly in digital domains.
- Also, anti-aliasing filtering is done partly in analog and partly digital.

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 13

Professor Andrew Singer
Department of Electrical and Computer Engineering

Discrete-Time Filter Design Methods

There are many types of digital filters:

- Frequency Selective: HP, LP, BP, BS, etc…
- Estimation/Prediction, neg $\tau(\omega)$, equalizers, LPC
- Adaptive Filters

$$x[n] \rightarrow \boxed{H(z)} \rightarrow y[n]$$

want H(z) realizable $\rightarrow$ described by LCCDE, H(z) rational

IIR DT Filter Design Methods

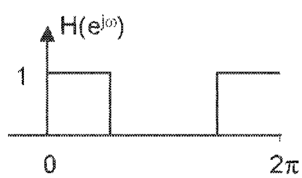
$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$$

Rational H(z) can be FIR or IIR. Today IIR

- Powerful mathematical analog design tools
- Less computation for given specifications
- Less phase control
- Stability

Ideal Frequency Selective

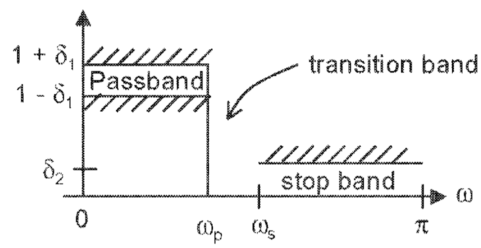
Is this a LP, BP, BS, HP, or AP filter?



Typically focus on $|H(e^{j\omega})| \cong |H_{id}(e^{j\omega})|$

Typical magnitude specification

Low-pass filter magnitude specification

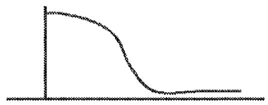


IIR Filters From Analog Design Methods

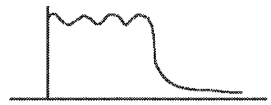
- Four-step process:

1. Specification of desired discrete-time filter
2. Map specs to continuous time
3. Design analog filter to meet CT specs using existing analog design mathematical tools
4. Transform analog filter to DT filter

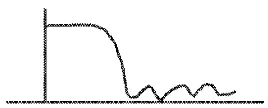
Prototype Analog Filters



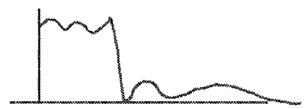
Butterworth
"maximally flat"



Chebychev I



Chebychev II



Elliptic

"0(ellip) < 0(Cheby) < 0(Butter)"

Desirable Mapping Properties

1) Stable analog $\rightarrow$ stable digital

   (LHP $\rightarrow$ inside u.c.)

2) $j\Omega$ axis $\rightarrow$ unit circle

$$H(s)\big|_{s=j\Omega} \rightarrow H(z)\big|_{z=e^{j\omega}}$$

A Few Possible Mapping Methods

- Impulse invariance: $h[n] = h_c(nT)$
- Map d/dt → differences in LCCDE
- Frequency warping by bilinear transformation

First consider Impulse invariance:

$$h[n] = h_c(nT)$$

ECE 451 - Lecture 13

8 of 19

Impulse Invariance

Write

$$H_c(s) = \sum_{k=1}^{p} \frac{A_k}{s - s_k} \rightarrow h_c(t) = \sum_{k=1}^{p} A_k e^{-s_k t} u(t)$$

$$\rightarrow h[n] = \sum_{k=1}^{p} A_k e^{s_k T n} u[n]$$

$\rightarrow$ pole @ $s = s_k$ $\rightarrow$ pole @ $z = e^{s_k T}$

$\rightarrow$ zeros?

ECE 451 - Lecture 13

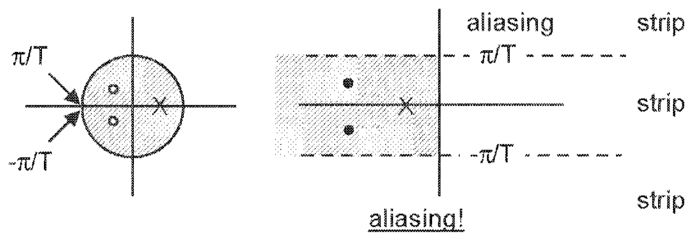
9 of 19

Impulse Invariance

$z = e^{sT} \rightarrow e^{j\omega} = e^{j\Omega T}, \quad \omega = \Omega T \mod 2\pi$

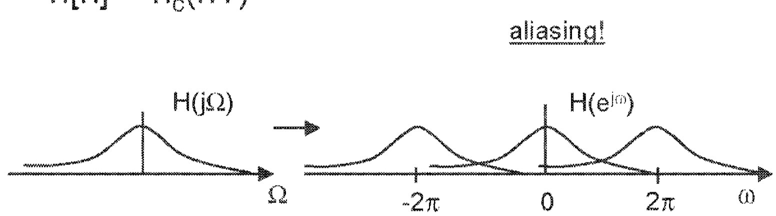
$z = e^{(\sigma + j\Omega)T} \Rightarrow$ LHP $\rightarrow$ inside u.c.

RHP $\rightarrow$ outside u.c.



Impulse Invariance = Sampling

$h[n] = h_c(nT)$



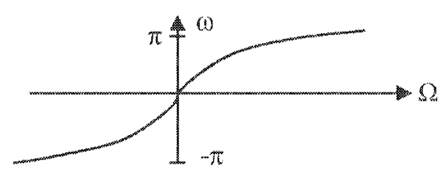
Bonus: what is $u(nT)$?? $\neq u[n]$!

Other Mappings

2) $\frac{d}{dT} \rightarrow$ differences: (homework)

$$s = j\Omega = \frac{1}{T}(1 - z^{-1}) \text{ or } = \frac{1}{T}(z^{-1} - 1)$$

3) BLT want entire $j\Omega$ axis $\rightarrow e^{j\omega}$. How?



Bilinear Transformation (BLT)

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right), \quad z = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s}$$

$$j\Omega = \frac{2}{T}\frac{e^{j\omega/2} - e^{-j\omega/2}}{e^{j\omega/2} + e^{-j\omega/2}} = j\frac{2}{T}\tan\left(\omega/2\right)$$

$$\boxed{\Omega = \frac{2}{T}\tan\left(\omega/2\right), \quad \omega = 2\tan^{-1}\left(\tfrac{T}{2}\Omega\right)}$$

Bilinear Transformation

- Preserves magnitude characteristics
- Warps freq axis

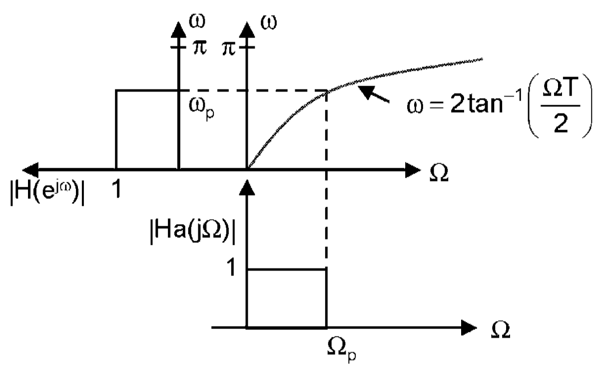
$$s = \sigma + j\Omega \rightarrow z = \frac{1 + \frac{T}{2}(\sigma + j\Omega)}{1 - \frac{T}{2}(\sigma + j\Omega)}$$

$$|z| = \frac{\sqrt{\left(1 + \frac{T}{2}\sigma\right)^2 + \Omega^2}}{\sqrt{\left(1 - \frac{T}{2}\sigma\right)^2 + \Omega^2}}$$

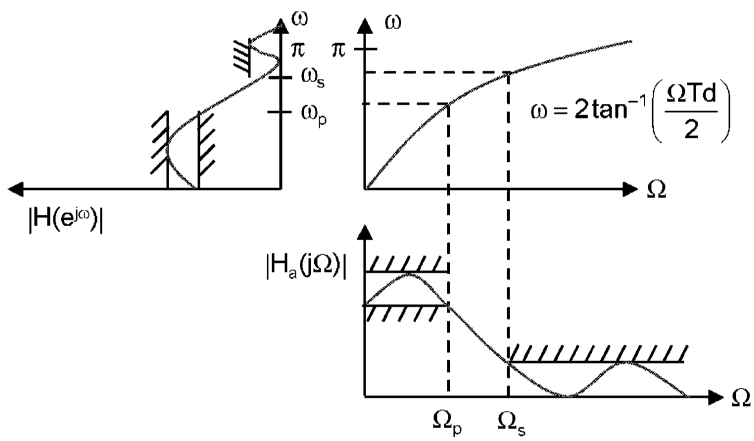
$$\therefore \sigma < 0 \rightarrow |z| < 1$$

Bilinear Transformation

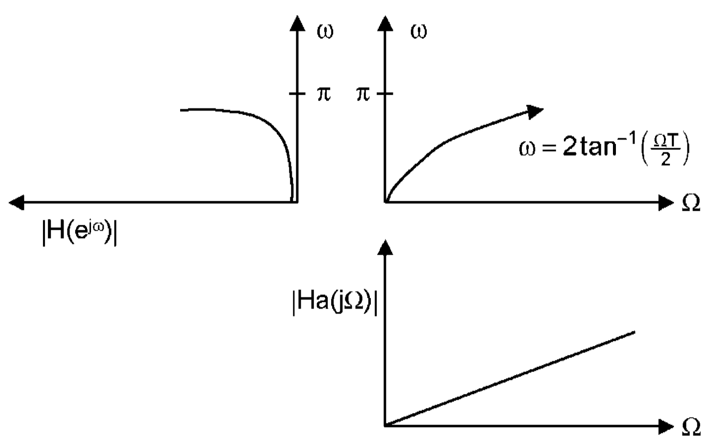
- Maps stable to stable (LHP to inside u.c.)
- Preserves L^∞ optimal analog prototype!



Bilinear Transformation



Bilinear Transformation



L^p Optimal Design

L^p optimal design $(1 < p < \infty)$

$$\xi_A^p = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^p d\omega$$

find $\{a_k\}_{k=1}^{N-1}, \{b_k\}_{k=0}^{M-1}$ to min ξ_A^p

- difficult, nonlinear in a_k, b_k, NL optimization
- needs good initial guess (e.g. Prony's)

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 14

Professor Andrew Singer
Department of Electrical and Computer Engineering

L^p Optimal Design

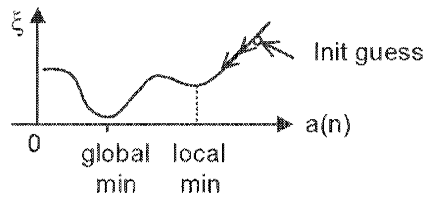
L^p optimal design $(1 < p < \infty)$

$$\xi_A^p = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^p d\omega$$

find $\{a_k\}_{k=1}^{N-1}, \{b_k\}_{k=0}^{M-1}$ to min ξ_A^p

- difficult, nonlinear in a_k, b_k, NL optimization
- needs good initial guess (e.g. Prony's)

NL Optimization for L^p Design



Deczky (1972) uses Fletcher Powell method.

Magnitude and Group Delay

Put $\xi = \alpha\, \xi_A + (1 - \alpha)\, \xi_\tau$

$$\xi_A = \sum_{i=1}^{L} W_A(\omega_i) \left\| H_d(e^{j\omega_i}) \right| - \left| A(e^{j\omega}) \right\|^{2m}$$

$$\xi_\tau = \sum_{i=1}^{L} w_\tau(\omega_i) \left| \tau(\omega_i) - \tau_d(\omega_i) \right|^{2q}$$

L^2 Optimal IIR Design

Find a[n], b[n] : $\min \sum_{n=-\infty}^{\infty} \left| h_d[n] - h[n] \right|^2$

or, equiv $\frac{1}{2\pi}\int_{-\pi}^{\pi} \left| H_d(e^{j\omega}) - H(e^{j\omega}) \right|^2 d\omega$

$\rightarrow$ again, very difficult nonlinear problem.

Linear Predictive Modeling

Let's solve an easier one:

$\rightarrow$ Linear prediction (prediction-error form)

For $H(z) = \frac{1}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$ (all pole or AR model)

$$h[n] = -\sum_{k=1}^{N-1} a_k h[n-k]$$

© 2001 Andrew Singer, University of Illinois at Urbana-Champaign, All Rights Reserved

Linear Predictive Modeling

$$\therefore \text{ find } a^* = \underset{\{a_k\}}{\arg\min} \sum_{n=-\infty}^{\infty} \left| h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right|^2$$

$\Rightarrow$ Linear in a_k!

$$0 = \frac{\partial \varepsilon}{\partial a_\ell} = 2 \sum_{n=-\infty}^{\infty} \left[h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right] h_d[n-\ell]$$

Linear Equations

$$\sum_{k=1}^{N-1} a_k \underbrace{\sum_{n=-\infty}^{\infty} h_d[n-k] h_d[n-\ell]}_{R_{hh}[k-l]} = - \underbrace{\sum_{n=-\infty}^{\infty} h_d[n] h_d[n-\ell]}_{R_{hh}[l]}$$

$\ell = 1, \ldots, N-1$

"Normal Equations"

$$\begin{bmatrix} R_{hh}[0] & \cdots & R_{hh}[N-2] \\ \vdots & \ddots & \\ R_{hh}[N-2] & & R_{hh}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = -\begin{bmatrix} R_{hh}[1] \\ \vdots \\ R_{hh}[N-1] \end{bmatrix}$$

$$R\vec{a} = -\vec{r}$$

$$\boxed{\vec{a}^* = -R^{-1}\vec{r}}$$

ARMA Modeling

How do we include ARMA (pole – zero)

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$$

Now

$$h[n] + \sum_{k=1}^{N-1} a_k h[n-k] = \begin{cases} b[n], & 0 \le n \le M-1 \\ 0, & \text{else} \end{cases}$$

Padé Approximation

* Match first M + N – 1 terms of h[n] exactly

Step 1:

$$h[n] + \sum_{k=1}^{M-1} a_k h[n-k] = 0 \quad \text{for } M \le n < M + N - 2$$

Solve for $\{a_k\}_{k=1}^{N-1}$ using N – 1 equations

Padé Approximation, cont'd

$$\begin{bmatrix} h_d[M-1] & \dots & h_d[M-N+1] \\ & & \\ h_d[M+N-3] & \dots & h_d[M-1] \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = - \begin{bmatrix} h_d[M] \\ \vdots \\ h_d[M+N-2] \end{bmatrix}$$

$$H \quad \vec{a} \quad = \quad -\vec{h}$$

$$\boxed{\vec{a}^* = -H^{-1}\,\vec{h}}$$

Padé Approximation, cont'd

Step 2:

$$b[n] = h[n] + \sum_{k=1}^{N-1} a_k^* h[n-k], \quad n = 0, \ldots, M-1$$

Now

$$e[n] = h_d[n] - h[n] = \begin{cases} 0, & 0 \le n \le M+N-2 \\ ?, & \text{else} \end{cases}$$

Prony's Method (1795)

Step 1:

find a_k to min $\sum_{n=M}^{\infty} \left| h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right|^2$

$$0 = \frac{\partial \varepsilon}{\partial a_\ell} \Rightarrow \sum_{k=1}^{N-1} a_k \underbrace{\sum_{n=M}^{\infty} h_d[n-k] h_d[n-\ell]}_{R_{hh}(k,\, \ell)} = \underbrace{-\sum_{n=M}^{\infty} h_d[n] h_d[n-\ell]}_{-R_{hh}(0,\, \ell)}$$

$R_{hh}(k, \ell)$ = $-R_{hh}(0, \ell)$

now depends on k & ℓ, can approx. with infinite sum

Prony's Method, cont'd

Step 2:

$$b[n] = h[n] + \sum_{k=1}^{N-1} a_k h[n-k] \qquad 0 \le n < M$$

$\rightarrow$ Better estimates for a[n] using linear prediction than Pade approximation

$\rightarrow$ Poor estimates of b[n]

Shank's Method (1967)

$$\delta[n] \rightarrow \boxed{\frac{1}{A(z)}} \rightarrow v[n] \rightarrow \boxed{B(z)} \rightarrow h[n]$$

Step 1: Find

$$\{a_k\}_{k=1}^{N-1} \text{ to min } \sum_{n=0}^{\infty} \left| h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right|^2$$

Using linear prediction (Prony's Method, M=0)

Shank's Method, cont'd

Step 2: Find v[n]:

$$v[n] = -\sum_{k=1}^{N-1} v[n-k]a_k^* + \delta[n]$$

$$\delta[n] \rightarrow \boxed{\frac{1}{A^*(z)}} \xrightarrow{v[n]} \boxed{B(z)} \rightarrow h[n] \approx h_d[n]$$

Shank's Method, cont'd

Step 3: Find an all zero (MA) filter such that

$$b^* = \underset{b_k}{\operatorname{argmin}} \sum_{n=0}^{\infty} \left| h_d[n] - \sum_{k=0}^{M-1} b_k v[n-k] \right|^2$$

$$\sum_{k=0}^{M-1} b_k \underbrace{\sum_{n=0}^{\infty} v[n-k]v[n-\ell]} = \sum_{n=0}^{\infty} h_d[n]v[n-\ell]$$

$$\ell = 0,\ldots,M-1, \quad \sum_{k=0}^{M-1} b_k \; R_{vv}[k,\ell] = R_{hv}[0,\ell]$$

$$\boxed{\vec{b}^* = R^{-1}\vec{r}}$$

AR / ARMA Modeling

Note:

None of these methods is a true L^2 optimal model!

All are related to Linear Prediction

Prony's (all pole) Method

$$H_d(z) = \frac{b_0}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$$

1) $h_d[0] = b_0$

2) Find a_k to min $\sum_{n=1}^{\infty} \left| h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right|^2$

Prony's (all pole) Method, cont'd

$$\frac{\partial \varepsilon}{\partial a_\ell} = 0 \Rightarrow \sum_{k=1}^{N-1} a_k \sum_{n=1}^{\infty} h_d[n-k] h_d[n-\ell] =$$

$$= -\sum_{n=1}^{\infty} h_d[n] h_d[n-\ell]$$

Solve for $\ell = 1, \ldots, N-1$

$$\sum_{k=1}^{N-1} R_{hh}(k,\ell) a_k = -R_{hh}(0,\ell)$$

Extended Prony's Method

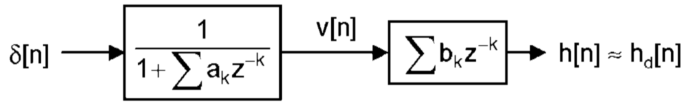
$$H_d(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}} \qquad \text{(pole-zero case)}$$

Now, do 2) first:

find a_k to min $\sum_{n=M}^{\infty} \left| h_d[n] + \sum_{k=1}^{N-1} a_k h_d[n-k] \right|^2$

Now 3) $b_k = h[k] + \sum_{\ell=1}^{N-1} a_\ell h_d[k-\ell], \qquad k = 0, \ldots, M-1$

Shank's Method



1) Compute a_k, k = 1, …, N - 1 using Prony's all-pole method.
2) Compute v[n]
3) Find b_k to min $\sum_{n=0}^{\infty} \left| h_d[n] - \sum_{k=0}^{M-1} b_k v[n-k] \right|^2$

Shank's Method, cont'd

Solve for ℓ = 0, …, M-1

$$\sum_{k=0}^{M-1} b_k \sum_{n=0}^{\infty} v[n-k]v[n-\ell] = \sum_{n=0}^{\infty} h_d[n]v[n-\ell]$$

$$\boxed{\sum_{k=0}^{M-1} b_k R_{vv}(k,\ell) = R_{hv}(0,\ell)}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 15

Professor Andrew Singer
Department of Electrical and Computer Engineering

FIR Filter Design

- Last time, IIR Filter design, based on analog filter design and transformations, L^p methods
- Now FIR filter design
  - Given a design specification, $h_d[n]$, or $H_d(e^{j\omega})$
  - Find h[n], s.t. h[n] = 0, n < 0, n > M,
  - "good", "optimal" methods

Finite-length Impulse Response

Assume h[n] of length M + 1

→ Guaranteed stable (why?)

→ Can have linear phase (how?)

→ Can be easily implemented / designed

Common Methods

- Window design methods
- Frequency sampling
- Computer L^P optimization methods
  - L^2
  - L^∞
  - mixed L^2 -L^∞ (popular - see rice website)

Why "do windows"?

Q: Why window design?

A1: Because it's "optimal" (not real reason)

A2: Because it's Easy!

Consider desinging an FIR filter using a "full-band unweighted L^2-optimal" design:

Full-band L^2-optimal FIR design

Optimality Criterion:

$$h^*[n] = \underset{h[n]}{\arg\min}\ \xi = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^2 d\omega$$

$$h[n] = \begin{cases} \text{nonzero}, & 0 \le n \le M \\ 0, & \text{else} \end{cases}$$

Parseval's Theorem

$$\xi = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^2 d\omega = \sum_{n=-\infty}^{\infty}\left|h_d[n]-h[n]\right|^2$$

$$= \underbrace{\sum_{n=0}^{M}\left|h_d[n]-h[n]\right|^2}_{} + \underbrace{\sum_{\substack{n=-\infty \\ n\neq[0,\ldots,M]}}^{\infty}\left|h_d[n]-0\right|^2}_{\text{Does not depend on } h[n]}$$

Does not depend on h[n]

$$= \sum_{n=0}^{M}\left|h_d[n]-h[n]\right|^2 + \sum_{\substack{n<0 \\ n>M}}\left|h_d[n]\right|^2$$

And the optimal FIR filter is…

Minimize ξ ? $\Rightarrow h[n] = \begin{cases} h_d[n], & 0 \le n \le M \\ 0, & \text{else} \end{cases}$

$$\Rightarrow h[n] = w[n]h_d[n],$$

$$\text{or, } h[n] = w[n]h_d[n-\tfrac{M}{2}] \text{ (why?)}$$

A Rectangular Window

$$w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{else} \end{cases}$$

"Rectangular window" or "truncation method"

Is it any good?

"Just because it's optimal doesn't mean it's good!"

What does the window do?

$$H(e^{j\omega}) = H_d(e^{j\omega}) \otimes W(e^{j\omega})$$

← periodic convolution

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

↑ complex valued functions

Maintain Linear Phase?

- Assume:
  - $h_d[n]$ symmetric about n=M/2 (integer or ½ integer)
  - $w[n]$ symmetric about n=M/2
- what is $\tau(\omega)$?

Linear Phase, cont'd

$$H_d(e^{j\omega}) = H_e(e^{j\omega})e^{-j\omega\frac{M}{2}}$$

$H_e(e^{j\omega})$, $W_e(e^{j\omega})$: real, even

$$h_e[n] = h_e\left[n - \tfrac{M}{2}\right]$$

for $\frac{M}{2}$ integer

what if not?

$$W(e^{j\omega}) = W_e(e^{j\omega})e^{-j\omega\frac{M}{2}}$$

$$\Rightarrow \boxed{H(e^{j\omega}) = A_e(e^{j\omega})e^{-j\omega\frac{M}{2}},}$$

$$A_e(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_e(e^{j\theta})W_e(e^{j(\omega-\theta)})d\theta$$

Example

Rectangular window

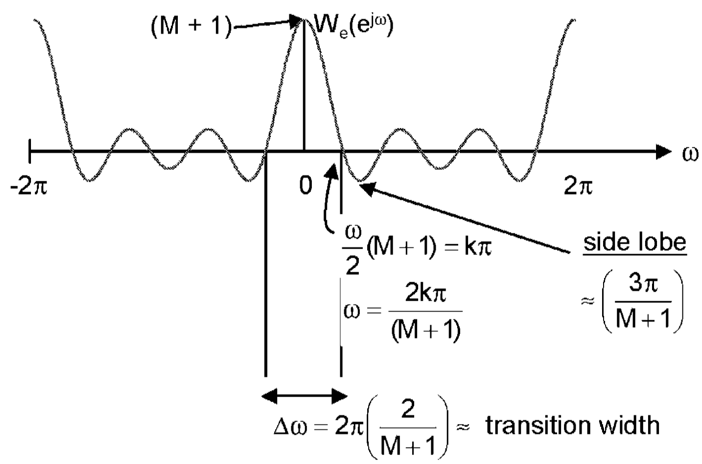
$$W(e^{j\omega}) = \sum_{n=0}^{M} (1)e^{-j\omega n} = \frac{1}{1-e^{-j\omega}} - \frac{e^{-j\omega(M+1)}}{1-e^{-j\omega}}$$

$$= \frac{e^{-j\omega\left(\frac{M+1}{2}\right)}\left(e^{j\omega\left(\frac{M+1}{2}\right)} - e^{-j\omega\left(\frac{M+1}{2}\right)}\right)}{e^{-j\omega\left(\frac{1}{2}\right)}\left(e^{j\left(\omega/2\right)} - e^{-j\left(\omega/2\right)}\right)}$$

$$= e^{-j\omega\left(\frac{M}{2}\right)} \frac{\sin\left(\omega\frac{M+1}{2}\right)}{\sin\left(\omega/2\right)}$$

Example, cont'd

"Periodic sinc"



Example, cont'd

First side lobe

$$\frac{\partial}{\partial\omega}\left(\frac{\sin\omega\left(\frac{M+1}{2}\right)}{\sin\left(\omega/2\right)}\right)=0$$

$$\omega\approx\frac{3\pi}{M+1}$$

$$\left|W(e^{j\omega})\right|=\sin\left(\frac{3\pi}{M+1}\left(\frac{M+1}{2}\right)\right)/\sin\left(\frac{3\pi}{(M+1)2}\right)$$

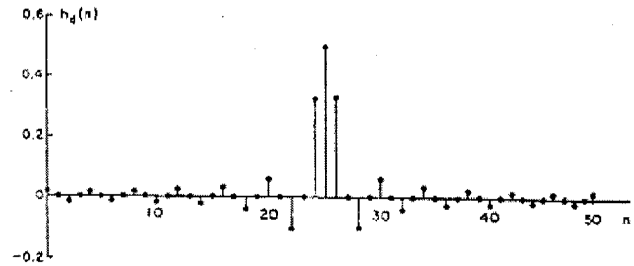
Example, cont'd

$$\approx\frac{|-1|}{\left(\frac{3\pi}{2(M+1)}\right)}$$

$$=\frac{2(M+1)}{3\pi}$$

$$\Rightarrow\frac{\left|w(e^{j0})\right|}{\left|w(e^{j\omega_{sl}})\right|}\approx\frac{M+1}{\frac{2(M+1)}{3\pi}}=\frac{3\pi}{2}\cong 13\text{dB}$$

Windowing an ideal lowpass filter



Truncated impulse response of an ideal lowpass filter. (Delay is 25 samples, total length is 51 samples, and cutoff frequency is $\omega_c = \pi/2$.)

Periodic Convolution

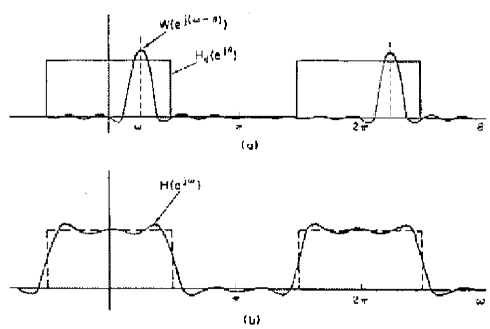


Figure 7.27 (a) Convolution process implied by truncation of the ideal impulse response. (b) Typical approximation resulting from windowing the ideal impulse response.

Commonly-used Windows

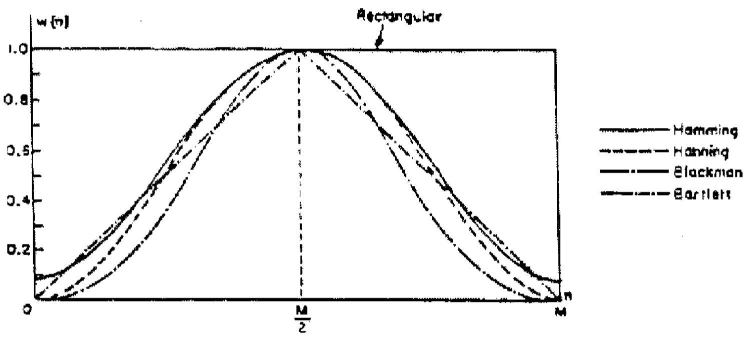


Figure 7.29 Commonly used windows.

$W(e^{j\omega})$

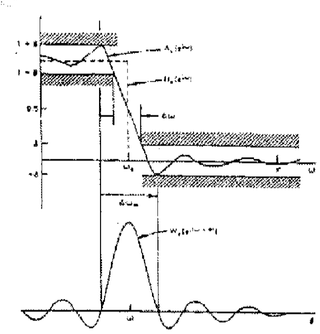


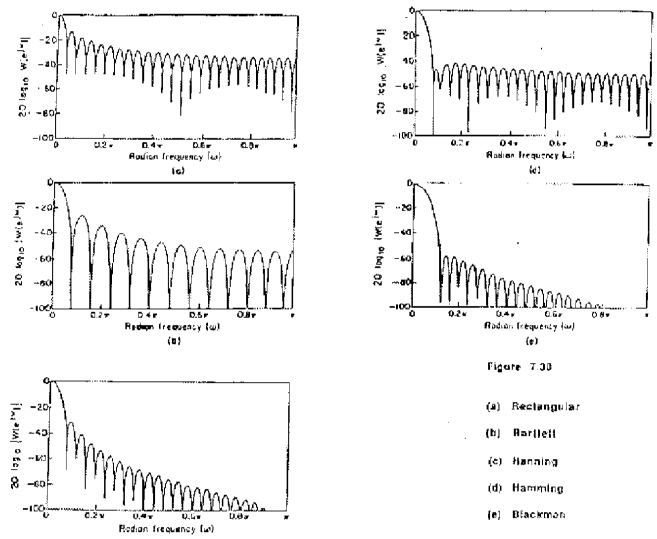
Figure 7.31

- Main lobe controls transition band. $\Delta\omega \approx 2\pi(2/(M+1))$
- Side lobes control passband and stopband ripple. $-20\log_{10}\delta \approx 20$db
- Passband and stopband ripple approximately equal over a wide range of cutoff frequencies.

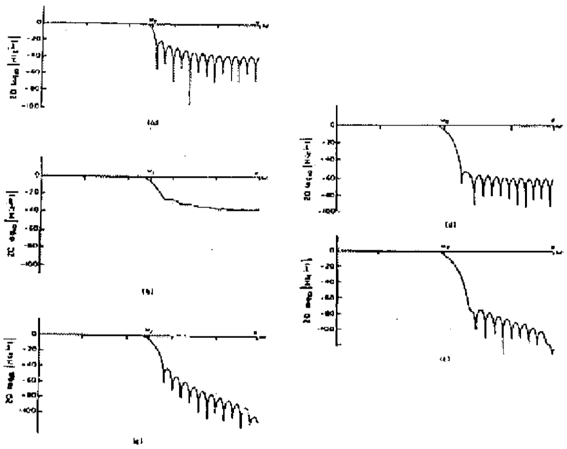
Ideal Properties for $w[n]$

For a given window length, $W(e^{j\omega})$ "most like an impulse" $\Rightarrow$ narrow main lobe, low sidelobes.

Frequency Responses of Windows



Frequency Responses cont'd



(a) Rectangular; (b) Bartlett; (c) Hanning; (d) Hamming; (e) Blackman.

Window Shopping

TABLE 7.2 COMPARISON OF COMMONLY USED WINDOWS

Window Type	Peak Sidelobe Amplitude (Relative)	Approximate Width of Mainlobe	Peak Approximation Error $20\log_{10}\delta$ (dB)	Equivalent Kaiser Window β	Transition Width of Equivalent Kaiser Window
Rectangular	−13	$4\pi/(M+1)$	−21	0	$1.81\pi/M$
Bartlett	−25	$8\pi/M$	−25	1.33	$2.37\pi/M$
Hanning	−31	$8\pi/M$	−44	3.86	$5.01\pi/M$
Hamming	−41	$8\pi/M$	−53	4.86	$6.27\pi/M$
Blackman	−57	$12\pi/M$	−74	7.04	$9.19\pi/M$

The Kaiser Window Family

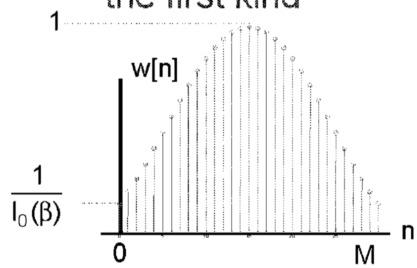
Class of windows parameterized by size and shape

$$w[n] = \begin{cases} I_0\left[\beta\sqrt{1-\left(\frac{n-\alpha}{\alpha}\right)^2}\right] \Big/ I_0(\beta) & 0 \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

The Kaiser Window Family, cont'd

$\alpha = M/2$ β: shape parameter

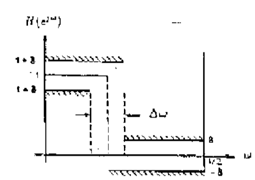
$I_0(\cdot)$: zeroth-order modified Bessel function of the first kind



β increasing
↓
$I_0(\beta)$ increasing
↓
smoother window

Kaiser Window

- Stop band spec's:
  - choose β
- Transition Band:
  - choose M



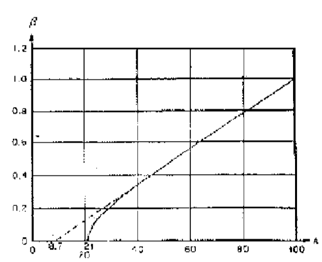
$A = -20\log_{10}\delta$

Filter Length $= M + 1$

$$M \geq \frac{A - 8}{2.285\Delta\omega}$$

Shape parameter:

$$\beta = \begin{cases} 0.1102(A - 8.7), & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & 21 \leq A \leq 50 \\ 0, & A \leq 21 \end{cases}$$



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 16

Professor Andrew Singer
Department of Electrical and Computer Engineering

Frequency Sampling Design

Design a length-M+1 FIR filter

$$H(e^{j\omega}) = \sum_{n=0}^{M} h[n]e^{-j\omega n} \qquad (1)$$

to fit exactly

$H_d(e^{j\omega_k})$ for M + 1 samples, ω_k

M + 1 equations (1), M + 1 unknowns h[n]

Frequency Sampling

1) $P(x) = a_0 + a_1x + \ldots + a_Mx^M$: polynomial

   - coefficients uniquely specified by M + 1 samples of P(x)

2) $h[n] = 0, n < 0, n > M \leftrightarrow H(e^{j\omega}) = \sum_{0}^{M} h[n](e^{-j\omega})^n$

   Mth order polynomial

   $\therefore h[n] \leftrightarrow M + 1$ "frequency samples"

3) Construct h[n] via polynomial interpolation

M+1 Linear Equations in h[n]

$$\underbrace{\begin{bmatrix} H_d(\omega_0) \\ \vdots \\ H_d(\omega_M) \end{bmatrix}}_{H} = \underbrace{\begin{bmatrix} 1 & e^{-j\omega_0} & e^{-j2\omega_0} \ldots & e^{-j\omega_0 M} \\ 1 & & & \\ \vdots & & & \\ 1 & e^{-j\omega_M} & e^{-j2\omega_M} \ldots & e^{-j\omega_M M} \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[M] \end{bmatrix}}_{h}$$

$$h^* = D^{-1} H$$

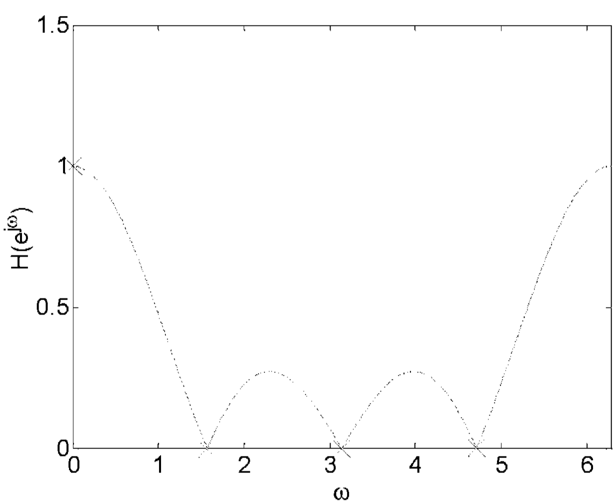
include linear phase!

$$e^{-j\omega_k\left(\frac{M}{2}\right)} H_e(\omega_k)$$

Example: 4 tap Low pass filter

```
>> M=3; w=2*pi*[0:M]/(M+1);
>> H=[1 0 0 0]';
>> D=exp(-j*w(:)*[0:M])
D =
   1.0000         1.0000                1.0000                1.0000
   1.0000         0.0000 - 1.0000i   -1.0000 - 0.0000i   -0.0000 + 1.0000i
   1.0000        -1.0000 - 0.0000i    1.0000 + 0.0000i   -1.0000 - 0.0000i
   1.0000        -0.0000 + 1.0000i   -1.0000 - 0.0000i    0.0000 - 1.0000i
>> h=D\H
h =
   0.2500 + 0.0000i
   0.2500 - 0.0000i
   0.2500 - 0.0000i
   0.2500 - 0.0000i
>> plot(w,abs(fft(h)),'x','Markersize',20)
```

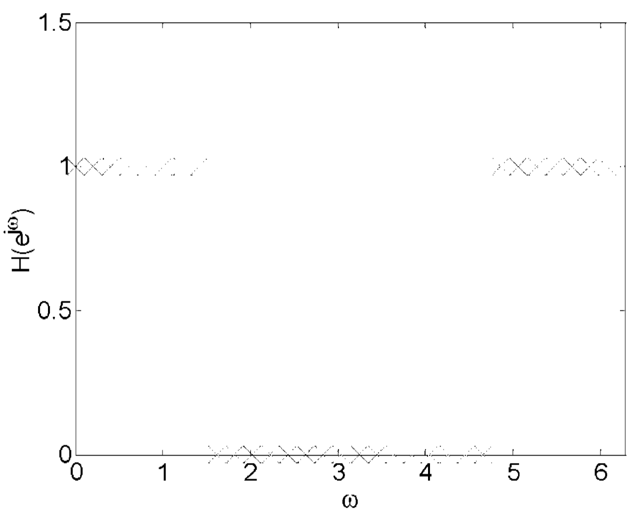

Resulting Frequency Response



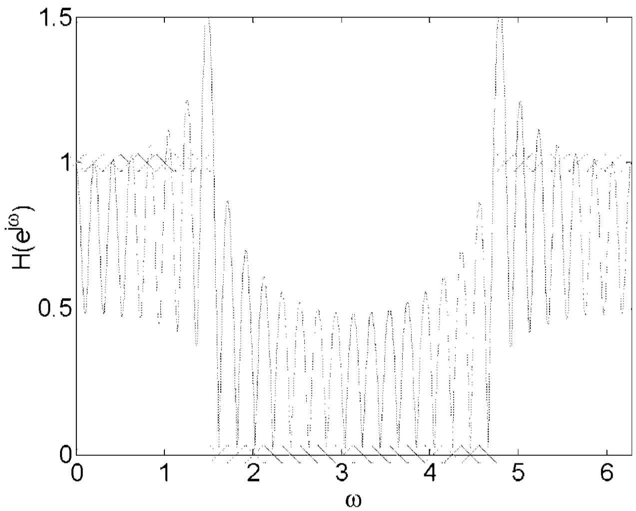
Another Example, now let M=30

```
>> M=30; w=2*pi*[0:M]/(M+1);

>> low=find(w<pi/2 | w>3*pi/2);     %% want a real filter

>> H=zeros(M+1,1);H(low)=ones(size(low)); %% LPF

>> h=D\H;

>> plot(w,abs(fft(h)),'x', 'Markersize',20)
>> xlabel('\omega')
>> ylabel('H(e^{j\omega})')
```


Resulting Frequency Response



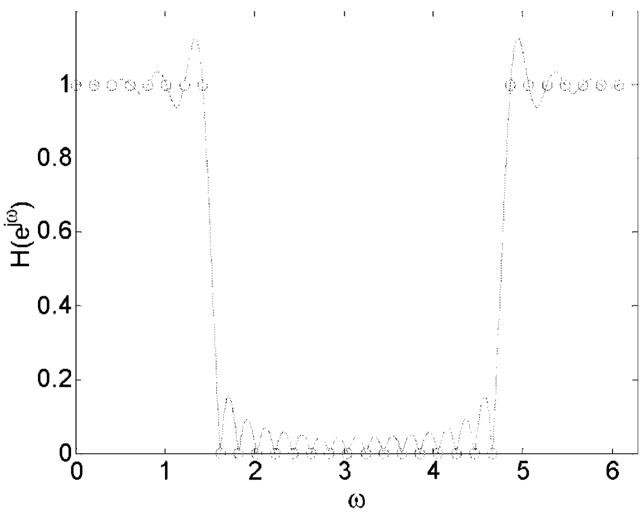
Is this what we wanted?



Let's try that again…

```
>> M=30; w=2*pi*[0:M]/(M+1);
>> H=zeros(M+1,1);H(low)=ones(size(low));

>> H=H.*exp(-j*w(:)*M/2);  % include linear phase this time

>> D=exp(-j*w(:)*[0:M]);
>> h=D\H;
>> plot(w,abs(fft(h)),'x',linspace(0,2*pi-
   2*pi/1024,1024),abs(fft(h,1024)),'Markersize',20)
>> xlabel('\omega')
>> ylabel('H(e^{j\omega})')
```

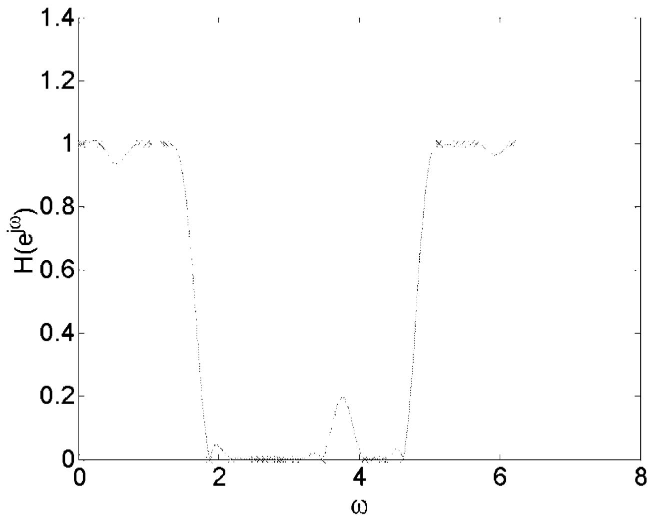

Resulting Frequency Response



Another Example, random locations

```
>> M=30; w=rand(M+1,1)*2*pi;  %% random freq's
>> low=find( (w<(pi/2)) | (w>(3*pi/2)));
>> H=zeros(M+1,1);H(low)=ones(size(low));
>> H=H.*exp(-j*w(:)*M/2);
>> D=exp(-j*w(:)*[0:M]);
>> h=D\H;
>> plot(w,abs(D*h),'x',linspace(0,2*pi-
   2*pi/1024,1024),abs(fft(h,1024)),'-','Markersize',8);
>> xlabel('\omega')
>> ylabel('H(e^{j\omega})')
```


Resulting Frequency Response



"DFT Matrix"

If $\omega_k = \dfrac{2\pi k}{M+1} \Rightarrow D = \text{DFT matrix}$

$$h^*[n] = \text{IDFT}\left(H_e(e^{j\omega})e^{-j\omega\left(\frac{M}{2}\right)}\right)$$

>> h=ifft(H); %% Matlab uses fft algorithm

Can also use more than M equations, but then

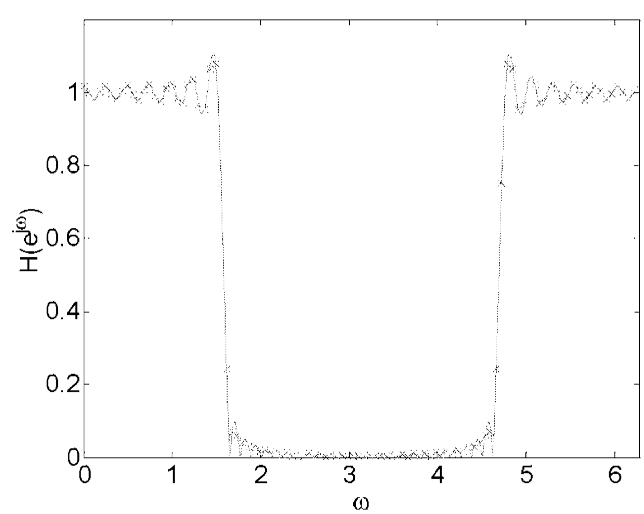
$$H(e^{j\omega_k}) \neq H_d(e^{j\omega_k})$$

may not, "least-squares" method, L^2 solution

Least Squares (#Eqs > #Unknowns)

```
>> M=100; w=2*pi*[0:M]/(M+1); %100 equations
>> low=find( (w<(pi/2)) | (w>(3*pi/2)));
>> H=zeros(M+1,1);H(low)=ones(size(low));
>> L=M/2;                              % length 50 filter
>> H=H.*exp(-j*w(:)*L/2);
>> D=exp(-j*w(:)*[0:L]);
>> h=D\H;
>> plot(w,abs(D*h),'x',linspace(0,2*pi-
   2*pi/1024,1024),abs(fft(h,1024)),'-');
```


Resulting Frequency Response



Weighted L^2 Error Design (WLS)

$$h^*[n] = \arg\min_{h[n]} \xi^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^2 d\omega$$

$$0 = \frac{\partial \varepsilon^2}{\partial h[n]} \Rightarrow$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\frac{\partial}{\partial h[n]}\left\{\left(H_d\left(e^{j\omega}\right) - H(e^{j\omega})\right)\left(H_d\left(e^{j\omega}\right) - H(e^{j\omega})\right)^*\right\}d\omega$$

Weighted Least Square Design, cont'd

$$\underbrace{\frac{-1}{2\pi}\int_{-\pi}^{\pi} W(\omega)H_d(e^{j\omega})e^{j\omega n}d\omega}_{-d[n]} + \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)H(e^{j\omega})e^{j\omega n}d\omega = 0$$

$$-d[n] + \sum_{\ell=0}^{M} h[\ell]\underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)e^{j\omega(n-\ell)}d\omega}_{M_{n,\ell}} = 0$$

Weighted Least Square Design, cont'd

$$\left[M_{[n,\ell]}\right]\begin{bmatrix} h[0] \\ \vdots \\ h[M] \end{bmatrix} = \begin{bmatrix} d[0] \\ \vdots \\ d[M] \end{bmatrix}$$

$$\boxed{\vec{h}^* = M^{-1}\vec{d}}$$

ECE 451 - Lecture 16 18 of 22

Example: Mth-order FIR LPF

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega(n-M/2)}, & |\omega| < \pi/2 \\ 0, & \text{else} \end{cases}$$

$$W(\omega) = \begin{cases} 100, & |\omega| < \pi/2 \\ 1, & \text{else} \end{cases}$$

$$d[n] = \tfrac{-1}{2\pi}\int_{-\pi/2}^{\pi/2} 100e^{-j\omega(n-M/2)}d\omega$$

$$M_{n,\ell} = \int_{-\pi/2}^{\pi/2} 100e^{j\omega(n-\ell)}d\omega + \int_{-\pi}^{-\pi/2} e^{j\omega(n-\ell)}d\omega + \int_{\pi/2}^{\pi} e^{j\omega(n-\ell)}d\omega$$

ECE 451 - Lecture 16 19 of 22

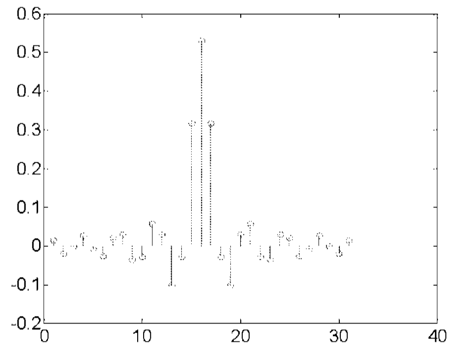
Example, M=30, cont'd

$$d[n] = 50\text{sinc}(\tfrac{\pi}{2}(n-\tfrac{M}{2})), \quad n = 0,\ldots,M$$

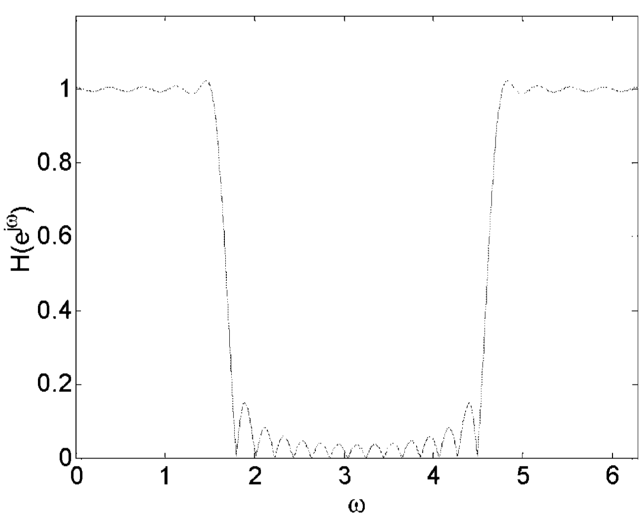
$$M_{n,\ell} = 50\text{sinc}(\tfrac{\pi}{2}(n-\ell)) - \tfrac{1}{2}\text{sinc}(\tfrac{\pi}{2}(n-\ell)) + \text{sinc}(\pi(n-\ell)),$$

$$n = 0,\ldots,M \qquad \ell = 0,\ldots,M$$

$$>> h = M \backslash d$$



Resulting Frequency Response



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 17

Professor Andrew Singer
Department of Electrical and Computer Engineering

L^∞ FIR Filter Design (min-max)

- Window design:

  Not equiripple, no control over relative PB/SB error

- LS design:

  Weighting permits control, but large peak errors may still occur

- Today L^∞ design:

$$\min_{h[n]} \max_{\omega} |E(\omega)|, \quad \omega \in F$$

Look At Type I FIR Filter

$$H(e^{jw}) = A_e(e^{jw})e^{-jw\frac{M}{2}}$$

M is even
- odd length filter

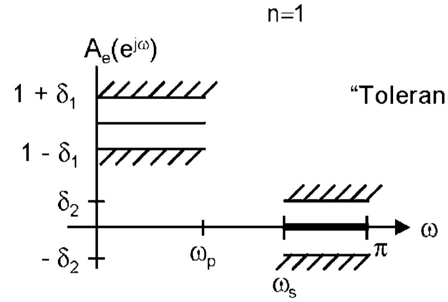
$$h[n] = h_e\left[n - \tfrac{M}{2}\right], \quad h_e[n] = h_e[-n],$$
$$= h[M-n]$$

$$A_e(e^{j\omega}) = \sum_{n=-L\ldots L} h_e[n]e^{-j\omega n}$$

Type I Fir Filter, cont'd

$$L = \frac{M}{2}, \text{ an integer}$$

$$\Rightarrow A_e(e^{j\omega}) = h_e[0] + \sum_{n=1}^{L} 2h_e[n]\cos(n\omega)$$



"Tolerance Scheme"

Parks & McClellan (1972)

L, ω_p, ω_s, δ_1/δ_2 fixed and δ_1 minimized.

- most commonly used method of FIR design
- uses polynomial approximation theory
- efficient, convenient computer algorithms available (MATLAB, Fortran, C, etc…)

Polynomial Approximation Problem

$$A_e(e^{j\omega}) = h_e[0] + \sum_{n=1}^{L} b_n \cos(\omega n)$$

L^{th}-degree trigonometric polynomial

$$A_e(e^{j\omega}) = h_e[0] + \sum_{n=1}^{L} a_n (\cos(\omega))^n = \underbrace{P(\cos(\omega))}$$

Using Tchebychev polynomials, $T_n(\cos\omega) = \cos\omega n$

$$A_e(e^{j\omega}) = P(x)\Big|_{x=\cos\omega}$$ ← Polynomial in x

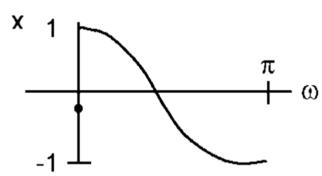
Tchebychev Polynomials

$$T_n(x) = \cos(n\cos^{-1}(x)),\quad T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

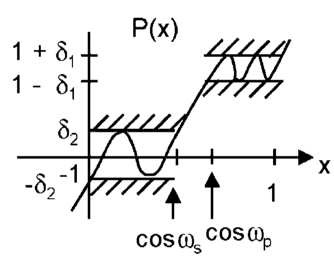
$$T_0 = 1$$

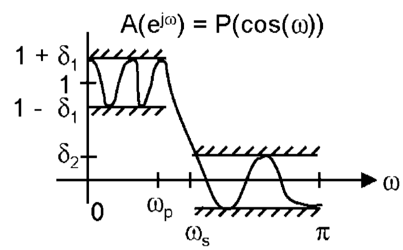
$$T_1 = x$$

$$P(x) = \sum_{k=0}^{L} a_k x^k \qquad x \in [-1, 1] \to \omega \in [0, \pi]$$



Polynomial Approximations





$$\to \text{Note}\quad P(\cos\omega)'\Big|_{\omega=\pi,0} = \left(P'(x)\Big|_{x=\cos\omega}\sin\omega\right)\Big|_{\omega=\pi,0}$$

$$= 0$$

zero slope at $\omega = 0, \pi$ by mapping, not nec. by extrema

Alternation Theorem Formulation

Define $E(\omega) = W(\omega)[H_d(e^{j\omega}) - A_e(e^{j\omega})]$

$$\updownarrow \quad x = \cos\omega$$

$$E_p(x) = W_p(x)[D_p(x) - P(x)]$$

Defined on union of closed intervals, e.g.

$$F = [0, \omega_p] \cup [\omega_s, \pi]$$

Alternation Theorem Formulation

$$\rightarrow F_p = [-1, \cos\omega_s] \cup [\cos\omega_p, 1]$$

$$W(\omega) = \begin{cases} \frac{1}{k}, & 0 \le \omega \le \omega_p \\ 1, & \omega_s \le \omega \le \pi \end{cases}, \quad K = \delta_1/\delta_2$$

Alternation Theorem

Let F_p denote the closed subset consisting of the disjoint union of closed subsets of the real axis x.

P(x) denotes an r^{th}-order polynomial

$$P(x) = \sum_{k=0}^{r} a_k x^k$$

Alternation Theorem, cont'd

Also, $D_P(x)$ denotes a given desired function of x that is continuous on F_P; $W_P(x)$ is a positive function, continuous on F_P, and $E_P(x)$ denotes the weighted error

$$E_P(x) = W_P(x)[D_P(x) - P(x)].$$

The maximum error ||E|| is defined as

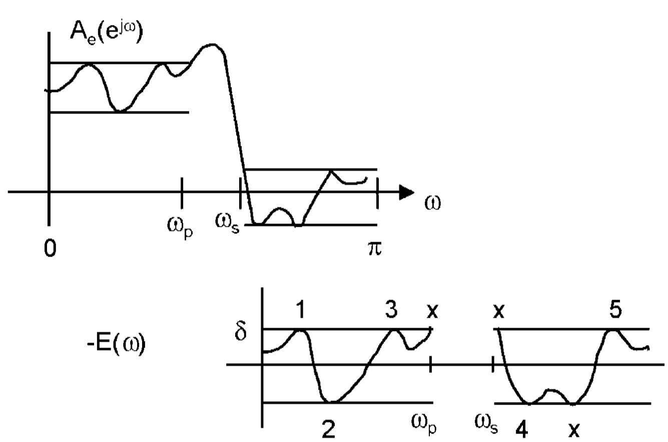
$$\|E\| = \max_{x \in F_P} |E_p(x)|$$

Alternation Theorem, cont'd

A necessary and sufficient condition that P(x) is the unique r^{th}-order polynomial that minimizes $||E||$ is that $E_P(x)$ exhibit at least (r + 2) alternations, i.e., there must exist at least (r + 2) values x_i in F_P such that $x_1 < x_2 < \ldots < x_{r+2}$ and such that

$$E_P(x_i) = -E_P(x_{i+1}) = \pm ||E|| \text{ for } i = 1, 2, \ldots, (r+1).$$

Counting Alternations



5 alternations

Optimal Type I Lowpass Filters

$$P(\cos\omega) = \sum_{k=0}^{L} a_k (\cos\omega)^k \quad x = \cos\omega,\ r = L$$

$$D_P(\cos\omega) = \begin{cases} 1, & \cos\omega_p \le \cos\omega \le 1 \\ 0, & -1 \le \cos\omega \le \cos\omega_s \end{cases}$$

Type I Lowpass Filters, cont'd

$$W_p(\cos\omega) = \begin{cases} \frac{1}{K} & \cos\omega_p \le \cos\omega \le 1 \\ 1 & -1 \le \cos\omega \le \cos\omega_s \end{cases} \qquad k = \frac{\delta_1}{\delta_2}$$

$$E_P(\cos\omega) = W_P(\cos\omega)[D_P(\cos\omega) - P(\cos\omega)]$$

min# alternations = L + 2 → "equiripple case"

Type I Lowpass Filters, cont'd

- L + 3 max
- ω_p, ω_s must be alternations
- all external points in bands must be alternations
- transition band monotonic

Type I Lowpass Filters, cont'd

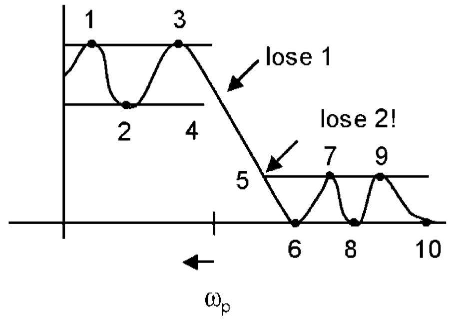
$\rightarrow$ L + 3 L^{th} order poly $\rightarrow$ L – 1 extrema

4 band edges $\rightarrow$ L – 1 + 4 = L + 3

$\rightarrow$ ω_p, ω_s **must be included**:

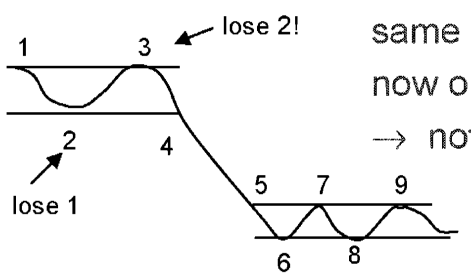
(same argument holds for ω_s)

Lose 1, Lose 2 for Type 1 Lowpass



Lose 1, Lose 2 Again

All external points must be alternations
(except poss. 0, π)



same argument.
now only L + 1 possible
$\rightarrow$ not enough.

Transition band must be monotonic
(homework problem)

Type II Lowpass, M odd, Even Length

$$h[n] = h[M-n]$$

$$H(e^{j\omega}) = e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} 2h[n]\cos\left(\omega\left(\frac{M}{2}-n\right)\right)$$

$$= e^{-j\omega\frac{M}{2}} \cos\left(\omega/2\right) P(\cos\omega)$$

Type II Lowpass, cont'd

$$D_P(\cos\omega) = \begin{cases} \frac{1}{\cos\omega}, & 0 \le \omega \le \omega_p \\ 0, & \omega_s \le \omega \le \pi \end{cases}$$

$$W(\omega) = \begin{cases} \frac{\cos(\omega/2)}{k}, & 0 \le \omega \le \omega_p \\ \cos\left(\omega/2\right), & \omega_s \le \omega \le \pi \end{cases}$$

The Parks-McClellan Algorithm

The optimal filter will satisfy the set of equations:

$$E(\omega_i) = W(\omega_i)[H_d(e^{j\omega_i}) - A_e(e^{j\omega_i})] = \pm\delta$$

Algorithm:

1. Initial guess of L+2 extremal frequencies,

   ω_i, i=1..L+2
2. Solve for polynomial coefficients and δ, using L+2 equations in L+2 unknowns.
3. Evaluate $A_e(e^{j\omega})$ and $E(\omega)$ on a dense set of frequencies and choose a new set of extremal frequencies.
4. Check whether the extremal frequencies have changed. If yes, goto step 2, if no, end.

More detail on step 2

The optimum filter will satisfy:

$$W(\omega_i)[H_d(e^{j\omega_i}) - A_e(e^{j\omega_i})] = (-1)^{i+1}\delta$$

$$i = 1, 2, \ldots, L + 2$$

More detail, cont'd

$x_i = \cos\omega_i$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^L & \frac{1}{W(\omega_1)} \\ 1 & x_2 & x_2^2 & \dots & x_2^L & \frac{-1}{W(\omega_1)} \\ \vdots & & & & & \\ 1 & x_{L+2} & x_{L+2}^2 & \dots & x_{L+2}^L & \frac{(-1)^{L+2}}{W(\omega_{L+2})} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_L \\ \delta \end{bmatrix} = \begin{bmatrix} H_d(e^{j\omega_1}) \\ H_d(e^{j\omega_2}) \\ \vdots \\ H_d(e^{j\omega_{L+2}}) \end{bmatrix}$$

$0(L^3)$ by standard methods, L log L by FFT

Another, more efficient method

Method #2 (More efficient)

For a given set of ω_k's :

$$\delta = \frac{\sum_{k=1}^{L+2} b_k H_d(e^{j\omega k})}{\sum_{k=1}^{L+2} \frac{b_k(-1)^{k+1}}{W(\omega_k)}}$$

$$b_k = \prod_{\substack{i=1 \\ i \neq k}}^{L+2} \frac{1}{(x_k - x_i)}$$

$$x_k = \cos\omega_k$$

Lagrange Interpolation

Use to construct $A_e(e^{j\omega})$ without solving for a_k:

$$A_e(e^{j\omega}) = P(\cos\omega) = \frac{\sum_{k=1}^{L+1} \frac{d_k}{(x - x_k)} P(\cos\omega_k)}{\sum_{k=1}^{L+1} \frac{d_k}{(x - x_k)}}$$

Lagrange Interpolation, cont'd

$$d_k = \prod_{\substack{i=1 \\ i \neq k}}^{L+1} \frac{1}{(x_k - x_i)}$$

$$P(x) = \sum_{k=1}^{L+1} P(x_k) \underbrace{\prod_{\substack{i=0 \\ i \neq k}}^{L} \frac{x - x_i}{x_k - x_i}}$$

$$= \begin{cases} 1, & x = x_k \\ 0, & x = x_j \;\; j \neq k \end{cases}$$

Lagrange Interpolation, cont'd

Application to step 2:

if δ known $\rightarrow$ (a) compute

$$A(\omega_k) = H_d(\omega_k) - \frac{(-i)^k \delta}{W(\omega_k)}$$

0(NL) operatives

(b) Obtain $A(e^{j\omega})$ on dense grid by Lagrange Interpolation w/o computing a_k's

Kaiser's Empirical Formula

Determine necessary filter length:

$$M = \frac{-10\log_{10}(\delta_1\delta_2) - 13}{2.324\Delta w}$$

Linear Phase FIR Filter Design

Method	Optimality Criterion	Advantage	Dis-advantage
Window design	L^2, full band "anything's optimal"	Easy! quick!	ad hoc
Freq Sampling	$A_e(e^{j\omega_k}) = H_d(e^{j\omega_k})$	Easy! quick!	Picket Fence
Least Squares	$\frac{1}{2\pi}\int W(\omega)\left\|H_d(e^{j\omega}) - A(e^{j\omega})\right\|^2 d\omega$	weighting some trades in PB/SB	Gibbs (L^2)
L^∞	$\max_{w \in F} W(w)\left[H_d(e^{jw}) - A(e^{jw})\right]$	minimax, control of band, arbitrary flexible design	hard for very long filters

Graphs

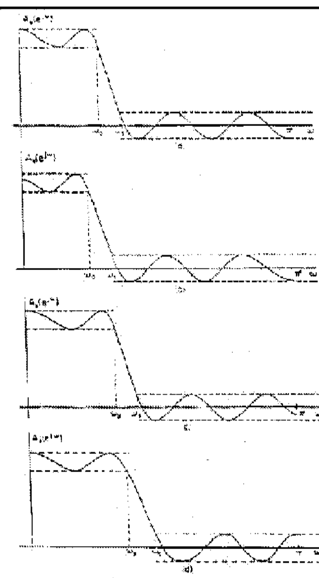


Figure 7.44 Possible optimum lowpass filter approximations for L=7. (a) L+3 alternations (extraripple case). (b) L+2 alternations (extremum at $\omega = \pi$). (c) L+2 alternations (extremum at $\omega = 0$). (d) L+2 alternations (extremum at both $\omega = 0$ and $\omega = \pi$).

The optimal filter will satisfy the set of equations

$$W(\omega_i)\,[H_d(e^{j\omega_i}) - A(e^{j\omega_i})] = +/- \delta$$

Algorithm:

1. Initial guess of (L+2) extremal frequencies $\omega_0, \omega_1, \ldots, \omega_{L+1}$

2. Solve for polynomial coefficients and δ. (L+2) equations in (L+2) unknowns.

3. Evaluate $A_e(e^{j\omega})$ or $E(\omega)$ on a dense set of frequencies and choose a new set of extremal frequencies

4. Check whether extremal frequencies changed. If yes go to step 2. If no, algorithm has converged.

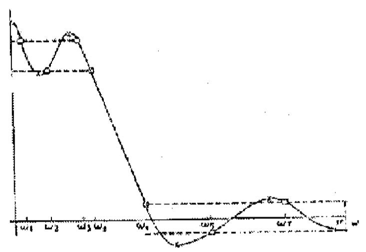


Figure 7.47 Illustration of the Parks-McClellan algorithm for equiripple approximation.

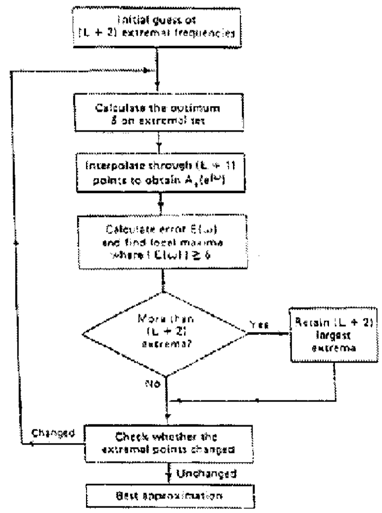


Figure 7.48 Flowchart of Parks-McClellan algorithm.

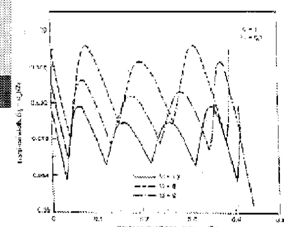
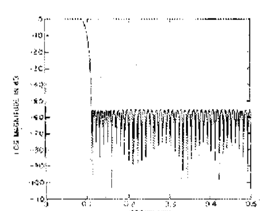
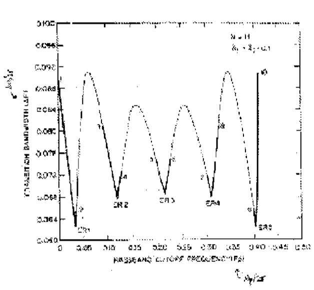


Figure 7.41 Dependence of transition width on cutoff frequency for optimum approximations of a lowpass filter.



Frequency response of optimal (minimax error) lowpass filter M=98. (Rabiner and Gold)



Transition bandwidth as a function of passband cutoff frequency for optimal lowpass filters, N = M + 1 (Rabiner and Gold)

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

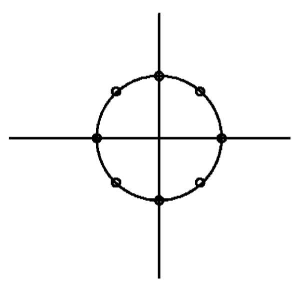
Lecture 18

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Sampling in Frequency

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

$$X(e^{j\omega_k}), \quad \omega_k = \frac{2\pi k}{N}$$



N = number of samples of DTFT

N samples of the DTFT

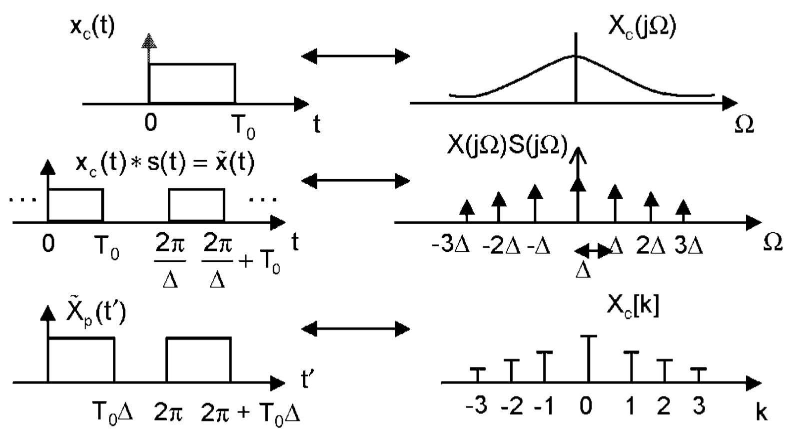
$$X[k] = X(e^{j\omega})\Big|_{\omega=\frac{2\pi k}{N}}$$

$$= \sum_{n=-\infty}^{\infty} x[n] e^{-j\frac{2\pi kn}{N}}$$

$$= \sum_{n=-\infty}^{\infty} x[n] W_N^{nk}, \quad W_N = e^{-j\frac{2\pi}{N}}$$

when does X[k] uniquely specify $X(e^{j\omega})$? (& ∴ x[n]?)

A Continuous-time Example



Fourier Series Coefficients

When are the FS sufficient?

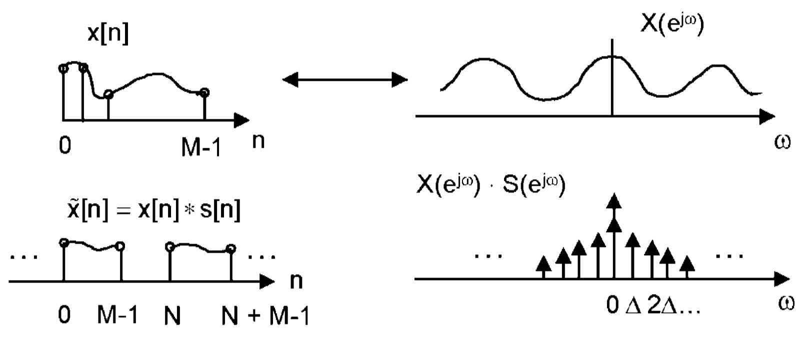
Can reconstruct iff $T_0\Delta < 2\pi$

$$\Delta < \frac{2\pi}{T_0}$$

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} X[k]e^{-j\frac{2\pi k}{T_0}t}$$

Need x(t) "time limited" to avoid aliasing in time.

Analogous Results / Discrete-Time

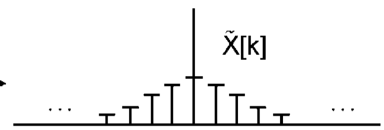


$$\frac{2\pi}{\Delta} = N \text{ equally-spaced samples}$$

Sufficient for recunstruction?

Need $\frac{2\pi}{\Delta} = N \geq M =$ length of x[n]

$\tilde{x}[n] \xleftrightarrow{\text{DFS}}$



Discrete Fourier Series

Invertible iff $\frac{2\pi}{\Delta} = N \geq M$ (avoid time-aliasing)

Discrete Fourier Series

Analysis equation

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\omega n} \Big|_{\omega = \frac{2\pi k}{N}}$$

N-periodic in k

Discrete Fourier Series Matrix

$$\begin{bmatrix} \tilde{X}[0] \\ \tilde{X}[1] \\ \tilde{X}[k] \\ \tilde{X}[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & & \cdots & W_N^0 \\ W_N^0 & W_N^1 & \cdots & W_N^{N-1} \\ \vdots & & & \\ W_N^0 & W_N^{N-1} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \tilde{x}[0] \\ \\ \vdots \\ \\ \tilde{x}[N-1] \end{bmatrix}$$

As a linear transformation

$$F[ij] = w_N^{ij}$$

$$\tilde{X} = F \ \tilde{x}$$

$$\tilde{x}[n] = F^{-1}\tilde{X}[k]$$

$$FF^{\#} = NI = F^{\#}F, \quad F^{\#} = F^{*}$$

(symmetric)

$$\tilde{x}[n] = \frac{1}{N}F^{\#}\tilde{X}[k]$$

Invertible linear
~ Unitary transformation
(not really) of
N-dimensional
vectorspace

Discrete Fourier Series

"Synthesis equations"

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-nk} = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{j\frac{2\pi k}{N} n}$$

weights ↑ ↑ basis functions (vectors)

Harmonically related

Properties of the DFS

Properties:

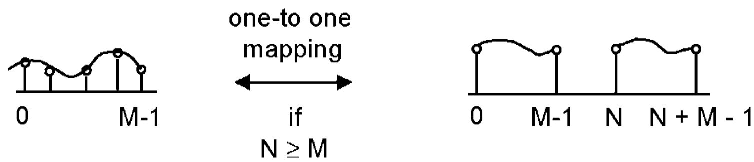
1. $\tilde{x}[n-m] \leftrightarrow e^{-j\frac{2\pi mk}{N}} \tilde{X}[k]$ shift

2. $e^{j\frac{2\pi \ell n}{N}} \tilde{x}[n] \leftrightarrow \tilde{X}[h-\ell]$ modulation

3. $\sum_{m=0}^{N-1} \tilde{x}[m]\tilde{y}[n-m] \leftrightarrow \tilde{X}[k]\tilde{Y}[k]$ periodic convolution

Finite Length Signals

Finite length vs. periodic signals



If N = M

→ Both have N degrees of freedom

Windowing a Periodic Signal

$$x[n] = \begin{cases} \tilde{x}[n], & 0 \le n \le N-1 \triangleq R_N[n]\tilde{x}[n] \\ 0, & \text{else} \end{cases}$$

↖ rect. window

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n + rN] \triangleq x[((n))_N]$$

Time aliasing by N

Discrete Fourier Transform (DFT)

The DFT of x[n] $\triangleq$ one period of DFS of $\tilde{x}[n]$

$$X[k] = \tilde{X}[k],\ k = 0, \ldots, N-1$$

Analysis

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \qquad 0 \le k \le N-1$$

Synthesis

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk} \qquad 0 \le n \le N-1$$

& not defined elsewhere or 0 by convention.

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 19

Professor Andrew Singer
Department of Electrical and Computer Engineering

Fourier Transform of Periodic Signals

$$\tilde{x}[n] \quad \text{periodic} \quad \rightarrow \quad \tilde{X}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \tfrac{2\pi}{N} \tilde{X}[k]\, \delta(\omega - \tfrac{2\pi k}{N})$$

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{j(2\pi/N)kn} = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] W_N^{kn}$$

How would you show this?

Discrete Fourier Transform (DFT)

The DFT of x[n] $\triangleq$ one period of DFS of $\tilde{x}[n]$

$$X[k] = \tilde{X}[k],\ k = 0, \ldots, N-1$$

Analysis

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \qquad 0 \le k \le N-1$$

Synthesis

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk} \qquad 0 \le n \le N-1$$

& not defined elsewhere or 0 by convention.

Relationship to the periodic signal

$$x[n] \leftrightarrow X(e^{j\omega})$$

$$\tilde{x}[n] \leftrightarrow \tilde{X}[k] = X(e^{j2\pi k/N})$$

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n + rN] \qquad \text{Periodic replication}$$

When does $\tilde{X}[k] = X(e^{j2\pi k/N})$ hold ?

Two cases to consider

Case 1:

$$x[n] = 0 \quad \text{outside} \quad 0 \le n \le N\text{-}1$$

$$\tilde{x}[n] = x[n] \qquad 0 \le n \le N-1$$

$$\tilde{x}[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}}$$

$$= X(e^{j\omega})\Big|_{\omega=\frac{2\pi k}{N}}$$

Case 2: x[n] Arbitrary Length

x[n] arbitrary length, consider $\tilde{X}[k]$:

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \overbrace{\left(\sum_{r=-\infty}^{\infty} x[n+rN] \right)}^{\tilde{x}[n]} e^{-j\frac{2\pi kn}{N}}$$

$$= \sum_{r=-\infty}^{\infty} \sum_{n=0}^{N-1} x[\underbrace{n+rN}_{m}] e^{-j\frac{2\pi kn}{N}}$$

$$= \sum_{r=-\infty}^{\infty} \sum_{rN}^{(r+1)N-1} x[m] e^{-j\frac{2\pi k(m-rN)}{N}}$$

Sampling in ω = Aliasing in time

$$= \sum_{r=-\infty}^{\infty} \sum_{m=rN}^{(r+1)N-1} x[m] e^{-j\frac{2\pi km}{N}}$$

$$= \sum_{m=-\infty}^{\infty} x[m] e^{-j\frac{2\pi km}{N}} = X(e^{j\omega})\Big|_{\omega=\frac{2\pi k}{N}}$$

$\therefore \tilde{X}[k] = X(e^{j\omega})\Big|_{\omega=\frac{2\pi k}{N}}$ independent of the length of x[n]!

When is x[n] recoverable?

$\rightarrow \tilde{x}[n]$ always recoverable from $\tilde{X}[k]$

$\rightarrow x[n]$ recoverable only if $x[n] = 0$ outside

$0 \leq n \leq N\text{-}1$

Time Aliasing

Suppose x[n] not finite length:

What is

$$\mathrm{IDFT}_N\left(X(e^{j2\pi k/N})R_N[k]\right)?$$

Answer:

$$= R_N[n]\sum_{r=-\infty}^{\infty} x[n+rN] = R_N[n]\tilde{x}[n]$$

Properties of DFT

$x_1[n] = N_1$ pt sequence

$x_2[n] = N_2$ pt sequence

$N > \max(N_1, N_2)$

$$x_1[n] \rightarrow X_1(e^{j\omega})$$

$$x_1[n-m] \rightarrow X_1(e^{j\omega})e^{-j\omega m} \rightarrow X_1[k]e^{-j\frac{2\pi km}{N}}$$

$$x_1[((n-m))_N] \leftarrow$$

Circular Convolution

$\Rightarrow X_1[k]X_2[k] \rightarrow$ one period of $\tilde{x}_1[n] \circledast \tilde{x}_2[n]$

$\Rightarrow$ circular convolution $x_1[n] \, \text{Ⓝ} \, x_2[n]$

Useful for:

- filtering
- spectral analysis

Summary

x[n] (arbitrary length) $\leftrightarrow$ $X(e^{j\omega})$ DTFT

DFS:

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n + rN] \leftrightarrow \tilde{X}[k] = X(e^{j\omega})\Big|_{\omega=\frac{2\pi k}{N}}$$

In general, x[n] not recoverable from

$\tilde{x}[n]$ or $\tilde{X}[k]$.

Recoverable?

If x[n] = 0, n < 0 and n > N – 1 then:

$$\tilde{x}[n] = x[((n))_N]$$

$$x[n] = \begin{cases} \tilde{x}[n] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

The DFS

DFS:

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-nk} \qquad W_N = e^{-j\frac{2\pi}{N}}$$

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{nk}$$

$$\tilde{x}[n-m] \leftrightarrow e^{-j\frac{2\pi}{N}km}\tilde{X}[k]$$

$$\sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m] \leftrightarrow \tilde{X}_1[k]\tilde{X}_2[k]$$

The Discrete Fourier Transform

IDFT:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-nk} \qquad 0 \le n \le N-1$$

DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} \qquad 0 \le k \le N-1$$

If x[n] not finite length, what is the IDFT of

$$\tilde{X}[k]R_N[k]?$$

Periodic convolution

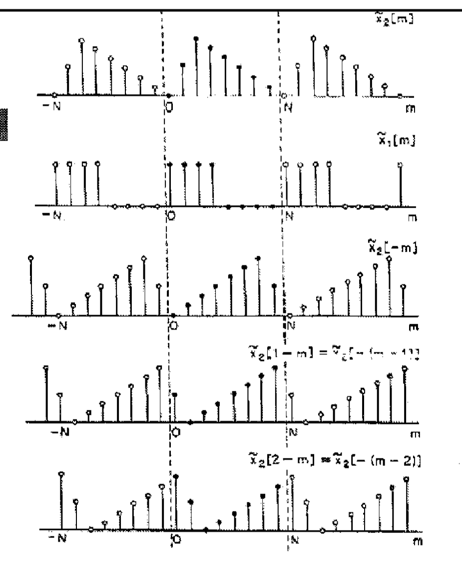


Figure 8.6 Procedure for forming the periodic convolution of two periodic sequences.

Circular shift

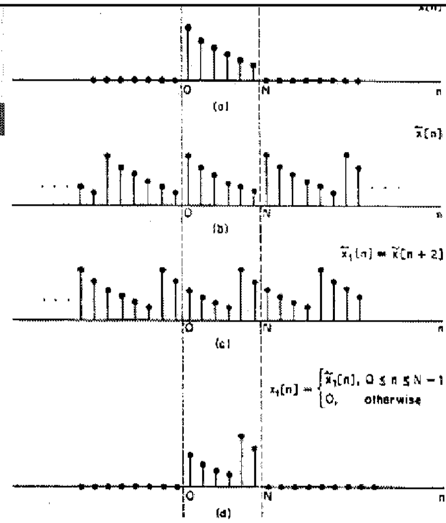


Figure 8.12 Circular shift of a finite-length sequence; i.e., the effect in the time domain of multiplying the DFT of the sequence by a linear phase factor.

Circular Convolution

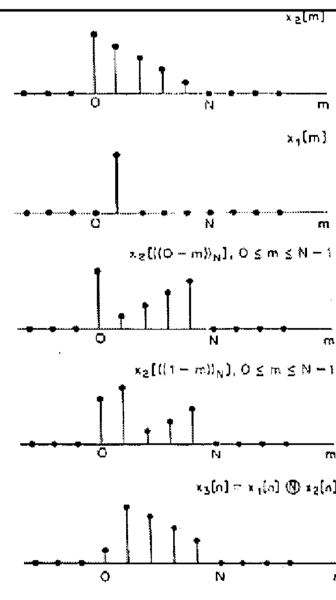


Figure 8.14 Circular convolution of a finite-length sequence $x_2[n]$ with a single delayed impulse.

Example

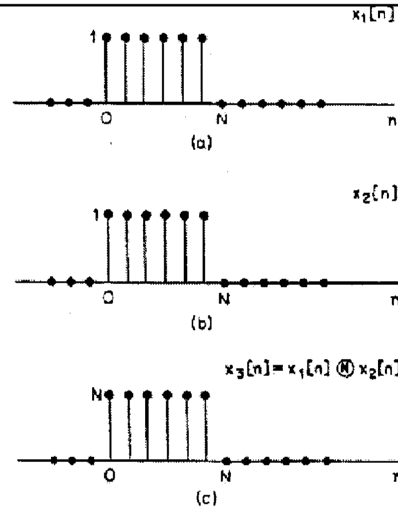


Figure 8.15 N-point circular convolution of two constant sequences of length N.

N Large enough

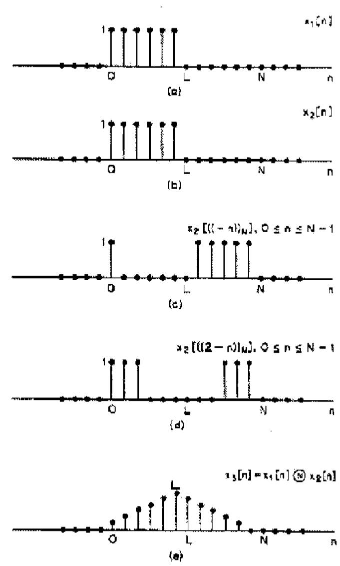
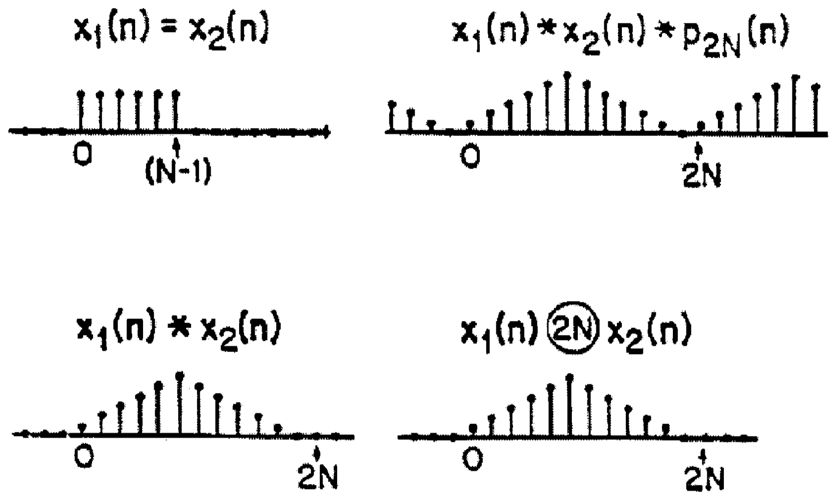
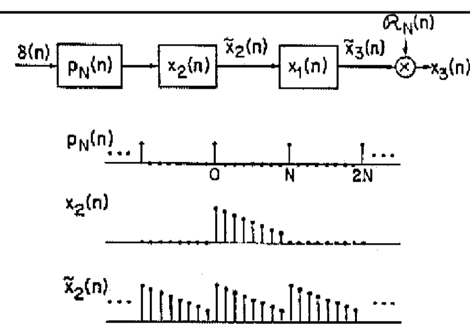


Figure 8.16 2L-point circular convolution of two constant sequences of length L.

Circular and Linear Convolution

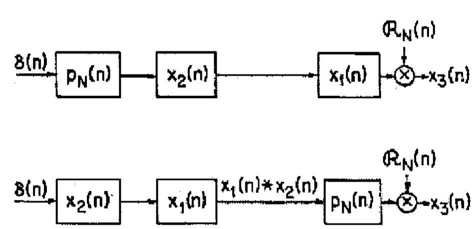


Circular Convolution??



Alias first?

Or convolve first?



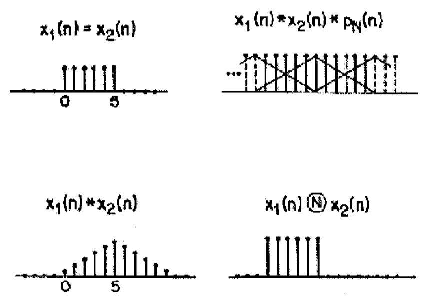
Convolve First:

"Circular Convolution =
Linear Convolution + Aliasing"

$$\hat{x}_3(n) = x_1(n) * x_2(n)$$

$$x_3(n) = x_1(n) \circledN x_2(n)$$

$$x_3(n) = \left[\sum_{r=-\infty}^{+\infty} \hat{x}_3(n+rN)\right]\mathcal{R}_N(n)$$



Fast Convolution

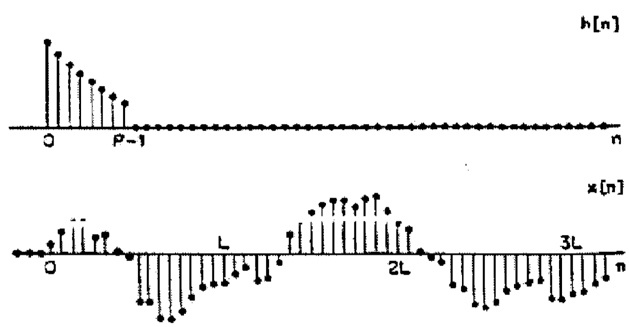


Figure 8.22

Overlap-Add

$x[n] = \sum_{k=0} x_k[n - kN]$ (long sequence)

$y[n] = \sum_{m=0}^{P} h[m]x[n - m]$ (length-P h[n])

$y[n] = \sum_{m=0}^{P} h[m] \sum_{k=0}^{\infty} x_k[n - m - kN]$

$$y[n] = \sum_{k=0}^{\infty} \underbrace{\sum_{m=0}^{P} h[m]x_k[n - m - kN]}_{h[n]*x_k[n-KN]}$$

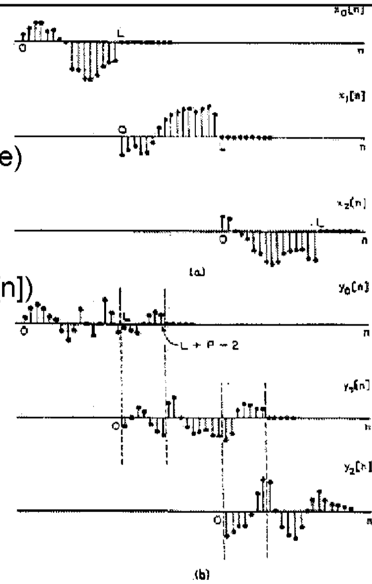


Figure 8.23

Overlap-Save

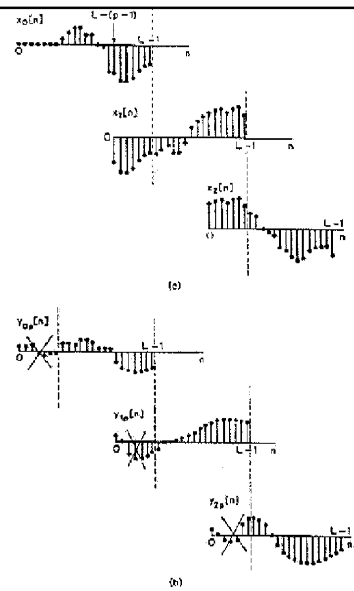
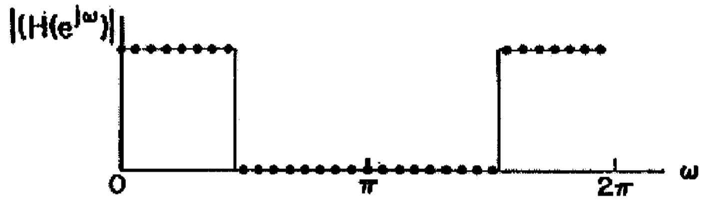


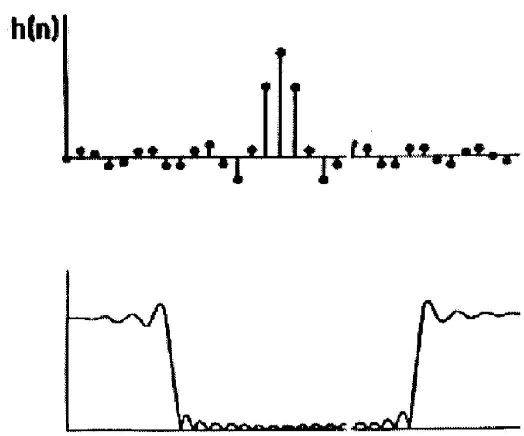
Figure 8.24

Recap on Frequency Sampling

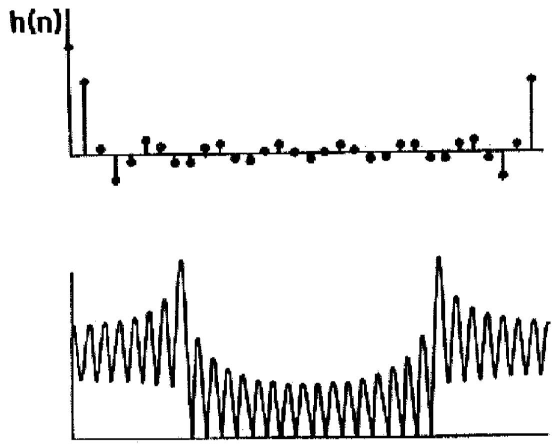


$$H(e^{j\omega_k}) = H_D(e^{j\omega_k}) \quad \omega_k = \frac{2\pi}{N}k \quad k = 0,1,\cdots,(N-1)$$

With Proper Linear Phase Term

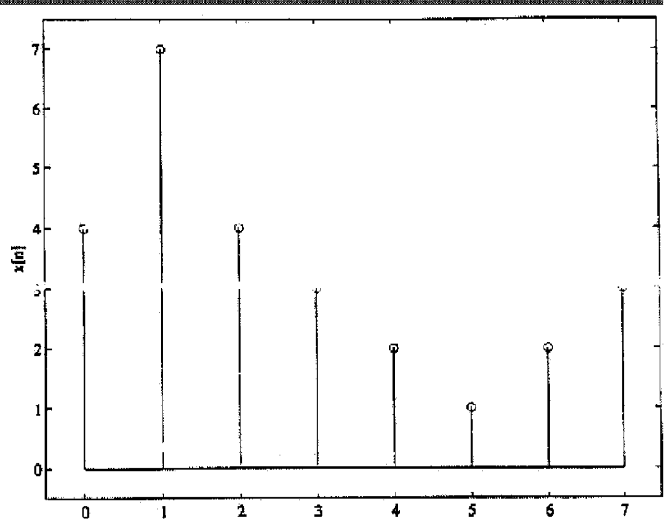


With Zero Phase Assumed, why?



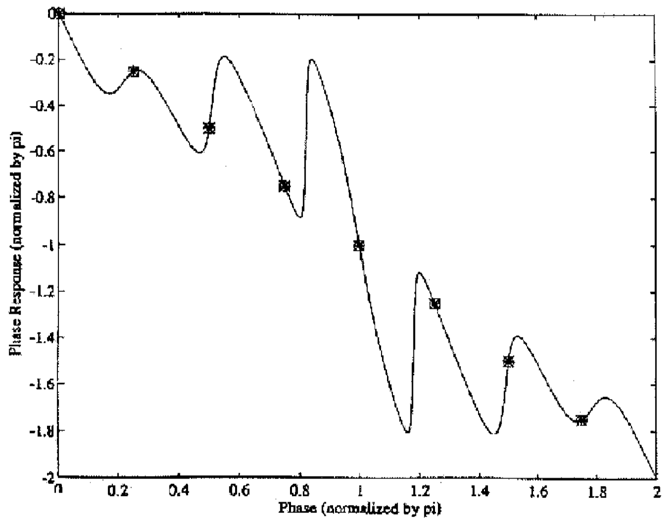
ECE 451 - Lecture 19

What will the phase look like?



ECE 451 - Lecture 19

Samples are Generalized Linear, why?



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 20

Professor Andrew Singer
Department of Electrical and Computer Engineering

Fast Fourier Transform Algorithm

Review FFT algorithms:

Computes the Discrete Fourier Transform

Decimation in Time/Decimation in Frequency

Direct Computation: $\theta(N^2)$

$$\text{Multiplies \& Adds} \begin{cases} 4N \text{ real mults} \\ 4N-2 \text{ real adds} \end{cases} \quad XN \cong 4N^2$$

$\rightarrow$ If all of them N lg N via FFT

Other Fast Algorithms

Q: What if you only want a few of them?
(i.e. << $\lg_2 N$ frequencies?)

Goertzel Algorithm: Reduces the constant of proportionality for direct computation

DFT: $$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, \qquad W_N = e^{-j\frac{2\pi}{N}}$$

Goertzel Algorithm (fun with W_N^{kN})

Note that:

$$W_N^{-kN} = e^{j\frac{2\pi kN}{N}} = e^{j2\pi k} = 1$$

$$\therefore X[k] = W_N^{-kN} \sum_{n=0}^{N-1} x[n] W_N^{nk} = \sum_{n=0}^{N-1} x[n] W_N^{-k(N-n)}$$

$$\Rightarrow X[k] = \left(x[n] * W_N^{-kn} \right)\Big|_{n=N}$$

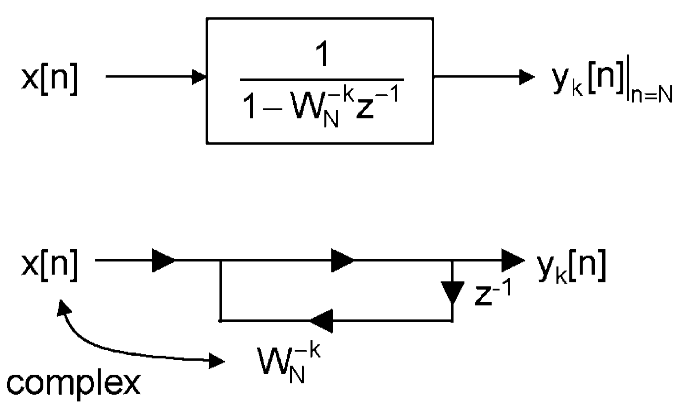
Implement DFT as a convolution

Compute:

$$y_k[n] = x[n] * g[n], \quad g[n] = W_N^{-kn} u[n]$$

$$X[k] = y_k[n]\big|_{n=N}$$

Block diagram



Complexity

$\Rightarrow$ Each $y_k[n]$ requires 4 Mults + 4 Adds

(Both 4s are labelled "real".)

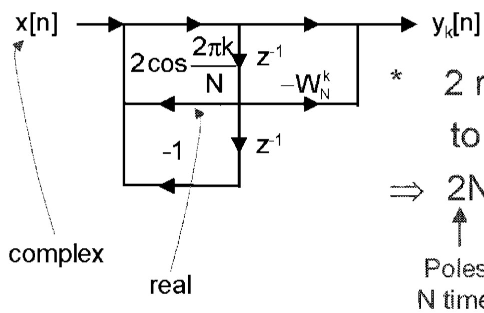
Need $y_k[1]$, $y_k[2]$, … $y_k[N]$ = 4N real MADS.

Manipulate the transfer function

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} = \frac{\left(1 - W_N^{k} z^{-1}\right)}{\left(1 - W_N^{-k} z^{-1}\right)\left(1 - W_N^{k} z^{-1}\right)}$$

$$= \frac{1 - W_N^{k} z^{-1}}{1 - z^{-1}\underbrace{\left(W_N^{-k} + W_N^{k}\right)}_{2\cos\frac{2\pi k}{N}} + z^{-2}}$$

Replace with Real poles



* 2 real mults/sample to implement poles

$\Rightarrow$ 2N + 4 = 2(N + 2) Mults

(↑ 2N: Poles N times; ↑ 4: zero once)

4(N + 1) real adds.

$\approx$ ½ direct computation

2 for the price of 1

Note we can also get X[N – k] by changing

$$W_N^k \to W_N^{-k}$$

$\Rightarrow$ ~2N mults + ~4N adds $\Rightarrow$ 2 DFT values, X[k] & X[N – k]

Implemented DFT as a convolution!

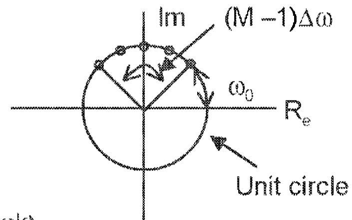
Chirp Transform Algorithm

Another method, more flexible, used/principle in SAR ⇒ Chip Transform

(CTA) = Chirp Transform Algorithm

x[n] length N

$X(e^{j\omega})$ = DTFT



Consider evaluating $X(e^{j\omega_k})$,

$\omega_k = \omega_0 + k\Delta\omega \quad k = 0, 1, \ldots, M - 1$

Chirp Transform

$$\text{e.g. } \omega_0 = 0, M = N, \Delta\omega = \frac{2\pi}{N} \Rightarrow \text{DFT samples.}$$

$$X(e^{j\omega_k}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega_k n}, \; k = 0, 1, \ldots, M-1$$

$$W \triangleq e^{-j\Delta\omega}, \qquad e^{-j\omega_k} = e^{-j\omega_0} W^k$$

$$X(e^{j\omega_k}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega_0 n} W^{nk}$$

(since $(k - n)^2 = k^2 - 2kn - n^2$)

$$nk = \tfrac{1}{2}[n^2 + k^2 - (k-n)^2]$$

CTA: Why are we doing this nk thing?

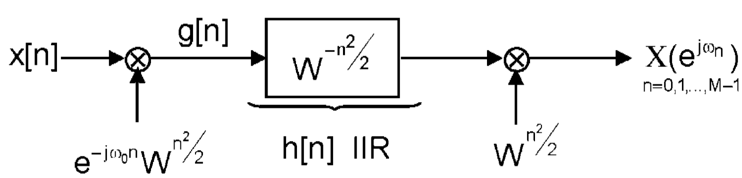
$$X(e^{j\omega_k}) = \sum_{n=0}^{N-1} \underbrace{x[n]e^{-j\omega_0 n}W^{n^2/2}}W^{k^2/2}W^{-(k-n)^2/2}$$

$$\text{put} \quad g[n] = x[n]e^{-j\omega_0 n}W^{n^2/2}$$

$$X(e^{j\omega_k}) = \left(\sum_{n=0}^{N-1} g[n]W^{-(k-n)^2/2}\right)W^{k^2/2}, k = 0, 1, \ldots, M-1$$

$$= W^{k^2/2}\left(g[n] * W^{-n^2/2}\right), \quad k = 0, 1, \ldots, M-1$$

Chirp, Filter, De-Chirp = DFT



CTA: do not need IIR h[n]

Note, only require

$$h[n] = \begin{cases} W^{-n^2/2}, & -[N-1] \le n \le M-1 \\ 0, & \text{else} \end{cases}$$

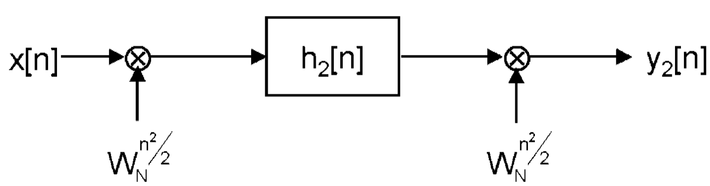
FIR is okay.

$$W^{n^2/2} = e^{-j\frac{\Delta\omega n^2}{2}} = e^{-j\left(\frac{\Delta\omega n}{2}\right)n}$$

Linear FM, or "Chirp"

CTA for computing the DFT:

To compute DFT, $\omega_0 = 0, \quad \Delta\omega = \frac{2\pi}{N}$



CTA-DFT

$$h_2[n] = \begin{cases} W_N^{-n^2/2}, & n = 1, 2, \ldots, 2N-1 \\ 0, & \text{else} \end{cases}$$

Can use FFT's (of length $L \geq 2N - 1$) to compute convolution:

$\Rightarrow$ For N = prime, can <u>still</u> use FFT's to get ~θ(NlgN) computational complexity!

Spiral contours, rather than arcs

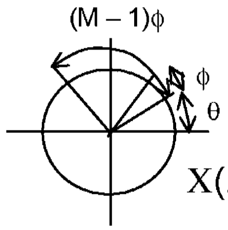
More general version Chirp Z-transform (CZT) (Rabiner et al. 1969)

Compute samples of

$$X(e^{j\omega})\Big|_{\text{spiral in z-plane}}$$

$$z_k = \underbrace{A_o e^{j\theta}}_{A} \underbrace{\left(W_0 e^{-j\phi}\right)^{-k}}_{W^{-k}}$$

Use the nk substitution again

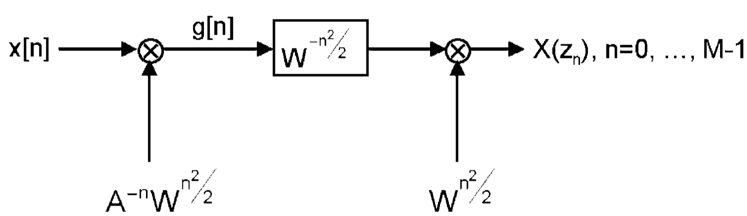


$$X(z_k) = \sum_{n=0}^{N-1} x[n]\left(A_0 e^{j\theta}\left(W_0 e^{-j\phi}\right)^{-k}\right)^{-n}$$

$$X(z_k) = \sum_{n=0}^{N-1} x[n] A^{-n} W^{nk}$$

$$= \sum_{n=0}^{N-1} x[n] A^{-n} W^{n^2/2} W^{k^2/2} W^{-(k-n)^2/2}$$

CTA Block Diagram



Decimation-in-frequency: Evens

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

k even $\qquad\qquad$ $k = 2r$

$$X(2r) = \sum_{n=0}^{N-1} x(n) \underbrace{W_N^{2nr}}_{\downarrow}$$

$$W_{N/2}^{nr}$$

Decimation-in-frequency: Evens

$$= \sum_{n=0}^{N/2-1} x(n) W_{N/2}^{nr} + \underbrace{\sum_{n=N/2}^{N-1} x(n) W_{N/2}^{nr}}_{\downarrow}$$

$$\sum_{n=0}^{N/2-1} x(n + \tfrac{N}{2}) W_{N/2}^{nr}$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n + \tfrac{N}{2}\right) \right] W_{N/2}^{nr}$$

$\frac{N}{2}$- point DFT of x(n) time-aliased

Decimation-in-frequency: Odds

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

k odd $\qquad k = 2r + 1$

$$X(2r+1) = \sum_{n=0}^{N-1} x(n) \underbrace{W_N^{n(2r+1)}}_{\downarrow}$$

$$W_N^n W_{N/2}^{nr}$$

Decimation-in-frequency: Odds

$$= \sum_{n=0}^{N/2-1} x(n) W_N^n W_{N/2}^{nr} + \underbrace{\sum_{n=N/2}^{N-1} x(n) W_N^n W_{N/2}^{nr}}$$

$$\sum_{n=0}^{N/2-1} x(n + \tfrac{N}{2}) W_N^{(n+N/2)} W_{N/2}^{nr}$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) W_N^n + x\left(n + \tfrac{N}{2}\right) W_N^{(n+N/2)} \right] W_{N/2}^{nr}$$

$\frac{N}{2}$- point DFT of $x(n)W_N^n$ time-aliased

Decimation-in-frequency, cont'd

even points

$$X(2r) = \sum_{n=0}^{N/2-1} \left[x(n) + x(n+\tfrac{N}{2})\right] W_{N/2}^{nr}$$

odd points

$$X(2r+1) = \sum_{n=0}^{N/2-1} \left[x(n)W_N^n + x(n+\tfrac{N}{2})W_N^{(n+N/2)}\right] W_{N/2}^{nr}$$

Decimation-in-frequency, cont'd

or

$$X(2r+1) = \sum_{n=0}^{N/2-1} \left[x(n) - x(n+\tfrac{N}{2})\right] W_N^n W_{N/2}^{nr}$$

because

$$W_N^{N/2} = e^{-j\pi} = -1$$

Decimation-in-frequency, cont'd

$$X(2r): \text{ alias then DFT: } \quad (\frac{N}{2})^2$$

$$X(2r+1): \text{ alias, modulate, then DFT: } (\frac{N}{2}) + (\frac{N}{2})^2$$

$$N^2 \rightarrow \frac{N}{2} + \frac{N^2}{2}$$

DIF FFT Example

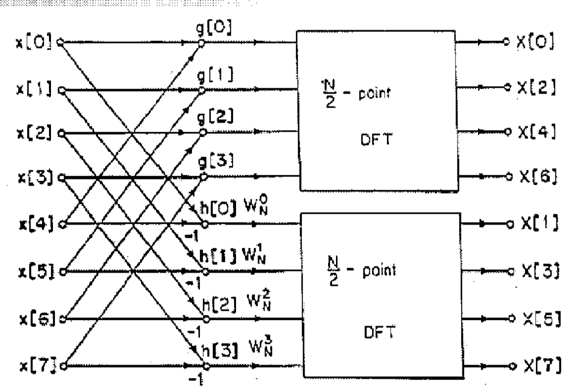
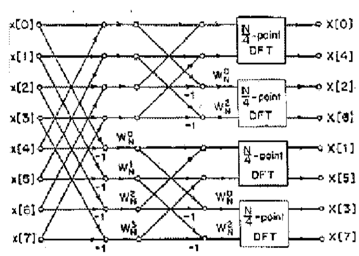


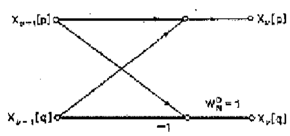
Figure 9.17

8-pt DFT: evens, odds

Example, cont'd

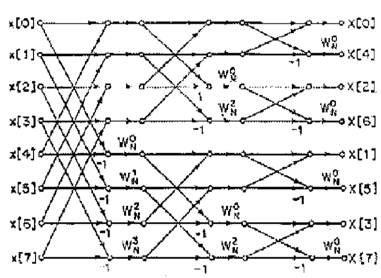


8-pt DFT: evens-evens, odds-odds

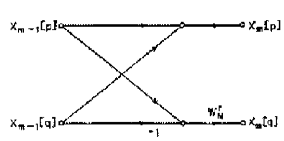


Single butterfly section

Example, cont'd

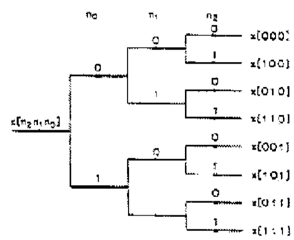


8-pt DIF DFT: complete butterfly flowgraph



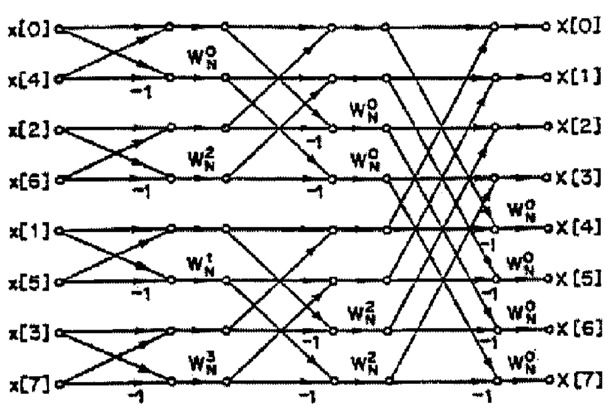
Ordering of the output

000 <= x[000]
001 <= x[100]
010 <= x[010]
011 <= x[110]
100 <= x[001]
101 <= x[101]
110 <= x[011]
111 <= x[111]



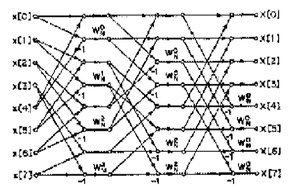
Bit-reversed order

Can manipulate flowgraph

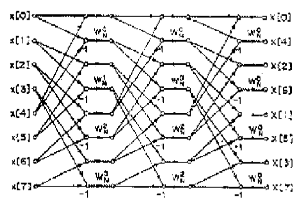


Bit-reversed input DIF FFT

Other (bad?) alternatives



Normal-order input/output



Fixed flowgraph stages

Decimation in Time

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$$

$$= \sum_{n \text{ even}} x[n] W_N^{nk} + \sum_{n \text{ odd}} x[n] W_N^{nk}$$

$$= \underbrace{\sum_{r=0}^{\frac{N}{2}-1} x[2r] W_{\frac{N}{2}}^{rk}}_{G[k]} + \underbrace{W_N^k \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_{\frac{N}{2}}^{rk}}_{H[k]}$$

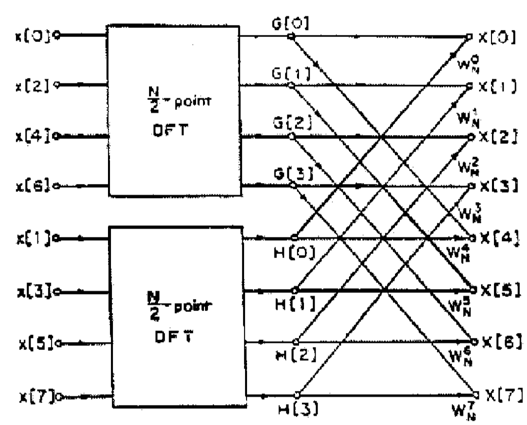
Decimation in Time, cont'd

$$X[k] = G[k] + W_N^k H[k]$$

$$G[k], H[k] \Rightarrow \tfrac{N}{2}\text{- point DFT's } \Rightarrow \text{ periodic}$$

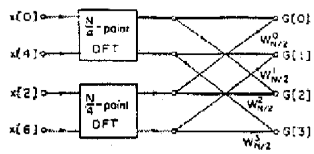
$$\left(\tfrac{N}{2}\right)^2 \times 2 + N = \tfrac{N^2}{2} + N$$

Decimation in Time, cont'd

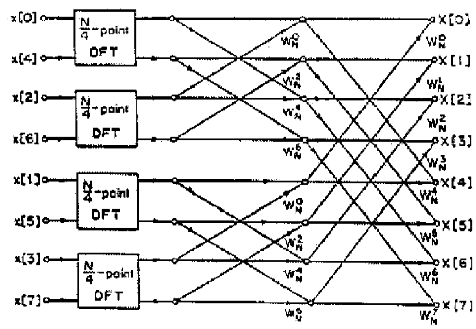


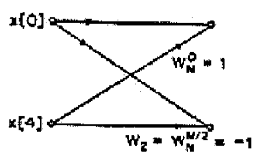
DFT of evens, DFT of odds, then (modulate and) alias

Decimation in Time, cont'd



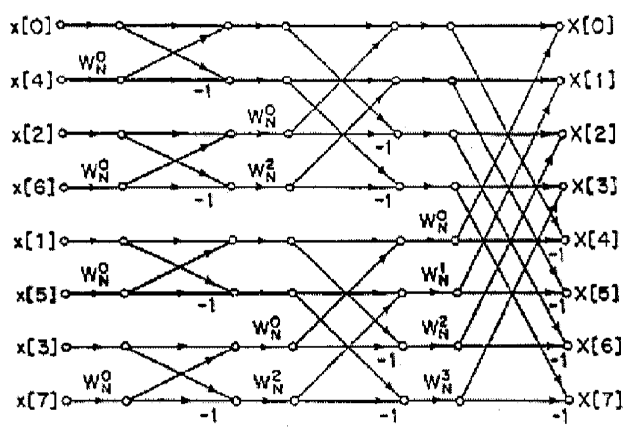
N/2 point as 2 N/4 points





N/4 point as 2 N/8 points

Decimation in Time, cont'd



8 point DIT FFT flowgraph

Efficient Convolution

Fourier Transform

Multiply

Inverse Transform

Efficient Fourier Transform

Fast Fourier Transform (FFT)

or Formulate as a convolution

chirp transform

Winograd Fourier transform

Efficient Convolution

Fourier transform

Multiply

Inverse Fourier transform

but:

Implement Fourier transform and

inverse as a convolution

And:

Do those convolutions using FFTs!

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 21

Professor Andrew Singer
Department of Electrical and Computer Engineering

Decimation-in-frequency: Evens

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

k even $\qquad\qquad$ $k = 2r$

$$X(2r) = \sum_{n=0}^{N-1} x(n) \underbrace{W_N^{2nr}}_{\downarrow}$$

$$W_{N/2}^{nr}$$

Decimation-in-frequency: Evens

$$= \sum_{n=0}^{N/2-1} x(n)W_{N/2}^{nr} + \underbrace{\sum_{n=N/2}^{N-1} x(n)W_{N/2}^{nr}}$$

$$\downarrow$$

$$\sum_{n=0}^{N/2-1} x(n+\tfrac{N}{2})W_{N/2}^{nr}$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n+\tfrac{N}{2}\right)\right]W_{N/2}^{nr}$$

$\frac{N}{2}$- point DFT of x(n) time-aliased

Decimation-in-frequency: Odds

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}$$

k odd $\qquad$ $k = 2r + 1$

$$X(2r+1) = \sum_{n=0}^{N-1} x(n)\underbrace{W_N^{n(2r+1)}}$$

$$\downarrow$$

$$W_N^n W_{N/2}^{nr}$$

Decimation-in-frequency: Odds

$$= \sum_{n=0}^{N/2-1} x(n) W_N^n W_{N/2}^{nr} + \underbrace{\sum_{n=N/2}^{N-1} x(n) W_N^n W_{N/2}^{nr}}_{\sum_{n=0}^{N/2-1} x(n+\frac{N}{2}) W_N^{(n+N/2)} W_{N/2}^{nr}}$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) W_N^n + x\left(n+\tfrac{N}{2}\right) W_N^{(n+N/2)} \right] W_{N/2}^{nr}$$

$\frac{N}{2}$- point DFT of $x(n)W_N^n$ time-aliased

Decimation-in-frequency, cont'd

even points

$$X(2r) = \sum_{n=0}^{N/2-1} \left[x(n) + x(n+\tfrac{N}{2}) \right] W_{N/2}^{nr}$$

odd points

$$X(2r+1) = \sum_{n=0}^{N/2-1} \left[x(n) W_N^n + x(n+\tfrac{N}{2}) W_N^{(n+N/2)} \right] W_{N/2}^{nr}$$

Decimation-in-frequency, cont'd

or

$$X(2r+1) = \sum_{n=0}^{N/2-1} \left[x(n) - x(n+\tfrac{N}{2}) \right] W_N^n W_{N/2}^{nr}$$

because

$$W_N^{N/2} = e^{-j\pi} = -1$$

Decimation-in-frequency, cont'd

$X(2r)$: alias then DFT: $(\frac{N}{2})^2$

$X(2r+1)$: alias, modulate, then DFT: $(\frac{N}{2}) + (\frac{N}{2})^2$

$$N^2 \rightarrow \frac{N}{2} + \frac{N^2}{2}$$

DIF FFT Example

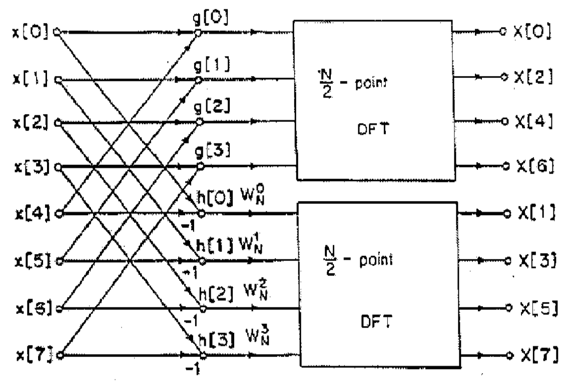
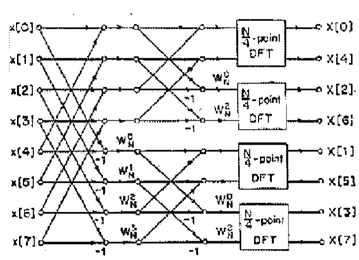


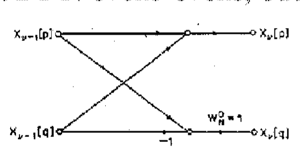
Figure 9.17

8-pt DFT: evens, odds

Example, cont'd

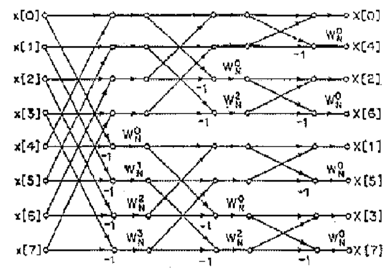


8-pt DFT: evens-evens, odds-odds

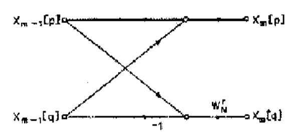


Single butterfly section

Example, cont'd

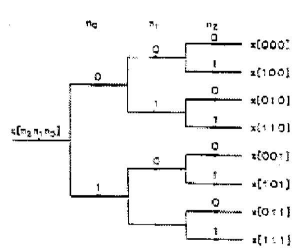


8-pt DIF DFT: complete butterfly flowgraph



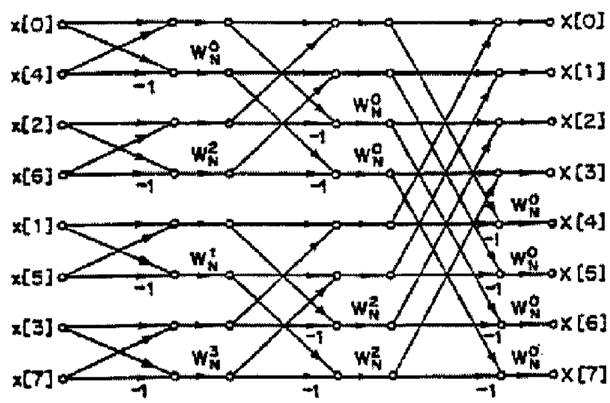
Ordering of the output

000 <= x[000]
001 <= x[100]
010 <= x[010]
011 <= x[110]
100 <= x[001]
101 <= x[101]
110 <= x[011]
111 <= x[111]



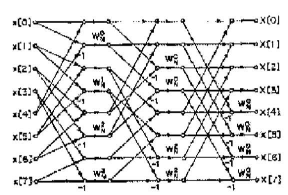
Bit-reversed order

Can manipulate flowgraph

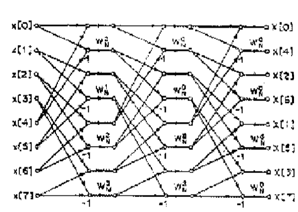


Bit-reversed input DIF FFT

Other (bad?) alternatives



Normal-order input/output



Fixed flowgraph stages

Decimation in Time

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk}$$

$$=\sum_{n\text{ even}}x[n]W_N^{nk}+\sum_{n\text{ odd}}x[n]W_N^{nk}$$

$$=\underbrace{\sum_{r=0}^{\frac{N}{2}-1}x[2r]W_{\frac{N}{2}}^{rk}}_{G[k]}+W_N^k\underbrace{\sum_{r=0}^{\frac{N}{2}-1}x[2r+1]W_{\frac{N}{2}}^{rk}}_{H[k]}$$

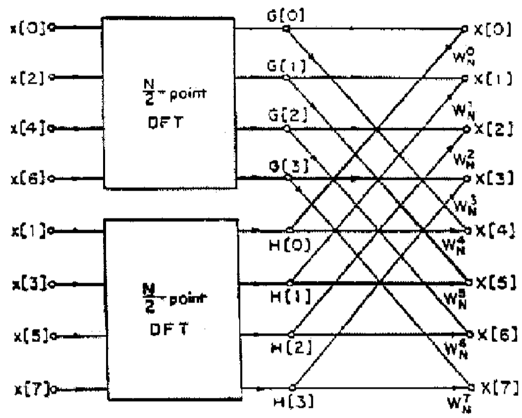
Decimation in Time, cont'd

$$X[k]=G[k]+W_N^kH[k]$$

$G[k], H[k] \Rightarrow \frac{N}{2}$- point DFT's $\Rightarrow$ periodic

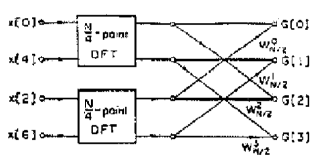
$$\left(\tfrac{N}{2}\right)^2\times 2+N=\tfrac{N^2}{2}+N$$

Decimation in Time, cont'd

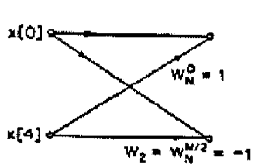


DFT of evens, DFT of odds, then (modulate and) alias

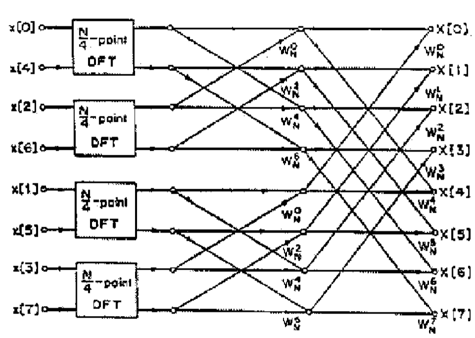
Decimation in Time, cont'd



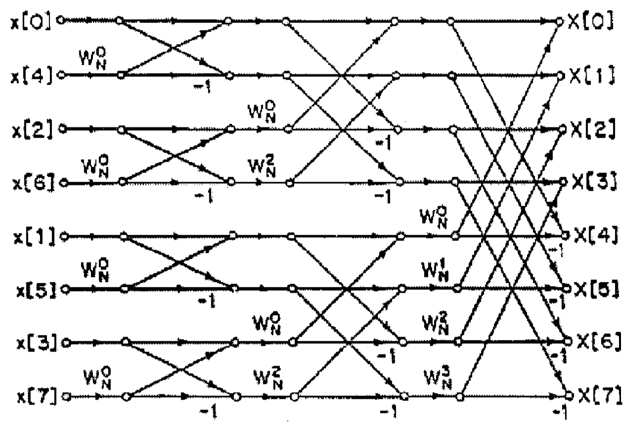
N/2 point as 2 N/4 points



N/4 point as 2 N/8 points



Decimation in Time, cont'd



8 point DIT FFT flowgraph

Efficient Convolution

Fourier Transform

Multiply

Inverse Transform

Efficient Fourier Transform

Fast Fourier Transform (FFT)

or Formulate as a convolution

chirp transform

Winograd Fourier transform

Efficient Convolution

Fourier transform

Multiply

Inverse Fourier transform

but:

Implement Fourier transform and

inverse as a convolution

And:

Do those convolutions using FFTs!

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 22

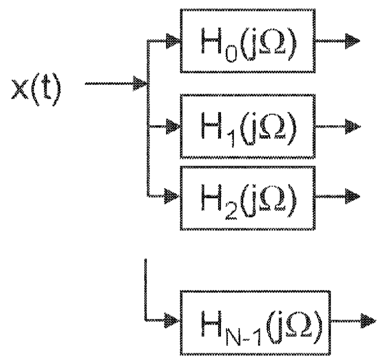
Professor Andrew Singer
Department of Electrical and
Computer Engineering

Spectral Analysis/ Estimation

Used for:
- Detection
- System Identification
- Time Series Analysis
- Coding
- Channel Modeling/Equalization
- Weiner Filtering/Matched Filtering

Filterbank Viewpoint

x(t) ⟶ Filterbank

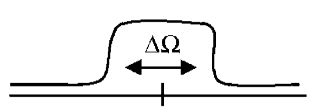


"Analysis Bank"

Examples

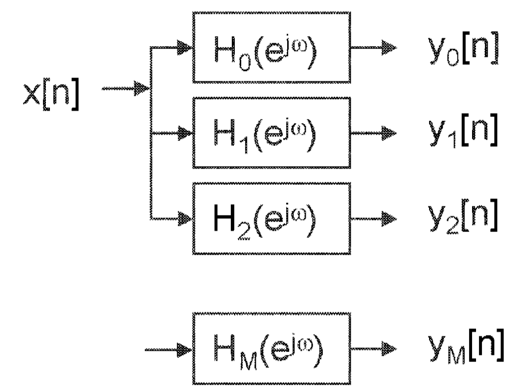
" Constant BW": $\Delta\Omega$ fixed

"Constant Q" $\frac{\Delta\Omega_k}{\Omega_k}$ fixed



Spectral Analysis, DT

DT Filterbanks:



DT Spectral Analysis: Prototype Filters

$$H_k(e^{j\omega}) = H_0\left(e^{j(\omega - \frac{2\pi k}{m})}\right)$$

$$h_k[n] = h_0[n]e^{j\omega_k n}$$

$$y_k[n] = \sum_{r=-\infty}^{\infty} h_k[n-r]x[r] = \sum_{r=-\infty}^{\infty} h_0[n-r]e^{j\omega_k (n-r)}x[r]$$

Prototype Filters, cont'd

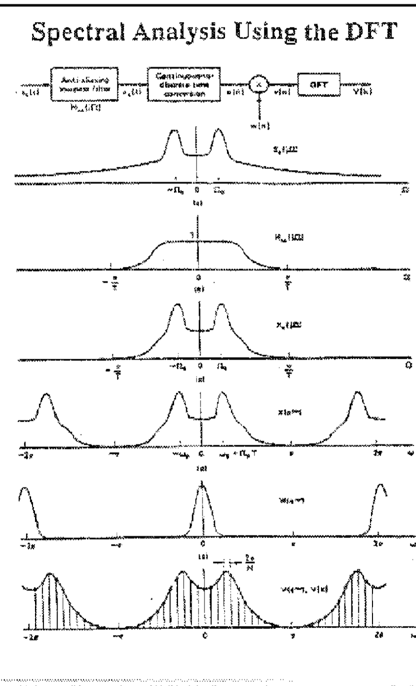
$$y_k[n] = e^{j\omega_k n} \sum_{r=-\infty}^{\infty} h_0[n-r]x[r]e^{-j\omega_k r}$$

$$= e^{j\omega_k n} \cdot \mathrm{DTFT}(h_0[n-r]x[r])_{w=\omega_k}$$

window signal

$$|y_k[n]| = |\mathrm{DTFT}(h_0[n-r]x[r])|_{w=\omega_k}$$

Using DFT



Example

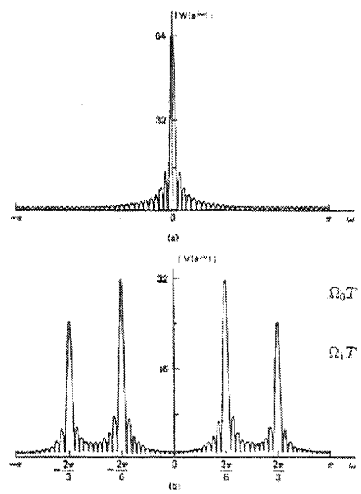
Text Example 11.1

$1/T = 10kHz$ w[n] rectangular window length 64

$T = 10^{-4}$

$s_c(t) = \cos \Omega_0 t + 0.75 \cos \Omega_1 t$

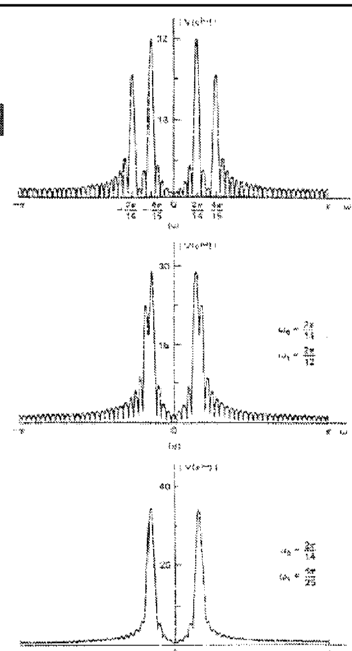
$x[n] = \cos \underbrace{(\Omega_0 T)}_{\omega_0} n + 0.75 \cos \underbrace{(\Omega_1 T)}_{\omega_1} n$



$\Omega_0 T = \frac{2\pi}{6}$

$\Omega_1 T = \frac{2\pi}{3}$

Example, cont'd



Fences

The "Picket Fence" Effect

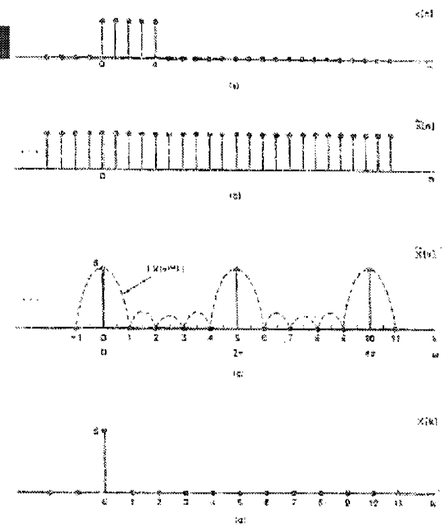


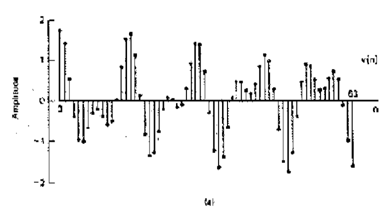
Figure 8.10

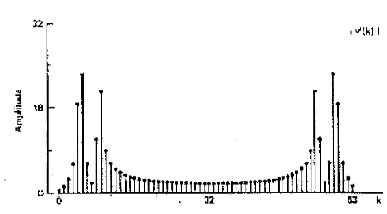
Example

Text Example 11.2

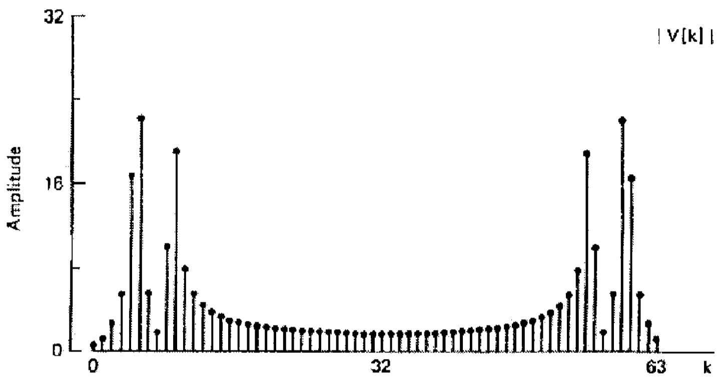
$x[n] = \cos\left(\frac{2\pi}{14}\right)n + 0.75\cos\left(\frac{4\pi}{15}\right)n$

$w[n]$ = 64-point rectangular window



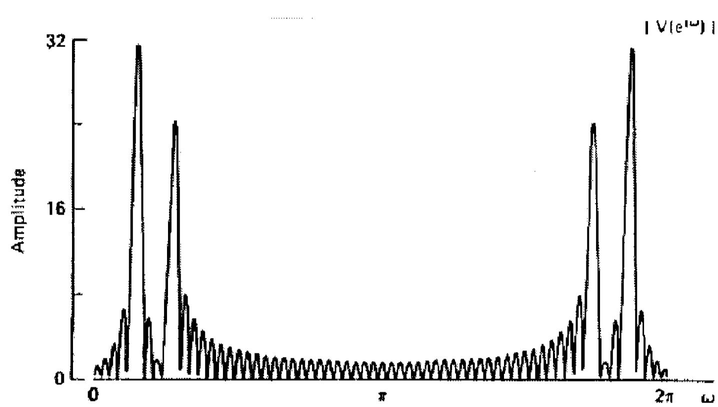


DFT of the windowed signal



(d)

DTFT of the windowed Signal

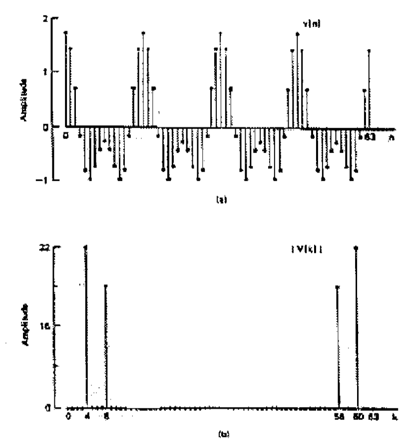


(f)

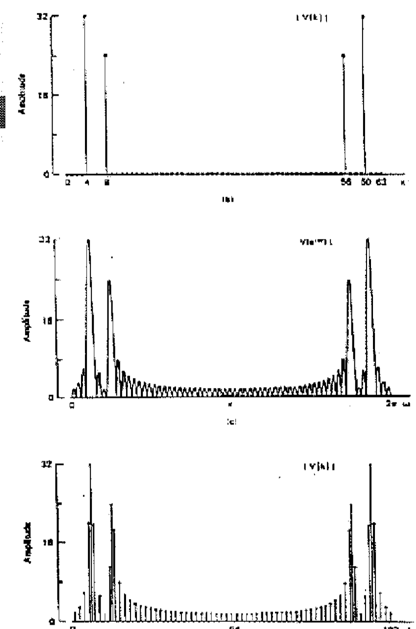
Again

Text Example 11.3

$$v[n] = \begin{cases} \cos\left(\frac{2\pi}{16}n\right) + 0.75\cos\left(\frac{2\pi}{8}n\right), & 0 \le n \le 63, \\ 0, & \text{otherwise}, \end{cases} \quad (11.16)$$



Picket Fence



Chirp Signals, sliding window DFT

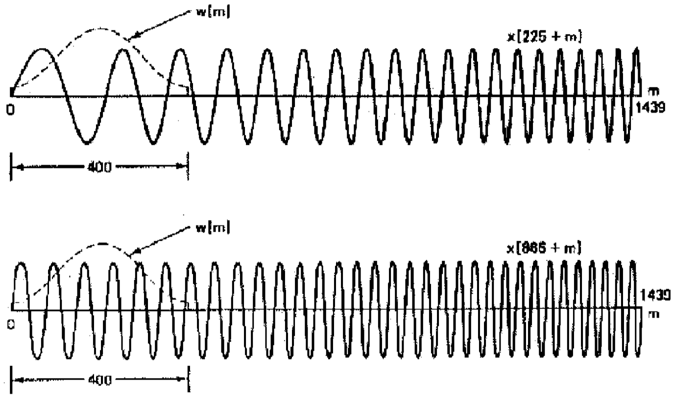


Figure 11.11 Two segments of the linear chirp signal $x[n] = \cos(2\pi \times 14 \times 10^{-6})n^2$ with the window superimposed.

Time-Frequency Plot for Chirp

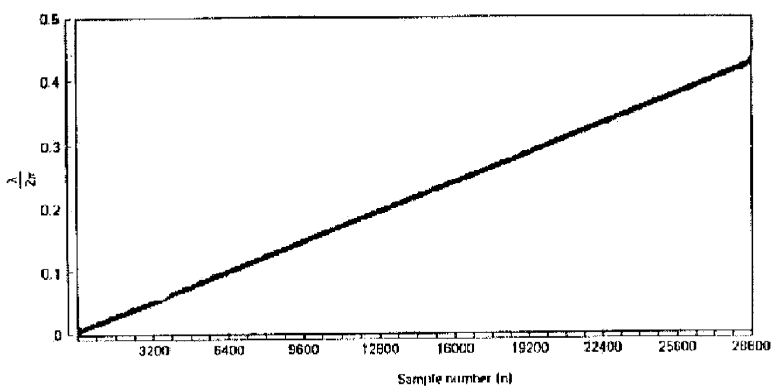


Figure 11.12 The magnitude of the time-dependent Fourier transform of $x[n] = \cos(2\pi \times 14 \times 10^{-6})n^2$ using a Hamming window of length 400.

Spectrograms for Speech

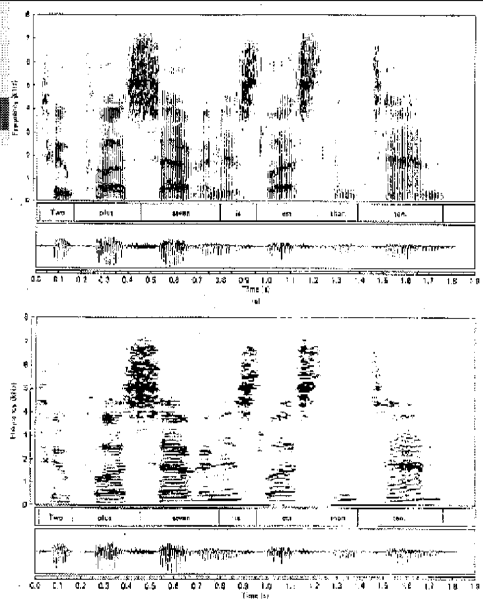


Figure 11.17 (a) Wideband spectrogram of waveform of Fig. 11.16. (b) Narrowband spectrogram.

Spectral Analysis of Random Signals

- WSS Random process x[n], with power spectral density $S_{xx}(e^{j\omega})$, x(t) BL with psd $S_{xx}(j\Omega)$
- How can we find $S_{xx}(e^{j\omega})=S_{xx}(j\omega/T)$
- $v[n]=R_L[n]x[n]$ (length L rectangular window)
- Compute $I(\omega)=(1/LU)\ |V(e^{j\omega})|^2$
- "Periodogram" approach

Spectral Estimation

Is $I(\omega)$ a "good" estimate of $S_{xx}(e^{j\omega})$?

$$E\{I(\omega)\} = E\left\{\frac{1}{LU}\left|V(e^{j\omega})\right|^2\right\} \underbrace{=}_{\text{easy to show}} \frac{1}{LU} S_{xx}(e^{j\omega}) * \left|W(e^{j\omega})\right|^2$$

$$\left|V(e^{j\omega})\right|^2 = V(e^{j\omega})V(e^{j\omega})^* = \sum_{m=-(L-1)}^{L-1} v[n] * v[-n]e^{-j\omega n}$$

$$= \sum_{n=-(L-1)}^{L-1} \underbrace{\left(\sum_{k=0}^{L-1} v[k-n]v[k]\right)}_{C_{vv}[n]} e^{-j\omega n}$$

Easy to Show

$$I(\omega) = \frac{1}{LU} \sum_{m=-(L-1)}^{L-1} C_{vv}[m]e^{-j\omega m}$$

$$C_{vv}[m] = \sum_{n=0}^{L-1} x[n]w[n]x[n+m]w[n+m]$$

$$= x[n]w[n] * x[-n]w[-n]$$

Use Deterministic Correlations

$$E[I(\omega)] = \frac{1}{LU}\sum_{-(L-1)}^{L-1} E(C_{vv}[m])e^{-j\omega m}$$

$$E(C_{vv}[m]) = \sum_{n=0}^{L-1} w[n]w[n+m]E(x[n]x[n+m])$$

$(C_{ww}[m] = w[n] * w[-n])$

$$= C_{ww}[m]R_{xx}[m]$$

for $R_L[n],\ C_{ww}[m] = \text{Bartlett window}$

By FT Modulation Property…

$$\therefore E\{I(\omega)\} = \frac{1}{2\pi LU}\int_{-\pi}^{\pi} S_{xx}(e^{j\theta})P_{ww}(e^{j(\omega-\theta)})d\theta$$

$$P_{ww}(e^{j\omega}) = \left|W(e^{j\omega})\right|^2$$

Biased!

Since

$$E(I(\omega)) = \frac{1}{LU} S_{xx}(e^{j\omega}) * \left|W(e^{j\omega})\right|^2 \neq S_{xx}(e^{j\omega})$$

$\Rightarrow$ Biased Estimate

as $L \rightarrow \infty \quad \left|W(e^{j\omega})\right|^2 \Rightarrow L\delta(\omega)$

Asymptotically Unbiased

If choose U s.t.

$$\frac{1}{2\pi LU} \int_{-\pi}^{\pi} \left|W(e^{j\omega})\right|^2 d\omega = 1$$

–or – $\quad U = \frac{1}{L} \sum_{n=0}^{L-1} (w[n])^2$

then $L \rightarrow \infty \quad I(\omega) \rightarrow S_{xx}(e^{j\omega})$

$\Rightarrow$ Asympotically unbiased

But is it good?

For $R_L[n] \Rightarrow U = 1$

Okay, it is unbiased, is it a “good” estimate?

$\Rightarrow$ Look at variance

is it consistent?

i.e. does variance $\to 0$ as $L \to \infty$

Variance doesn’t vanish

ICB shown (hard to do)

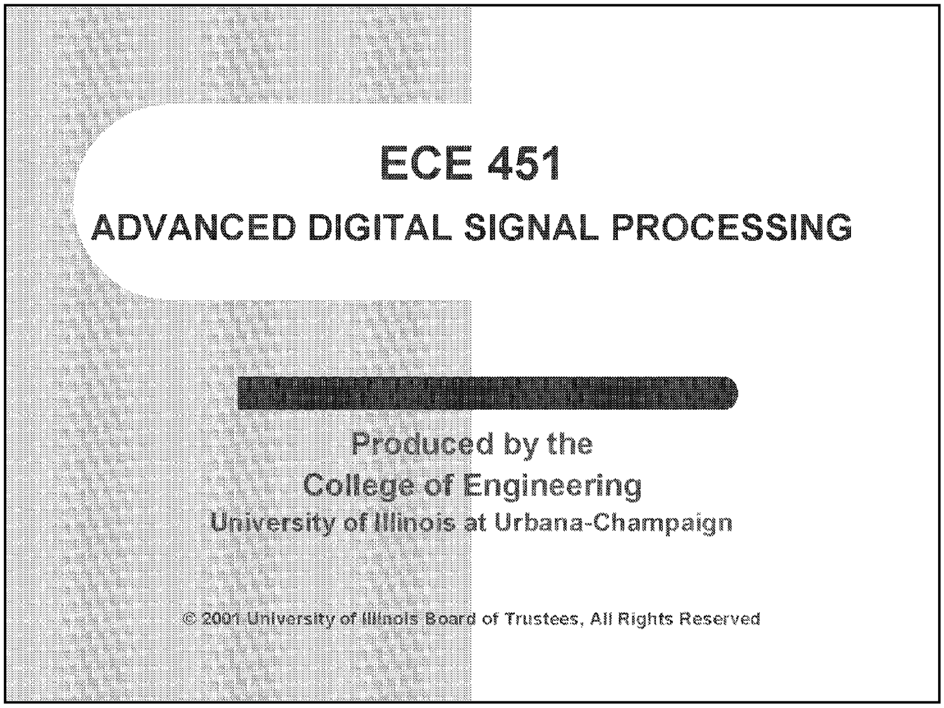
$$\mathrm{Var}\{I(\omega)\} \simeq (S_{xx}(e^{j\omega}))^2$$

$\Rightarrow$ does not go to zero

$\Rightarrow$ not consistent

$L \to \infty \to$ longer DFT $\to$ doesn’t get any better

Periodogram is Not a “good” estimate.


ECE 451
ADVANCED DIGITAL SIGNAL PROCESSING
Produced by the
College of Engineering
University of Illinois at Urbana-Champaign
© 2001 University of Illinois Board of Trustees, All Rights Reserved

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 23

Professor Andrew Singer
Department of Electrical and Computer Engineering

Spectral Analysis of Random Signals

- WSS Random process:
  - x[n] with power spectral density $S_{xx}(e^{j\omega})$
  - x(t) bandlimited with PSD $S_{xx}(j\Omega)$
- How can we find $S_{xx}(e^{j\omega})=S_{xx}(j\omega/T)$?
- $v[n]=R_L[n]x[n]$ (length L rectangular window)
- Compute $I(\omega)=(1/LU)\,|V(e^{j\omega})|^2$
- "Periodogram" approach

Spectral Estimation

Is $I(\omega)$ a "good" estimate of $S_{xx}(e^{j\omega})$?

$$E\{I(\omega)\} = E\left\{\frac{1}{LU}\left|V(e^{j\omega})\right|^2\right\} \underbrace{=}_{\text{easy to show}} \frac{1}{LU} S_{xx}(e^{j\omega}) * \left|W(e^{j\omega})\right|^2$$

$$\left|V(e^{j\omega})\right|^2 = V(e^{j\omega})V(e^{j\omega})^* = \sum_{m=-(L-1)}^{L-1} (v[n] * v[-n]) e^{-j\omega n}$$

$$= \sum_{n=-(L-1)}^{L-1} \underbrace{\left(\sum_{k=0}^{L-1} v[k-n]v[k]\right)}_{C_{vv}[n]} e^{-j\omega n}$$

Easy to Show

$$I(\omega) = \frac{1}{LU} \sum_{m=-(L-1)}^{L-1} C_{vv}[m] e^{-j\omega m}$$

$$C_{vv}[m] = \sum_{n=0}^{L-1} x[n]w[n]x[n+m]w[n+m]$$

$$= x[n]w[n] * x[-n]w[-n]$$

Use Deterministic Correlations

$$E[I(\omega)] = \frac{1}{LU} \sum_{-(L-1)}^{L-1} E(C_{vv}[m]) e^{-j\omega m}$$

$$E(C_{vv}[m]) = \sum_{n=0}^{L-1} w[n]w[n+m]E(x[n]x[n+m])$$

$$(C_{ww}[m] = w[n] * w[-n])$$

$$= C_{ww}[m]R_{xx}[m]$$

for $R_L[n]$, $C_{ww}[m] = \text{Bartlett window}$

By FT Modulation Property…

$$E\{I(\omega)\} = \frac{1}{LU} \sum_{m=-(L-1)}^{L-1} C_{ww}[m]R_{xx}[m]e^{-j\omega m}$$

$$= \frac{1}{LU2\pi} \int_{-\pi}^{\pi} S_{xx}(e^{j\theta}) P_{ww}(e^{j(\omega-\theta)}) d\theta$$

$$= \frac{1}{LU} S_{xx}(e^{j\omega}) * \left| W(e^{j\omega}) \right|^2$$

Biased!

Since

$$E(I(\omega)) = \tfrac{1}{LU} S_{xx}(e^{j\omega}) * \left|W(e^{j\omega})\right|^2 \neq S_{xx}(e^{j\omega})$$

$\Rightarrow$ Biased Estimate

as $L \to \infty \quad \left|W(e^{j\omega})\right|^2 \Rightarrow L\delta(\omega)$

Asymptotically Unbiased

If choose U s.t.

$$\frac{1}{2\pi LU}\int_{-\pi}^{\pi}\left|W(e^{j\omega})\right|^2 d\omega = 1$$

–or – $\quad U = \frac{1}{L}\sum_{n=0}^{L-1}(w[n])^2$

then $L \to \infty \quad I(\omega) \to S_{xx}(e^{j\omega})$

$\Rightarrow$ Asymptotically unbiased

But is it good?

For $R_L[n] \Rightarrow U = 1$

Okay, it is asymptotically unbiased,

but is it a "good" estimate?

$\Rightarrow$ Look at the variance: is it consistent?

i.e. does variance $\rightarrow 0$ as $L \rightarrow \infty$

Variance does not vanish

It can be shown that

$$\mathrm{Var}\{I(\omega)\} \simeq (S_{xx}(e^{j\omega}))^2$$

$\Rightarrow$ does not go to zero

$\Rightarrow$ not consistent

$L \rightarrow \infty \rightarrow$ longer DFT $\rightarrow$ doesn't get any better

Periodogram is not a "good" estimate.

Ergodic Assumption

Assume also that the signal is **ergodic** in mean & variance

$\rightarrow$ ensemble averages = time averages

$$g[n] : \text{mean } m_g : \hat{m}_g = \tfrac{1}{L}\sum_{0}^{L-1} g[n] = \text{"sample mean"}$$

$$\text{ergodic} : \hat{m}_g \xrightarrow{L\rightarrow\infty} m_g \text{ , similarly for variance}$$

Sample Mean

$$E(\hat{m}_g) = m_g \quad \text{unbiased}$$

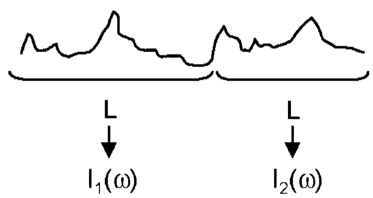
$$\text{var}(\hat{m}_g) \rightarrow 0 \text{ as } L \rightarrow \infty \text{ (consistent)}$$

$$\text{if } g[n] \text{ white} : \text{var}(\hat{m}_g) = \frac{1}{L}\sigma_g^2$$

Periodogram Averaging

$$E(I(\omega)) = \tfrac{1}{LU} S_{xx}(e^{j\omega}) * \underbrace{\left|W(e^{j\omega})\right|^2}_{S_{WW}(e^{j\omega})}$$

$$\text{define} \quad \hat{I}(\omega) = \tfrac{1}{K}\sum_{i=1}^{K} I_i(\omega)$$



Averaging Reduces Variance

"Periodogram Averaging"

$$E(\hat{I}(\omega)) = E(I_i(\omega)) = S_{xx}(e^{j\omega}) * S_{WW}(e^{j\omega})\tfrac{1}{LU}$$

$\rightarrow$ biased, but can be asymptotically unbiased.

$$\text{Var}\left[\hat{I}(\omega)\right] = \tfrac{1}{K}\text{Var}[I_i(\omega)] \cong \tfrac{1}{K}(S_{xx}(e^{j\omega}))^2$$

$\rightarrow 0$ as $K \rightarrow \infty$

$\Rightarrow$ consistent

Effect of Varying the Block Size

Length Q, if segments don't overlap: $K = \frac{Q}{L}$

As $L\uparrow$ $K\downarrow$ i.e. as $S_{ww}(e^{j\omega}) \to \delta(\omega)$ $K \downarrow 0$

as $K \to \infty$ $S_{ww}(e^{j\omega})$ blurs more

Tradeoff Variance vs. Resolution

Trade off: <u>Resolution</u> vs. <u>Variance Reduction</u>

large L large K

"Welch": different windows, overlapping "Welch Technique" (1970)

Can overlap, e.g. ≈ 50%, and get improvement in both, more than that and they are too correlated to get reduction in σ_x^2

Method 2: Indirect (Blackman & Tukey)

Estimate $R_{xx}[m]$ from x[n] first, then take

$$\hat{S}_{xx}(e^{j\omega}) = \text{DTFT}(R_{xx}[m])$$

$$\hat{R}_{xx}[m] = \text{ estimate of } E\{x[n]x[n+m]\}$$

Start with v[n],

$$v[n] = x[n]R_L[n] (= x[n]w[n] \text{ more generally})$$

Sample Correlation

$$R_{vv}[m] = \sum_{n=0}^{Q-1} v[n]v[n+m]$$

$$R_{vv}[m] = \begin{cases} \sum_{n=0}^{Q-|m|-1} x[n]x\left[n+|m|\right], & |m| \le Q-1 \\ 0, & \text{else} \end{cases}$$

Estimate of Autocorrelation Function

Try:

$$\hat{R}_{xx}[m] = \frac{1}{Q} R_{vv}[m]$$

$$E\{\hat{R}_{xx}[m]\} = \frac{1}{Q} \sum_{n=0}^{Q-|m|-1} E\{x[n]x\left[n+|m|\right]\}, \quad |m| \le Q-1$$

Almost Unbiased

$$= \frac{1}{Q}(Q - |m|) R_{xx}[m], \qquad |m| \le Q-1$$

$$= \begin{cases} \left(\dfrac{Q-|m|}{Q}\right) R_{xx}[m], & |m| \le Q-1 \\ 0, & \text{else} \end{cases}$$

$\Rightarrow$ biased, but almost unbiased for small m

An Unbiased Alternative

Could use:

$$\frac{1}{Q-|m|}\sum_{n=0}^{Q-|m|-1} x[n]x\left[n+|m|\right] \Rightarrow \text{unbiased}$$

Estimate of the Spectrum

$$\hat{S}_{xx}(e^{j\omega}) = \sum_{m=-(M-1)}^{M-1} w_c[n]\hat{R}_{xx}[m]e^{-j\omega m}$$

Note:

$$\hat{R}_{xx}[m] \overset{\text{DTFT}}{\leftrightarrow} \frac{1}{Q}\left|V(e^{j\omega})\right|^2 = I(\omega)$$

Resulting Spectrogram Estimate

$$\therefore \hat{S}_{xx}(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} I(\theta)W_c(e^{j(\omega-\theta)})d\theta$$

periodogram ($I(\theta)$)

then smooth the result by averaging adjacent frequencies ($W_c(e^{j(\omega-\theta)})$)

Retaining Positivity Constraint

Note

$\hat{S}_{xx}(e^{j\omega}) \not\geq 0$ not necessary unless

$W_c(e^{j\omega}) \geq 0 \quad |\omega| < \pi$ (e.g. Gaussian window)

Periodogram averaging $\Rightarrow \hat{I}(\omega) \geq 0$

by definition

Asymptotically Unbiased, Consistent

$$E(\hat{S}_{xx}(e^{j\omega})) \cong \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})W_c(e^{j(\omega-\theta)})d\theta$$

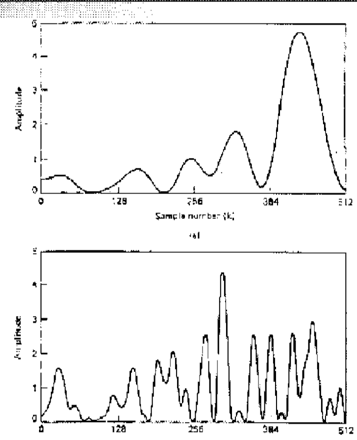
Biased but, if $w_c[0] = 1 \Rightarrow$ asymptotically unbiased

$$\mathrm{Var}(\hat{S}_{xx}(e^{j\omega})) \cong \underbrace{\left(\frac{1}{Q}\sum_{-(M-1)}^{M-1} W_c^2[m]\right)}_{\text{Variance-reduction factor}} S_{xx}^2(e^{j\omega})$$

Variance-reduction factor

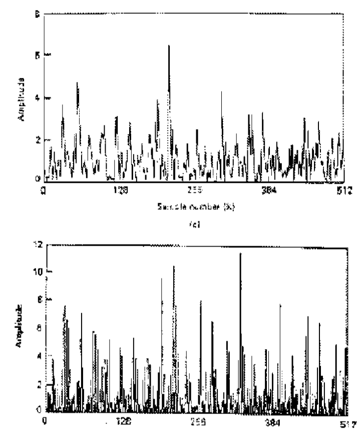
$Q \to \infty \Rightarrow$ consistent

Periodogram Examples



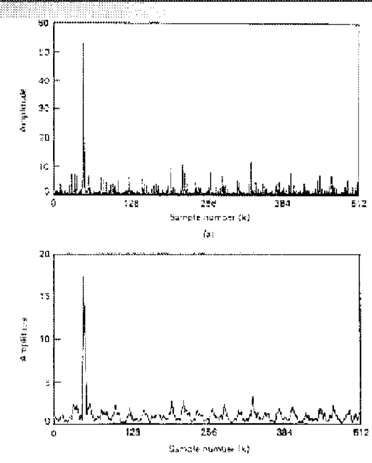
Periodograms for windows of length 16 and 64 for white noise

Note, no improvement with L



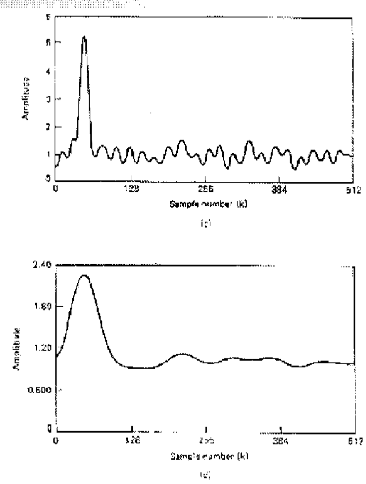
Periodograms for windows of length 256 and 1024 for white noise

Periodogram Averaging Examples



Averaging, Q=1024, (a) L=1024, K=1
L=256 (b), K=7

Averaging, cont'd



(c) K=31, L=64 (d) K=127, L=16

Notes on Power Spectral Densities

$$x(t) \longrightarrow \boxed{H_{\in}(j\Omega)} \longrightarrow x_{\in}(t)$$

var $x_{\in}(t) = S_{xx}(j\Omega)$

$$H_{\in}(j\Omega) = \begin{cases} \sqrt{2\pi / \in}, & |\Omega - \Omega_0| < \in / 2 \\ 0, & \text{else} \end{cases}$$

$$S_{xx}(j\Omega_0) \triangleq \lim_{\in \to 0} \text{Var } x_{\in}(t) = \lim_{\in \to 0} E\left\{(x_{\in}(t) - E\{x_{\in}(t)\})^2\right\}$$

Power Spectral Densities, cont'd

Theorem:

$$S_{xx}(j\Omega) = FT\{c_{xx}(\tau)\} = \int_{-\infty}^{\infty} c_{xx}(\tau)e^{-j\Omega\tau}d\tau$$

$$c_{xx}(\tau) = E\{(x(t) - \overline{x}(t))(x(t+\tau) - \overline{x}(t+\tau))\}$$

Power Spectral Densities, cont'd

Proof: Let $T_{xx}(j\Omega) = FT\{c_{xx}(\tau)\}$

$$C_{x_\in x_\in}(\tau) = h_\in(\tau) * h_\in(-\tau) * C_{xx}(\tau)$$

$$T_{x_\in x_\in}(j\Omega) = |H_\in(j\Omega)|^2 T_{xx}(j\Omega)$$

Power Spectral Densities, cont'd

$$\text{Var } x_\in(t) = C_{x_\in x_\in}(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} T_{x_\in x_\in}(j\Omega)d\Omega$$

$$\cong \frac{1}{2\pi}\cdot\frac{2\pi}{\in}\in T_{xx}(j\Omega_0)$$

$$\underbrace{\lim_{\in\to 0}\text{var } x_\in(t)} = T_{xx}(j\Omega_0)$$

$$S_{xx}(j\Omega_o) = T_{xx}(j\Omega_0), \qquad \text{QED}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 24

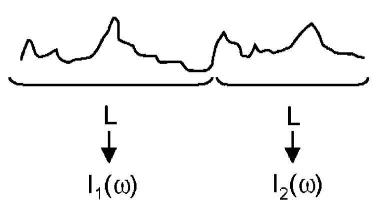
Professor Andrew Singer

Department of Electrical and Computer Engineering

Periodogram Averaging

$$E(I(\omega)) = \frac{1}{LU} S_{xx}(e^{j\omega}) * \underbrace{\left|W(e^{j\omega})\right|^2}_{S_{WW}(e^{j\omega})}$$

define $\hat{I}(\omega) = \frac{1}{K}\sum_{i=1}^{K} I_i(\omega)$



Averaging Reduces Variance

"Periodogram Averaging"

$$E(\hat{I}(\omega)) = E(I_i(\omega)) = S_{xx}(e^{j\omega}) * S_{ww}(e^{j\omega})\tfrac{1}{LU}$$

$\rightarrow$ biased, but can be asymptotically unbiased.

$$\mathrm{Var}\left[\hat{I}(\omega)\right] = \tfrac{1}{K}\mathrm{Var}[I_i(\omega)] \cong \tfrac{1}{K}(S_{xx}(e^{j\omega}))^2$$

$\rightarrow 0$ as $K \rightarrow \infty$

$\Rightarrow$ consistent

Effect of Varying the Block Size

Length Q, if segments don't overlap: $K = \dfrac{Q}{L}$

As $L\uparrow$ $K\downarrow$ i.e. as $S_{ww}(e^{j\omega}) \rightarrow \delta(\omega)$ $K \downarrow 0$

as $K \rightarrow \infty$ $S_{ww}(e^{j\omega})$ blurs more

Tradeoff Variance vs. Resolution

Trade off: Resolution vs. Variance Reduction

large L   large K

"Welch": different windows, overlapping "Welch Technique" (1970)

Can overlap, e.g. ≈ 50%, and get improvement in both, more than that and they are too correlated to get reduction in σ_x^2

Method 2: Indirect (Blackman & Tukey)

Estimate $R_{xx}[m]$ from x[n] first, then take

$$\hat{S}_{xx}(e^{j\omega}) = \mathrm{DTFT}(R_{xx}[m])$$

$$\hat{R}_{xx}[m] = \text{ estimate of } E\{x[n]x[n+m]\}$$

Start with v[n],

$$v[n] = x[n]R_L[n](= x[n]w[n] \text{ more generally})$$

Sample Correlation

$$R_{vv}[m] = \sum_{n=0}^{Q-1} v[n]v[n+m]$$

$$R_{vv}[m] = \begin{cases} \sum_{n=0}^{Q-|m|-1} x[n]x\left[n+|m|\right], & |m| \le Q-1 \\ 0, & \text{else} \end{cases}$$

Estimate of Autocorrelation Function

Try:

$$\hat{R}_{xx}[m] = \frac{1}{Q} R_{vv}[m]$$

$$E\{\hat{R}_{xx}[m]\} = \frac{1}{Q} \sum_{n=0}^{Q-|m|-1} E\{x[n]x\left[n+|m|\right]\}, \ |m| \le Q-1$$

Almost Unbiased

$$= \frac{1}{Q}\left(Q - |m|\right)R_{xx}[m], \qquad |m| \le Q-1$$

$$= \begin{cases} \left(\dfrac{Q-|m|}{Q}\right) R_{xx}[m], & |m| \le Q-1 \\ 0, & \text{else} \end{cases}$$

$\Rightarrow$ biased, but almost unbiased for small m

An Unbiased Alternative

Could use:

$$\frac{1}{Q-|m|}\sum_{n=0}^{Q-|m|-1} x[n]x\left[n+|m|\right] \Rightarrow \text{unbiased}$$

Estimate of the Spectrum

$$\hat{S}_{xx}(e^{j\omega}) = \sum_{m=-(M-1)}^{M-1} w_c[n]\hat{R}_{xx}[m]e^{-j\omega m}$$

Note:

$$\hat{R}_{xx}[m] \overset{\text{DTFT}}{\leftrightarrow} \frac{1}{Q}\left|V(e^{j\omega})\right|^2 = I(\omega)$$

Resulting Spectrogram Estimate

$$\therefore \hat{S}_{xx}(e^{j\omega}) = \tfrac{1}{2\pi}\int_{-\pi}^{\pi} I(\theta)W_c(e^{j(\omega-\theta)})d\theta$$

periodogram ($I(\theta)$)

then smooth the result by averaging adjacent frequencies ($W_c(e^{j(\omega-\theta)})$)

Retaining Positivity Constraint

Note

$\hat{S}_{xx}(e^{j\omega}) \not\geq 0$ not necessary unless

$W_c(e^{j\omega}) \geq 0 \quad |\omega| < \pi$ (e.g. Gaussian window)

Periodogram averaging $\Rightarrow \quad \hat{I}(\omega) \geq 0$

by definition

Asymptotically Unbiased, Consistent

$$E(\hat{S}_{xx}(e^{j\omega})) \cong \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})W_c(e^{j(\omega-\theta)})d\theta$$

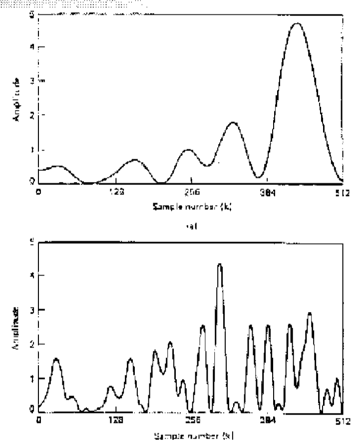
Biased but, if $w_c[0] = 1 \Rightarrow$ asymptotically unbiased

$$\mathrm{Var}(\hat{S}_{xx}(e^{j\omega})) \cong \underbrace{\left(\frac{1}{Q}\sum_{-(M-1)}^{M-1} W_c^2[m]\right)}_{} S_{xx}^2(e^{j\omega})$$

Variance-reduction factor

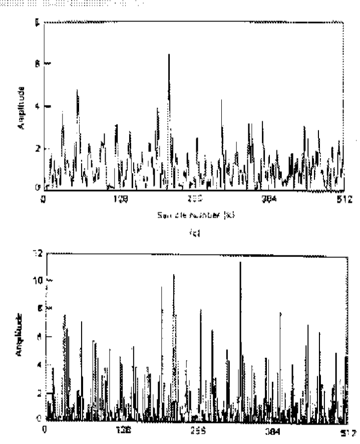
$Q \to \infty \Rightarrow$ consistent

Periodogram Examples



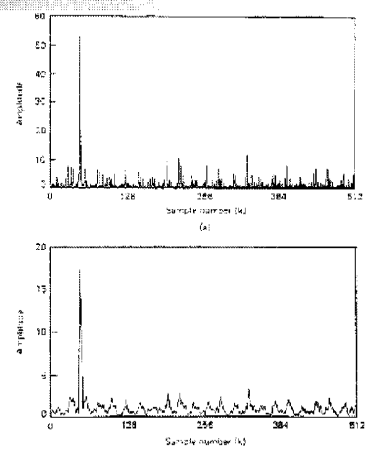
Periodograms for windows of length
16 and 64 for white noise

Note, no improvement with L



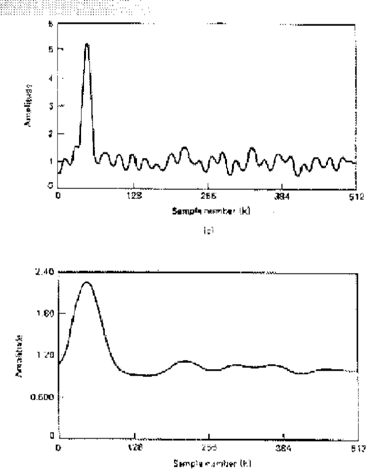
Periodograms for windows of length
256 and 1024 for white noise

Periodogram Averaging Examples



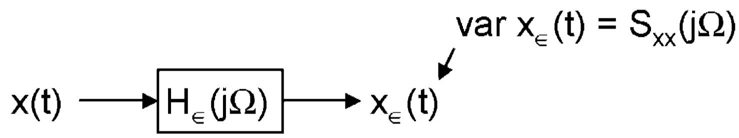
Averaging, Q=1024, (a) L=1024, K=1
L=256 (b), K=7

Averaging, cont'd



(c) K=31, L=64 (d) K=127, L=16

Notes on Power Spectral Densities



$$H_\in(j\Omega) = \begin{cases} \sqrt{2\pi/\in}, & |\Omega - \Omega_0| < \in/2 \\ 0, & \text{else} \end{cases}$$

$$S_{xx}(j\Omega_0) \triangleq \lim_{\in \to 0} \text{Var } x_\in(t) = \lim_{\in \to 0} E\left\{(x_\in(t) - E\{x_\in(t)\})^2\right\}$$

Power Spectral Densities, cont'd

Theorem:

$$S_{xx}(j\Omega) = FT\{c_{xx}(\tau)\} = \int_{-\infty}^{\infty} c_{xx}(\tau) e^{-j\Omega\tau} d\tau$$

$$c_{xx}(\tau) = E\{(x(t) - \overline{x}(t))(x(t+\tau) - \overline{x}(t+\tau))\}$$

Power Spectral Densities, cont'd

Proof: Let $T_{xx}(j\Omega) = FT\{c_{xx}(\tau)\}$

$$C_{x_\in x_\in}(\tau) = h_\in(\tau) * h_\in(-\tau) * C_{xx}(\tau)$$

$$T_{x_\in x_\in}(j\Omega) = \left|H_\in(j\Omega)\right|^2 T_{xx}(j\Omega)$$

Power Spectral Densities, cont'd

$$\text{Var } x_\in(t) = C_{x_\in x_\in}(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} T_{x_\in x_\in}(j\Omega)d\Omega$$

$$\cong \frac{1}{2\pi}\cdot\frac{2\pi}{\in}\in T_{xx}(j\Omega_0)$$

$$\underbrace{\lim_{\in\to 0} \text{var } x_\in(t)} = T_{xx}(j\Omega_0)$$

$$S_{xx}(j\Omega_o) = T_{xx}(j\Omega_0), \qquad \text{QED}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

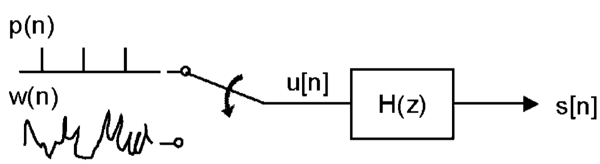
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 25

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Linear Prediction



<u>Model for speech</u>:

- analysis, synthesis
- u[n] = input (not "unit step")

Time-Series Analysis

- Spectral Estimation (parametric)
- Prediction
- Coding (quantization, ADPCM)
- Wiener Filtering

Model

past values → unknown input

$$s[n] = -\sum_{k=1}^{P} a_k s[n-k] + G\sum_{\ell=0}^{P} b_\ell u[n-\ell],\ b_0 = 1$$

$$a_k \quad 1 \le k \le p, \quad b_\ell \quad 1 \le \ell \le q, \quad G = \text{Gain}$$

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1+\sum_{\ell=1}^{q} b_\ell z^{-\ell}}{1+\sum_{k=1}^{p} a_k z^{-k}}$$

System FCN For model

AR/MA Modeling

"Pole-zero" Model

- all-zero model, $a_k = 0, \ 1 \le k \le q$ (MA)
- all-pole model, $b_\ell = 0, \ 1 \le k \le p$ (AR)

Pole-zero = ARMA

Focus on AR (all-pole) models

$$s[n] = -\sum_{k=1}^{p} a_k s[n-k] + Gu[n]$$

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

Predictor

$$s[n] \longrightarrow \boxed{h[n]} \longrightarrow \hat{s}[n]$$

For example: $\hat{s}[n] = -\sum_{k=1}^{p} a_k s[n-k]$

→ Pth order linear predictor

→ $h[k] = -a_k, \quad k = 1, \ldots, p$

→ Strictly causal: $h[k] = 0, \ k \leq 0$ (why?)

→ Negative sign convenient

Linear Predictor

How do we design a linear predictor?

- If I had one, I could encode s[n] by transmitting $\{a_k\}_{k=1}^{p}$ quantized and $\{e[n]\}_{n=0}^{N}$ also quantized, where $e[n] = s[n] - \hat{s}[n]$

↑ "prediction error" "residual"

⇒ LPC → "linear predictive coding"

Formal Problem Statement

(Deterministic Case)

Given $\{x[n]\}_{n=0}^{N-1}$, find a_k, $k = 1, \dots p$

Which would have predicted each sample of

s[n] well via:

$$\hat{s}[n] = \sum_{k=1}^{p} a_k s[n-k]$$

Method of Least Squares

Predictor: $$\hat{s}[n] = -\sum_{k=1}^{p} a_k s[n-k]$$

Prediction error: $$e[n] = s[n] - \hat{s}[n]$$

$$= s[n] + \sum_{k=1}^{p} a_k s[n-k]$$

Let $$E = \sum_{n=0}^{N-1} e^2[n] = \sum_{n=0}^{N-1} \left(s[n] + \sum_{k=1}^{p} a_k s[n-k] \right)^2$$

Method of Least Squares, cont'd

Determine $\{a_k\}_{k=1}^{p}$ to minimize E

Note: when $n \le p$

$*$ don't have data

e.g. $\hat{s}[0] = \sum_{k=1}^{p} a_k s[-k]$

Method of Least Squares, cont'd

→ Can be handled many ways

- assume $s[n] = 0, \; n < 0$
  "auto correlation method"
- do not use where we do not have data
  "covariance method"
- $N \to \infty$ both are equivalent

Method of Least Squares, cont'd

$$E = \sum_{n=0}^{N-1}\left(s[n] + \sum_{k=1}^{p} a_k s[n-k] \right)^2$$

$$\frac{\partial E}{\partial a_\ell} = 0, \qquad \ell = 1, \ldots, p$$

Normal Equations

$$\sum_{k=1}^{p} a_k \sum_{n=0}^{N-1} s[n-k]s[n-\ell] = -\sum_{n=0}^{N-1} s[n]s[n-\ell] \qquad (1)$$

$$\ell = 1, \ldots, p \quad \Rightarrow \quad \text{"Normal Equations"}$$

Solve p linear equations in p unknowns

Look at the error for these a_k's

$$E = \sum_{n=0}^{N-1}\left(s^2[n] + \sum_{k=1}^{p} a_k s[n-k] \sum_{\ell=1}^{p} a_\ell s[n-\ell] + 2\sum_{k=1}^{p} a_k s[n]s[n-k] \right)$$

$$= \sum_{n=0}^{N-1} s^2[n] + \sum_{k=1}^{p} \underbrace{\left(\sum_{\ell=1}^{p} a_\ell \sum_{n} s[n-k]s[n-\ell] \right)}_{(1)} a_k + 2\sum_{k=1}^{p} a_k \sum_{n} s[n]s[n-k]$$

$$= \sum_{n=0}^{N-1} s^2[n] + \sum_{k=1}^{p} a_k \left(-\sum_{n} s[n]s[n-k] \right) + 2\sum_{k=1}^{p} a_k \sum_{n} s[n]s[n-k]$$

$$E = \sum_{n=0}^{N-1} s^2[n] + \sum_{k=1}^{p} a_k \sum_{n=0}^{N-1} s[n]s[n-k]$$

Minimum LS Error

In terms of deterministic correlations

Define

$$\hat{R}_{xx}[k,\ell] = \sum_{n=0}^{N-1} x[n-k]x[n-\ell] = \hat{R}_{xx}[\ell,k]$$

$$(1) \quad \Rightarrow \sum_{k=1}^{p} \hat{R}_{ss}[k,\ \ell] a_k = -\hat{R}_{ss}[0,\ \ell] \qquad \ell = 1, \ldots, p$$

"Augmented Normal Equations"

1) $\sum_{k=1}^{p} a_k \hat{R}_{ss}[k, \ell] = -\hat{R}_{ss}[0, \ell] \qquad \ell = 1, \ldots, p$

2) $E_p = \hat{R}_{ss}[0, 0] + \sum_{k=1}^{p} a_k \hat{R}_{ss}[k, 0]$

Autocorrelation Method

Note: for n'=(n-k)

$\hat{R}_{ss}[k, \ell] = \sum_{n'=-k}^{N-1-k} s[n']s[n'-m], \quad \text{for } m = (\ell - k)$

= convolution of s[n] with s[-n]

And, a function of the difference, (ℓ-k), only if x[n] = 0 outside [0, N – 1]

$\hat{R}_{ss}[k, \ell] = \hat{R}_{ss}[k - \ell] = \hat{R}_{ss}[\ell - k]$

Autocorrelation Normal Equations

1) $\sum_{k=1}^{p} a_k \hat{R}_{ss}[k-\ell] = -\hat{R}_{ss}[\ell]$

2) $"E_p" = E = \hat{R}_{ss}[0] + \sum_{k=1}^{p} a_k \hat{R}_{ss}[k]$

In Matrix Form

$$\begin{bmatrix} \hat{R}_{ss}[0] & \hat{R}_{ss}[1] & \cdots & \hat{R}_{ss}[p-1] \\ \hat{R}_{ss}[1] & \hat{R}_{ss}[0] & \cdots & \hat{R}_{ss}[p-2] \\ \vdots & \ddots & \ddots & \vdots \\ \hat{R}_{ss}[p-1] & \cdots & \hat{R}_{ss}[1] & \hat{R}_{ss}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} \hat{R}_{ss}[1] \\ \hat{R}_{ss}[2] \\ \vdots \\ \hat{R}_{ss}[p] \end{bmatrix}$$

↑

Symmetric, Toeplitz $\triangleq$ diagonals equal

Complexity

- $O(p^3)$ to solve
- can speed up to $O(p^2)$ using special prop's "Levinson Durbin" algorithm
- use this to generate efficient structures for LPC → Lattice Filters

Random Signals

$$E = E\{e^2[n]\}$$

1) $\Rightarrow i = 1, \ldots, p$

$$\sum_{k=1}^{p} a_k E\{s[n-k]s[n-i]\} = -E\{s[n]s[n-i]\}$$

2) $$E_p = E\{s^2[n]\} + \sum_{k=1}^{p} E\{s[n]s[n-k]\}$$

Random Signals, cont'd

WSS: $E\{s[n]s[n-k]\} = R_{ss}[k]$

$E\{s[n-k]s[n-\ell]\} = R_{ss}[k-\ell]$

$\Rightarrow$ same as before!

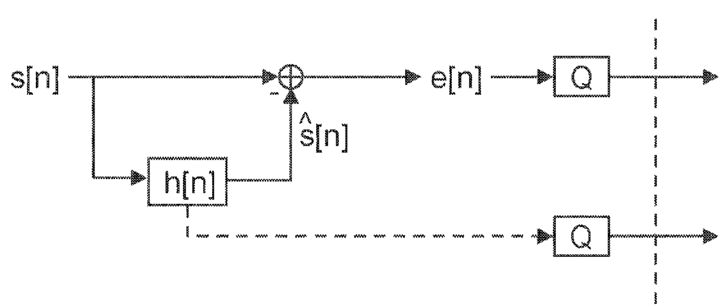
Nonstationary:

same as covariance method

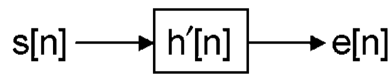
$\rightarrow$ use $R_{ss}[k, \ell]$ $\leftarrow$

Linear Predictive Coding (speech)

LPC:



Prediction error filter



$$h'[n] = \delta[n] - h[n] = \delta[n] + \sum_{k=1}^{p} a_k \delta[n-k]$$

$$H'(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$$

Start with the ACNE's

(1)

$$\begin{bmatrix} \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p-1] \\ \vdots & & & \\ \hat{R}[p-1] & & & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} \hat{R}[1] \\ \vdots \\ \hat{R}[p] \end{bmatrix}$$

Move all to one side

$$\left[\begin{array}{c|cccc} \hat{R}[1] & \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p-1] \\ \hat{R}[2] & \hat{R}[1] & & & \\ \vdots & \vdots & & & \\ \hat{R}[p] & \hat{R}[p-1] & & & \hat{R}[0] \end{array}\right] \left[\begin{array}{c} +1 \\ \hline a_1 \\ \vdots \\ a_p \end{array}\right] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Add the equation for the MMSE

(2)

$$\begin{bmatrix} \hat{R}_{ss}[0] & \hat{R}_{ss}[1] & \cdots & \hat{R}_{ss}[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} E_p \end{bmatrix}$$

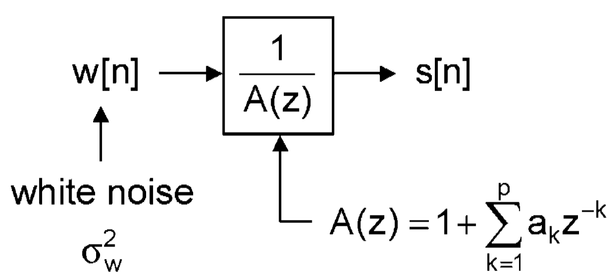
Augmented Normal Equations

$$\begin{bmatrix} (2) \\ \hdashline (1) \end{bmatrix} \Rightarrow \begin{bmatrix} \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p] \\ \hdashline \hat{R}[1] & \hat{R}[0] & \cdots & \hat{R}[p-1] \\ \vdots & \ddots & & \\ \hat{R}[p] & & & \hat{R}[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} E_p \\ \hdashline 0 \\ \\ \end{bmatrix}$$

$p+1$ eqn's $\quad p+1$ unknowns: $\{a_k\}_{k=1}^{p},\ E_p$

AR / All-Pole Modeling

Model for s[n]:



$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$$

= all-pole model

AR / All-Pole Modeling Problem

$\rightarrow$ Given $\{s[n]\}_{n=0}^{N-1}$, apply (1), (2)

$$N \rightarrow \infty \quad \hat{R}_{ss}[k] \rightarrow R_{ss}[k] = E\{s[n]s[n+k]\}$$

Case: $P = p \qquad \underset{\text{algorithm}}{a_k} \rightarrow \underset{\text{model}}{a_k}$

$P \leq p \qquad \underset{\text{algorithm}}{a_k} \rightarrow \begin{cases} \underset{\text{model}}{a_k}, & k = 1, \ldots, P \\ 0, & k = P+1, \ldots, p \end{cases}$

$E_p \rightarrow \sigma_w^2$

AR Spectral Estimation

- "Parametric"
- "High Resolution"

$$\hat{S}_{xx}(e^{j\omega}) = \frac{\sigma_w^2}{\left|A(e^{j\omega})\right|^2}$$

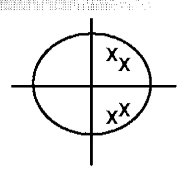
AR Spectral Estimation, cont'd

Model sequence x[n] as p^{th} order AR process and estimate parameters by linear prediction!

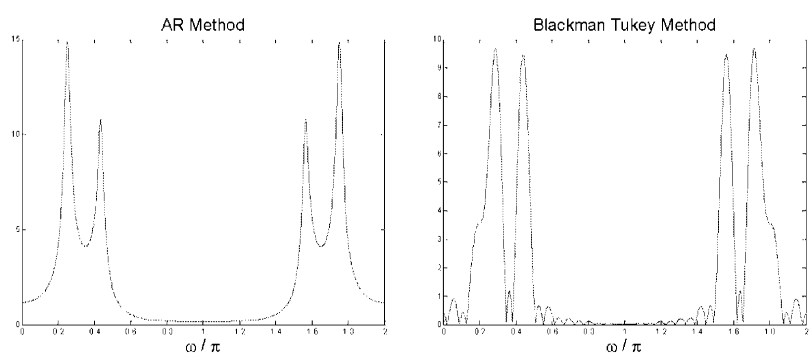
Spectral Estimate:

$$\hat{S}_{xx}(e^{j\omega}) = \frac{E_p}{\left|1 + \sum_{k=1}^{p} a_k e^{-jk\omega}\right|^2}$$

Example



$\hat{S}_{xx}(e^{j\omega})$



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 26

Professor Andrew Singer
Department of Electrical and Computer Engineering

Linear Prediction: Random Signals

$$E = E\{e^2[n]\}$$

1) $\Rightarrow i = 1, \ldots, p$

$$\sum_{k=1}^{p} a_k E\{s[n-k]s[n-i]\} = -E\{s[n]s[n-i]\}$$

2) $$E_p = E\{s^2[n]\} + \sum_{k=1}^{p} E\{s[n]s[n-k]\}$$

Random Signals: Stationarity

WSS: $E\{s[n]s[n-k]\} = R_{ss}[k]$

$E\{s[n-k]s[n-\ell]\} = R_{ss}[k-\ell]$

$\Rightarrow$ same as before!

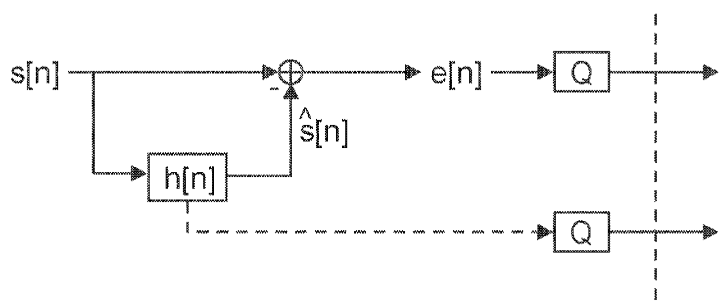
Nonstationary:

same as covariance method

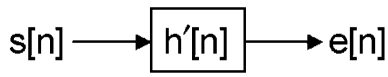
$\rightarrow$ use $R_{ss}[k, \ell]$ $\leftarrow$

Linear Predictive Coding (speech)

LPC:



Prediction error filter



$$h'[n] = \delta[n] - h[n] = \delta[n] + \sum_{k=1}^{p} a_k \delta[n-k]$$

$$H'(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$$

Start with the ACNE's

(1)

$$\begin{bmatrix} \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p-1] \\ \vdots & & & \\ \hat{R}[p-1] & & & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} \hat{R}[1] \\ \vdots \\ \hat{R}[p] \end{bmatrix}$$

Move all to one side

$$\begin{bmatrix} \hat{R}[1] & \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p-1] \\ \hat{R}[2] & \hat{R}[1] & & & \\ \vdots & \vdots & & & \\ \hat{R}[p] & \hat{R}[p-1] & & & \hat{R}[0] \end{bmatrix} \begin{bmatrix} +1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Add the equation for the MMSE

(2)

$$\begin{bmatrix} \hat{R}_{ss}[0] & \hat{R}_{ss}[1] & \cdots & \hat{R}_{ss}[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} E_p \end{bmatrix}$$

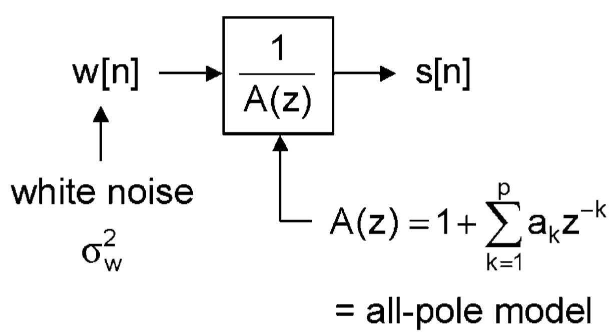
Augmented Normal Equations

$$\begin{bmatrix}(2)\\ \hline (1)\end{bmatrix} \Rightarrow \begin{bmatrix}\hat{R}[0] & \hat{R}[1] & \dots & \hat{R}[p]\\ \hline \hat{R}[1] & \hat{R}[0] & \dots & \hat{R}[p-1]\\ \vdots & \ddots & & \\ \hat{R}[p] & & & \hat{R}[0]\end{bmatrix}\begin{bmatrix}1\\ a_1\\ \vdots\\ a_p\end{bmatrix} = \begin{bmatrix}E_p\\ \hline 0\\ \\ \end{bmatrix}$$

$p+1$ eqn's $\quad p+1$ unknowns: $\{a_k\}_{k=1}^{p},\ E_p$

AR / All-Pole Modeling

Model for s[n]:



AR / All-Pole Modeling Problem

$\rightarrow$ Given $\{s[n]\}_{n=0}^{N-1}$, apply (1), (2)

$$N \rightarrow \infty \quad \hat{R}_{ss}[k] \rightarrow R_{ss}[k] = E\{s[n]s[n+k]\}$$

Case: $P = p \qquad \underset{\text{algorithm}}{a_k} \rightarrow \underset{\text{model}}{a_k}$

$P \leq p \qquad \underset{\text{algorithm}}{a_k} \rightarrow \begin{cases} \underset{\text{model}}{a_k}, & k = 1, \ldots, P \\ 0, & k = P+1, \ldots, p \end{cases}$

$E_p \rightarrow \sigma_w^2$

AR Spectral Estimation

- "Parametric"
- "High Resolution"

$$\hat{S}_{xx}(e^{j\omega}) = \frac{\sigma_w^2}{\left|A(e^{j\omega})\right|^2}$$

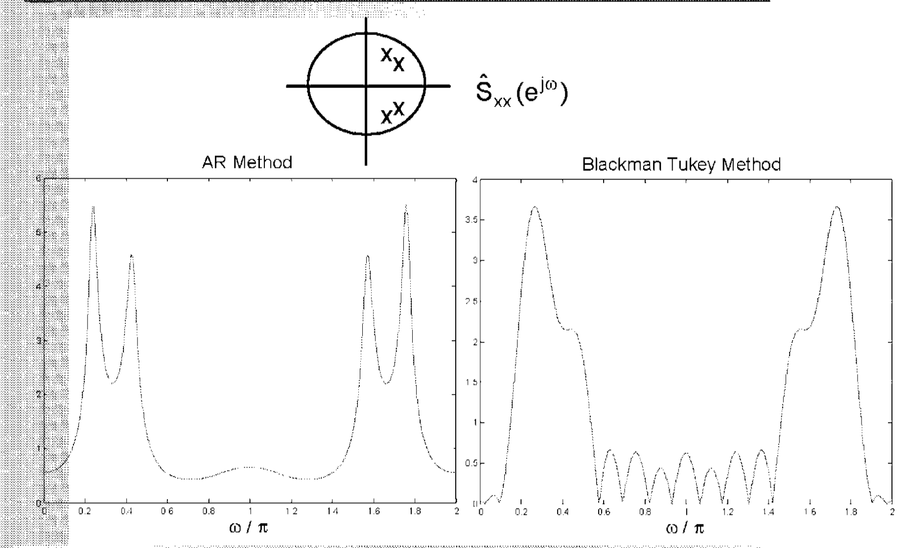
AR Spectral Estimation, cont'd

Model sequence x[n] as p^{th} order AR process and estimate parameters by linear prediction!

Spectral Estimate:

$$\hat{S}_{xx}(e^{j\omega}) = \frac{E_p}{\left|1 + \sum_{k=1}^{p} a_k e^{-jk\omega}\right|^2}$$

Example



How to solve ACNE's efficiently

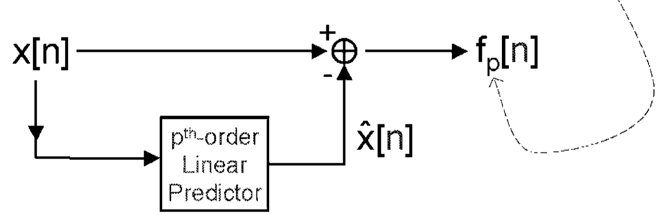
Start with Augmented Normal Equations:

$$\begin{bmatrix} \hat{R}[0] & \hat{R}[1] & \cdots & \hat{R}[p] \\ \hat{R}[1] & \hat{R}[0] & \cdots & \hat{R}[p-1] \\ \vdots & & \ddots & \vdots \\ \hat{R}[p] & \hat{R}[p-1] & \cdots & \hat{R}[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} E \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

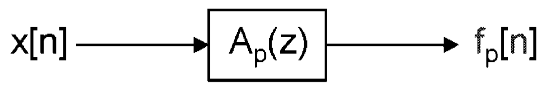
Forward Prediction Error

Order-explicit notation: p^{th}-order

$$\hat{x}[n] = -\sum_{k=1}^{p} a_p[k]x[n-k]$$

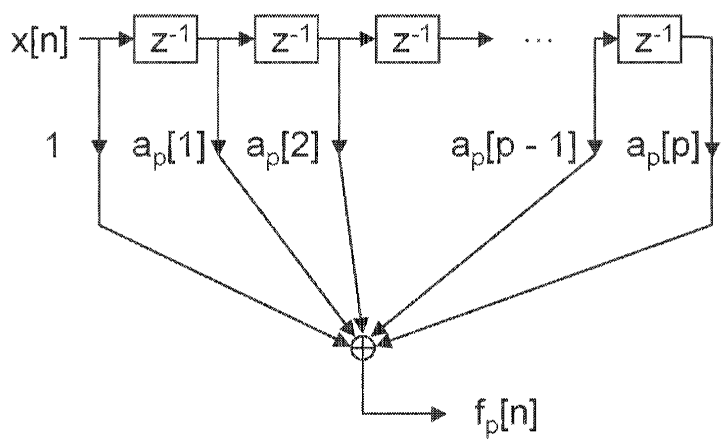


Formal Prediction-Error Filter

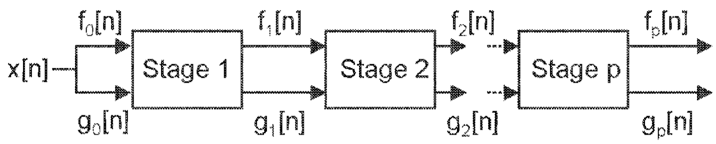


$$A_p(z) = \sum_{k=0}^{p} a_p[k] z^{-k}, \qquad a_p[0] = 1$$

Transversal Filter for $A_p(z)$



Equivalent Lattice Filter Form



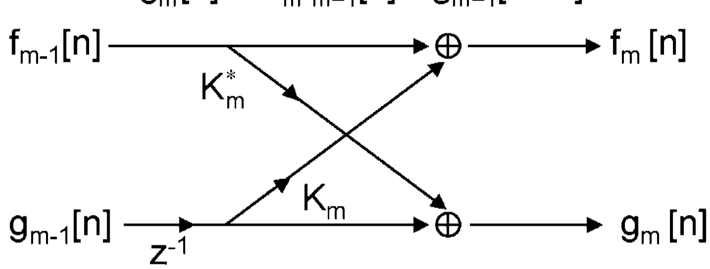
A lattice implementation can be derived from the given A(z)

Order-Recursion

$$f_0[n] = g_0[n] = x[n]$$

$$f_m[n] = f_{m-1}[n] + K_m g_{m-1}[n-1]$$

$$g_m[n] = K_m^* f_{m-1}[n] + g_{m-1}[n-1]$$



Single lattice filter stage

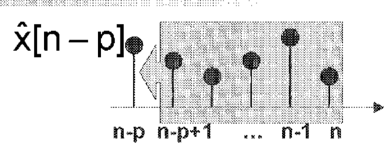
Order-Recursion, cont'd

By equivalence:

$$F_p(z) = A_p(z)X(z)$$

$$A_p(z) = \frac{F_p(z)}{X(z)} = \frac{F_p(z)}{F_0(z)}$$

Backward Linear Prediction



$$\hat{x}[n-p] = -\sum_{k=0}^{p-1} b_p[k]x[n-k]$$

$$g_p[n] = x[n-p] + \sum_{k=0}^{p-1} b_p[k]x[n-k]$$

$$= \sum_{k=0}^{p} b_p[k]x[n-k], \qquad b_p[p] = 1$$

Backward Linear Prediction

⇒ Same as forward prediction of $x'[n]=x[-n]$

⇒ Note that $R_{xx}[m] = R_{x'x'}[m]$, $\therefore E_p^b = E_p^f$

⇒ $b_p[k] = a_p[p-k] \quad k = 0, \ldots, p$

⇒ Can be realized with <u>same</u> lattice filter

⇒ If $x[n]$ is complex, then:

$$R_{xx}[m] = R^*_{x'x'}[m], \quad b_p[k] = a^*_p[p-k] \quad k = 0, \ldots, p$$

Forward and Backward Prediction

$$R_{x'x'}[m] = E\{x'[n]x'[n+m]^*\} = E\{x[-n]x[-n-m]^*\}$$

$$= E\{x[k]x[k-m]^*\} = E\{x[n]^* x[n+m]\} = R^*_{xx}[m]$$

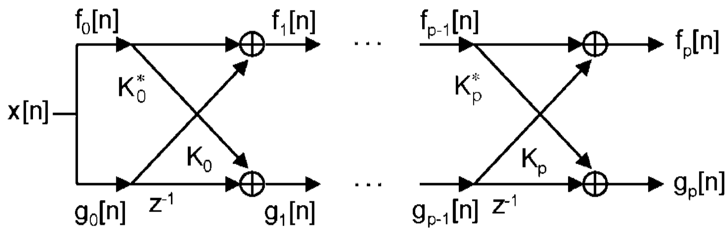
$$\hat{x}'[n] = \sum_{k=1}^{p} a_k x'[n-k] \quad \text{(assume real)}$$

$$\hat{x}[-n] = \sum_{k=1}^{p} a_k x[k-n]$$

$$\hat{x}[n-p] = \sum_{k=1}^{p} a_k x[k+n-p] = \sum_{\ell=p}^{1} \underbrace{a_{p-\ell}}_{b_\ell} x[n-\ell]$$

$$b_p[k] = a_p[p-k] \quad k = 0, \ldots, p$$

1 Lattice, two predictors!



Backward Prediction Lattice

$$G_p(z) = B_p(z)X(z)$$

$$B_p(z) = \frac{G_p(z)}{X(z)} = \frac{G_p(z)}{G_0(z)}$$

Backward Linear Prediction

Since $b_p[k] = a_p^*[p-k]$:

$$B_p(z) = \sum_{k=0}^{p} a_p^*[p-k]z^{-k}$$

$$= z^{-p}\sum_{k=0}^{p} a_p^*[k]z^{k}$$

$$= z^{-p}A_p^*(z^{-1})$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

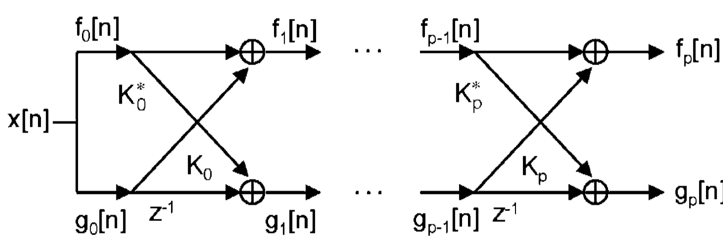
ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 27

Professor Andrew Singer

Department of Electrical and
Computer Engineering

Return to the Predictor Lattice



Prediction Error Recursion

$$F_0(z) = G_0(z) = X(z)$$

$$(1) \quad F_m(z) = F_{m-1}(z) + K_m z^{-1} G_{m-1}(z) \qquad m = 1,\ldots,p$$

$$G_m(z) = K_m^* F_{m-1}(z) + z^{-1} G_{m-1}(z) \qquad m = 1,\ldots,p$$

Prediction-Error Transfer Functions

$$\frac{(1)}{X(z)} \Rightarrow A_0(z) = B_0(z) = 1$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \qquad m = 1,\ldots,p$$

$$B_m(z) = K_m^* A_{m-1}(z) + z^{-1} B_{m-1}(z) \qquad m = 1,\ldots,p$$

Recursion for A(z) , B(z), and K_m

$\Rightarrow$ (2)

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m z^{-1} \\ K_m^* & z^{-1} \end{bmatrix} \begin{bmatrix} A_{m-1}(z) \\ B_{m-1}(z) \end{bmatrix}$$

Use (2) to get $\{a_m[k]\}_{k=1}^{m} \leftrightarrow \{K_m\}_{m=1}^{p}$

Lattice Structure is "efficient"

Lattice Filter with

$\{K_m\}_{m=1}^{p} \Rightarrow$ p direct form structures $\{\{a_\ell[k]\}_{k=1}^{\ell}\}_{\ell=1}^{p}$

$\downarrow$ $\Downarrow$

only p coeffs vs. $\dfrac{p(p+1)}{2}$ coeffs

Order-recursive

Reflection coeff's are embedded

optimal order k $\Rightarrow K_1, \ldots, K_k$

optimal order k + 1 $\rightarrow \underbrace{K_1, \ldots, K_k}_{\text{same!}}, K_{k+1}$

From reflection coeff's to $A_m(z)$

$$a_m[0] = 1$$

$$a_m[m] = K_m$$

$$a_m[k] = a_{m-1}[k] + K_m a^*_{m-1}[m-k]$$

$$= a_{m-1}[k] + a_m[m] a^*_{m-1}[m-k]$$

$$1 \le k \le m-1$$

$$m = 1, \ldots, p$$

From $A_m(z)$'s to reflection coeff's

$$K_m = a_m[m]$$

$$a_{m-1}[k] = \frac{a_m[k] - K_m b_m[k]}{1 - |K_m|^2}$$

$$= \frac{a_m[k] - a_m[m] a_m^*[m-k]}{1 - |a_m[m]|^2}$$

$$m = p, \ldots, 1$$

Levison-Durbin Algorithm

$$R_p = \begin{bmatrix} R_{xx}[0] & R_{xx}[1] & \cdots & R_{xx}[p-1] \\ R_{xx}[1] & & & \\ \vdots & & & \\ R_{xx}[p-1] & R_{xx}[p-2] & \cdots & R_{xx}[0] \end{bmatrix}$$

$R_p[i, j] = R_p[j, i]$ symmetric

$R_p[i, j] = R_{xx}[i - j]$ toeplitz

Order-Recursive Solution

Normal Equations:

$$\sum_{k=0}^{p} a_p[k]R_{xx}[\ell - k] = 0 \qquad \ell = 1, 2, \ldots, p$$

$$a_p[0] = 1$$

Augment with:

$$\sum_{k=0}^{p} a_p[k]R_{xx}[\ell - k] = \begin{cases} E_p^f, & \ell = 0 \\ 0, & \ell = 1, 2, \ldots, p \end{cases}$$

Order-Recursive Solution, cont'd

Start with m = 1:

$$a_1[1] = \frac{-R_{xx}[1]}{R_{xx}[0]}, \quad E_1^f = R_{xx}[0] + a_1[1]R_{xx}[-1]$$

$$= R_{xx}[0](1 - |a_1[1]|^2)$$

$$a_1[1] = K_1 \qquad \text{(first reflection coeff.)}$$

Order-Recursive Solution, cont'd

Now let m = 2:

$$a_2[1]R_{xx}[0] + a_2[2]R_{xx}[1] = -R_{xx}[1]$$

$$a_2[1]R_{xx}[1] + a_2[2]R_{xx}[0] = -R_{xx}[2]$$

$$\Rightarrow a_2[2] = -\frac{R_{xx}[2] + a_1[1]R_{xx}[1]}{E_1^f}$$

$$a_2[1] = a_1[1] + a_2[2]a_1[1]$$

Vector update of $a_m[k]$'s

In general:

$$\vec{a}_m = \begin{bmatrix} a_m[1] \\ a_m[2] \\ \vdots \\ a_m[m] \end{bmatrix} = \begin{bmatrix} \\ \vec{a}_{m-1} \\ \cdots \\ 0 \end{bmatrix} + \begin{bmatrix} \\ \vec{d}_{m-1} \\ \cdots \\ K_m \end{bmatrix}$$

 © 2001 Andrew Singer, University of Illinois at Urbana-Champaign, All Rights Reserved

Partitioning the Correlation Matrix

Partition R_m

$$R_m = \begin{bmatrix} R_{m-1} & \vec{r}^{\,b}_{m-1} \\ \vec{r}^{\,bT}_{m-1} & R_{xx}[0] \end{bmatrix}$$

$$\left(\vec{r}^{\,b}_{m-1}\right)^T = \vec{r}^{\,bT}_{m-1} = \left[R_{xx}[m-1] \;\; R_{xx}[m-2] \;\; \ldots \;\; R_{xx}[1]\right]$$

$$\alpha^b_{\uparrow} = \text{"flip up-down", or "reverse order"}$$

Write mth-order in terms of m-1st-order

Now: $R_m \vec{a}_m = -\vec{r}_m$ becomes

$$\begin{bmatrix} R_{m-1} & \vec{r}^{\,b}_{m-1} \\ \vec{r}^{\,bT}_{m-1} & R_{xx}[0] \end{bmatrix} \left(\begin{bmatrix} \vec{a}_{m-1} \\ \cdots \\ 0 \end{bmatrix} + \begin{bmatrix} \vec{d}_{m-1} \\ \cdots \\ K_m \end{bmatrix} \right) = - \begin{bmatrix} \vec{r}_{m-1} \\ \cdots \\ R_{xx}[m] \end{bmatrix}$$

“divide & conquer” step of
Levinson-Durbin Alg.

Partitioned Matrix Equations become

1) $R_{m-1}\vec{a}_{m-1} + R_{m-1}\vec{d}_{m-1} + K_m\vec{r}^{\,b}_{m-1} = -\vec{r}_{m-1}$

2) $\vec{r}^{\,bT}_{m-1}\vec{a}_{m-1} + \vec{r}^{\,bT}_{m-1}\vec{d}_{m-1} + K_m R_{xx}[0] = -R_{xx}[m]$

but

$$R_{m-1}\vec{a}_{m-1} = -\vec{r}_{m-1}$$

Solve for update terms

1) $\Rightarrow \vec{d}_{m-1} = -K_m R^{-1}_{m-1}\vec{r}^{\,b}_{m-1}$

$$= K_m a^b_{m-1}$$

2) $\Rightarrow K_m = \dfrac{-\left(R_{xx}[m] + \vec{r}^{\,bT}_{m-1}\vec{a}_{m-1}\right)}{R_{xx}[0] + \vec{r}^{\,bT}_{m-1}\vec{a}^{\,b}_{m-1}}$

Putting It All Together in $\theta(p^2)$ ops.

Levinson Durbin Algorithm:

$$a_m[m] = K_m = \frac{-\left(R_{xx}[m] + \vec{r}_{m-1}^{\,bT}\vec{a}_{m-1}\right)}{R_{xx}[0] + \vec{r}_{m-1}^{\,bT}\vec{a}_{m-1}^{\,b}}$$

$$= -\frac{\left(R_{xx}[m] + \vec{r}_{m-1}^{\,bT}\vec{a}_{m-1}\right)}{E_m^f}$$

$$a_m[k] = a_{m-1}[k] + K_m a_{m-1}[m-k]$$

$$= a_{m-1}[k] + a_m[m] a_{m-1}[m-k]$$

$$k = 1,\ 2,\ \cdots,\ m-1;\ m = 1,\ 2,\ \cdots,\ p$$

Levinson-Durbin Algorithm

Note: Same as recursion for predictor coefficients from the Lattice Filter

⇒ Lattice implements Levinson-Durbin

or,

Levinson-Durbin produces <u>optimum Lattice</u> filter, too!

Note how the prediction errors change

$$E_f^m = R_{xx}[0] + \sum_{k=1}^{m} a_m[k] R_{xx}[-k]$$

$$= E_{m-1}^f \left[1 - |a_m[m]|^2 \right]$$

$$= E_{m-1}^f \left(1 - \underbrace{|K_m|^2}_{\leq 1} \right), \quad k = 1, 2, \ldots, p$$

$$E_0^f \geq E_1^f \geq E_2^f \geq \ldots \geq E_p^f$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

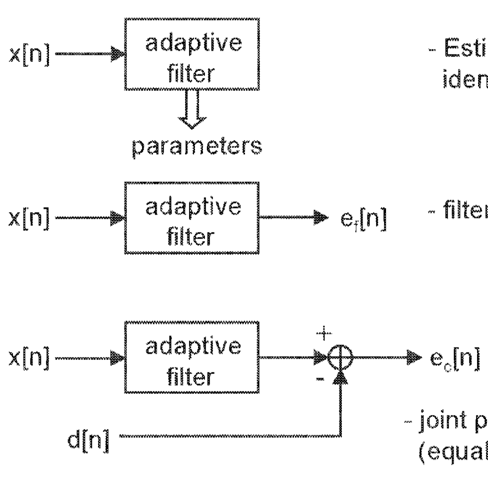
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 28

Professor Andrew Singer
Department of Electrical and Computer Engineering

Adaptive Filtering



- Estimation / modeling/ system identification
- filter, predictor
- joint process estimator (equalization/ noise cancellation)

Adaptive Filtering

* Recursive / Sequential

- Solve for optimal / good filter at each n
- update based on observations
- want algorithms to be fast / stable (sometimes conflicting goals)

Adaptive Filtering Methods

Techniques:

- Deterministic LS
- Statistical MMSE
- Gradient (deterministic / stochastic)

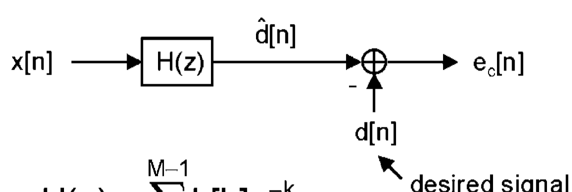
Outline of Adaptive Filters

1) Wiener Filters / MMSE FIR filter design
2) LMS Algorithm
   – simplest / most commonly used
3) Stability / Performance of LMS
4) Applications of adaptive filters
5) Advanced adaptive filter algorithms

Wiener Filter / MMSE Solution

General problem formulation:

Given x[n], d[n], find h[n] such that e[n] is minimized in the mean square



$$H(z) = \sum_{k=0}^{M-1} h[k] z^{-k}$$

x[n], d[n] WSS Random Processes

x[n], d[n] zero-mean, Jointly WSS

$$\Rightarrow R_{xx}[m] = E\{x[n]x[n-m]\} = R_{xx}[-m]$$

$$R_{xd}[m] = E\{x[n]d[n-m]\}$$

$$= R_{dx}[-m]$$

$$R_{xd}[n, m] = R_{xd}[n-m]$$

↑ xcorr ↑ wss

Goal

Find $\{h[n]\}_{n=0}^{M-1}$ to

$$\min \varepsilon = E\{e^2[n]\} = E\left\{\left(d[n] - \underbrace{\sum_{k=0}^{M-1} h[k]x[n-k]}_{\hat{d}[n]}\right)^2\right\}$$

Solution

Method 1:

$0 \le \ell \le M-1$

$$0 = \frac{\partial \varepsilon}{\partial h[\ell]} = E\left\{\left(d[n] - \sum_{k=0}^{M-1} h[k]x[n-k]\right)(-2)x[n-\ell]\right\}$$

$$0 = E\{d[n]x[n-\ell]\} - \sum_{k=0}^{M-1} h[k]E\{x[n-k]x[n-\ell]\}$$

$$0 \le \ell \le M-1 \qquad \sum_{k=0}^{M-1} h[k]R_{xx}[k-\ell] = R_{dx}[\ell]$$

Wiener Filter

$$0 \le \ell \le M-1 \qquad \sum_{k=0}^{M-1} h[k]R_{xx}[k-\ell] = R_{dx}[\ell]$$

Stochastic ACNE's

- → R_{xx} symmetric toeplitz matrix
- → assume positive definite ⇒ nonsingular (usually true)
- ⇒ Levinson-Durbin algorithm to solve it!

ACNEs Revisited

$$\begin{bmatrix} R_{xx}[0] & R_{xx}[1]\cdots & R_{xx}[M-1] \\ R_{xx}[1] & & \\ \vdots & & \\ R_{xx}[M-1] & \cdots & R_{xx}[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[M-1] \end{bmatrix} = \begin{bmatrix} R_{dx}[0] \\ R_{dx}[1] \\ \vdots \\ R_{dx}[M-1] \end{bmatrix}$$

$$R_{xx} \quad \vec{h} \quad = \quad \vec{p}$$

↑ symmetric, toeplitz

usually assume pos. def. (is non neg. def)

$\vec{h}_{opt} = R^{-1}\vec{p}$ (use Levinson-Durbin Algorithm $\theta(m^2)$ operations.

Adaptive Filter Notation

In vector notation:

$$e_c[n] = d[n] - \vec{h}^T \vec{x}[n]$$

where:

$$\vec{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-M+1] \end{bmatrix} \qquad \vec{h} = \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[M-1] \end{bmatrix}$$

Vector Form

$$\varepsilon = E\{e_c^2[n]\} = E\{(d - \vec{h}^T\vec{x}[n])^2\}$$

$$= E\{d^2[n]\} - 2\vec{h}^T E\{d[n]\vec{x}[n]\} + E\{\vec{h}^T\vec{x}[n]\vec{x}[n]^T\vec{h}[n]\}$$

$$= E\{d^2[n]\} - 2\vec{h}^T E\{d[n]\vec{x}[n]\} + \vec{h}^T E\{\vec{x}[n]\vec{x}^T[n]\}\vec{h}^T$$

Wiener Filter Solution

$$\varepsilon = R_{dd}[0] - 2\vec{h}^T\vec{p} + \vec{h}^T R_{xx}\vec{h}$$

$$\nabla_{\vec{h}}\varepsilon = \begin{bmatrix} \partial\varepsilon / \partial h[0] \\ \vdots \\ \partial\varepsilon / \partial h[M-1] \end{bmatrix} = -2\vec{p} + (R_{xx} + R_{xx}^T)\vec{h}$$

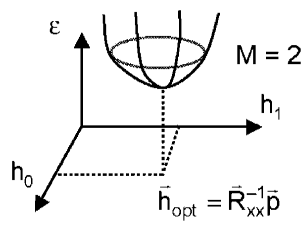
$$= -2\vec{p} + 2R_{xx}\vec{h} = 0$$

$$R_{xx}\vec{h} = \vec{p}!!$$

Define the gradient
(could also complete the square!)

MMSE Gradient Algorithm

$\varepsilon = R_{dd}(0) - 2\vec{p}^T\vec{h} + \vec{h}^T R_{xx}\vec{h}$ $\leftarrow$ quadratic fcn of $\vec{h}$



$\varepsilon_{min} = R_{dd}(0) - \vec{p}^T R_{xx}^{-1}\vec{p}$

Levinson Durbin Alg: $\theta(m^2)$

Steepest Decent Approach

$-\nabla_{\vec{h}}\varepsilon$ points in direction of "steepest decent"

Guess $\vec{h}^{(0)}$

use $\vec{h}^{(i+1)} = \vec{h}^{(i)} - \mu\nabla_{\vec{h}}\varepsilon$

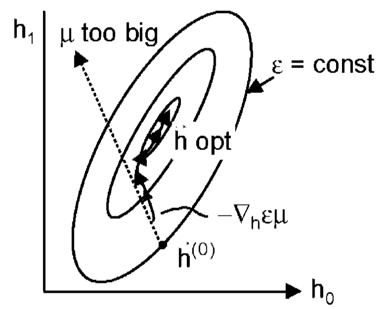
$$\vec{h}^{(i+1)} = \vec{h}^{(i)} - \mu(2R_{xx}\vec{h}^{(i)} - 2\vec{p})$$

$$= \vec{h}^{(i)} - 2\mu(R_{xx}\vec{h}^{(i)} - \vec{p})$$

↑ $\theta(m^2)$ operations still!

Steepest Decent Approach, cont'd

$\vec{h}^{\infty} = \vec{h}$ opt (hopefully!)



Steepest Decent Approach, cont'd

Will be the same governing convergence for the LMS Algorithm with assumptions

$$\vec{h}^{(i+1)} = \vec{h}^{(i)} + 2\mu(\vec{p} - R_{xx}\vec{h}^{(i)})$$

$$= (I - 2\mu R_{xx})\vec{h}^{(i)} + 2\mu\vec{p}$$

let $\vec{q}^{(i)} = \vec{h}^{(i)} - \vec{h}_{opt}$

Steepest Decent Approach, cont'd

$$\vec{q}^{(i+1)} = (I - 2\mu R_{xx})\vec{h}^{(i)} + 2\mu\vec{p} - R_{xx}^{-1}\vec{p}$$

$$= (I - 2\mu R_{xx})\vec{h}^{(i)} - (I - 2\mu R_{xx})R_{xx}^{-1}\vec{p}$$

$$\leftarrow = (I - 2\mu R_{xx})\vec{q}^{(i)}$$

Will return here next time to study convergence!

Approximate Solution

(Stochastic Gradient)

Q: What if $\nabla_h \varepsilon$ is not available?

- i.e. don't know R_{xx}!

1) Estimate

$$\hat{R}_{xx} = \sum_{N} \vec{x}[n]\vec{x}[n]^T$$

$\rightarrow$ expensive, still $\theta(m^2)$

LMS Algorithm

2) Approximate

$$\nabla_h \varepsilon = 2R_{xx}\vec{h} - 2\vec{p}$$

$$= 2E\{\vec{x}[n]\vec{x}[n]^T\vec{h} - \vec{x}[n]d[n]\}$$

$$= 2E\left\{\vec{x}[n]\big(\underbrace{\vec{x}[n]^T\vec{h} - d[n]}_{-e[n]}\big)\right\}$$

$$\nabla_h \varepsilon = -2E\{\vec{x}[n]e[n]\}$$

Instantaneous Gradient Approximation

LMS (Widrow 1960)

use

$$\nabla_{\vec{h}} \varepsilon \cong \underbrace{-2\vec{x}[n]e[n]}_{\theta(m)}$$

LMS Algorithm, cont'd

Guess $\vec{h}^{(0)}$

$$e[n] = d[n] - \vec{h}^{(n)T}\vec{x}[n]$$

$$\vec{g}[n] = -2\vec{x}[n]e[n]$$

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} - \mu\vec{g}[n]$$

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu e[n]\vec{x}[n]$$

Convergence

Q: 1) if x, d jointly WSS does $\vec{h}^{(i)} \to \vec{h}_{opt}$?

A: Need to put $\mu = \mu[n]$

And $\sum\mu[n] = \infty$

$\sum\mu^2[n] < \infty$

$\Rightarrow \vec{h}^{(n)} \to \vec{h}_{opt}$

Robbins-Munro "Stochastic Approximation"

LMS Convergence Speed

2) If it converges, how fast?

Can we optimally select $\mu[n]$?

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 29

Professor Andrew Singer
Department of Electrical and Computer Engineering

MMSE Optimal Filter

- Find $$\vec{h}_{opt} = \underset{\vec{h}}{\arg\min}\, \mathcal{E} = E\left\{\left(d - \hat{d}[n]\right)^2\right\}$$
$$= \underset{\vec{h}}{\arg\min}\, E\left\{\left(d - \vec{h}^T\vec{x}[n]\right)^2\right\}$$
- Can solve directly using $O(M^2)$ operations (how?)
$$R_{xx}\vec{h}_{opt} = \vec{p}$$

Steepest Decent Approach

$-\nabla_{\vec{h}}\varepsilon$ points in direction of "steepest decent"

Guess: $\vec{h}^{(0)}$

update: $\vec{h}^{(i+1)} = \vec{h}^{(i)} - \mu\nabla_{\vec{h}}\varepsilon$

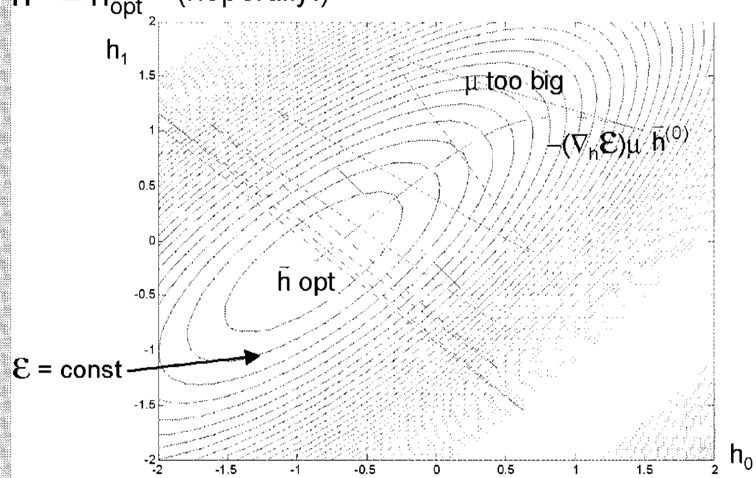
$$\vec{h}^{(i+1)} = \vec{h}^{(i)} - \mu(2R_{xx}\vec{h}^{(i)} - 2\vec{p})$$

$$= \vec{h}^{(i)} - 2\mu(R_{xx}\vec{h}^{(i)} - \vec{p})$$

$O(M^2)$ operations still!

2-tap Example

$\vec{h}^{\infty} = \vec{h}_{opt}$ (hopefully!)



Convergence?

$$\vec{h}^{(i+1)} = \vec{h}^{(i)} + 2\mu(\vec{p} - R_{xx}\vec{h}^{(i)})$$

$$= (I - 2\mu R_{xx})\vec{h}^{(i)} + 2\mu\vec{p}$$

let $\vec{q}^{(i)} = \vec{h}^{(i)} - \vec{h}_{opt}$

Evolution of the error vector

$$\vec{q}^{(i+1)} = \underbrace{(I - 2\mu R_{xx})\vec{h}^{(i)} + 2\mu\vec{p}}_{\vec{h}^{(i+1)}} - \underbrace{R_{xx}^{-1}\vec{p}}_{\vec{h}_{opt}}$$

$$= (I - 2\mu R_{xx})\vec{h}^{(i)} - (I - 2\mu R_{xx})R_{xx}^{-1}\vec{p}$$

$$= (I - 2\mu R_{xx})\vec{q}^{(i)}$$

$$= (I - 2\mu R_{xx})^{(i+1)}\vec{q}^{(0)}$$

Evolution of the error vector

$$\vec{q}^{(i+1)} = (I - 2\mu R_{xx})^{(i+1)}\vec{q}^{(0)}$$

- Error vector converges to 0 when
  $(I-2\mu R_{xx})^{(i+1)} = A^{(i+1)} \rightarrow 0$
- Steepest decent method converges when $A^{(i+1)} \rightarrow 0$
- We will return to this when we consider convergence of the LMS algorithm

Approximate Solution

(Stochastic Gradient)

Q: What if $\nabla_h \mathcal{E}$ is not available?

- i.e. don't know R_{xx}!

1) We can estimate

$$\hat{R}_{xx} = \sum_{N} \vec{x}[n]\vec{x}[n]^T$$

$\rightarrow$ expensive, still $O(M^2)$

LMS Algorithm

2) Approximate the Gradient

$$\nabla_{\vec{h}}\mathcal{E} = 2R_{xx}\vec{h} - 2\vec{p}$$

$$= 2E\{\vec{x}[n]\vec{x}[n]^T\vec{h} - \vec{x}[n]d[n]\}$$

$$= 2E\left\{\vec{x}[n]\underbrace{\left(\vec{x}[n]^T\vec{h} - d[n]\right)}_{-e[n]}\right\}$$

$$\nabla_{\vec{h}}\mathcal{E} = -2E\{\vec{x}[n]e[n]\}$$

Instantaneous Gradient Approximation

"Least Mean Square" (LMS) Algotithm (Widrow 1960)

use "stochastic gradient"

$$\nabla_{\vec{h}}\mathcal{E} \cong \underbrace{-2\vec{x}[n]e[n]}_{O(M)}$$

LMS Algorithm

Initialize: $\vec{h}^{(0)}$

Update: $e[n] = d[n] - \vec{h}^{(n)T}\vec{x}[n]$

$$\vec{g}[n] = -2\vec{x}[n]e[n]$$

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} - \mu\vec{g}[n]$$

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu\vec{x}[n]e[n]$$

Convergence

Q: 1) if x[n], d[n] jointly WSS does $\vec{h}^{(i)} \to \vec{h}_{opt}$?

A: Need to put $\mu = \mu[n]$

And $\sum\mu[n] = \infty$

$\sum\mu^2[n] < \infty$

$\Rightarrow \vec{h}^{(n)} \to \vec{h}_{opt}$

Robbins-Munro “Stochastic Approximation”

LMS Convergence Speed

2) If it converges, how fast?

Can we optimally select μ, or $\mu[n]$?

What forms of convergence?

- Convergence in the mean
- Convergence in the mean-square

LMS Algorithm

1) Initialize: $\vec{h}^{(0)}$

2) Repeat: $\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu e[n]\vec{x}[n]$

$$= \vec{h}^{(n)} + 2\mu[d[n] - \vec{x}[n]^T \vec{h}^{(n)}]\vec{x}[n]$$

LMS Algorithm: Mean Analysis

1) Assume x[n], d[n] JWSS, 0-mean

$$E\{\vec{h}^{(n)}\} \overset{?}{\rightarrow} \vec{h}_{opt} = R_{xx}^{-1}\vec{p}$$

$$E\{\vec{h}^{(n+1)}\} = E\{\vec{h}^{(n)}\} + 2\mu E\{[d[n] - \vec{x}[n]^T \vec{h}^{(n)}]\vec{x}[n]\}$$

$$= E\{(I - 2\mu\vec{x}[n]\vec{x}[n]^T)\vec{h}^{(n)}\} + 2\mu E\{d[n]\vec{x}[n]\}$$

LMS Mean Analysis is Mean

Since $\vec{h}^{(n)} = f(\vec{x}[n])$ nonlinear, time varying

function of a stochastic process

$\Rightarrow$ Very hard to analyze!

If $\mu << 1 \Rightarrow$ lots of averaging

$\vec{h}$ changes slowly about mean

recent data $\approx$ independent of $\vec{h}^{(n)}$

"Independence" Assumptions

- Assume $\vec{h}^{(n)}$ independent of x[t], t = 0, 1, …, n

Blatantly false! $\vec{h}^{(n)}$ dependent on x[n] even if x[n] is white!

→ More rigorous analysis has shown that $E\{\vec{h}^{(n)}\}$ converges when μ is sufficiently small with weaker (bounded memory) assumptions. e.g. Macchi & Eweda 1985

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 30

Professor Andrew Singer
Department of Electrical and Computer Engineering

LMS Algorithm

1) Initialize: $\vec{h}^{(0)}$

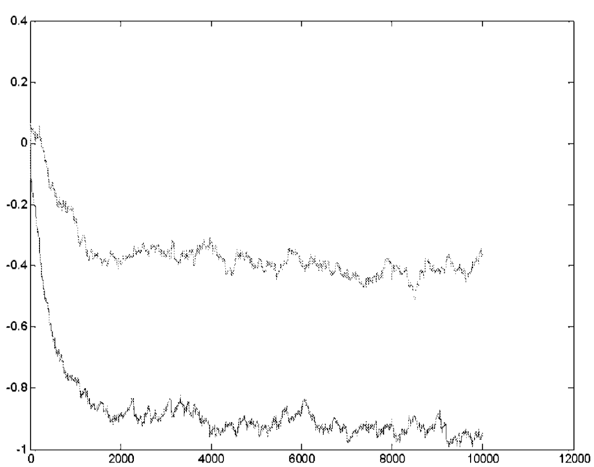
2) Repeat: $\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu e[n]\vec{x}[n]$

$$= \vec{h}^{(n)} + 2\mu[d[n] - \vec{x}[n]^T \vec{h}^{(n)}]\vec{x}[n]$$

LMS Algorithm in Matlab

```
% create a signal x[n] to be predicted
x=filter(1,[1 .9 .4],randn(10000,1));

% LMS 2nd order prediction algorithm
% storing the filter coefficients each time

for k=3:10000
  xhat(k)=h(:,k)'*x(k-1:-1:k-2);
  e(k)=x(k)-xhat(k);
  h(:,k+1)=h(:,k)+2*(.001)*x(k-1:-1:k-2)*e(k);
end
```


LMS Filter Coefficients



LMS Algorithm Convergence

Make assumptions: x[n], d[n] JWSS, 0-mean random processes

Q: Does $\vec{h}^{(n)}$ converge?

To what?

How fast?

LMS Algorithm, Convergence in Mean

1) Mean Analysis:

$$E\{\vec{h}^{(n)}\} \overset{?}{\rightarrow} \vec{h}_{opt} = R_{xx}^{-1}\vec{p}$$

$$E\{\vec{h}^{(n+1)}\} = E\{\vec{h}^{(n)}\} + 2\mu E\{[d[n] - \vec{x}[n]^T \vec{h}^{(n)}]\vec{x}[n]\}$$

$$= E\{(I - 2\mu\vec{x}[n]\vec{x}[n]^T)\vec{h}^{(n)}\} + 2\mu E\{d[n]\vec{x}[n]\}$$

"Independence" Assumptions

- Assume $\vec{h}^{(n)}$ independent of x[t], t = 0, 1, …, n

Blatantly false! $\vec{h}^{(n)}$ dependent on x[n] even if x[n] is white!

→ More rigorous analysis has shown that $E\{\vec{h}^{(n)}\}$ → converges when μ is sufficiently small with out these assumptions. Macchi & Eweda 1985

Using Independence Assumptions

$$E\left\{\vec{h}^{(n+1)}\right\} \approx (I - 2\mu \underbrace{E\{\vec{x}[n]\vec{x}[n]^T\}}_{R_{xx}})E\left\{\vec{h}^{(n)}\right\} + 2\mu \underbrace{E\{\vec{x}[n]d[n]\}}_{\vec{p}}$$

$$E\left\{\vec{q}^{n+1}\right\} = \underbrace{(I - 2\mu R_{xx})E\left\{\vec{h}^{(n)}\right\} + 2\mu\vec{p}}_{E\left\{\vec{h}^{(n+1)}\right\}} - \underbrace{R_{xx}^{-1}\vec{p}}_{\vec{h}_{opt}}$$

$$= (I - 2\mu R_{xx})E\left\{\vec{h}^{(n)}\right\} - (I - 2\mu R_{xx})R_{xx}^{-1}\vec{p}$$

$$= (I - 2\mu R_{xx})E\left\{\vec{q}^{n}\right\}$$

Diagonalize R_{xx}

$R_{xx} = V\Lambda V^T$ symm. (assumed Pos. Def.)

$\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$

V = ortho-normal matrix

$$\vec{v}_i^T \vec{v}_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$$\rightarrow V^T = V^{-1}$$

Change of Basis

Change of variables from:

$$\vec{q}^n = \vec{h}^{(n)} - \vec{h}_{opt}$$

"Change of Basis" or "rotation of coordinates"

$$\begin{cases} \tilde{q}^n = V^T \vec{q}^n \\ \tilde{x}[n] = V^T \vec{x}[n] \end{cases}$$

Independent Coordinates

Now $E\{\tilde{x}[n]\tilde{x}[n]^T\} = V^T R_{xx} V$

$$= V^T V \Lambda V^T V$$

$$= \Lambda \leftarrow \text{ diagonal}$$

Coordinates of $\tilde{x}[n]$ are independent

"KL" transform

$$E\{\tilde{q}^{n+1}\} = (I - 2\mu\Lambda)E\{\tilde{q}^n\}$$

n Uncoupled Modes

$$E\{\tilde{q}_j^{n+1}\} = (1 - 2\mu\lambda_j)^{n+1}\tilde{q}_j^0$$

j^{th} component

For convergence: $-1 < 1 - 2\mu\lambda_j < 1$

$$-2 < -2\mu\lambda_j < 0$$

$$\frac{1}{\lambda_j} > \mu, \forall j$$

Mean coefficient vector error: $E\{\vec{q}^n\} = VE\{\tilde{q}^n\}$

Convergence in the Mean

$$\Rightarrow E\{\vec{q}^{n}\} \to 0 \qquad \text{exponentially}$$

$$\text{provided} \qquad \mu < \frac{1}{\lambda_{max}}$$

Sure, but how fast?

j^{th}-mode time constant:

$$\tau_j = \text{time } E\{q_j^n\} = \frac{1}{e} q_j^0$$

$$\to \tau_j \approx \frac{-1}{\ell n(1-2\mu\lambda_j)} \approx \frac{1}{2\mu\lambda_j}$$

Convergence time limited by λ_{min}

$$\therefore E\{\vec{h}^n\} \approx \vec{h}_{opt} + \underbrace{V(I - 2\mu\Lambda)^n \tilde{q}^0}$$

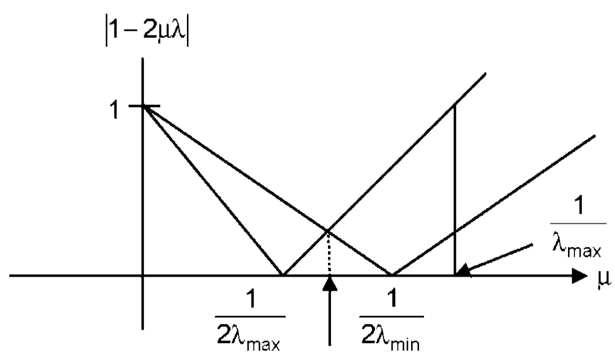
error term $\rightarrow 0$ for sufficiently small μ

convergence time limited by

$\boxed{\lambda_{min}}$

"slowest direction" of convergence.

Choose μ for Fastest Convergence



$$\mu_{opt} = \frac{1}{\lambda_{min} + \lambda_{max}} \Rightarrow \text{fastest convergence}$$

Modes

Now modes for λ_{max} & λ_{min} go as ~

$$\left(\frac{\frac{\lambda_{max}}{\lambda_{min}}-1}{\frac{\lambda_{max}}{\lambda_{min}}+1}\right)^n$$

"e-value" spread

Spectral Variation

Fact

$$\min_{\omega} S_{xx}(e^{j\omega}) < \lambda_i < \max_{\omega} S_{xx}(e^{j\omega})$$

$$\lambda_{max} \underset{M\to\infty}{\longrightarrow} \max_{\omega} S_{xx}(e^{j\omega})$$

$$\lambda_{min} \underset{M\to\infty}{\longrightarrow} \min_{\omega} S_{xx}(e^{j\omega})$$

$\Rightarrow$ Lots of spectral variation in x[n]

- $\rightarrow$ slow convergence!

Change the step-size with time

Normalized step size:

$$2\mu[n] = \frac{1}{\sigma^2[n]} = \frac{1}{(1-\alpha)\sigma^2[n-1] + x^2[n]}$$

$$\left(\text{Note}: \hat{\sigma}_n^2 = \hat{R}_{xx}(0) = \frac{\sum \lambda_i}{N}\right)$$

$$0 < \alpha << 1 \quad \text{as } n \to \infty \; E\left\{\frac{1}{2\mu[n]}\right\} \to \frac{E\{x^2[n]\}}{\alpha}$$

Adaptive Step Size

Replace 2μ with

$$E\{2\mu[n]\} \approx \frac{1}{E\{\sigma_n^2\}}$$

$$2\mu[n] = \frac{1}{\sigma^2[n]} = \frac{1}{\frac{1}{N}\sum_{j=1}^{N} \lambda j}$$

Adaptive Step Size, cont'd

$$\sigma_0^2 = \left(\frac{\alpha}{R_{xx}(0)}\right)^{-1} \Rightarrow E\{2\mu(n)\} \approx \frac{\alpha}{R_{xx}(0)}$$

$$\Rightarrow \tau_j \approx \frac{R_{xx}(0)}{\alpha\lambda_j}$$

$$= \frac{\sum_{j=1}^{n}\lambda_j}{N\alpha\lambda_j} = \frac{\lambda_{avg}}{\alpha\lambda_j}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 31

Professor Andrew Singer
Department of Electrical and Computer Engineering

LMS Algorithm: Mean Convergence

1) Initialize: $\vec{h}^{(0)}$

2) Repeat: $\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu\left(d[n] - \vec{x}[n]^T \vec{h}^{(n)}\right)\vec{x}[n]$

$\Rightarrow E\{\vec{h}^n\} \rightarrow \vec{h}_{opt}$ exponentially

provided $\mu < \dfrac{1}{\lambda_{max}}$

Sure, but how fast?

j^{th}-mode time constant:

$$\tau_j = \text{time it takes for } E\{q_j^n\} = \tfrac{1}{e} q_j^0$$

$$E\{q_j^\tau\} = q_j^0 (1 - 2\mu\lambda_j)^\tau = \tfrac{1}{e} q_j^0$$

$$\tau_j \ln(1 - 2\mu\lambda_j) = -1$$

$$\rightarrow \tau_j = \frac{-1}{\ln(1 - 2\mu\lambda_j)} \approx \frac{1}{2\mu\lambda_j}$$

Convergence time limited by λ_{min}

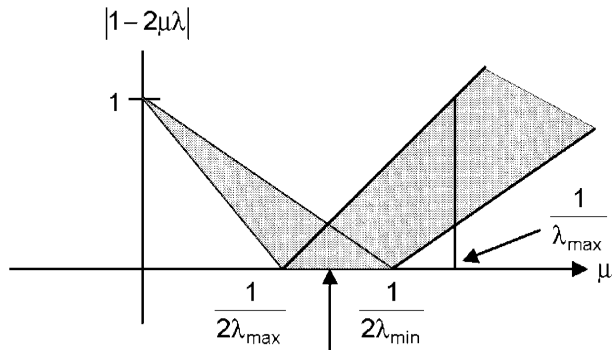
$$\therefore E\{\vec{h}^n\} \approx \vec{h}_{opt} + \underbrace{V(I - 2\mu\Lambda)^n \tilde{q}^0}$$

error term $\rightarrow 0$ for sufficiently small μ

convergence time limited by $\boxed{\lambda_{min}, \mu}$

- "slowest direction" of convergence
- speed up by decreasing $|1\text{-}2\mu\lambda_j|$ for all j

Choose μ for Fastest Convergence



$$\mu_{opt} = \frac{1}{\lambda_{min} + \lambda_{max}} \Rightarrow \text{fastest convergence}$$

Slowest Modes for μ_{opt}

Now modes for λ_{max} & λ_{min} go as ~

$$\left|1 - 2\left(\frac{1}{\lambda_{min} + \lambda_{max}}\right)\lambda_m\right|^n = \left(\frac{\frac{\lambda_{max}}{\lambda_{min}} - 1}{\frac{\lambda_{max}}{\lambda_{min}} + 1}\right)^n$$

"e-value" spread

Spectral Variation

Fact: $\min_{\omega} S_{xx}(e^{j\omega}) < \lambda_i < \max_{\omega} S_{xx}(e^{j\omega})$

(can you show this?)

$\lambda_i = \frac{v_i^H R_{xx} v_i}{v_i^H v_i}$

$$\lambda_{max} \underset{M\to\infty}{\longrightarrow} \max_{\omega} S_{xx}(e^{j\omega})$$

$$\lambda_{min} \underset{M\to\infty}{\longrightarrow} \min_{\omega} S_{xx}(e^{j\omega})$$

$\Rightarrow$Lots of spectral variation in x[n]

$\rightarrow$ slow convergence!

Change the step-size with time

Normalized step size:

$$2\mu[n] = \frac{1}{\sigma^2[n]} = \frac{1}{(1-\alpha)\sigma^2[n-1] + x^2[n]}$$

$$\left(\text{Note}: \hat{\sigma}_n^2 = \hat{R}_{xx}(0) = \frac{\sum \lambda_i}{N}\right)$$

$$0 < \alpha << 1 \quad \text{as } n \to \infty \ E\left\{\frac{1}{2\mu[n]}\right\} \to \frac{E\{x^2[n]\}}{\alpha}$$

Adaptive Step Size

Replace 2μ with

$$2\mu[n] = \frac{1}{\sigma^2[n]} = \frac{1}{(1-\alpha)\sigma^2[n-1] + x^2[n]}$$

$$E\{2\mu[n]\} \approx \frac{1}{E\{\sigma_n^2\}}$$

Adaptive Step Size, cont'd

$$E\{\sigma_n^2\} = \left(\frac{\alpha}{R_{xx}(0)}\right)^{-1} \Rightarrow E\{2\mu(n)\} \approx \frac{\alpha}{R_{xx}(0)}$$

$$\Rightarrow \tau_j \approx \frac{R_{xx}(0)}{\alpha\lambda_j}$$

$$= \frac{\sum_{j=1}^{n}\lambda_j}{N\alpha\lambda_j} = \frac{\lambda_{avg}}{\alpha\lambda_j}$$

Output MSE 2^{nd} Order Convergence

First: Given $\vec{h}^{(n)}$: (i.e. condition on $\vec{h}^{(n)}$)

$$\varepsilon[n] \triangleq E\{e^2[n]\}$$

$$= E\left\{(d[n] - \vec{h}^{(n)T}\vec{x}[n])^2\right\}$$

$$= E\left\{d^2[n] - 2d[n]\vec{h}^{(n)T}\vec{x}[n] + \vec{h}^{(n)T}\vec{x}[n]\vec{x}[n]^T\vec{h}^{(n)}\right\}$$

$$= E\{d^2[n]\} - 2\vec{h}^{(n)T}\vec{p} + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

Output MMSE

Note: $\varepsilon_{min} = E\{(d[n] - \vec{h}_{opt}^T\vec{x}[n])^2\}$

$$= E\{d^2[n]\} - 2\vec{h}_{opt}^T\vec{p} + \vec{h}_{opt}^T R_{xx}\vec{h}_{opt}$$

$$= E\{d^2[n]\} - 2(R_{xx}^{-1}\vec{p})^T\vec{p} + (R_{xx}^{-1}\vec{p})^T R_{xx}(R_{xx}^{-1}\vec{p})$$

$$= E\{d^2[n]\} - 2\vec{p}^T R_{xx}^{-1}\vec{p} + \vec{p}^T R_{xx}^{-1}R_{xx}R_{xx}^{-1}\vec{p}$$

$$\boxed{\varepsilon_{min} = E\{d^2[n]\} - \vec{p}^T R_{xx}^{-1}\vec{p}}$$

Output MSE, cont'd

$$\varepsilon[n] = E\{d^2[n]\} - 2\vec{h}^{(n)T}\vec{p} + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

$$= \varepsilon_{min} + \vec{p}^T R_{xx}^{-1}\vec{p} - 2\vec{h}^{(n)}\vec{p} + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

$$= \varepsilon_{min} + \left(\vec{h}^{optT}R_{xx}\right)R_{xx}^{-1}\left(R_{xx}\vec{h}^{opt}\right) -$$

$$- 2\vec{h}^{(n)}\left(R_{xx}\vec{h}^{opt}\right) + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

Output MSE, cont'd

$$= \varepsilon_{min} + \vec{h}^{optT}R_{xx}\vec{h}^{opt} - 2\vec{h}^{(n)T}R_{xx}\vec{h}^{opt} + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

$$= \varepsilon_{min} + \left(\vec{h}^{(n)} - \vec{h}^{opt}\right)^T R_{xx}\left(\vec{h}^{(n)} - \vec{h}^{opt}\right)$$

Output MSE, cont'd

$$\varepsilon[n] = \varepsilon_{min} + \vec{q}^{(n)^T} R_{xx} \vec{q}^{(n)}$$

(Now this is for the deterministic gradient algorithm, i.e. $\vec{h}^{(n)}$ was assumed given.)

now take

$$E_h\{\varepsilon[n]\} = E\{\varepsilon_{min} + \vec{q}^{(n)T} R_{xx} \vec{q}^{(n)}\}$$

Output MSE Expression

$$= \varepsilon_{min} + E\{\vec{q}^{(n)T} V \Lambda V^T \vec{q}^{(n)}\}$$

$$= \varepsilon_{min} + E\{\tilde{q}^{(n)T} \Lambda \tilde{q}^{(n)}\}$$

$$E\{\varepsilon[n]\} = \varepsilon_{min} + \sum_{i=1}^{N} \lambda_i E\{(\tilde{q}_i^{(n)})^2\}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 32

Professor Andrew Singer
Department of Electrical and Computer Engineering

Output MSE 2nd Order Convergence

First: Given $\vec{h}^{(n)}$: (i.e. condition on $\vec{h}^{(n)}$)

$$\varepsilon[n] \triangleq E\{e^2[n]\}$$

$$= E\left\{(d[n] - \vec{h}^{(n)T}\vec{x}[n])^2\right\}$$

$$= E\left\{d^2[n] - 2d[n]\vec{h}^{(n)T}\vec{x}[n] + \vec{h}^{(n)T}\vec{x}[n]\vec{x}[n]^T\vec{h}^{(n)}\right\}$$

$$= E\{d^2[n]\} - 2\vec{h}^{(n)T}\vec{p} + \vec{h}^{(n)T}R_{xx}\vec{h}^{(n)}$$

Output MMSE

Note: $\varepsilon_{min} = E\{(d[n] - \vec{h}_{opt}^T \vec{x}[n])^2\}$

$$= E\{d^2[n]\} - 2\vec{h}_{opt}^T \vec{p} + \vec{h}_{opt}^T R_{xx} \vec{h}_{opt}$$

$$= E\{d^2[n]\} - 2(R_{xx}^{-1}\vec{p})^T \vec{p} + (R_{xx}^{-1}\vec{p})^T R_{xx} (R_{xx}^{-1}\vec{p})$$

$$= E\{d^2[n]\} - 2\vec{p}^T R_{xx}^{-1} \vec{p} + \vec{p}^T R_{xx}^{-1} R_{xx} R_{xx}^{-1} \vec{p}$$

$$\boxed{\varepsilon_{min} = E\{d^2[n]\} - \vec{p}^T R_{xx}^{-1} \vec{p}}$$

Output MSE, cont'd

$$\varepsilon[n] = E\{d^2[n]\} - 2\vec{h}^{(n)T} \vec{p} + \vec{h}^{(n)T} R_{xx} \vec{h}^{(n)}$$

$$= \varepsilon_{min} + \vec{p}^T R_{xx}^{-1} \vec{p} - 2\vec{h}^{(n)} \vec{p} + \vec{h}^{(n)T} R_{xx} \vec{h}^{(n)}$$

$$= \varepsilon_{min} + \left(\vec{h}_{opt}^T R_{xx}\right) R_{xx}^{-1} \left(R_{xx} \vec{h}_{opt}\right) -$$

$$- 2\vec{h}^{(n)} \left(R_{xx} \vec{h}_{opt}\right) + \vec{h}^{(n)T} R_{xx} \vec{h}^{(n)}$$

Output MSE, cont'd

$$= \varepsilon_{min} + \vec{h}_{opt}^T R_{xx} \vec{h}_{opt} - 2\vec{h}^{(n)T} R_{xx} \vec{h}_{opt} + \vec{h}^{(n)T} R_{xx} \vec{h}^{(n)}$$

$$= \varepsilon_{min} + \left(\vec{h}^{(n)} - \vec{h}_{opt}\right)^T R_{xx} \left(\vec{h}^{(n)} - \vec{h}_{opt}\right)$$

Output MSE, cont'd

$$\varepsilon[n] = \varepsilon_{min} + \vec{q}^{(n)^T} R_{xx} \vec{q}^{(n)}$$

(Now this is for the deterministic gradient algorithm, i.e. $\vec{h}^{(n)}$ was assumed given.)

now take

$$E_h\{\varepsilon[n]\} = E\{\varepsilon_{min} + \vec{q}^{(n)T} R_{xx} \vec{q}^{(n)}\}$$

Output MSE Expression

$$= \varepsilon_{min} + E\{\vec{q}^{(n)T} V \Lambda V^T \vec{q}^{(n)}\}$$

$$= \varepsilon_{min} + E\{\tilde{q}^{(n)T} \Lambda \tilde{q}^{(n)}\}$$

$$\boxed{E\{\varepsilon[n]\} = \varepsilon_{min} + \sum_{i=1}^{N} \lambda_i E\{(\tilde{q}_i^{(n)})^2\}}$$

Coefficient Error Covariance Matrix

Define $Q^{(n)} = E\{\tilde{q}^{(n)} \tilde{q}^{(n)T}\}$

LMS:

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} + 2\mu e[n]\vec{x}[n]$$

$$\vec{h}^{(n+1)} - \vec{h}_{opt} = \vec{h}^{(n)} - \vec{h}_{opt} + 2\mu e[n]\vec{x}[n]$$

$$\vec{q}^{(n+1)} = \vec{q}^{(n)} + 2\mu e[n]\vec{x}[n]$$

Error Covariance Matrix

$$V^T\vec{q}^{(n+1)} = V^T\vec{q}^{(n)} + 2\mu e[n]V^T\vec{x}[n]$$

$$\tilde{q}^{(n+1)} = \tilde{q}^{(n)} + 2\mu e[n]\tilde{x}[n]$$

$$Q^{(n+1)} = E\{\tilde{q}^{(n+1)}\tilde{q}^{(n+1)T}\} = E\{\tilde{q}^{(n)}\tilde{q}^{(n)T} + 2\mu \underbrace{e[n]\tilde{x}[n]\tilde{q}^{(n)T}}_{1}$$

$$+ 2\mu \underbrace{\tilde{q}^{(n)}e[n]\tilde{x}[n]^T}_{2} \qquad + 4\mu^2 \underbrace{e^2[n]\tilde{x}[n]\tilde{x}[n]^T}_{3}\}$$

Take them one term at a time

$$Q^{(n+1)} = E\left\{\tilde{q}^{(n)}\tilde{q}^{(n)T}\right\} + 2\mu(1) + 2\mu(2) + 4\mu^2(3)$$

$$(2) \quad = E\{\tilde{q}^{(n)}e[n]\tilde{x}[n]^T\}$$

$$= E\{\tilde{q}^{(n)}(d[n] - \vec{h}^{(n)T}\vec{x}[n])\vec{x}[n]^T V\}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 33

Professor Andrew Singer

Department of Electrical and Computer Engineering

Error Covariance Matrix

$$V^T\vec{q}^{(n+1)} = V^T\vec{q}^{(n)} + 2\mu e[n]V^T\vec{x}[n]$$

$$\tilde{q}^{(n+1)} = \tilde{q}^{(n)} + 2\mu e[n]\tilde{x}[n]$$

$$Q^{(n+1)} = E\{q^{(n+1)}q^{(n+1)T}\} = E\{q^{(n)}q^{(n)T} + \underbrace{2\mu e[n]x[n]q^{(n)T}}_{1}$$

$$\underbrace{\quad {}^{(n)} \quad {}^{T}}_{2} \qquad \underbrace{{}^{2}\,{}^{2} \quad {}^{T}}_{3}$$

Take them one term at a time

$$Q^{(n+1)} = E\left\{\tilde{q}^{(n)}\tilde{q}^{(n)T}\right\} + 2\mu(1) + 2\mu(2) + 4\mu^2(3)$$

$$(2) \quad = E\{\tilde{q}^{(n)} e[n]\tilde{x}[n]^T\}$$

$$= E\{\tilde{q}^{(n)}(d[n] - \vec{h}^{(n)T}\vec{x}[n])\vec{x}[n]^T V\}$$

Output MSE term (2) , cont'd

$$= E\{\tilde{q}^{(n)}(d[n] - (\vec{h}_{opt}^T + \vec{q}^{(n)T})\vec{x}[n])\vec{x}[n]^T V\}$$

$$= E\{\tilde{q}^{(n)}(d[n]\dot{x}^T[n] - \vec{h}_{opt}^T\dot{x}[n]\dot{x}^T[n])\}V$$

$$- E\{\tilde{q}^{(n)}\vec{q}^{(n)T}\dot{x}[n]\dot{x}^T[n]V\}$$

Output MSE, term (2) cont'd

$$= E\{\tilde{q}^{(n)}\}E\{\underbrace{d[n]\vec{x}[n]^T - \vec{h}_{opt}^T\vec{x}[n]\vec{x}[n]^T}_{0}\}V$$

$$- E\{\tilde{q}^{(n)}\vec{q}^{(n)T}\vec{x}[n]\vec{x}[n]^T V\}$$

$$= -E\{\tilde{q}^{(n)}\vec{q}^{(n)T}\vec{x}[n]\vec{x}[n]^T V\}$$

Output MSE, term (2) cont'd

$$= -E\{\tilde{q}^{(n)}\vec{q}^{(n)T}VV^T\vec{x}[n]\vec{x}[n]^T V\}$$

$$= -E\{\tilde{q}^{(n)}\tilde{q}^{(n)T}V^T\vec{x}[n]\vec{x}[n]^T V\}$$

$$= -E\{\tilde{q}^{(n)}\tilde{q}^{(n)T}\}V^T R_{xx} V$$

Output MSE, term (2) cont'd

$$(2) = -Q^{(n)}\Lambda = (1)^T$$

$$(1) = -\Lambda^T Q^{(n)T} = -\Lambda Q^{(n)}$$

(3) $E\{e^2[n]\tilde{x}[n]\tilde{x}[n]^T\}$ ⟵ product of 4 variables

Output MSE, term (3) cont'd

If Gaussian:

Use Moment factoring:

$$E\{x_1\, x_2\, x_3\, x_4\} = E\{x_1\, x_2\}\, E\{x_3\, x_4\} + E\{x_1\, x_3\}\, E\{x_2\, x_4\} + E\{x_1\, x_4\}\, E\{x_2\, x_3\}$$

Output MSE, term (3) cont'd

If not Gaussian $\Rightarrow$ Trouble

More independence assumptions. (BFA)

Note: Can do with more sophisticated analysis, see Gardner 1984, but this is easier and the results are similar

BFA Justification

as $h \rightarrow h^{\infty}$,

e[n] becomes random with respect to x[n]

$$E\{x[n]e[n]\} = E\{x[n](d[n] - x[n]^T h^{opt})\}$$

$$= p - R_{xx} h^{opt}$$

This is the orthogonality principle: error $\perp$ data

Output MSE, term (3) cont'd

$$(3) \quad = E\{e^2[n]\tilde{x}[n]\tilde{x}[n]^T\}$$

$$= E\{e^2[n]\}E\{\tilde{x}[n]\tilde{x}[n]^T\}$$

$$= E\{e^2[n]\}V^T R_{xx} V$$

$$= E\{e^2[n]\}V$$

Putting them all together

$$3) \quad = \left(\varepsilon_{min} + \sum_{j=1}^{N} \lambda_j Q^{(n)}[j,j] \right) \Lambda$$

$$Q^{(n+1)} = Q^{(n)} + 2\mu(-\Lambda Q^{(n)}) + 2\mu(-Q^{(n)}\Lambda) + $$
$$+ 4\mu^2 \left(\varepsilon_{min} + \sum_{j=1}^{N} \lambda_j Q^{(n)}[j,\ j] \right) \Lambda$$

Matrix difference equation

Error Covariance Matrix Difference Equation

$$Q^{(n+1)} = (I - 2\mu\Lambda)Q^{(n)} - 2\mu Q^{(n)}\Lambda$$

$$+ 4\mu^2\left(\varepsilon_{min} + \sum_{j=1}^{N}\lambda_j Q^{(n)}[j,j]\right)\Lambda$$

Diagonal Terms of Matrix

Diagonial terms: (decoupled)

$$Q^{(n+1)}[j, j] = (1 - 4\mu\lambda_j)Q^{(n)}[j, j] +$$

$$+ 4\mu^2\left[\varepsilon_{min} + \sum^{N}\lambda_\ell Q^{(n)}[\ell, \ell]\right]\lambda_j$$

If it converges, then $Q^{(n)} \rightarrow Q^{\infty}$

Diagonal terms, cont'd

$$Q^{\infty}[j,j] = (1 - 4\mu\lambda_j)Q^{\infty}[j,j] +$$

$$+ 4\mu^2\left[\varepsilon_{min} + \sum_{\ell=1}^{N}\lambda_\ell Q^{\infty}[\ell,\ell]\right]\lambda_j$$

$$Q^{\infty}[j,j](1 - 1 + 4\mu\lambda_j) = 4\mu^2\lambda_j\left[\varepsilon_{min} + \sum_{\ell}\lambda_\ell Q^{\infty}[\ell,\ell]\right]$$

ECE 451 - Lecture 33 14 of 21

Diagonal Terms are Decoupled

$$Q^{\infty}[j,j] = \mu\left[\varepsilon_{min} + \sum_{\ell}\lambda_\ell Q^{\infty}[\ell,\ell]\right]$$ ← independent of j!

$$\therefore Q^{\infty}[j,j]\left[1 - \mu\sum_{\ell}\lambda_\ell\right] = \mu\varepsilon_{min} \qquad (?)$$

$$E\{\tilde{q}_j^2\} \rightarrow Q^{\infty}[j,j] = \frac{\mu\varepsilon_{min}}{1 - \mu\sum_{\ell}\lambda_\ell} \qquad j = 1,\ldots,N$$

ECE 451 - Lecture 33 15 of 21

Output MSE, cont'd

$$\therefore \underset{n\to\infty}{E\{e^2[n]\}} = \varepsilon_{min} + \sum_{j=1}^{N} \lambda_j E\{\tilde{q}_j^{\infty 2}\}$$

$$= \varepsilon_{min}\left(1 + \sum_{j=1}^{N} \lambda_j \left(\frac{\mu}{1-\mu\sum_{\ell}\lambda_{\ell}}\right)\right)$$

ECE 451 - Lecture 33 16 of 21

Output MSE, cont'd

$$E\{e^2[\infty]\} = \frac{\varepsilon_{min}}{1-\mu\sum_{\ell}\lambda_{\ell}} = \frac{\varepsilon_{min}}{1-\mu N R_{xx}[0]}$$

$$\Rightarrow \frac{1}{1-\mu N R_{xx}[0]} = \text{"misadjustment factor"}$$

↑ with μ, trade-off MSE vs. speed.

ECE 451 - Lecture 33 17 of 21

Output MSE, cont'd

Now check when it converges.

Off diagonals: $Q_{ij}^{(n+1)} = (1 - 4\mu\lambda_i)Q_{ij}^{(n)}$

decay to zero if $|1 - 4\mu\lambda_i| < 1$

$$\Rightarrow \mu < \frac{1}{2\lambda_{max}}$$

Output MSE, cont'd

Diagonals:

$$Q_{jj}^{(n+1)} = (1 - 4\mu\lambda_j)Q_{jj}^{(n)} + 4\mu^2\lambda_j \sum_{\ell} \lambda_\ell Q_{\ell\ell}^{(n)} + 4\mu^2 \varepsilon_{min}\lambda_j$$

Requires
$$\mu < \frac{1}{\sum \lambda_\ell} = \frac{1}{tr(R)} = \frac{1}{NR_{xx}(0)}$$

Output MSE, cont'd

$$= \frac{1}{N\sigma_x^2}$$

$$\mu < \frac{1}{N\sigma_x^2}$$

(more rigorous analysis leads to $\mu < \frac{1}{3N\sigma_x^2}$)

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

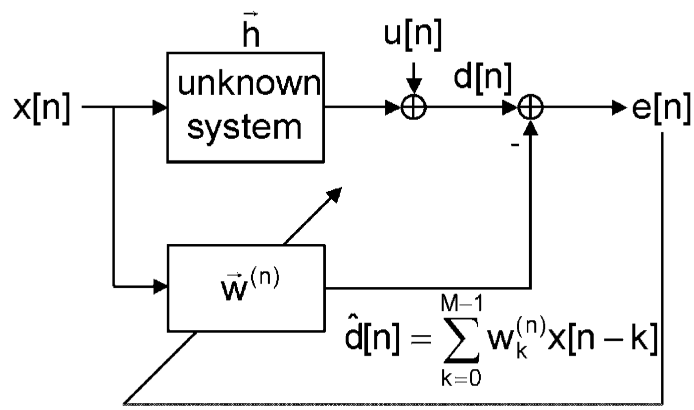
Lecture 34

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Applications of Adaptive Filtering

- Amazing variety of applications
- Same algorithm (LMS) can apply in each
- Need to define d(n) and x(n) appropriately

System Identification



System Identification, cont'd

Know x[n], measure d[n], e[n]
for $\vec{h}$ FIR, u[n] white (uncorrelated with x[n])

$$R^{-1}\vec{p} = w_{opt}$$

$$d[n] = x[n] * h[n] + u[n]$$

$$\underbrace{E\{d[n]\vec{x}[n]\}}_{\vec{p}} = E\{(\vec{h}^T\vec{x}[n] + u[n])\vec{x}[n]\}$$

$$= E\{\vec{x}[n]\vec{x}^T[n]\vec{h}\} + E\{u[n]\vec{x}[n]\}$$

$$= R\vec{h} + 0$$

Converges to the true system

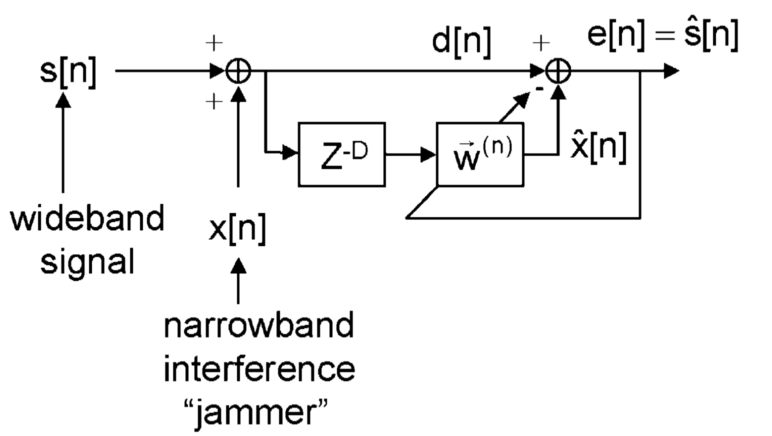
$$\Rightarrow w_{opt} = R^{-1}\vec{p} = R^{-1}(R\vec{h}) = \vec{h}$$

$$\vec{w}^{(n)} \to \vec{h}$$

- if μ small enough
- if M = length (h)

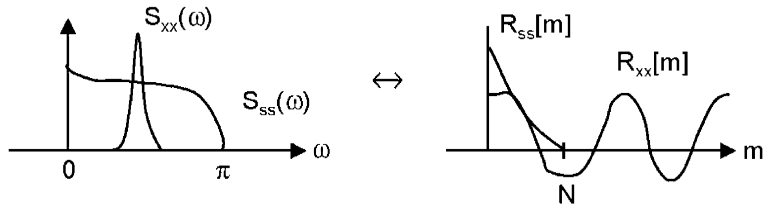
  "model-order selection" problem

Narrowband Interference Suppression



Interference Suppression, cont'd

How does this work?



When D > N, the interference in d[n] is predictable from d[n-D], but s[n] is not!

Interference Suppression, cont'd

$$\vec{w}_{opt} = R_{vv}^{-1}\vec{p}$$

$$\vec{v}[n] = \vec{d}[n-D] = [d[n-D] \ldots d[n-D-M-1]]^T$$

$$R_{vv} = E\left\{\vec{d}[n-D]\vec{d}[n-D]^T\right\}$$

$$= E\left\{(\vec{s}[n-D]+\vec{x}[n-D])(\vec{s}[n-D]+\vec{x}[n-D])^T\right\}$$

$$= R_{ss} + R_{xx} \text{ (since x[n],s[n] are uncorrelated)}$$

Interference Suppression, cont'd

$$\vec{p} = E\left\{d[n]\vec{v}[n]\right\} = E\left\{d[n]\left(\vec{s}[n-D] + \vec{x}[n-D]\right)\right\}$$

$$= E\left\{d[n]\vec{S}[n-D]\right\} + E\left\{d[n]\vec{x}[n-D]\right\}$$

$$= E\left\{\left(s[n] + x[n]\right)\vec{s}[n-D]\right\} + E\left\{\left(s[n] + x[n]\right)\vec{x}[n-D]\right\}$$

Interference Suppression, cont'd

$$E\left\{\left(s[n] + x[n]\right)\vec{s}[n-D]\right\} + E\left\{\left(s[n] + x[n]\right)\vec{x}[n-D]\right\}$$

$$= \begin{bmatrix} R_{ss}[D] \\ R_{ss}[D+1] \\ \vdots \\ R_{ss}[D+M-1] \end{bmatrix} + 0 + 0 + \begin{bmatrix} R_{xx}[D] \\ R_{xx}[D+1] \\ \vdots \\ R_{xx}[D+M-1] \end{bmatrix}$$

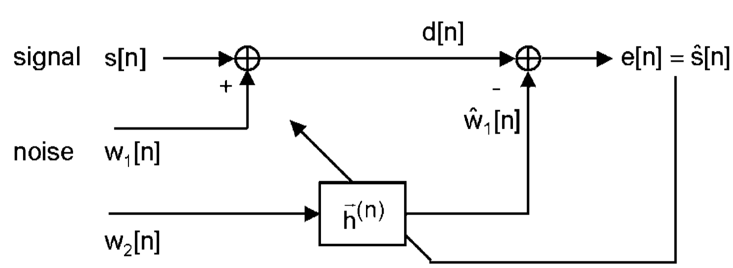
Signal s[n] has short memory

$$\cong \begin{bmatrix} R_{xx}[D] \\ \vdots \quad \vdots \\ R_{xx}[D+M-1] \end{bmatrix}$$

$$\vec{w}_{opt} = (R_{ss} + R_{xx})^{-1} \begin{bmatrix} R_{xx}[D] \\ \vdots \quad \vdots \\ R_{xx}[D+M-1] \end{bmatrix}$$

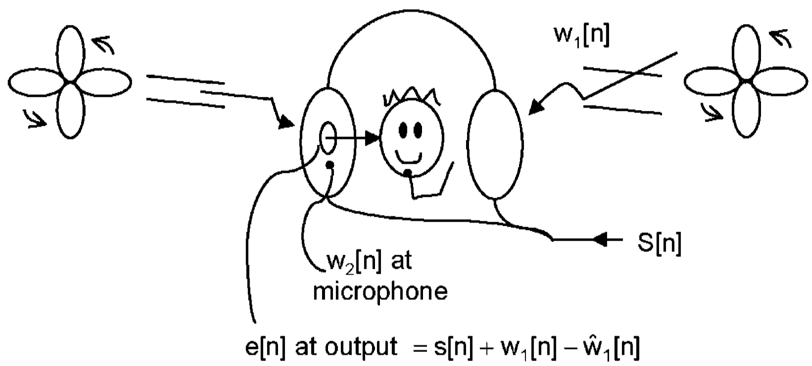
= MMSE estimate of x[n] from v[n]!

Active Noise Cancellation



Measured noise correlated with $w_1[n]$ but not s[n]

Active Noise Cancellation, cont'd



Active Noise Cancellation, cont'd

Also used in

- cars
- volumes! subs, tanks, planes etc…
- classroom!
- video conferencing

Adaptive Noise Cancellation, cont'd

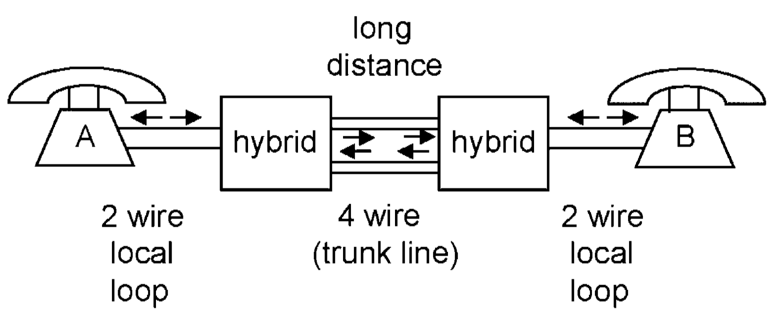
$$\vec{h}_{opt} = R_{w_2 w_2}^{-1}\vec{p} = E\left\{\vec{w}_2[n]\vec{w}_2[n]^T\right\}^{-1} E\left\{d[n]\vec{w}_2[n]\right\}$$

$$= R_{w_2 w_2}^{-1}\left(\cancelto{0}{\vec{r}_{sw_2}} + \vec{r}_{w_1 w_2}\right)$$

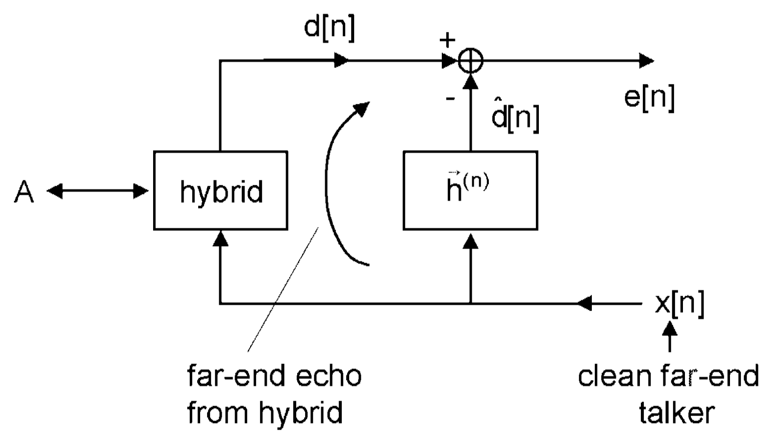
$$= R_{w_2 w_2}^{-1}\vec{r}_{w_1 w_2} = \text{MMSE}$$ estimate of w_1 based on w_2

Echo Cancellation

In telephone networks

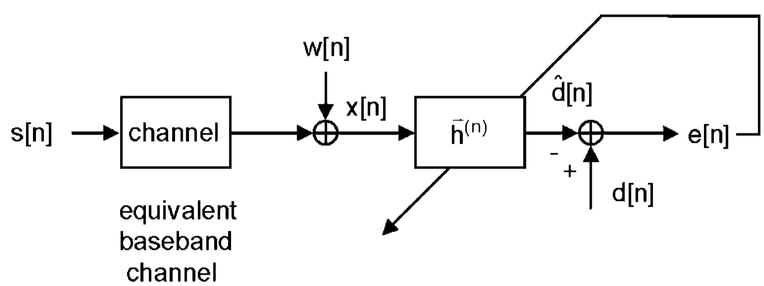


Echo Cancellation, cont'd



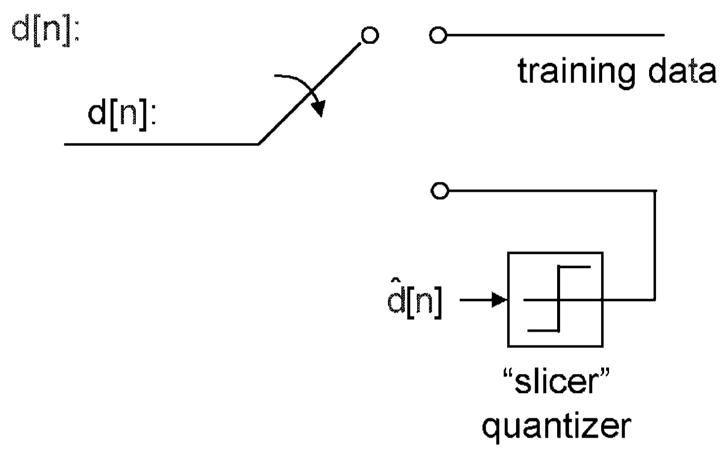
Digital Communications

Linear equalizer as receiver



How can you already know d[n]?

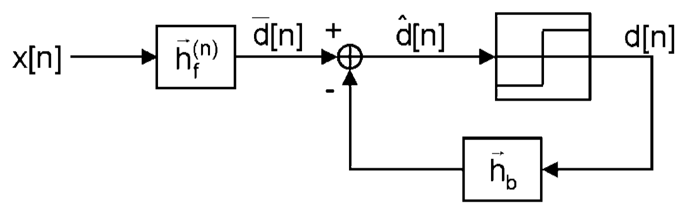
Decision-directed adaptation



Typical Communication Channels

Channel	varying	length
telephone	slowly	few taps
HF radio	rapidly	many taps
wireless	rapidly	few taps
underwater Acoustics	rapidly	many taps

Decision Feedback Equalizer



$$\vec{h} = \begin{bmatrix} \vec{h}_f \\ \vec{h}_b \end{bmatrix} \qquad \vec{X} = \begin{bmatrix} \vec{x}[n] \\ \vec{d}[n] \end{bmatrix}$$

Other Adaptive Filtering Algorithms

Variations on LMS:

a. Signed LMS $\vec{h}^{n+1} = h^n + 2\mu\,\mathrm{sgn}(e[n])\vec{x}[n]$

or $e[n]\,\mathrm{sgn}(\vec{x}[n])$

or $e[n]\,\mathrm{sgn}(e[n])\,\mathrm{sgn}[x[n]]$

- faster to compute, but slower convergence

- larger misadjustment error

Overview, cont'd

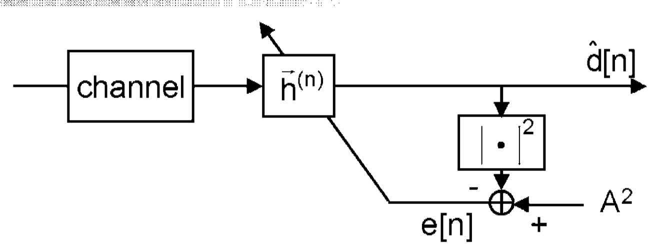
b. Normalized LMS
$$\mu^{(n)} = \frac{\frac{1}{2}}{\vec{x}[n]^T \vec{x}[n] + \varepsilon}, \quad \varepsilon > 0$$

or
$$\mu^{(n)} = \frac{1}{(1-\alpha)\hat{\sigma}_x^2 + x^2[n]}$$

- converges quickly
- can express results for convergence in terms of α!

$$\boxed{\alpha < \alpha_{max} < \frac{2}{N}}$$

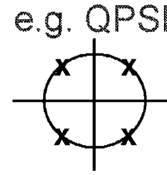
"Property-Restoration" Algorithms



CMA (Godard Algorithm)

→ d[n] not available, but $|d[n]|^2 = A^2$, e.g. QPSK

→ Wire tapping, "blind" equalization



ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

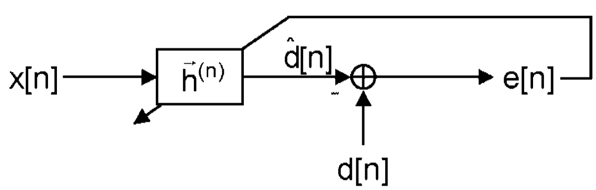
ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 35

Professor Andrew Singer
Department of Electrical and Computer Engineering

Weighted Least Squares Adaptive Filtering

Return to the deterministic case



$$\min \varepsilon[N] = \sum_{n=0}^{N} w_{N-n}\left(d[n] - \hat{d}[n]\right)^2$$

Solve for the optimum filter h

$$\frac{\partial \varepsilon[N]}{\partial h_\ell} \Rightarrow \text{ACNE's}$$

$$\vec{h}_{opt} = \hat{R}_{xx}^{-1}\hat{\vec{p}}$$

$$\hat{R}_{xx} = \sum_{n=0}^{N} w_{N-n}\vec{x}[n]\vec{x}^T[n]$$

$$\hat{\vec{p}} = \sum_{n=0}^{N} w_{N-n} d[n]\vec{x}[n]$$

Taking the gradient of the error

$$\nabla_{\vec{h}}\varepsilon[N] = 2\sum_{n=0}^{N} w_{N-n}\left(d[n] - \vec{h}^T\vec{x}[n]\right)\vec{x}^T[n] = 0$$

$$\vec{h}^T \underbrace{\left(\sum_{n=0}^{N}\vec{x}[n]\vec{x}[n]^T\omega_{N-n}\right)}_{\hat{R}_{xx}} = \underbrace{\sum_{n=0}^{N} w_{N-n} d[n]\vec{x}^T[n]}_{\hat{\vec{p}}^T}$$

$$\hat{R}_{xx}\vec{h}_{opt} = \hat{\vec{p}}$$

Weighting function w

For exponential window, $W_{N-n} = \alpha^{N-n}$

$$\hat{R}_{xx}[N] = \alpha \hat{R}_{xx}[N-1] + \vec{x}[n]\vec{x}[n]^T$$

$$N_{eff} \cong \frac{1}{1-\alpha}$$

$$\hat{\vec{p}}[N] = \alpha \hat{\vec{p}}[N-1] + d[n]\vec{x}[n]$$

Can recursively compute $\hat{R}_{xx}[n], \hat{\vec{p}}[n]$
to obtain $\vec{h}_{opt}^{(n)}$

Recursive Least Squares

$$\vec{h}_{opt}^{(n)} = \hat{R}_{xx}[n]^{-1}\hat{\vec{p}}[n]$$

$$\hat{R}_{xx}[n]\vec{h}_{opt}^{(n)} = \hat{\vec{p}}[n]$$

$$= \alpha \hat{\vec{p}}[n-1] + d[n]\vec{x}[n]$$

$$= \underbrace{\alpha \hat{R}_{xx}[n-1]}\vec{h}_{opt}^{(n-1)} + d[n]\vec{x}[n]$$

$$\hat{R}_{xx}\vec{h}_{opt}^{(n)} = \left(\hat{R}_{xx}[n] - \vec{x}[n]\vec{x}[n]^T\right)\vec{h}_{opt}^{(n-1)} + d[n]\vec{x}[n]$$

Recursive Least Squares, cont'd

$$\vec{h}_{opt}^{(n)} = \left(I - \hat{R}_{xx}[n]^{-1}\vec{x}[n]\vec{x}[n]^T\right)h_{opt}^{(n-1)} + \hat{R}_{xx}^{-1}[n]d[n]\vec{x}[n]$$

$$= \vec{h}_{opt}^{(n-1)} + \hat{R}_{xx}^{-1}[n]\left(-\vec{x}[n]\vec{x}_{[n]}^T h_{opt}^{(n-1)} + d[n]\vec{x}[n]\right)$$

$$= \vec{h}_{opt}^{(n-1)} + \hat{R}_{xx}^{-1}[n]\left(d[n] - \hat{d}[n]\right)\vec{x}[n]$$

Recursive Least Squares, cont'd

$$= \vec{h}_{opt}^{(n-1)} + \hat{R}_{xx}^{-1}[n]\left(d[n] - \hat{d}[n]\right)\vec{x}[n]$$

$$\Rightarrow \boxed{\vec{h}_{opt}^{(n)} = \vec{h}_{opt}^{(n-1)} + \hat{R}_{xx}^{-1}[n]e[n]\vec{x}[n]}$$

RLS algorithm

$\theta(N^2)$ if done right: N^2 RLS; $\theta(N^3)$ also popular for numerical reasons

Matrix Inversion Lemma

$$\left(A + BCB^T\right)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + B^T A^{-1} B\right)^{-1} B^T A^{-1}$$

$\Rightarrow$ Update $\hat{R}_{xx}^{-1}[n]$ directly:

$$\hat{R}_{xx}^{-1}[n] = \left(\hat{R}_{xx}[n-1]\alpha + \vec{x}[n]\vec{x}[n]^T\right)^{-1}$$

$$= \left(A + BB^T\right)^{-1}$$

Matrix Inversion Lumma, cont'd

$$\hat{R}_{xx}^{-1}[n] = \frac{\hat{R}_{xx}^{-1}[n-1]}{\alpha} -$$

$$\frac{\hat{R}^{-1}[n-1]}{\alpha}\dot{x}[n]\left(I + \dot{x}[n]^T \frac{R_{xx}^{-1}[n-1]}{\alpha}\dot{x}[n]\right)^{-1} x[n]^T \frac{\hat{R}_{xx}^{-1}[n-1]}{\alpha}$$

Riccati Equation

Example

$y(t) = a_1 + a_2 t + a_3 t^2/2$

find a_1, a_2, a_3 in LS sense for

$y(t_n)$, n = 1, 2, …

$$\vec{h} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \hat{y}(t_n) = \vec{h}^{(n)T} \vec{x}[t_n] \qquad \vec{x}[t_n] = \begin{bmatrix} 1 \\ t_n \\ t_n^2/2 \end{bmatrix}$$

Example, cont'd

$$\vec{h}^{(n)} = \vec{h}^{(n-1)} + \hat{R}_n^{-1}(y(t_n) - \hat{y}(t_n)) \begin{bmatrix} 1 \\ t_n \\ t_n^2/2 \end{bmatrix}$$

$$\hat{R}_n = \hat{R}_{n-1} + \begin{bmatrix} 1 \\ t_n \\ t_n^2/2 \end{bmatrix} [1 \quad t_n \quad t_n^2/2] \qquad (\alpha = 1)$$

Prior knowledge, bootstrapping

How can we include prior guess/initialization?

$$\vec{h}^{(0)} = \vec{h}_{init}$$

$$\hat{R}_0 = I \qquad \Rightarrow \min\left(h^{(n)} - h^{(0)}\right)^T \left(h^{(n)} - h^{(0)}\right) w_n$$

$$+ \sum_{n=1}^{N} w_{N-n} \left(d[n] - \hat{d}[n]\right)^2$$

Sometimes $\hat{R}_0 = \delta I \leftarrow$ "diagonal loading"

"prior knowledge" $\Rightarrow$

better numerical properties

Exact RLS has a vector gain

Kalman gain $\hat{R}[n]^{-1}\vec{x}[n] = \vec{K}_n$

$$\vec{h}^{(n)} = \vec{h}^{(n-1)} + \vec{K}_n \left(d[n] - \hat{d}[n]\right)$$

Variations on RLS Themes

scalar gain $g_n = r_n^{-1}$

$$\vec{h}^{(n)} = \vec{h}^{(n-1)} + g_n \vec{x}[n]e[n]$$

Replace $\hat{R}_{xx}[n]$ with $r_n I$, $r_n = \frac{1}{M}\mathrm{tr}\{R_{xx}[n]\}$

"average trace"

$$\mathrm{tr}\left(\hat{R}_n\right) = \mathrm{tr}\left(\alpha \hat{R}_{n-1}\right) + \mathrm{tr}\left(\vec{x}[n]\vec{x}[n]^T\right)$$

$$\boxed{r_n = \alpha\, r_{n-1} + \tfrac{1}{M}\left\|\vec{x}[n]\right\|^2}$$ scalar update

Scalar Kalman Gain RLS

$$\begin{cases} \vec{h}^{(n)} = \vec{h}^{(n-1)} + \mu_n \vec{x}[n]e[n] \\ \mu_n = \dfrac{1}{\hat{\sigma}_x^2[n]} \quad \hat{\sigma}_x^2[n] = \alpha\hat{\sigma}_x^2[n-1] + \frac{1}{M}\sum_{k=1}^{M} x[n-k]^2 \end{cases}$$

→ Adaptive Step Size LMS Algorithm ←

Variations on RLS Themes, cont'd

Also $g_n = r_n^{-1}, \quad r_n = \text{const} = \text{LMS}$ (fixed step size)

$$\vec{h}^{(n+1)} = \vec{h}^{(n)} + \mu \,\text{sign}\big(e[n]\big)\vec{x}[n]$$

$$\vec{h}^{(n)} + \mu e[n]\,\text{sign}\big(\vec{x}[n]\big)$$

$$\vec{h}^{(n)} + \mu \,\text{sign}\big(e[n]\big)\,\text{sign}\big(\vec{x}[n]\big)$$

Faster! larger misadjustment error

RLS on Continuous Measurements

Continuous measurement signals

$$d(t) = h(t)^T \vec{x}(t) \qquad \min_h \int_0^T e^2(t)dt$$

$$\vec{h}_{opt} = \left(\int_0^T \vec{x}(t)\vec{x}(t)^T dt\right)^{-1}\left(\int_0^T \vec{x}(t)d(t)dt\right)$$

e.g. $$y(t) = a_1 + a_2 \sin\frac{2\pi t}{T} + a_3 \cos\frac{8\pi t}{T}$$

$$\min_{\vec{h}} \int_0^T \big(y(t) - \hat{y}(t)\big)^2 dt$$

Example

$$\hat{h} = \begin{bmatrix} \int_0^T dt & \int_0^T \sin\frac{2\pi t}{T} dt & \int_0^T \cos\frac{8\pi t}{T} dt \\ \int_0^T \sin\frac{2\pi t}{T} & \int_0^T \sin^2\frac{2\pi t}{T} & \int_0^T \sin\frac{2\pi t}{T}\cos\frac{8\pi t}{T} dt \\ \int_0^T \cos\frac{2\pi t}{T} dt & \int_0^T \sin\frac{2\pi t}{T}\cos\frac{8\pi t}{T} & \int_0^T \cos^2\frac{8\pi t}{T} dt \end{bmatrix}^{-1}$$

$$\begin{bmatrix} \int_0^T y(t)dt \\ \int_0^T \sin\frac{2\pi t}{T} y(t)dt \\ \int_0^T \cos\frac{8\pi t}{T} y(t)dt \end{bmatrix}$$

Continuous-time RLS

$$\vec{x}(t) = \begin{bmatrix} 1 \\ \sin\frac{2\pi t}{T} \\ \cos\frac{8\pi t}{T} \end{bmatrix}$$

$$\text{CT RLS} \Rightarrow \frac{d}{dt}\vec{h}^{(t)} = \hat{R}^{-1}(t)\vec{x}(t)\left(y(t) - \vec{h}^{(t)T}\vec{x}(t)\right)$$

$$\frac{d}{dt}\hat{R}(t) = \vec{x}(t)\vec{x}(t)^T$$

Can solve NL LS problems, too

Nonlinear LS adaptive filtering to solve NL equations!

$y = F(x) + e$ nonlinear function

$\min(y - F(x))^2$

Linearize the problem about a guess

Guess: $\hat{x}_0$

$$y = F(\hat{x}_0) + \left.\frac{dF}{dx}\right|_{x=\hat{x}_0}(x - \hat{x}_0) + \ldots + e$$

$$\left(y - F(\hat{x}_0)\right) \cong \left.\frac{dF}{dx}\right|_{x=\hat{x}_0}(x - \hat{x}_0) + e$$

pick $\hat{x}_1$ to min e^2

Gauss-Newton Iterations, NLLS

linear equation in $(x - \hat{x}_0)$

$$A \cong B(x - \hat{x}_0) + e$$

$$(\hat{x}_1 - \hat{x}_0) = \left(\left(\frac{dF}{dx}\Big|_{x=\hat{x}_0} \right)^T \left(\frac{dF}{dx}\Big|_{x=\hat{x}_0} \right) \right)^{-1} \left(\frac{dF}{dx}\Big|_{x=\hat{x}_0} \right)^T (y - F(\hat{x}_0))$$

Gauss-Newton iterations, or NLLS

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the

College of Engineering

University of Illinois at Urbana-Champaign

ECE 451

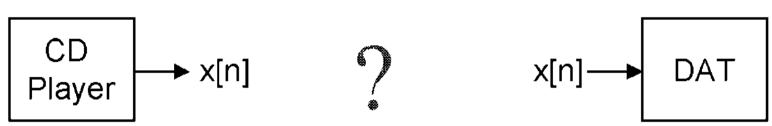
ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 36

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Multirate System Fundamentals

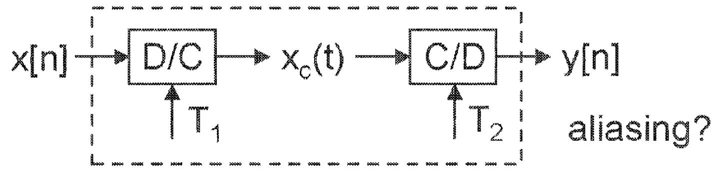
Suppose you want to use a DAT to record a CD

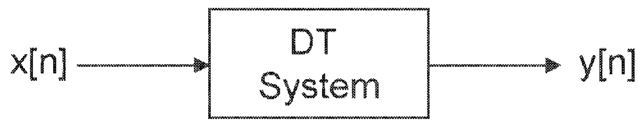


$$\frac{1}{T_1} = 44.1 \text{ kHz} \qquad \frac{1}{T_2} = 48\text{kHz}$$

Three solutions

1) 44.1 ≈ 48 ☹

2) 

3) 

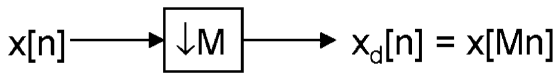
Slower Rate

How about 44.1 kHz → 22.05 kHz ?

$x[n] = x_c(nT)$

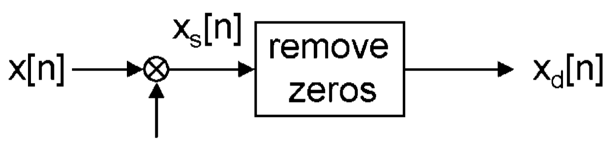
$y[n] = x_c(n(2T)) = x_c(2nT) = x[2n]$

Sample rate compression, model



"compressor"

Model:



$$s[n] = \sum_k \delta[n - Mk]$$

Modulated signal

$$X_s(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]s[n]e^{-j\omega n}$$

Rewrite s[n] as $\frac{1}{M}\sum_{k=0}^{M-1} e^{j\frac{2\pi kn}{M}} = \begin{cases} 1, & n = \ell M \\ 0, & \text{else} \end{cases}$

Spectrum of modulated signal

$$X_s(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\left(\frac{1}{M}\sum_{k=0}^{M-1} e^{j\frac{2\pi kn}{M}}\right)e^{-j\omega n}$$

$$= \frac{1}{M}\sum_{k=0}^{M-1}\left(\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n + j\frac{2\pi kn}{M}}\right)$$

$$\boxed{X_s(e^{j\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{j\left(\omega - \frac{2\pi k}{M}\right)}\right)}$$

aliasing @ $\frac{2\pi k}{M}$

Now remove the zeros

$$X_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_s[nM]e^{-j\omega n}$$

$$= \sum_{k=-\infty}^{\infty} x_s[k]e^{-j\frac{\omega k}{M}} \qquad k = nM$$

$$= X_s\left(e^{j\frac{\omega}{M}}\right)$$

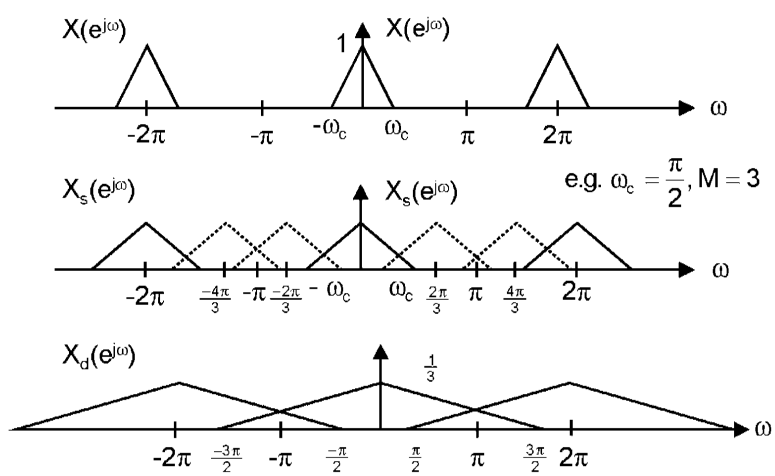
Sample rate compression

$$\therefore \quad X_d(e^{j\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi k}{M}\right)}\right)$$

Aliasing @ $\frac{2\pi k}{M}$, scaled by $\frac{1}{M}$,

stretched by $\frac{1}{M}$

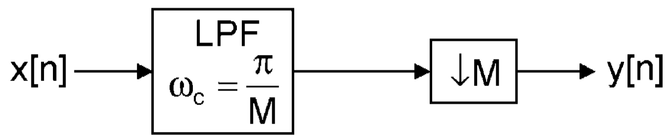
Example, cont'd



Reducing the sampling rate

Aliasing if $X(e^{j\omega}) \neq 0, \quad |\omega| > \pi/M$

$\rightarrow$ can use an anti-aliasing filter to avoid aliasing



"down-sampling / decimation"

Now, increase the sampling rate

Suppose you sampled at rate $1 / T_1$

$\rightarrow x[n] = x_c(nT_1)$ want $y[n] = x_c(nT_2)$,

$$T_2 = \frac{T_1}{L}, \quad y[n] = \begin{cases} x\left[\frac{n}{L}\right], & n = kL \\ ???, & \text{else} \end{cases}$$

$\rightarrow$ 22.05 kHz $\rightarrow$ 44.1 kHz

$\rightarrow$ Interpolate! Linear? BL? Spline? ZOH?

Sample rate expansion

$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow X_e[n]$$

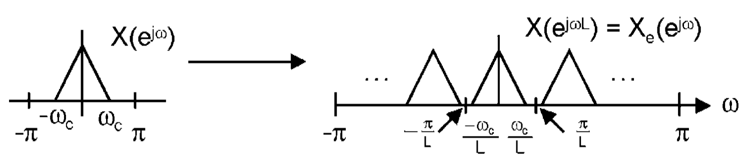
$$x_e[n] = \begin{cases} x[nk], & n = k\ell \\ 0, & \text{else} \end{cases}$$

Puts L-1 zeros between samples of x[n]

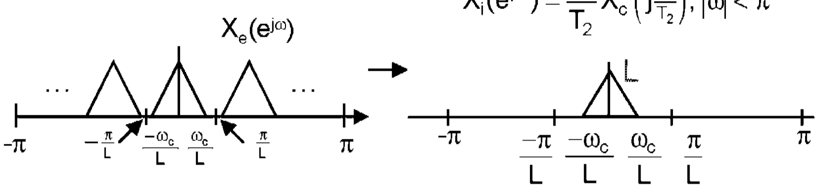
$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_e[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x[n/L]e^{-j\omega n}$$

$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega kL} \Rightarrow \boxed{X_e(e^{j\omega}) = X(e^{j\omega L})}$$

Example

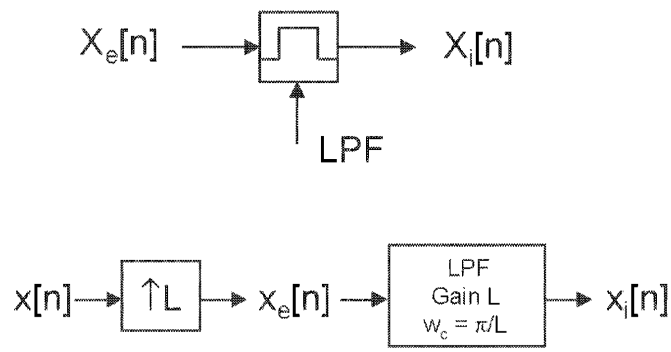


$$X_i(e^{j\omega}) = \frac{1}{T_2} X_c\left(j\tfrac{\omega}{T_2}\right), |\omega| < \pi$$



What you would have had with sample rate $T_2 = T_1/L$

From expansion to interpolation



Band-limited interpolation

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin \pi(n-kL)/L}{\pi(n-kL)/L}$$

$$x_i[n] = \sum_{k=-\infty}^{\infty} x_e[k] \frac{\sin \pi(n-k)/L}{\pi(n-k)/L}$$

$$= \sum_{\substack{k=-\infty \\ k=mL}}^{\infty} x\left[\tfrac{k}{L}\right] \sin \frac{\pi(n-k)/L}{\pi(n-k)/L}$$

$$= \sum_{m=-\infty}^{\infty} x[m] \sin \frac{\pi(n-mL)/L}{\pi(n-mL)/L}$$

⇐ Band-limited interpolation

Interpolation filters

Interpolation filter definition =

$$h_i[n] = \begin{cases} 1, & n = 0 \\ 0, & n = \pm L, \pm 2L, \pm 3L, \ldots \end{cases} = \frac{\sin(\pi n/L)}{(\pi n/L)} = h_{lp}[n]$$

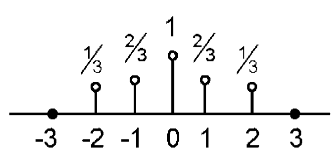
Bandlimited interpolation

"Zero ISI" , preserves original samples at kL

Linear interpolation

Linear interpolator

$$h_{\ell i}[n] = \begin{cases} \left|\frac{L-n}{L}\right|, & |n| \le L \\ 0, & \text{else} \end{cases}$$



L=3

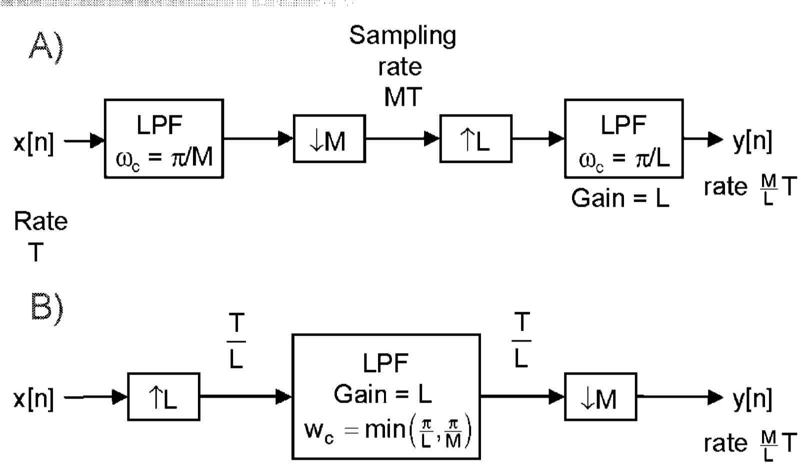
→ Linearly interpolates between the samples

Band-limited interpolation filter

Band-limited Interpolator definition

$$h_i[n] = \frac{\sin(\pi \, n/L)}{(\pi \, n/L)} = \begin{cases} 1, & n = 0 \\ 0, & n = \pm L, \pm 2L, \ldots \end{cases}$$

Change rate T to rate MT/L



Up then down, down then up?

A) Is not good! Why?

B) Is better!

CD $\rightarrow$ DAT?

44.1 kHz 48kHz

$$\frac{1}{T_1} = 44.1 \times 10^3 \quad \frac{1}{T_2} = 48 \times 10^3$$

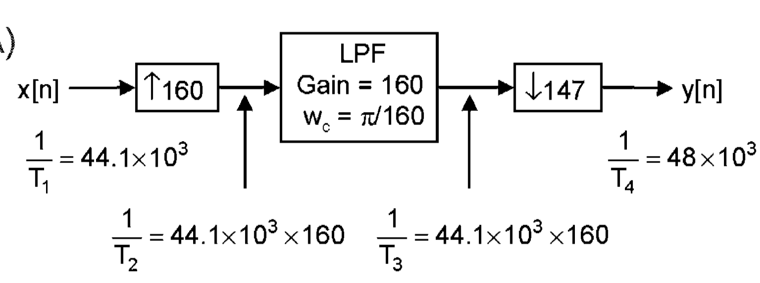
$$T_2 = \frac{M}{L} T_1 = \frac{441}{480} T_1$$

Up then down

Need to increase rate by $\frac{L}{M} = \frac{480}{441}$

$$\frac{L}{M} = \frac{480}{441} = \frac{160}{147}$$

A)

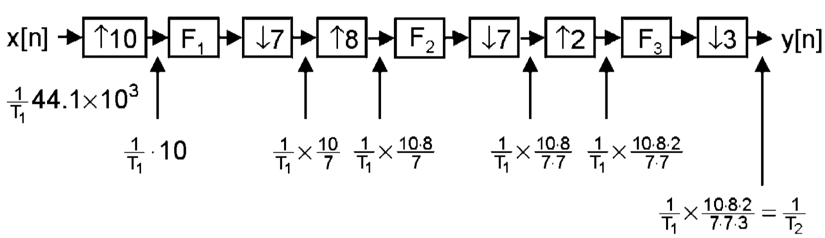


Internal rate gets too high

Too much storage / computation

$$\frac{160}{147} = \frac{10 \cdot 8 \cdot 2}{7 \cdot 21} = \frac{10}{7} \cdot \frac{8}{7} \cdot \frac{2}{3}$$

Better approach to s.r. conversion



F_1 = ?, F_2 = ?, F_3 = ?

Max rate = $\frac{1}{T_1} \times 10 = 441 \times 10^3$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 37

Professor Andrew Singer
Department of Electrical and
Computer Engineering

Multirate System Fundamentals

Last time, we had for $z = e^{j\omega}$:

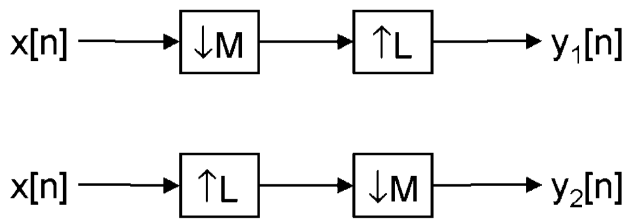
Sample rate expansion:

$$X_e(z) = X\left(z^L\right),$$

Sample rate compression:

$$X_d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} W_M^k\right), \qquad W_M = e^{-j\frac{2\pi}{M}}$$

Does order matter?



$$y_2[n] \overset{?}{=} y_1[n] \quad \forall x[n]?$$

Look in the frequency domain

$$Y_1(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{L}{M}} W_M^k\right)$$

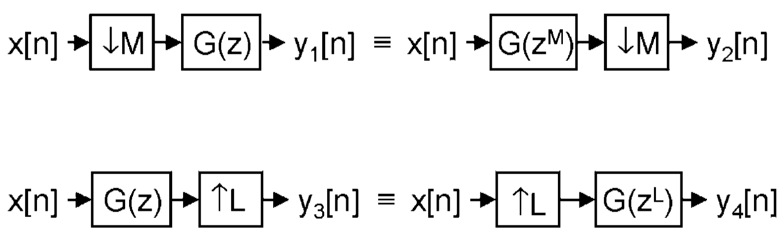
$$Y_2(z) = \frac{1}{M}\sum_{k=0}^{M-1} X_e\left(z^{\frac{1}{M}} W_M^k\right) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{L}{M}} W_M^{kL}\right)$$

$$Y_1(z) = Y_2(z) \text{ iff } \left\{e^{-j\frac{2\pi k}{M}}\right\}_{k=0}^{M-1} = \left\{e^{-j\frac{2\pi kL}{M}}\right\}_{k=0}^{M-1}$$

only true when M, L are coprime, i.e. GCD(M, L) = 1

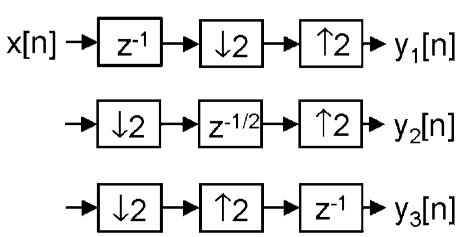
Noble Identities

(For G(z) rational)



Less Noble

If G(z) not rational, $\Rightarrow$ not true in general.



$$x[n] = \delta[n] \rightarrow y_1[n] = 0$$

$$y_3[n] = \delta[n-1] \rightarrow\leftarrow$$

Polyphase Representation

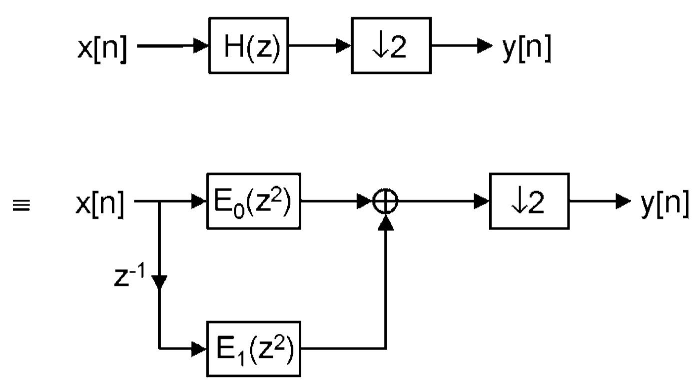
$$H(z) = \sum_n h(n) z^{-n} = \sum_n h[2n] z^{-2n} + \sum_n h[2n+1] z^{-(2n+1)}$$

$$= E_0(z^2) + z^{-1} E_1(z^2)$$

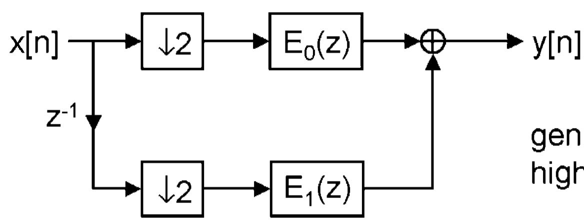
$$E_0(z) = \sum_n h[2n] z^{-n}, \quad E_1(z) = \sum_n h[2n+1] z^{-n}$$

$$\Rightarrow H(z) = E_0(z^2) + z^{-1} E_1(z^2)$$

Polyphase Decimator Structure



Polyphase Decimator



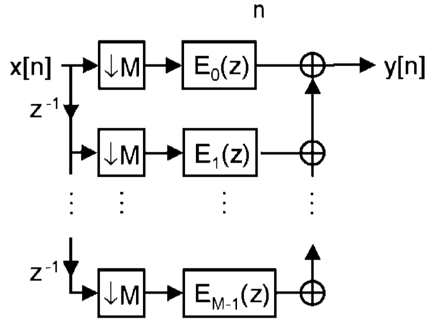
generalizes to higher orders

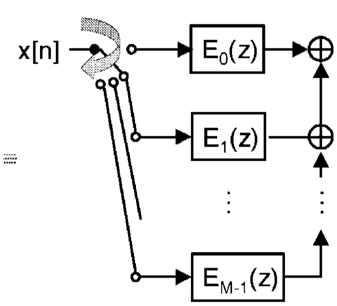
Operates at ½ the peak rate of standard decimation filter

Higher-order Polyphase Decimator

$$H(z) = E_0(z^M) + z^{-1}E_1(z^M) + \ldots + z^{-(M-1)}E_{M-1}(z^M),$$

$$E_i(z) = \sum_n h[M_{n+i}]z^{-n}$$

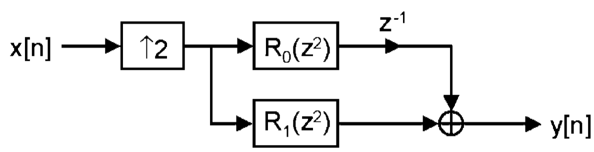


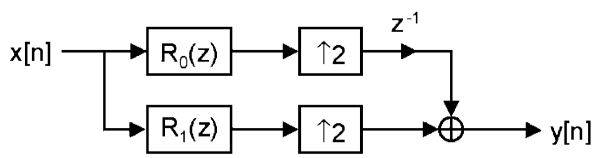


Interpolators

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

$$= z^{-1}R_0(z^2) + R_1(z^2), \quad R_0 = E_1 \quad R_1 = E_0$$

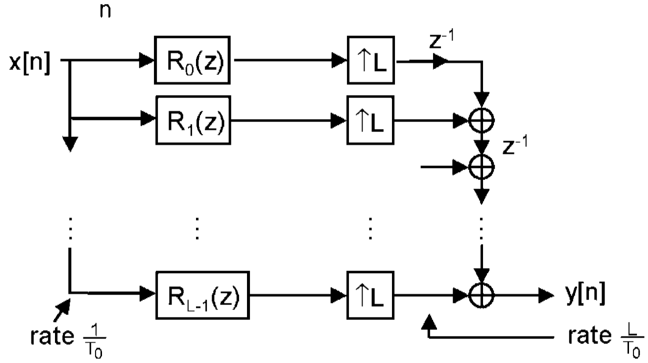




Polyphase Interpolator Structure

$$H(z) = z^{-(L-1)}R_0(z^L) + \ldots + z^{-1}R_{L-2}(z^L) + R_{L-1}(z^L)$$

$$R_i(z) = \sum_n h[Ln + L - i] = E_{L-i}(z)$$



Compuational Savings

If h[n] is length N

$$\Rightarrow \left(\frac{N}{L}\right) \text{ taps per filter}$$

x L filters

$$\text{x} \quad \frac{1}{T_0}\frac{\text{samples}}{\text{second}} = \frac{N}{L} \cdot L \cdot \frac{1}{T_0} = \frac{N}{T_0}$$

In comparison to...

Dumb way:

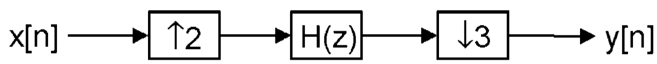
$$N \times \frac{L}{T_0} = \frac{NL}{T_0} \Rightarrow \text{savings of factor of L.}$$

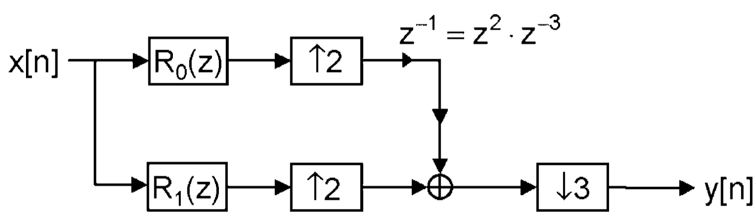
↑

filter at higher rate after zero-insertion

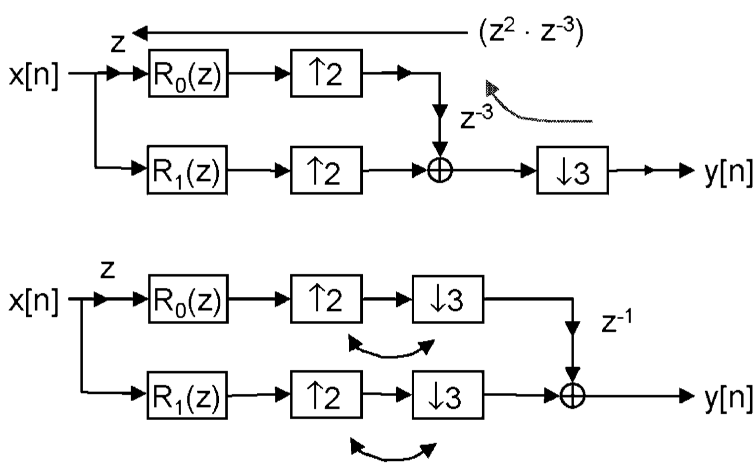
Fractional Rate Conversion

$$\frac{L}{M} = \frac{2}{3}$$

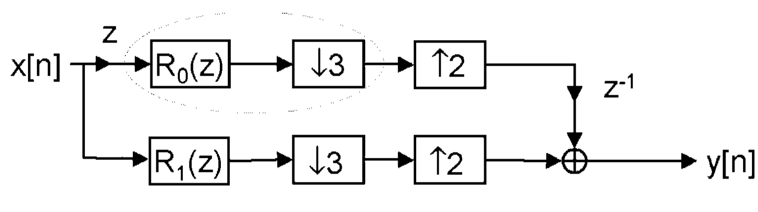




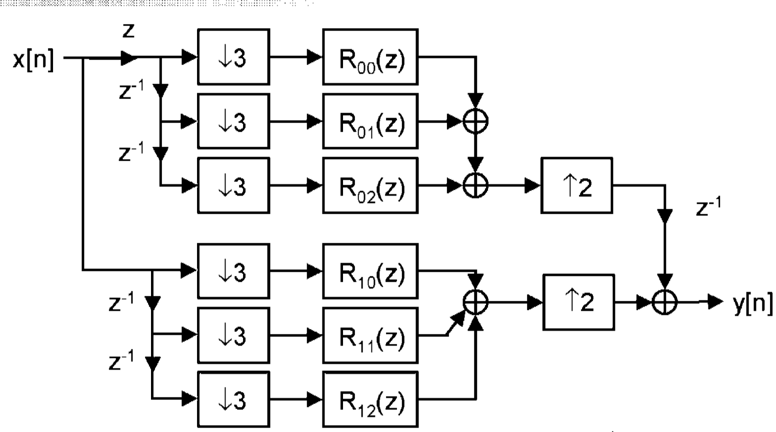
Fractional Rate Conversion, cont'd



Fractional Rate Conversion, cont'd



Fractional Rate Conversion, cont'd



$\frac{N+1}{M}$ x's/sample $\quad$ $\frac{N+1-L}{M}$ +'s/sample $\quad$ vs LN/sample

$\rightarrow$ savings O(ML)

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

ECE 451

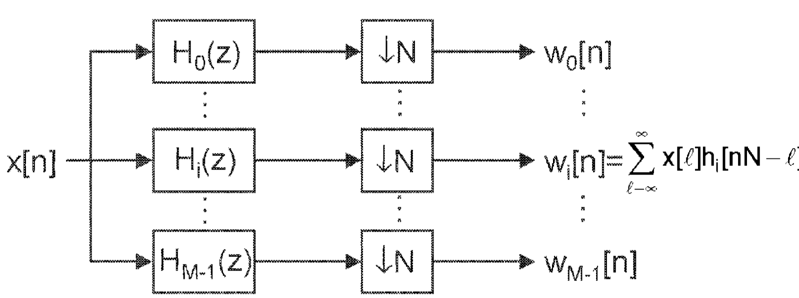
ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 38

Professor Andrew Singer
Department of Electrical and Computer Engineering

Multirate Filterbanks

M-band Filterbank: (e.g. STFT/DFT Filterbanks)



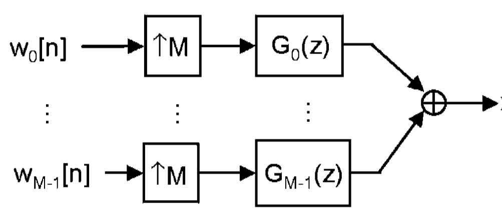
- For N>M, cannot recover x[n] from $w_i[n]$ (why?)
- "Critically sampled" filterbank, for N=M

Invertible Filterbank

For what $H_i(z)$ can x[n] be recovered from $w_i[n]$?

- Yes! Two simple examples you know:
  - Ideal band pass filters
  - Ideal delay filters, i.e. $h_i[n]=\delta[n-i]$
- Are there others?
- How do we find them?

Reconstruction of x[n]



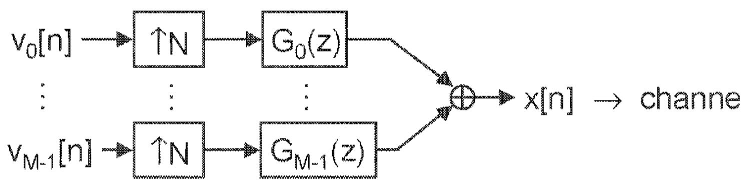
$$\hat{x}[n] = \sum_{k=0}^{M-1} \sum_{\ell=-\infty}^{\infty} w_k[\ell] g_k[n-\ell M]$$

Can view as basis expansion for x[n] in $w_k[n]$

If reconstruction is not possible, how can we choose $G_i(z)$ so $\hat{x}[n] \approx x[n]$?

$\rightarrow$ coding, compression of audio and video

Multiplexing



- Cannot separate $v_i[n]$ if N < M (why?)
- For N = M, "spectrally efficient" multiplexing

Signal Separation

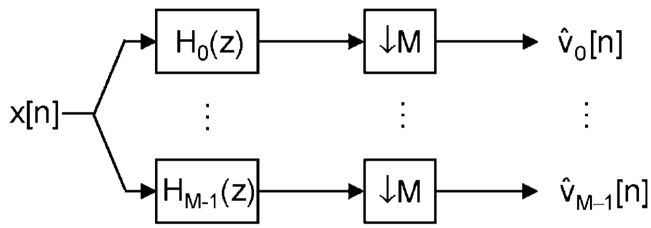
What are the conditions on $G_i(z)$ such that $v_i[n]$ can be recovered from x[n]?

→ Ideal bandpass filters will work

→ Ideal delays, $G_i(z) = z^{-i}$, will work

→ How can we find/specify others?

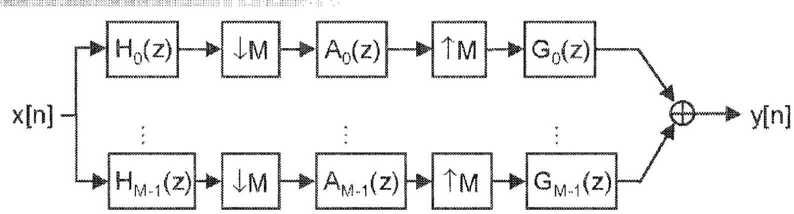
De-Multiplexing



This idea is used in digital communications:

- xDSL modems use this idea (DMT)
- CDMA/TDMA/FDMA cellular telephony
- Wireless data networks (OFDM)

An M-Channel Filterbank Example



For which $H_i(z)$, $A_i(z)$, $G_i(z)$ does

$$y[n] = x[n] * f[n] \text{ for some } f[n]?$$

Example, cont'd

→ Ideal bandpass filters

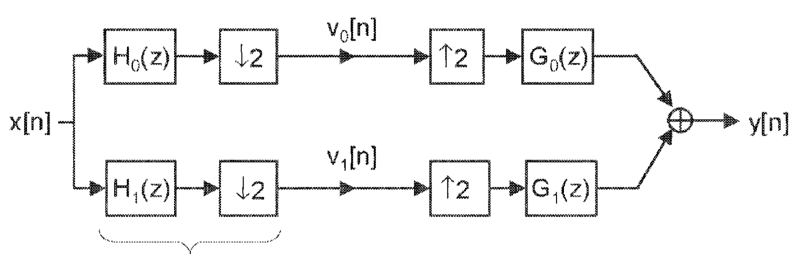
→ Ideal delays, z^{-i}

→ How do we find/specify others?

- Given a desired f[n], how are the $A_i(z)$'s specified?

    → Multirate adaptive filters

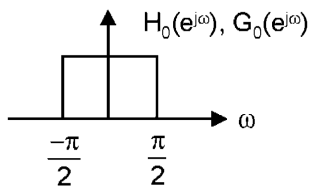
2 Channel Critically-Sampled Filterbanks

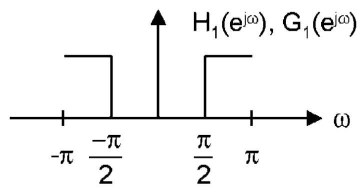


"analysis bank" "synthesis bank"

2 Channel Filterbank, cont'd

For example:





Filterbank Properties

- In general, linear and time-varying
- Actually, a linear periodically time-varying (LPTV) system

$x(n) \rightarrow y(n) \Rightarrow x(n - 2n_0) \rightarrow y(n - 2n_0)$

Filterbank Properties, cont'd

- What are conditions on $H_i(z)$, $G_i(z)$ such that the overall system is:

  I. Time-Invariant, also called "aliasing-cancellation" $y[n] = x[n] * f[n]$ for some $f[n]$

  II. Distortion-Free, "perfect reconstruction", "PR", "Biorthogonal"

  $y[n] = x[n]$ "strict sense"

  $y[n] = \alpha x[n - \ell]$, some α, ℓ

Filterbank Properties, cont'd

III. Energy-Preserving "orthogonal transformation", "generalized allpass" "lossless"

$$\sum x^2[n] = \sum y^2[n]$$

For LTI, true iff $f[n]$ is an allpass filter (why?)

Filterbank Properties, cont'd

IV. Internally orthogonal / energy preserving

- Orthogonal analysis

$$\sum_m \sum_n v_m^2[n] = \sum_n x^2[n]$$

- Orthogonal synthesis

$$\sum_m \sum_n v_m^2[n] = \sum_n y^2[n]$$

Filterbank Properties, cont'd

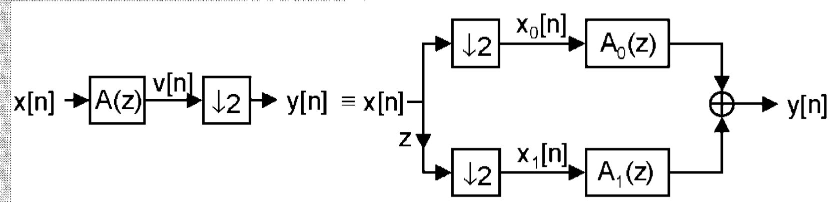
Desirable Properties / System constraints:

I. FIR $H_i(z)$, $G_i(z)$ "FIR" system

II. Generalized linear phase
h[n], g[n] have symmetry

III. Fast algorithms / efficient implementations

Polyphase Representation: Analysis



$A(z) = A_0(z^2) + z\,A_1(z^2)$

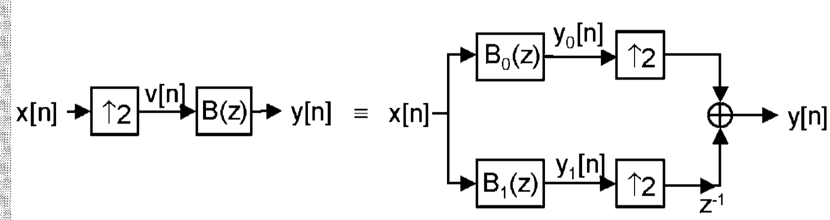
$X(z) = X_0(z^2) + z^{-1}\,X_1(z^2)$

Note sign convention

$$V(z) = X(z)A(z)$$
$$= [X_0(z^2)A_0(z^2) + X_1(z^2)A_1(z^2)] + z^{-1}[A_0(z^2)X_1(z^2) + z^2\,X_0(z^2)A_1(z^2)]$$

$\therefore Y(z) = X_0(z)A_0(z) + X_1(z)A_1(z)$ (why?)

Polyphase Representation: Synthesis



$B(z) = B_0(z^2) + z^{-1}\,B_1(z^2)$

$\therefore Y(z) = B(z)X(z^2)$

$$= B_0(z^2)X(z^2) + z^{-1}\,B_1(z^2)X(z^2)$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

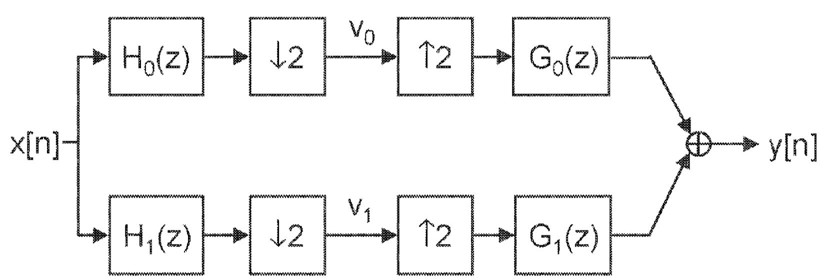
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

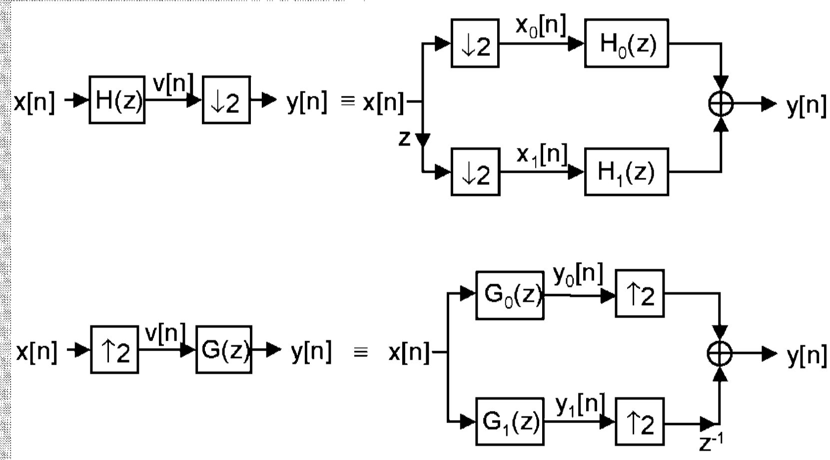
Lecture 39

Professor Andrew Singer
Department of Electrical and
Computer Engineering

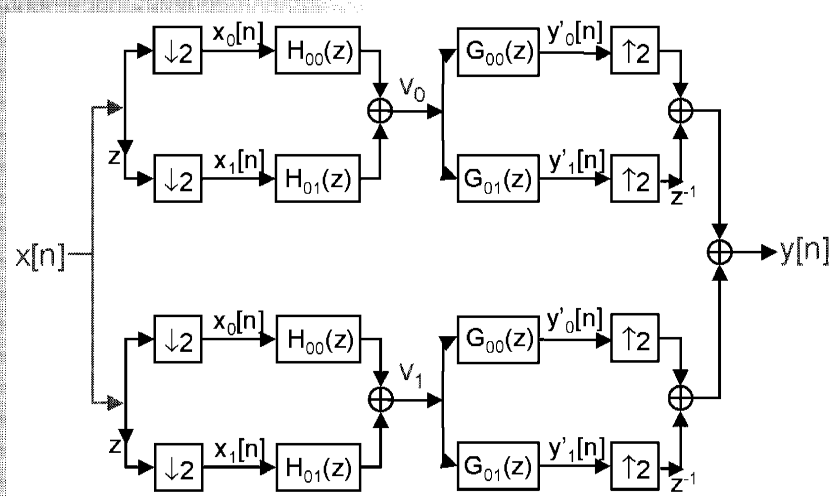
Two Channel Multirate Filterbank



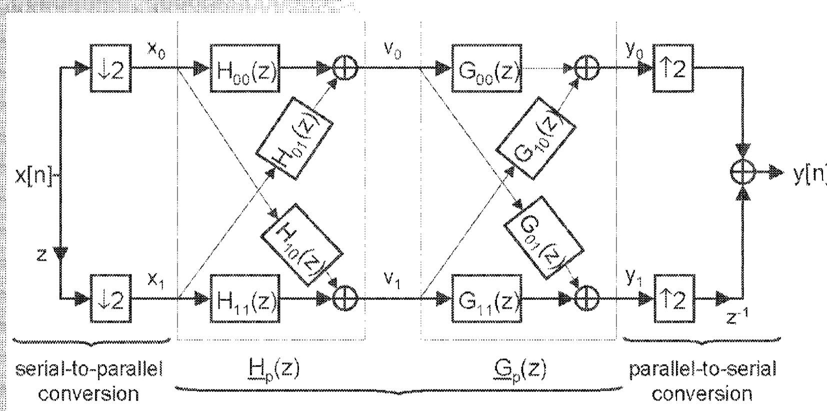
Polyphase Representation



Polyphase Characterization of MRFBs



Polyphase Characterization of MRFBs



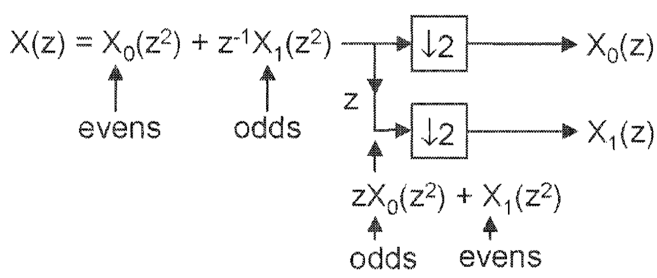
Multi-input Multi-output (MIMO) LTI system

$$\underline{T}_p(z) = \underline{G}_p(z)\ \underline{H}_p(z)$$

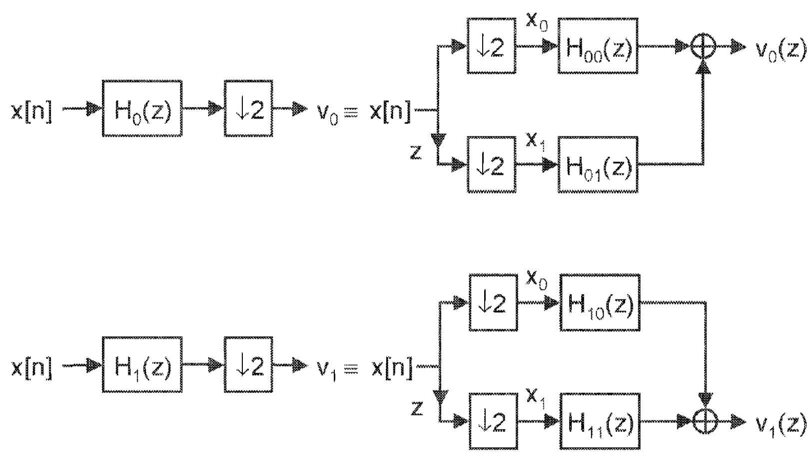
Polyphase Analysis

$H_i(z) = H_{i0}(z^2) + z\,H_{i1}(z^2) \qquad i = 0, 1$

$G_i(z) = G_{i0}(z^2) + z^{-1}G_{i1}(z^2)$



Polyphase Analysis



MIMO Analysis Transfer Matrix

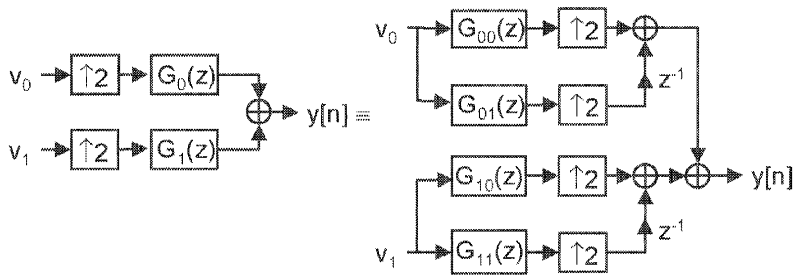
$$\begin{pmatrix} V_0(z) \\ V(z) \end{pmatrix} = \begin{bmatrix} X_0(z)H_{00}(z) + X_1(z)H_{01}(z) \\ X_0(z)H_{10}(z) + X_1(z)H_{11}(z) \end{bmatrix} =$$

$$\underbrace{\begin{bmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{bmatrix}}_{\triangleq \underline{H}_p(z)} \begin{pmatrix} X_0(z) \\ X_1(z) \end{pmatrix}$$

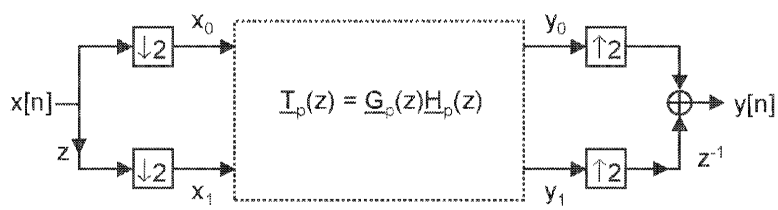
MIMO Synthesis Transfer Matrix

$$\begin{pmatrix}Y_0(z)\\ Y_1(z)\end{pmatrix} = \begin{bmatrix} G_{00}(z)V_0(z)+G_{10}(z)V_1(z)\\ G_{01}(z)V_0(z)+G_{11}(z)V_1(z)\end{bmatrix} = \underbrace{\begin{bmatrix}G_{00}(z) & G_{10}(z)\\ G_{01}(z) & G_{11}(z)\end{bmatrix}}_{\triangleq \underline{G}_p(z)} \begin{pmatrix}V_0(z)\\ V_1(z)\end{pmatrix}$$

Polyphase Synthesis



MIMO Transfer Function



$$Y(z) = Y_0(z^2) + z^{-1}Y_1(z^2)$$

MIMO Transfer Function

$$\begin{pmatrix} Y_0(z) \\ Y_1(z) \end{pmatrix} = \begin{bmatrix} T_{00}(z) & T_{01}(z) \\ T_{10}(z) & T_{11}(z) \end{bmatrix} \begin{pmatrix} X_0(z) \\ X_1(z) \end{pmatrix} =$$

$$= \begin{pmatrix} T_{00}(z)X_0(z) + T_{01}(z)X_1(z) \\ T_{10}(z)X_0(z) + T_{11}(z)X_1(z) \end{pmatrix}$$

Polyphase Output

$$Y(z) = \underbrace{\left[T_{00}(z^2)X_0(z^2) + T_{01}(z^2)X_1(z^2)\right]}_{Y_0(z^2)} +$$

$$\underbrace{z^{-1}\left[T_{10}(z^2)X_0(z^2) + T_{11}(z^2)X_1(z^2)\right]}_{z^{-1}Y_1(z^2)}$$

Polyphase Output

$$= X_0(z^2)\left[T_{00}(z^2) + z^{-1}T_{10}(z^2)\right] +$$

$$z^{-1}X_1(z^2)\left[zT_{01}(z^2) + T_{11}(z^2)\right]$$

$$= \left[X_0(z^2) + z^{-1}X_1(z^2)\right]\left[F_0(z^2) + zF_1(z^2)\right]$$

If LTI

$$\boxed{F_0(z) = T_{00}(z) = T_{11}(z),}$$

$$zF_1(z^2) = zT_{01}(z^2) = z^{-1}T_{10}(z^2) \Rightarrow$$

$$\boxed{F_1(z) = T_{01}(z) = z^{-1}T_{10}(z)}$$

Aliasing Cancellation Condition

$\therefore$ If LTI $\Rightarrow$

$$\underline{T}_p(z) = \begin{bmatrix} F_0(z) & F_1(z) \\ zF_1(z) & F_0(z) \end{bmatrix}$$

$$\underline{Y}(z) = \underline{T}_p(z)\underline{X}(z),$$

$$Y(z) = \underbrace{[F_0(z^2) + zF_1(z^2)]}_{F(z)}\underbrace{[X_0(z^2) + z^{-1}X_1(z^2)]}_{X(z)}$$

$F_0(z), F_1(z) =$ Polyphase comp. of F(z).

LTI / Aliasing Cancellation

- If the system is time-invariant (aliasing cancelled) then $\underline{T}_p(z)$ is pseudo-circulant, i.e.

$$\underline{T}_p(z) = \begin{bmatrix} F_0(z) & F_1(z) \\ zF_1(z) & F_0(z) \end{bmatrix}$$

- If $\underline{T}_p(z)$ is pseudo-circulant, then the system is TI.

T.I. $\Leftrightarrow$ $\underline{T}_p(z)$ is pseudo-circulant.

Perfect Reconstruction Examples

- For an M×M system, $\underline{T}_p(z)$ is circulant with a z factor in each term below main diagonal.

Example:

$$\text{P.R.} \quad F(z) = z^{-\ell} \qquad (\text{choose } \ell \text{ even})$$

$$F(z) = \left(z^{-\ell/2}\right)^2 + 0$$

$$F_0(z) = z^{-\ell/2}, \; F_1(z) = 0$$

Examples, cont'd

$$\underline{T}_p = \begin{bmatrix} z^{-\ell/2} & 0 \\ 0 & z^{-\ell/2} \end{bmatrix}$$

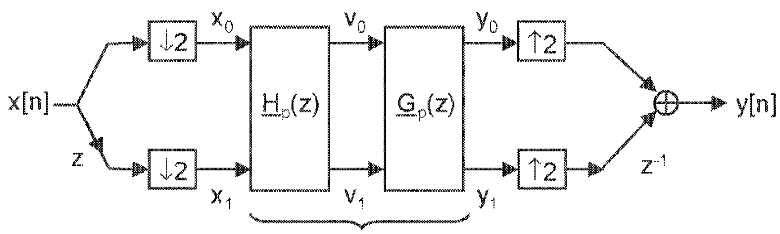
Others:

$$\underline{T}_p = I, \; \underline{T}_p = \begin{bmatrix} 0 & 1 \\ z & 0 \end{bmatrix}, \text{ or delayed versions}$$

 © 2001 Andrew Singer, University of Illinois at Urbana-Champaign, All Rights Reserved

2-Channel Multirate Filterbanks

Polyphase Representation:



$$\underline{T}_p(z) = \underline{G}_p(z)\underline{H}_p(z)$$

MRFB Properties

1) T.I. iff

$$\underline{T}_p(z) = \begin{bmatrix} F_0(z) & F_1(z) \\ zF_1(z) & F_0(z) \end{bmatrix} \quad \text{pseudo-circulant}$$

system function $\longrightarrow$ $F(z) = F_0(z) + zF_1(z^2)$

MRFB Properties, cont'd

2) P.R. iff

$$F(z) = I \rightarrow \underline{T}_p(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$F(z) = \alpha z^{-\ell} \rightarrow \underline{T}_p(z) = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \alpha z^{-\ell/2}, & \ell \text{ even} \\ \begin{bmatrix} 0 & 1 \\ z & 0 \end{bmatrix} \alpha z^{-(\ell+1)/2}, & \ell \text{ odd} \end{cases}$$

MRFB Properties, cont'd

3) Analysis bank orthogonal

iff $\underline{H}_p^H \underline{H}_p = I$, $\underline{H}_p(z)$ paraunitary

(unitary for all ω)

4) Synthesis bank orthogonal

iff $\underline{G}_p(z)$ paraunitary

5) Overall system is energy preserving if $\underline{T}_p(z)$ paraunitary

MRFB Properties, cont'd

6) If $H_0(z)$, $H_1(z)$ are orthogonal, then for P.R.

$$\underline{G}_p \underline{H}_p = I \Rightarrow \underline{G}_p = \underline{H}_p^{-1} = \underline{H}_p^H$$

(time reversal for real)

7) If $H_0(z)$, $H_1(z)$ are FIR, P.R.? with FIR G_0, G_1?

$$\underline{G}_p(z) = \underline{H}_p(z)^{-1} = \underbrace{\frac{1}{|\underline{H}_p|}\begin{bmatrix} H_{11} & -H_{01} \\ -H_{10} & H_{00} \end{bmatrix}}_{\text{FIR}}$$

$$\Rightarrow |\underline{H}_p| = \beta z^{-k}$$

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

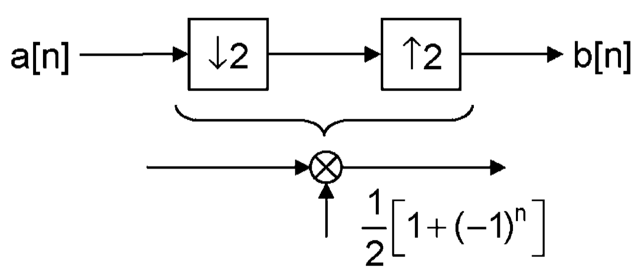
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 40

Professor Andrew Singer
Department of Electrical and Computer Engineering

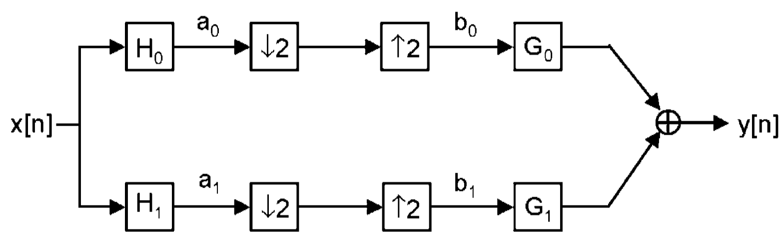
Modulation-Domain Characterization of MRFB's



$$B(e^{j\omega}) = \frac{1}{2}\left[A(e^{j\omega}) + A(e^{j(\omega-\pi)})\right]$$

$$B(z) = \frac{1}{2}[A(z) + A(-z)]$$

MRFB in Modulation-Domain



$$Y(z) = \begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \begin{bmatrix} B_0(z) \\ B_1(z) \end{bmatrix}$$

Y(z) from Modulation-Domain

$$Y(z) = \tfrac{1}{2}\begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \begin{bmatrix} A_0(z) + A_0(-z) \\ A_1(z) + A_1(-z) \end{bmatrix}$$

$$= \tfrac{1}{2}\begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \begin{bmatrix} H_0(z)X(z) + H_0(-z)X(-z) \\ H_1(z)X(z) + H_1(-z)X(-z) \end{bmatrix}$$

$$= \tfrac{1}{2}\begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix}$$

$$= \tfrac{1}{2}\underline{G}(z)\underline{H}_m(z)\underline{X}_m(z)$$

"Modulation matrix" (alias components)

Properties of MRFBs

1) T.I. iff $\underline{G}(z)\underline{H}_m(z) = [2F(z) \quad 0]$

$(G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0)$

$\rightarrow$ Given $H_0(z)$, $H_1(z)$, and desired $F(z)$, then

$$\underline{G}(z) = [2F(z) \;\; 0]\underline{H}_m^{-1}(z)$$

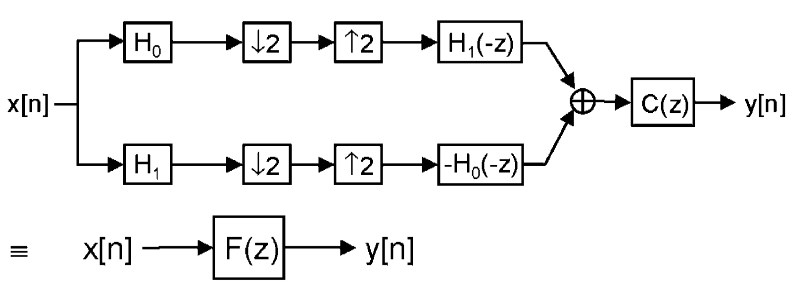
$$[G_0(z) \;\; G_1(z)] = [2F(z) \;\; 0]\begin{bmatrix} H_1(-z) & -H_0(-z) \\ -H_1(z) & H_0(z) \end{bmatrix}\frac{1}{|\underline{H}_m(z)|}$$

Modulation-Domain, LTI

$$\Rightarrow G_0(z) = C(z)H_1(-z)$$

$$G_1(z) = -C(z)H_0(-z)$$

$$, \quad C(z) = \frac{2F(z)}{\det \underline{H}_m(z)}$$



Modulation-Domain, LTI

For

$$\underline{G}_m(z) = \begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix}$$

$$\text{LTI} \leftrightarrow \underbrace{\underline{G}_m(z)\underline{H}_m(z)}_{\underline{T}_m(z)} = 2\begin{bmatrix} F(z) & 0 \\ 0 & F(-z) \end{bmatrix}$$

"aliasing cancellation" MTX

Perfect Reconstruction

2) P.R.

$$T_m(z) = 2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, 2\begin{bmatrix} z^{-\ell} & 0 \\ 0 & -z^{-\ell} \end{bmatrix}, 2\begin{bmatrix} z^{-\ell} & 0 \\ 0 & z^{-\ell} \end{bmatrix}$$

ℓ odd $\quad$ ℓ even

3) $H_0(z)$, $H_1(z)$ FIR, P.R. with FIR $G_0(z)$, $G_1(z)$?

$$\det H_m(z) = \beta z^{-k}$$

Modulation / Polyphase Domains

$\underline{H}_p \leftrightarrow \underline{H}_m$? $\underline{G}_p \leftrightarrow G_m$?

$H_i(z) = H_{i0}(z^2) + zH_{i1}(z^2)$ $i = 0, 1$

$\rightarrow H_i(-z) = H_{i0}(z^2) + (-z)H_{i1}(z^2) =$

$= H_{i0}(z^2) - zH_{i1}(z^2)$

$$\therefore \begin{bmatrix} H_i(z) & H_i(-z) \end{bmatrix} = \begin{bmatrix} H_{i0}(z^2) & H_{i1}(z^2) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ z & -z \end{bmatrix}$$

Modulation / Polyphase Matrices

$$\therefore \underline{H}_m(z) = \underline{H}_p(z^2) \underbrace{\begin{bmatrix} 1 & 1 \\ z & -z \end{bmatrix}}$$

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & z \end{bmatrix}}_{\Delta(z)} \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{W}$$

delay chain on diagonal

Paraunitary, $\Delta(z)\Delta^H(1/z^*)=I$

$W^{-1} = \frac{1}{2}W$

almost unitary (DFT)

Modulation / Polyphase Relations

∴ $\underline{H}_m(z)$ ~paraunitary iff $\underline{H}_p(z)$ is (i.e. $\underline{H}_m^H(1/z^*)\underline{H}_m(z) = NI$)

∴ Analysis bank orthogonal iff $H_m(z)$ ~paraunitary

Can also derive relations for $\underline{G}_p$ and $\underline{G}_m$

$$G_m(z) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}}_{\Delta(z^{-1})} \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{W} \underline{G}_p(z^2)$$

Quadrature Mirror Filters (QMFs)

→ Convenient for H_0, H_1, to have mirror symmetry about $\omega = \frac{\pi}{2}$

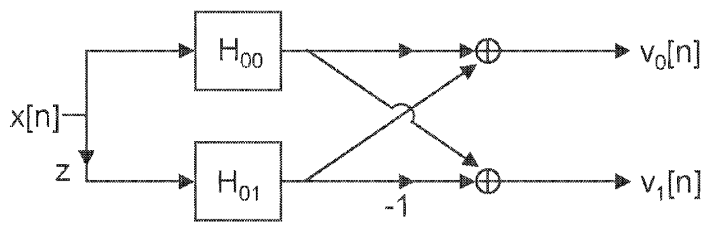
$H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)})$ then

$$H_0(e^{j\omega}) = H_{00}(e^{j2\omega}) + e^{j\omega}H_{01}(e^{j2\omega})$$

$$H_1(e^{j\omega}) = H_{00}(e^{j2\omega}) - e^{j\omega}H_{01}(e^{j2\omega})$$

$$\therefore \underline{H}_p = \begin{bmatrix} H_{00} & H_{01} \\ H_{00} & -H_{01} \end{bmatrix} \quad \text{(note odds)}$$

Efficient Implementation



For QMFs to be LTI

Alias cancellation:

$$G_0(z) = H_1(-z)C(z) \overset{\text{QMF}}{=} H_0(z)C(z)$$

$$G_1(z) = -C(z)H_0(-z)$$

$$C(z) = \frac{2F(z)}{\det H_m(z)} = \frac{2F(z)}{H_0(z)H_1(-z) - H_0(-z)H_1(z)}$$

$$\overset{\text{QMF}}{=} \frac{2F(z)}{H_0^2(z) - H_0^2(-z)}$$

Perfect Reconstruction / FIR QMFs

*For P.R. & FIR analysis/synthesis ($F(z) = \alpha z^{-\ell}$)

$$\Rightarrow \boxed{H_0^2(z) - H_0^2(-z) = \beta z^{-k}}$$

1) k must be odd (homework)
2) Only possible FIR $H_0(z) = C_0 z^{-2n_0} + C_1 z^{-(2n_1-1)}$
3) Many IIR solutions
4) Useful approximate linear phase FIR exist

Conjugate Quadrature Filters (CQF)

(perf. reconst. & orthogonal analysis)

P.R. & O.A.

$$\overbrace{\underline{G}_m(e^{j\omega})\underline{H}_m(e^{j\omega}) = 2I} \qquad \overbrace{\underline{H}_m^H(e^{j\omega})\underline{H}_m(e^{j\omega}) = 2I}$$

P.R. $\Rightarrow \underline{G}_m(e^{j\omega}) = H_m(e^{j\omega})^H$ (if real then time-reversal)

$$\therefore \; G_m^H(e^{j\omega})G_m(e^{j\omega}) = 2I \quad \underline{\underline{\text{O.S.}}}$$

CQF Orthogonality Implications

$$H_m^H(e^{j\omega})H_m(e^{j\omega}) = 2I$$

$$\begin{bmatrix} |H_0(e^{j\omega})|^2 + |H_0(-e^{j\omega})|^2 & D(e^{j\omega}) \\ D^*(e^{j\omega}) & |H_1(e^{j\omega})|^2 + |H_1(-e^{j\omega})|^2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$D(e^{j\omega}) = H_0(e^{j\omega})H_1^*(e^{j\omega}) + H_0(-e^{j\omega})H_1^*(-e^{j\omega}) = 0 \quad \star$$

ECE 451 - Lecture 40 16 of 19

CQF Key Result

If H_0, H_1 FIR, then $\star$ holds iff

$$H_1(e^{j\omega}) = -e^{-jk\omega}H_0^*(-e^{j\omega}), \quad \text{some odd } k$$

$$\Rightarrow |H_1(e^{j\omega})|^2 = |H_0(e^{j(\omega-\pi)})|^2$$

$$\Rightarrow \underbrace{|H_0(e^{j\omega})|^2}_{P(e^{j\omega})} + \underbrace{|H_0(e^{j(\omega-\pi)})|^2}_{P(e^{j(\omega-\pi)})} = 2$$

$P(z) + P(-z) = 2$ (P(z) “half band”)

ECE 451 - Lecture 40 17 of 19

CQF Properties

1) $h_0[n]$ must be of even length (homework)

2) Design: Choose halfband $P(z)$
   $\Rightarrow$ obtain $H_0(z)$ via spectral factorization

3) Allpass $H_0(z)$?

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

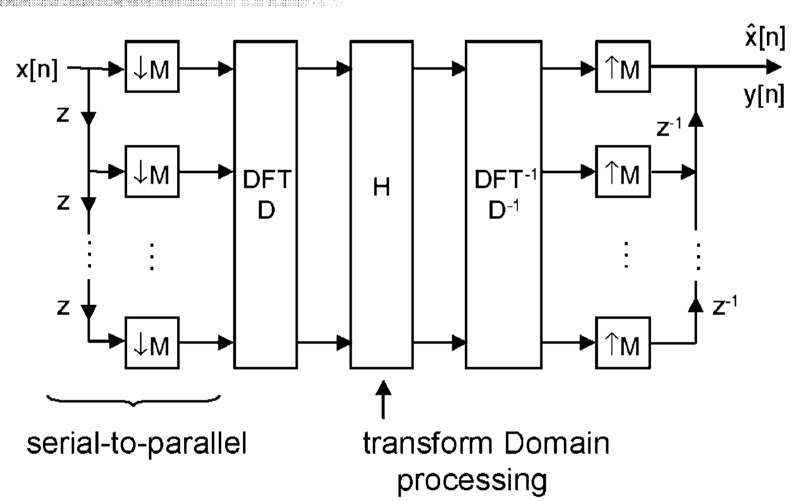
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 41

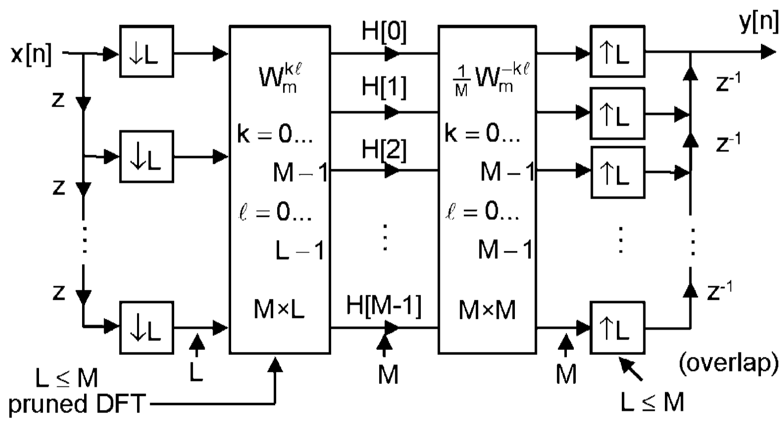
Professor Andrew Singer
Department of Electrical and Computer Engineering

Block Processing



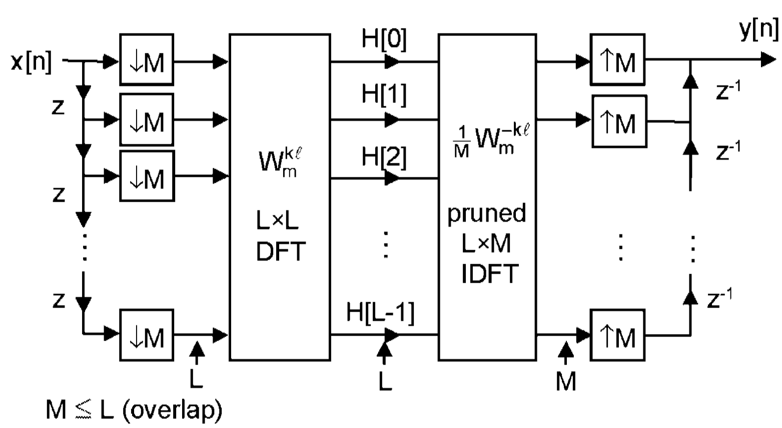
Block Convolution (OA)

L = Blocksize, P = filter length, $M = L + P - 1 \geq L$ = DFTsize

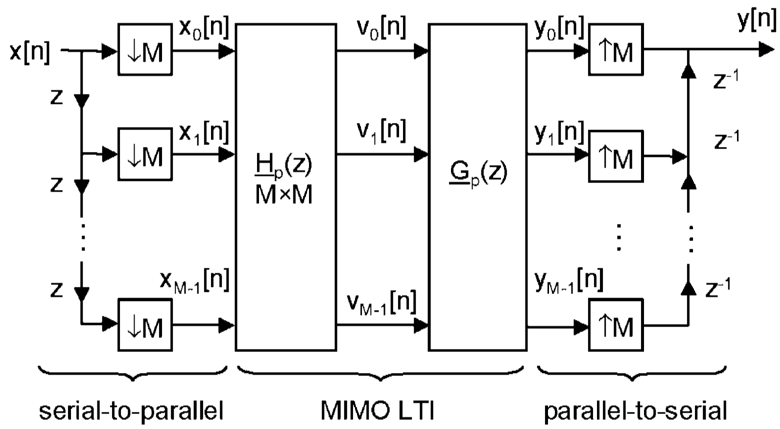


Block Convolution (OS)

L = DFT size, $M = L - P + 1 \leq L$ block spacing
P = filter length



Polyphase Representation



Polyphase Representation, cont'd

$$\underline{H}_p(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) \ldots & H_{0M-1}(z) \\ H_{10}(z) & & \vdots \\ \vdots & & \\ H_{M-10}(z) & \ldots & H_{M-1M-1}(z) \end{bmatrix}$$

$$\underline{G}_p(z) = \begin{bmatrix} G_{00}(z) & G_{10}(z) \ldots & G_{M-10}(z) \\ G_{01}(z) & & \vdots \\ \vdots & & \\ G_{0M-1}(z) & \ldots & G_{M-1M-1}(z) \end{bmatrix}$$

Polyphase Representation, cont'd

$$\underline{T}_p(z) = \underline{G}_p(z)\underline{H}_p(z)$$

0) Linear always

1) T.I. iff $\underline{T}_p(z)$ pseudo-circulant

$$\underline{T}_p(z) = \begin{bmatrix} F_0(z) & F_1(z) & \dots & F_{M-1}(z) \\ zF_{M-1}(z) & F_0(z) & \dots & F_{M-2}(z) \\ & & \ddots & \\ zF_1(z) & zF_2(z) & \dots & F_0(z) \end{bmatrix}$$

Polyphase Representation, cont'd

$$Y(z) = F(z)X(z),$$

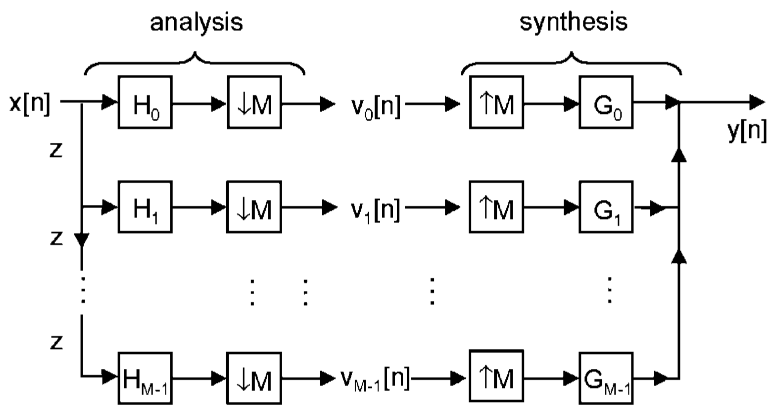
$$F(z) = F_0(z^M) + zF_1(z^M) + z^2F(z^M) + \dots + z^{M-1}F_{M-1}(z^M)$$

2) P.R., $F(z) = \alpha z^{-\ell}$

3) Analysis bank energy preserving iff $H_p(z)$ paraunitary

$$\underline{H}_p(e^{j\omega})^H \underline{H}_p(e^{j\omega}) = \underset{\forall\omega}{I}, \qquad \underline{H}_p(1/z^*)^H \underline{H}_p(z) = I$$

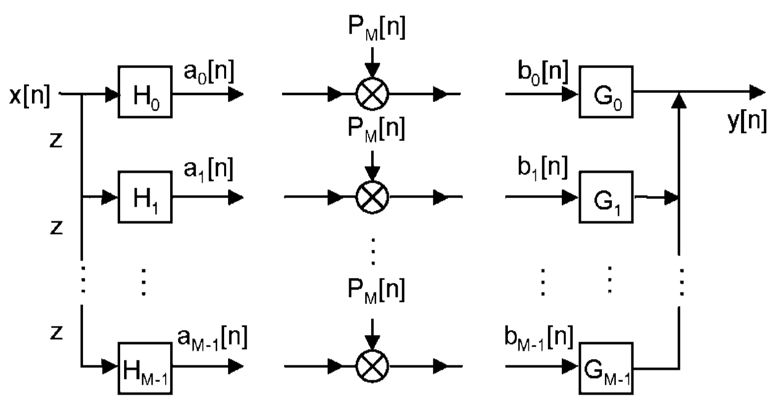
M-band Multirate Filterbank



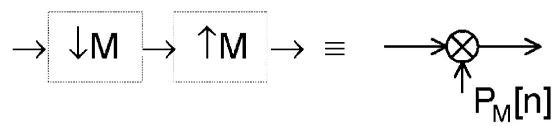
e.g.
IBPF's

Aliasing → LPTV system (period M)

Modulation-Domain Representation



Modulation Signal $P_M[n]$



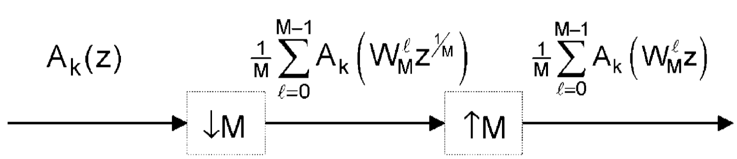
$$P_M[n] = \left[1 + W_M^{-n} + W_M^{-2n} + \ldots + W_M^{-(M-1)n}\right]\frac{1}{M}$$

$$W_m = e^{-j2\pi/M}$$

$$P_M[n] = \tfrac{1}{M}\sum_{k=0}^{M-1} W_M^{-kn}$$

Modulation Effect

$$B_k(z) = \tfrac{1}{M}\left[A_k(z) + A_k(W_M z) + \ldots + A_k(W_M^{M-1} z)\right]$$



Modulation-Domain Rep., cont'd

$$Y(z) = \tfrac{1}{M}\underbrace{\left[G_0(z)\ldots G_{M-1}(z)\right]}_{\underline{G}(z)} \times$$

$$\underbrace{\begin{bmatrix} H_0(z) & H_0(W_M z) \ldots & H_0(W_M^{M-1} z) \\ H_1(z) & H_1(W_M z) & \vdots \\ \vdots & \ddots & \\ H_{M-1}(z) & \ldots & H_{M-1}(W_m^{M-1} z) \end{bmatrix}}_{\underline{H}_M(z)} \underbrace{\begin{bmatrix} X(z) \\ X(W_M z) \\ \vdots \\ X(W_M^{M-1} z) \end{bmatrix}}_{\underline{X}_M(z)}$$

T.I. iff $G(z)\ H_M(z) = [MF(z)\ 0\ \ldots\ 0]$

Polyphase-Modulation Relationship

$$\underline{H}_M(e^{j\omega}) = \underline{H}_p(e^{j2\omega}) \underbrace{\begin{bmatrix} e^{j\omega} & & 0 \\ & \ddots & \\ 0 & & e^{j(M-1)\omega} \end{bmatrix}}_{\Delta(e^{j\omega})} [W]$$

$W_{[k\ell]} = W_M^{k\ell}$

$\Delta^H \Delta = I \qquad W^H W = MI$

1) P.R. iff $\det \underline{H}_p(e^{j\omega}) \neq 0 \ \ \forall\omega$

2) FIR analysis $\rightarrow$ P.R. possible w/ FIR synthesis iff $\det \underline{H}_p(z) = \beta z^{-k}$

The End

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign

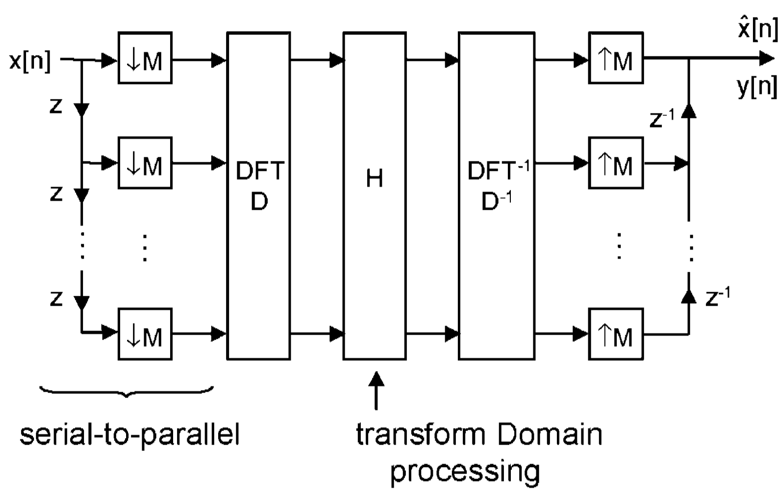
ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Lecture 41

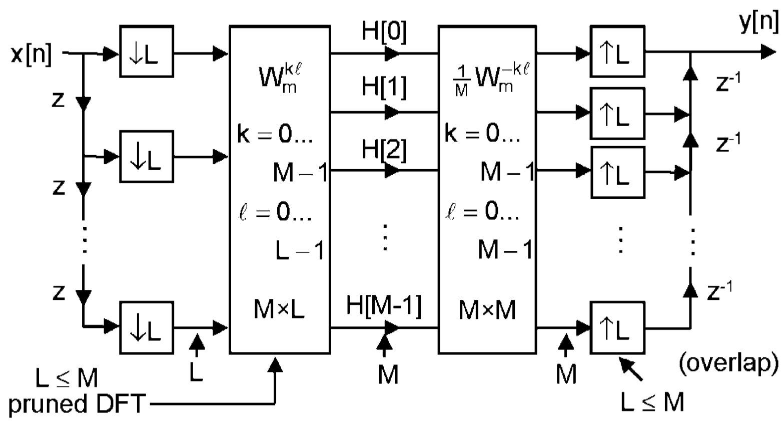
Professor Andrew Singer
Department of Electrical and
Computer Engineering

Block Processing



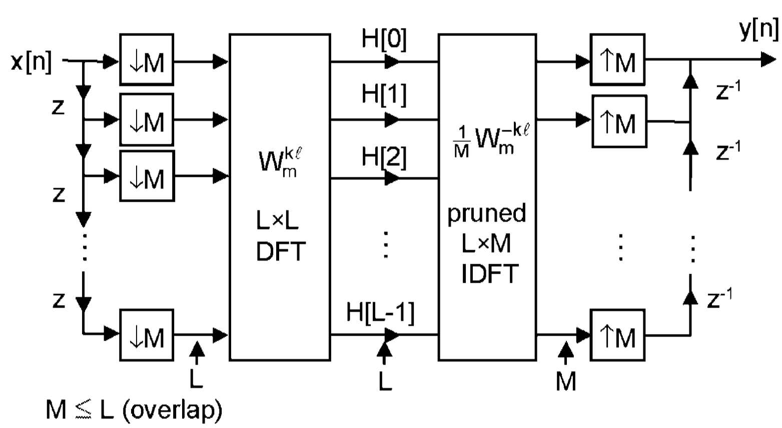
Block Convolution (OA)

L = Blocksize, P = filter length, M = L + P - 1 ≥ L = DFTsize

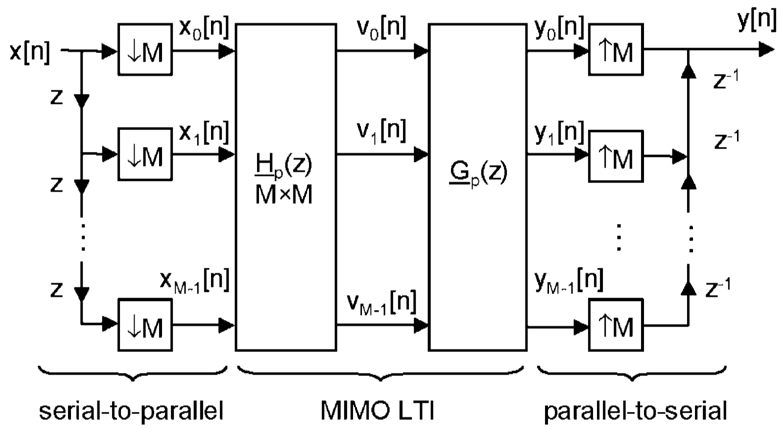


Block Convolution (OS)

L = DFT size, M = L - P + 1 ≤ L block spacing
P = filter length



Polyphase Representation



Polyphase Representation, cont'd

$$\underline{H}_p(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) \ldots & H_{0M-1}(z) \\ H_{10}(z) & & \\ \vdots & & \vdots \\ H_{M-10}(z) & \ldots & H_{M-1M-1}(z) \end{bmatrix}$$

$$\underline{G}_p(z) = \begin{bmatrix} G_{00}(z) & G_{10}(z) \ldots & G_{M-10}(z) \\ G_{01}(z) & & \\ \vdots & & \vdots \\ G_{0M-1}(z) & \ldots & G_{M-1M-1}(z) \end{bmatrix}$$

Polyphase Representation, cont'd

$$\underline{T}_p(z) = \underline{G}_p(z)\underline{H}_p(z)$$

0) Linear always

1) T.I. iff $\underline{T}_p(z)$ pseudo-circulant

$$\underline{T}_p(z) = \begin{bmatrix} F_0(z) & F_1(z) & \dots & F_{M-1}(z) \\ zF_{M-1}(z) & F_0(z) & \dots & F_{M-2}(z) \\ & & \ddots & \\ zF_1(z) & zF_2(z) & \dots & F_0(z) \end{bmatrix}$$

Polyphase Representation, cont'd

$$Y(z) = F(z)X(z),$$

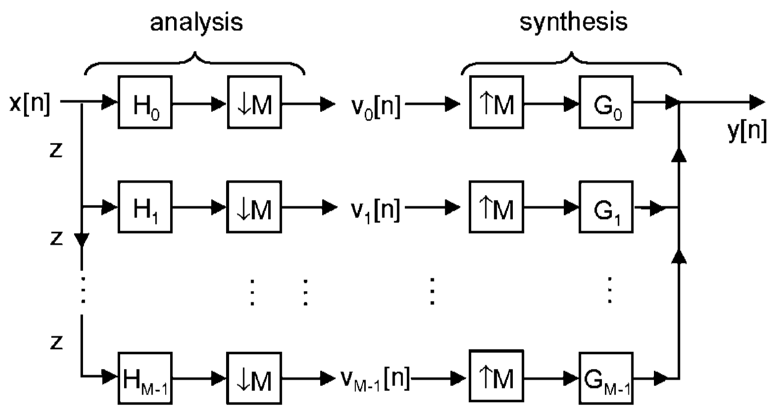
$$F(z) = F_0(z^M) + zF_1(z^M) + z^2F(z^M) + \dots + z^{M-1}F_{M-1}(z^M)$$

2) P.R., $F(z) = \alpha z^{-\ell}$

3) Analysis bank energy preserving iff $H_p(z)$ paraunitary

$$\underline{H}_p(e^{j\omega})^H \underline{H}_p(e^{j\omega}) = \underset{\forall\omega}{I}, \qquad \underline{H}_p(1/z^*)^H \underline{H}_p(z) = I$$

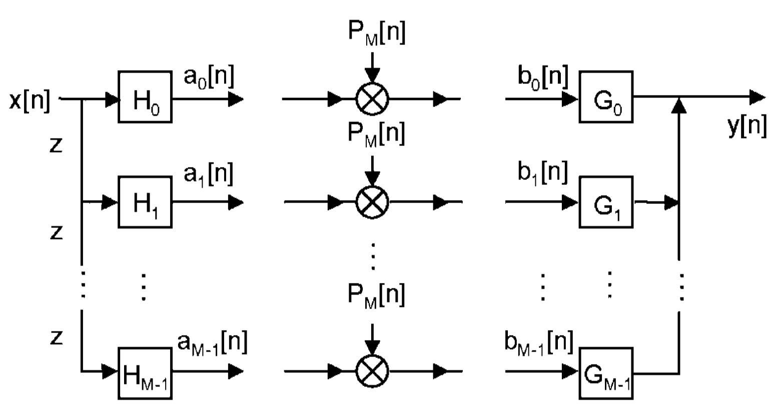
M-band Multirate Filterbank



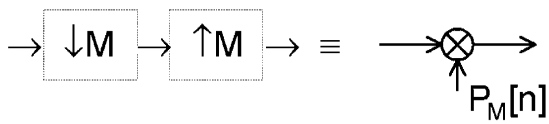
e.g.
IBPF's

Aliasing → LPTV system (period M)

Modulation-Domain Representation



Modulation Signal $P_M[n]$



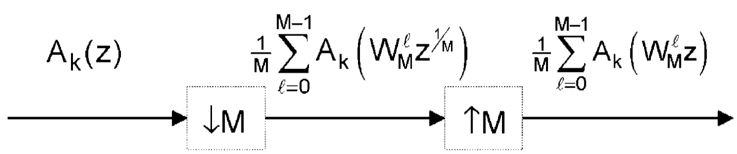
$$P_M[n] = \left[1 + W_M^{-n} + W_M^{-2n} + \ldots + W_M^{-(M-1)n}\right]\frac{1}{M}$$

$$W_m = e^{-j2\pi/M}$$

$$P_M[n] = \tfrac{1}{M}\sum_{k=0}^{M-1} W_M^{-kn}$$

Modulation Effect

$$B_k(z) = \tfrac{1}{M}\left[A_k(z) + A_k(W_M z) + \ldots + A_k(W_M^{M-1} z)\right]$$



Modulation-Domain Rep., cont'd

$$Y(z) = \frac{1}{M}\underbrace{\left[G_0(z) \ldots G_{M-1}(z)\right]}_{\underline{G}(z)} \times$$

$$\underbrace{\begin{bmatrix} H_0(z) & H_0(W_M z) & \ldots & H_0(W_M^{M-1} z) \\ H_1(z) & H_1(W_M z) & & \vdots \\ \vdots & & \ddots & \\ H_{M-1}(z) & \ldots & & H_{M-1}(W_m^{M-1} z) \end{bmatrix}}_{\underline{H}_M(z)} \underbrace{\begin{bmatrix} X(z) \\ X(W_M z) \\ \vdots \\ X(W_M^{M-1} z) \end{bmatrix}}_{\underline{X}_M(z)}$$

T.I. iff $G(z)\ H_M(z) = [MF(z)\ 0\ \ldots\ 0]$

Polyphase-Modulation Relationship

$$\underline{H}_M(e^{j\omega}) = \underline{H}_p(e^{j2\omega}) \underbrace{\begin{bmatrix} e^{j\omega} & & 0 \\ & \ddots & \\ 0 & & e^{j(M-1)\omega} \end{bmatrix}}_{\Delta(e^{j\omega})} [W]$$

$$W_{[k\ell]} = W_M^{k\ell}$$

$$\Delta^H \Delta = I \qquad W^H W = MI$$

1) P.R. iff $\det \underline{H}_p(e^{j\omega}) \neq 0 \quad \forall \omega$

2) FIR analysis → P.R. possible w/ FIR synthesis iff $\det \underline{H}_p(z) = \beta z^{-k}$

The End

ECE 451

ADVANCED DIGITAL SIGNAL PROCESSING

Produced by the
College of Engineering
University of Illinois at Urbana-Champaign