Adaptive Signal Processing

by
Andreas Spanias,
Professor and Director SenSIP Consortium
Department of Electrical Engineering
Arizona State University, Tempe, AZ 85287-5706

July 2007

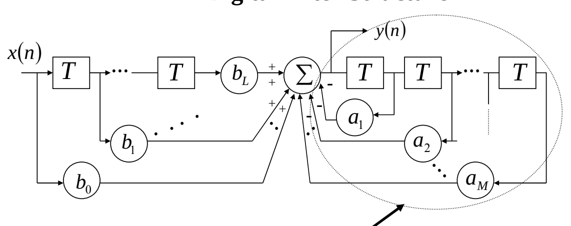
Presented to
European Thematic Network, ERASMUS-TN,
DOCTORAL EDUCATION IN COMPUTING (DEC)
and the the University of Cyprus

spanias@asu.edu

Phone: 480 965 1837
http://www.eas.asu.edu/~spanias

Background in DSP

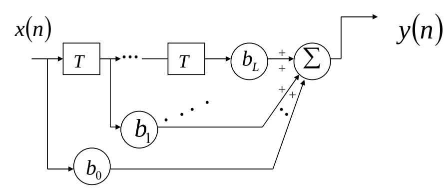
IIR Digital Filter Structure



feedback

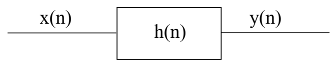
$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{i=1}^{M} a_i y(n-i)$$

FIR Digital Filter Structure



$$y(n) = \sum_{i=0}^{L} b_i x(n-i)$$

Two i/p-o/p Equations for Digital Filters



One can compute the output using the *convolution sum*

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)$$

or by using the *difference equation*

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{i=1}^{M} a_i y(n-i)$$

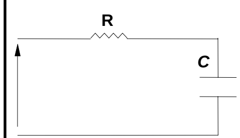
Remark: The impulse response *h(n)* can be determined by solving the difference equation.

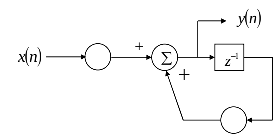
From the Laplace Transform to the Z-Transform

$$H_a(s) \Rightarrow H_d(z)$$

s-domain transfer function **z-domain transfer function**

$$H_a(s) = \frac{1}{1+sRC} \longrightarrow H_d(z) = \frac{b_o}{1+a_1 z^{-1}}$$





The Z-Transform - Definition

Given the signal: $x(n)$

its z-transform is $X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$

For causal signals, i.e., $x(n) = 0 \quad for \quad n < 0$

$$X(z) = \sum_{n=0}^{\infty} x(n) z^{-n}$$

The Transfer Function

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{i=1}^{M} a_i y(n-i)$$

$$X(z)\left(\sum_{i=0}^{L} b_i \; z^{-i}\right) = Y(z)\left(1 + \sum_{i=1}^{M} a_i \; z^{-i}\right)$$

The transfer function *H(z)* is defined as:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_L z^{-L}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}}$$

$$H(z) = \frac{\sum_{i=0}^{L} b_i \; z^{-i}}{1 + \sum_{i=1}^{M} a_i \; z^{-i}}$$

Note that feedback terms are in the denominator

The Frequency Response Function

The transfer function is

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_L z^{-L}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

by evaluating on the unit circle, i.e. for $z = e^{j\Omega}$

$$H(e^{j\Omega}) = \frac{b_0 + b_1 e^{-j\Omega} + \dots + b_L e^{-jL\Omega}}{1 + a_1 e^{-j\Omega} + \dots + a_M e^{-jM\Omega}}$$

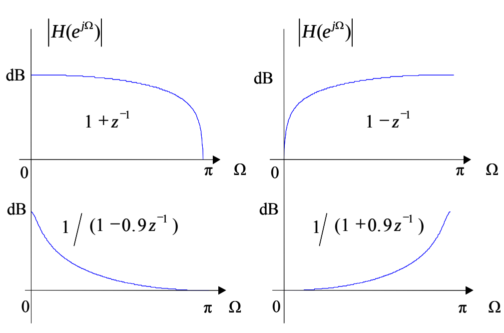
Poles and Zeros and Stability

$$H(z) = G\frac{(z-\zeta_1)(z-\zeta_2)\dots(z-\zeta_L)}{(z-p_1)(z-p_2)\dots(z-p_M)} = G\frac{\prod_{i=1}^{L}(z-\zeta_i)}{\prod_{i=1}^{M}(z-p_i)}$$

For stability of causal systems all the poles must be inside the unit circle, that is

$$|p_i| < 1 \quad \textit{for all} \quad i = 1, 2, \dots, M$$

Frequency Responses



Random Signal Processing - Some Definitions

The mean is defined as:

$$\mu_x = E[x(n)] = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)$$

Statistical expectation; for ergodic signals it is computed as a time average

The variance $\sigma_x^2 = E[(x(n)-\mu_x)(x(n)-\mu_x)] = E[x^2(n)] - \mu_x^2$

The variance is a measure of dispersion from the mean

The Autocorrelation

$$r_{xx}(m) = E[x(n+m)x(n)] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n+m)x(n)$$

Basic Properties: $r_{xx}(-m) = r_{xx}(m)$ $\qquad r_{xx}(0) \geq |r_{xx}(m)|$

uncorrelated i/p (e.g., white noise) → **Digital Filter** → **correlated o/p (e.g., colored noise)**

The filter acts as a correlator

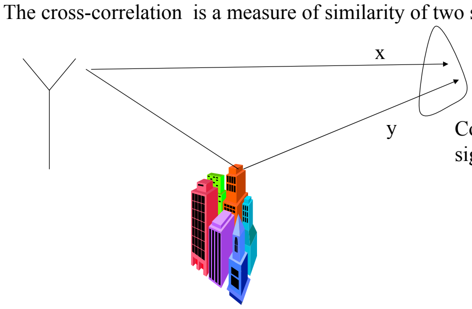
The autocorrelation is a measure of predictability of the signal, i.e., a correlated signal would be one whose future values can be predicted from past values

Mean, variance, and autocorrelation with J-DSP

The Cross-correlation and the Cross-covariance

$r_{xy}(m) = E[x(n+m)y(n)]$ $\qquad r_{xy}(m) = r_{yx}(-m)$

The cross-correlation is a measure of similarity of two signals.



The Cross-correlation of the Output Process

$x(n)$ → **Digital Filter** → $y(n)$

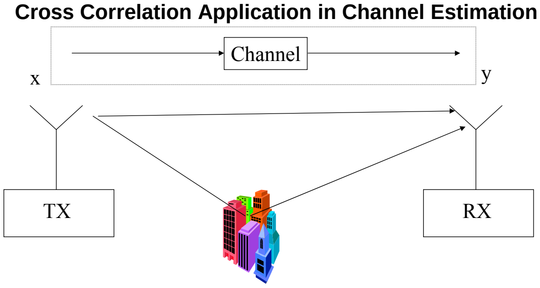
$$r_{yx}(m) = E[y(n+m)x(n)]$$

$$r_{yx}(m) = \sum_{i=-\infty}^{\infty} h(i) r_{xx}(m-i)$$

Another form

$$r_{yx}(m) = h(m) * r_{xx}(m)$$

Cross Correlation Application in Channel Estimation



The channel is often modeled by a transfer function that can be determined by measuring the statistics of the signal at the receiver. Hence, by sending a training "white" sequence 'x' a cross-correlation r_{yx} is measured and from that a channel impulse response

Cross Correlation Application in Channel Estimation (2)

$$r_{yx}(m) = \sum_{i=0}^{L} r_{xx}(m-i)\, b(i)$$

$$\begin{bmatrix} r_{yx}(0) \\ r_{yx}(1) \\ r_{yx}(2) \\ . \\ . \\ r_{yx}(L) \end{bmatrix} = \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) & \ldots & r_{xx}(1-L) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) & \ldots & r_{xx}(2-L) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \ldots & r_{xx}(3-L) \\ . & . & . & \ldots & . \\ . & . & . & \ldots & . \\ r_{xx}(L) & r_{xx}(L-1) & r_{xx}(L-2) & \ldots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ . \\ . \\ b(L) \end{bmatrix}$$

Calculating Autocorrelations

There are two kinds of autocorrelation estimates, i.e.,

Given an N-point data record $\{x(n), n = 0,1,2,\ldots, N-1\}$

a. Unbiased Estimates

$$\hat{r}(m) = \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} x(n+|m|)x(n)$$

b. Biased Estimates

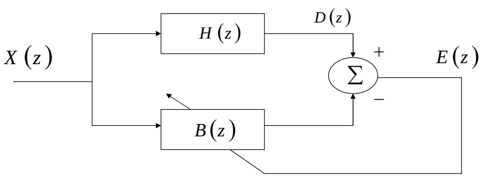
$$\tilde{r}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n+|m|)x(n)$$

$$-(N-1) \le m \le N-1$$

Preferred in most applications

ADAPTIVE FILTERING AND ITS APPLICATIONS

Adaptive System Identification - Adaptive Filtering



The idea is to make the filter B(z) behave like H(z)

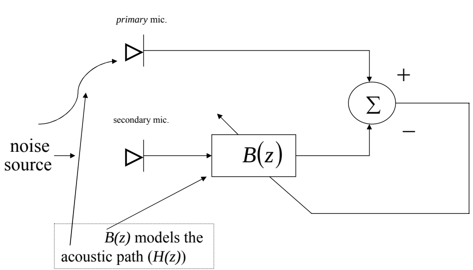
To do this we try to get B(z) to give the same output as H(z)

System identification arises in many applications, such as
adaptive noise and echo cancellation, channel equalization, active sound reduction, smart antennas, etc

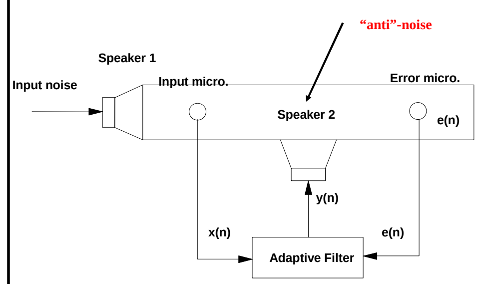
Adaptive System Identification

- The key problems in adaptive system identification lie in the choice of the adaptive filtering algorithm. In particular, the following issues are of concern
- presence of persistent and rich in frequency input (how good is the environment)
- adaptation speed (choice of adaptation gain)
- complexity of the algorithm (real time issues)
- quality of the final filter estimate (misadjustment)

Adaptive Noise Cancellation



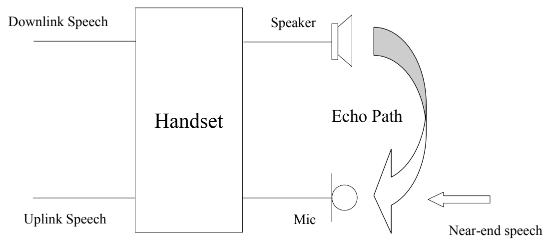
Active Noise Cancellation in Mufflers and Air Ducts



Filter adapts such that speaker generates "anti"-noise to cancel the noise and make e(n) small

Adaptive filter minimizes e(n)

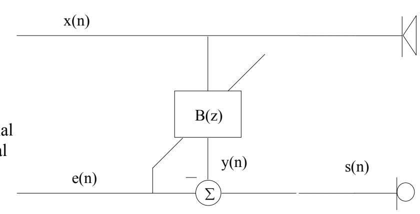
Acoustic Echo in Telephones



Ref: "Acoustic Echo Control: An Application of Very-High-Order Adaptive Filters," IEEE Signal Processing Magazine, July 1999

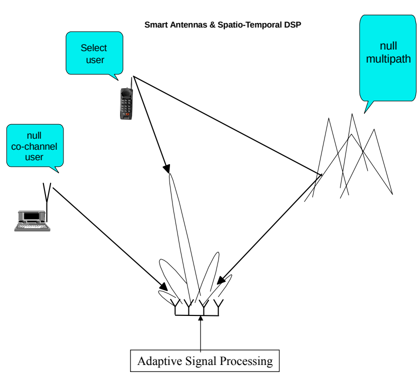
The Adaptive Echo Canceller

x - reference signal
s - reflected signal
e - error

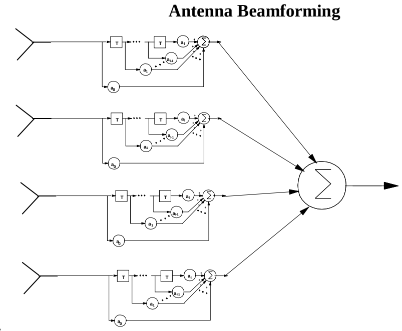


Echo cancellation is achieved by subtracting a replica of the echo from the reflected signal *s(n)*

Smart (Adaptive) Antennas



Antenna Beamforming

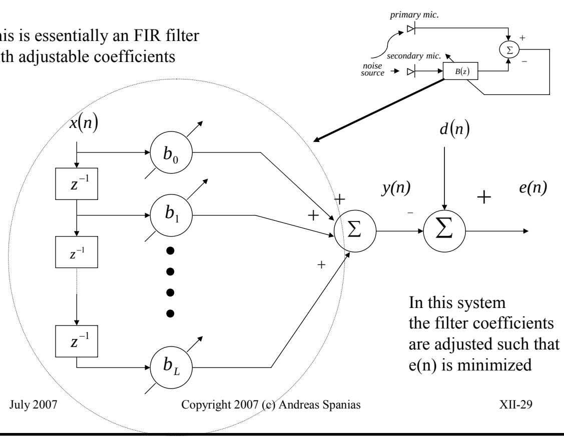


References
[1] B.D. VanVeen and K.M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," IEEE ASSP Magazine, Vol.2, No. 2, pp. 4-24, April 1988

[2] J.J. Shynk and M.D. Zotlowski, "Highlights of Signal Processing for Communications: Co-Channel Signal Separation and Adaptive Beamforming," Signal Processing Magazine, Vol. 16, No.2, March 1999

The Adaptive Linear Combiner

This is essentially an FIR filter with adjustable coefficients



In this system the filter coefficients are adjusted such that e(n) is minimized

The Adaptive Linear Combiner (Cont.)

$$e(n) = d(n) - y(n)$$

$$e(n) = d(n) - \sum_{i=0}^{L} b_i x(n-i)$$

$$\underline{x}(n) = \begin{bmatrix} \mathrm{x}(n) \\ x(n-1) \\ x(n-2) \\ . \\ . \\ . \\ \mathrm{x}(n-L) \end{bmatrix} \qquad \underline{\mathrm{b}} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ \mathrm{b}_{\mathrm{L}} \end{bmatrix}$$

The Mean Square Error (MSE)

The Error $e(n) = d(n) - \underline{x}^T(n)\underline{b}$

The MSE $\in = E\left[e^2(n)\right] = E\left[(d(n) - y(n))^2\right]$

Using vector notation, i.e.,

$$E\left[e^2(n)\right] = E\left[d^2(n)\right] - 2\mathrm{E}\left[\mathrm{d}(n)\underline{x}^T(n)\right]\underline{b} + \underline{\mathrm{b}}^T\,\mathrm{E}\left[\underline{\mathrm{x}}(n)\underline{x}^T(n)\right]\underline{\mathrm{b}}$$

where

$$\underline{p} = E\left[d(n)\underline{x}(n)\right] \quad \text{cross correlation vector}$$

$$\underline{\mathrm{R}} = E\left[\underline{x}(n)\underline{x}^T(n)\right] \quad \text{autocorrelation matrix}$$

MSE Solution

Minimizing $\in$, i.e., $\underline{\nabla}_{\in} = \dfrac{\partial \in}{\partial \underline{b}} = \underline{0}$

we get $\underline{b}^0 = \underline{R}^{-1}\underline{p}$

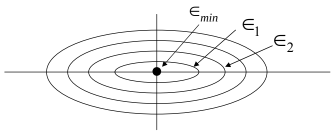
$$\underline{R} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \dots & r_{xx}(l) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(l-1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(l-2) \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ r_{xx}(l) & r_{xx}(l-1) & r_{xx}(l-2) & \dots & r_{xx}(0) \end{bmatrix}$$

Remarks on the MSE Solution

Which implies that the error becomes uncorrelated with the inputs.
If the input is white noise then

$$\underline{R} = \sigma_x^2 \underline{I} \qquad \underline{b}^0 = \underline{p} / \sigma_x^2$$

If R is a 2x2 matrix then the MSE can be described in terms of ellipses corresponding to constant MSE contours. MSE is minimum at the center of the ellipses.

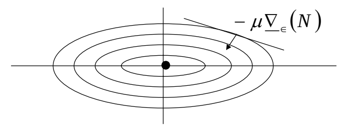


Gradient Adaptive Filtering Algorithms

The Steepest Descent Algorithm (SDA)

$$\underline{b}(N+1) = \underline{b}(N) - \mu \underline{\nabla}_{\in}(N)$$

the SDA converges if $\quad 0 < \mu < \dfrac{1}{\lambda_{\max}(\underline{R})}$



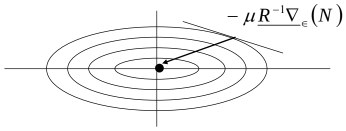
The LMS is a steepest-descent algorithm where the gradient is calculated only from the present error sample

The Newton Algorithm (NA)

$$\underline{b}(N+1) = \underline{b}(N) - \mu R^{-1} \underline{\nabla}_{\in}(N)$$

the NA converges if

$$0 < \mu < 1$$



NA is faster than SDA in terms of convergence but requires a matrix inverse in every iteration

Excess Mean Square Error and Misadjustment

This is defined at steady-state, i.e., after the adaptive transients vanish. The Excess MSE is due to noise in the adaptive process. The Excess MSE provides a measure of the difference between the actual and the optimal performance over time.

$$\textit{Excess}\ \mathrm{MSE} = \mathrm{E}\left[\in(\mathrm{k}) - \in_{\min}\right]$$

The misadjustment M is normalized measure of the difference between the actual and the optimal performance.

$$M = \frac{\textit{Excess}\ \mathrm{MSE}}{\in_{\min}}$$

THE SEQUENTIAL AND THE BLOCK LMS ALGORITHMS

July 2007 Copyright 2007 (c) Andreas Spanias XII-37

The Sequential LMS Algorithm

The LMS algorithm is due to Widrow. It is a steepest descent type of algorithm that uses an estimate of the gradient instead of the true gradient.

The SDA : $$\underline{b}(N+1)=\underline{b}(N)-\mu\underline{\nabla}_{\in}(N)$$

The LMS : $$\underline{b}(N+1)=\underline{b}(N)-\mu\hat{\underline{\nabla}}_{\in}(N)$$

where $$\hat{\underline{\nabla}}_{\in}(N)=-2e(N)\underline{x}(N)$$

July 2007 Copyright 2007 (c) Andreas Spanias XII-38

Convergence of the LMS

$$\underline{b}(N+1)=\underline{b}(N)+2\mu e(N)\underline{x}(N)$$

It can be shown that

$$\lim_{N\to\infty}\{E[\underline{b}(N)]\}=\underline{b}^0$$

if

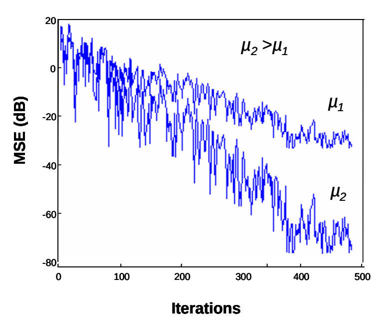
$$0<\mu<\frac{1}{\lambda_{\max}}$$

A more conservative convergence condition

$$0<\mu<\frac{1}{tr(r)}$$

July 2007 Copyright 2007 (c) Andreas Spanias XII-39

Simulations LMS



July 2007 Copyright 2007 (c) Andreas Spanias XII-40

Misadjustment of the LMS

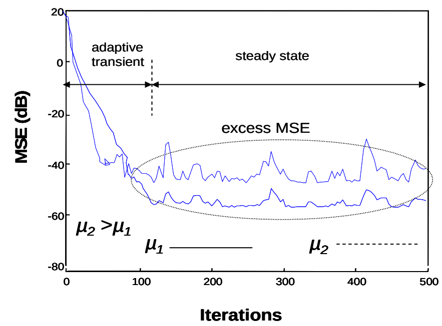
It can be shown [Windrow and Stearns] that

$$M \approx \mu \mathrm{tr}(\underline{r})$$

Longest Learning time constant

$$T_{mse} = \frac{1}{4\mu\lambda_{\min}}$$

Simulations LMS – Misadjustment and Excess MSE



The LMS / Newton Algorithm

$$\underline{b}(N+1) = \underline{b}(N) - \mu\lambda_{ave}\, \underline{r}^{-1}\hat{\underline{\nabla}}_{\in}(N)$$

The stability condition

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

$$M \approx \mu\, tr(\underline{r}) \qquad T_{mse} = \frac{1}{4\mu\lambda_{ave}}$$

Adaptive Filter Simulation Program (1)

```
Example 2.26 (file name: dsp2_26.m)   Page 1
% Compute PSD from LPC - Ex 2.26
clear
N=500; % total number of iterations
x=randn(N,1);%white Gaussian noise generated
Nf=256;
theta=[(2*pi/Nf).*[0:Nf-1]]; % precompute the set of discrete frequencies
L=2; %order of adaptive filter L=no. of coeff.-1
bhat(1:L+1)=0% initialized adaptive filter coefficients
mu=0.01 ;% step size

%*******************************************************

% Fixed "unknown" filter
b=[1 1 1];
a=[1];
d = filter(b,a,x);% filter output

% Form output of the adaptive filter and calculate error

H=freqz(b,a,theta); % compute the frequency response of H(z)
jj=1;%initialize counter

%*******************************************************
```

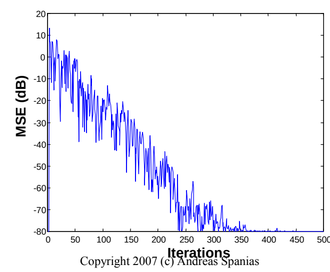
Adaptive Filter Simulation Program (2)

Example 2.26 (file name: dsp2_26.m)   Page 2

```
for n=L+1:N
   y(n)=0;%initialize
   for l=1:L+1
      y(n)=y(n)+bhat(l)*x(n-l+1);%compute output of adaptive filter
   end
   e(n)=d(n)-y(n);%compute error

   for l=1:L+1
      bhat(l)=bhat(l)+2*mu*e(n)*x(n-l+1);%adapt filter coefficients

   end
   jj=jj+1;%counter for plotting every 10 iterations

   if jj==10; %plot frequency responses every 10 iterations

   Hh=freqz(bhat,1,theta); % compute the frequency response (Hh(z))
    plot(theta,20*log10(abs(Hh)),theta,20*log10(abs(H))); % plot PSDs
    xlabel('frequency index')
    ylabel('Magnitude in (dB)')
    jj=0;
    pause;% press space bar to see the next frequency reponse
end
end
plot(10*log10(e.*e+0.00000001));%  plot sample mse vs iterations
ylabel('MSE (dB)')
xlabel('Iteration')
```


Adaptive Filter Simulation Program (3)

The simulation program in gives the MSE curve as a function of iteration

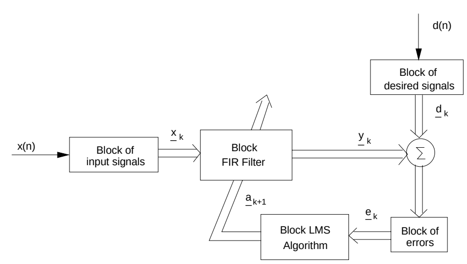


Vectorized Simulation of Adaptive Filter

Example 2.26a (file name: dsp2_26a.m)

```
for n=L+1:N
   xv=x(n:-1:n-L);
   yh=bhat*xv; %computer o/p of adaptive filter
   e(n)=d(n)-yh;   %compute error
   bhat=bhat+2*mu*e(n)*xv';%adapt filter coefficients

   jj=jj+1;%counter for plotting every 10 iterations

   if jj==10; %plot frequency responses every 10 iterations

   Hh=freqz(bhat,1,theta); % compute the frequency response (Hh(z))
    plot(theta,20*log10(abs(Hh)),theta,20*log10(abs(H))); % plot PSDs
    xlabel('frequency index')
    ylabel('Magnitude in (dB)')
    jj=0;
    pause;% press space bar to see the next frequency response
end
end
plot(10*log10(e.*e+0.00000001));%  plot sample mse vs iterations
ylabel('MSE (dB)')
xlabel('Iteration')
```


Block Algorithms



Block Algorithms (Cont.)

The block LMS algorithm, as opposed to the sequential LMS algorithm, minimizes a block (vector) of errors:

$$\in(k) = E\left[\underline{e}^T(k)\underline{e}(k)\right] \qquad \underline{e}(k) = \underline{d}(k) - \underline{y}(k)$$

or explicitly

$$\begin{bmatrix} e(ks) \\ e(ks+1) \\ e(ks+2) \\ . \\ . \\ . \\ e(ks+N-1) \end{bmatrix} = \begin{bmatrix} d(ks) \\ d(ks+1) \\ d(ks+2) \\ . \\ . \\ . \\ d(ks+N-1) \end{bmatrix} - \begin{bmatrix} y(ks) \\ y(ks+1) \\ y(ks+2) \\ . \\ . \\ . \\ y(ks+N-1) \end{bmatrix}$$

Block Algorithms (Cont.)

The output vector can be written as

$$\begin{bmatrix} \mathbf{y(ks)} \\ \mathbf{y(ks+1)} \\ \mathbf{y(ks+2)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{y(ks+N-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x(ks)} & \mathbf{x(ks-1)} & \mathbf{x(ks-2)} & \mathbf{..\ x(ks-L)} \\ \mathbf{x(ks+1)} & \mathbf{x(ks)} & \mathbf{x(ks-1)} & \mathbf{..\ x(ks-L+1)} \\ \mathbf{x(ks+2)} & \mathbf{x(ks+1)} & \mathbf{x(ks)} & \mathbf{..\ x(ks-L+2)} \\ \cdot & \cdot & \cdot & \cdot\cdot\ \ \cdot \\ \cdot & \cdot & \cdot & \cdot\cdot\ \ \cdot \\ \cdot & \cdot & \cdot & \cdot\cdot\ \ \cdot \\ \mathbf{x(ks+N-1)} & \mathbf{x(ks+N-2)} & \mathbf{x(ks+N-3)} & \mathbf{..\ x(ks+N-L-1)} \end{bmatrix} \begin{bmatrix} \mathbf{b_0(k)} \\ \mathbf{b_1(k)} \\ \mathbf{b_2(k)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{b_L(k)} \end{bmatrix}$$

The block error $\underline{e}(k) = \underline{d}(k) - \underline{x}_B(k)\underline{b}(k)$

The BLOCK LMS Algorithm

$$\underline{e}(k) = \underline{d}(k) - \underline{x}_B(k)\underline{b}(k)$$

$$\underline{b}(k+1) = \underline{b}(k) + 2\mu\,\underline{x}_B^T(k)\underline{e}(k)$$

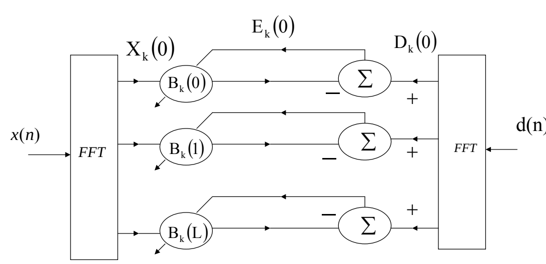
The stability and misadjustment[Clark et al]

$$0 < \mu < \frac{1}{\lambda_{\max}(\underline{R}_B)} \qquad M \approx \mu tr(\underline{R}_B)$$

$$\underline{R}_B = E[\underline{x}_B(k)\underline{x}^T{}_B(k)]$$

The Frequency Domain Adaptive Filter (FDAF)

This is due to [Dentino et al]



$L = N - 1$

The (FDAF) (Cont.)

$$\underline{E}(k) = \underline{D}(k) - \underline{X}_d(k)\underline{B}(k)$$

The coefficient update expression for the FDAF is given by

$$\underline{B}(k+1) = \underline{B}(k) + 2\mu \underline{X}_d^H(k)\underline{E}(k)$$

The stability condition is $$0 < \mu < \frac{1}{\lambda_{\max}(\underline{R}_d)}$$

or $$0 < \mu < \frac{1}{\max_i \left(E\left[|X_k(i)|^2\right]\right)}$$

Remarks on the FDAF

The use of the FFT for fast convolution is associated with periodic (circular) as opposed to linear convolution.

The optimal solution for the FDAF is not the same as that of the LMS or the BLMS. In order to see the differences between the two solutions one has to examine the time domain equivalent of the FDAF.

The FDAF is a block frequency domain algorithm associated with a block time-domain circulant matrix.

The FLMS Algorithm

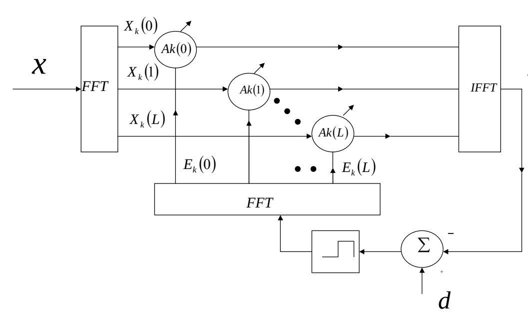
The FLMS algorithm is a frequency domain algorithm which is equivalent to the BLMS algorithm.

The FLMS uses 2N-point (augmented) FFTs to implement an Nth order BLMS adaptive filter in the frequency domain.

All the convolutions are done properly in the frequency domain using the overlap and save technique [Oppenheim].

In addition, the gradient is constrained such that it corresponds to that of the BLMS.

The FLMS Algorithm (2)



The Choice of the Convergence factor μ

Much work has been done in choosing the convergence factor or the step size of gradient algorithms.

A procedure for optimizing the convergence factors was proposed by [Mikhael and Yassa]. Time-varying convergence factors for the LMS and the BLMS were proposed by [Mikhael and Wu].

Time-varying convergence factors for frequency domain algorithms were porpsed by [Mikhael and Spanias].

Normalized convergence factors were also proposed by [Mansour and Gray].

The Choice of the Convergence factor μ (Cont.)

It can be shown [Yassa] that

$$\mu_k = \frac{\underline{\nabla}^T(k)\underline{\nabla}(k)}{2\underline{\nabla}^T(k)\underline{R}\underline{\nabla}(k)}$$

Choice of μ for the LMS (Cont.)

$$\mu_N = \frac{1}{2\sum_{i=0}^{L} x^2(N-i)}$$

Choice of μ for the BLMS

$$\underline{b}(k+1) = \underline{b}(k) - 2\mu_k \underline{x}_B^T(k)\underline{e}(k)$$

$$\mu_k = \frac{\underline{e}^T(k)\underline{x}_B(k)\underline{x}_B^T(k)\underline{e}(k)}{2e^T(k)\underline{x}_B(k)\underline{x}_B^T(k)\underline{x}_B(k)\underline{x}_B^T(k)\underline{e}(k)}$$

The last two equations describe the Optimum Block Algorithm (OBA)

Choice of μ for the FDAF

$$\underline{B}(k+1) = \underline{B}(k) - 2\mu_k \underline{X}_d^H(k)\underline{E}(k)$$

$$\mu_k = \frac{\sum_{i=0}^{N-1} |E_k(i)|^2 |\mathrm{X}_k(i)|^2}{2\sum_{i=0}^{N-1} |E_k(i)|^2 |\mathrm{X}_k(i)|^4}$$

Another frequency-depended gain is given below

$$\mu_k(i) = \frac{1}{\frac{2}{L}\sum_{j=k-L}^{k} |\mathrm{X}_k(i)|^2}$$

Choice of μ for the FDAF and FLMS

In addition an individual μ for each coefficient was proposed by [Mikhael and Spanias]. This is given by

$$\mu_k(i) = \frac{1}{\frac{2}{L}\sum_{j=k-L}^{k} |X_k(i)|^2}$$

THE DATA STRUCTURE OF THE AUTOCORRELATION MATRIX AND THE RLS

$$\underline{R} = E\left[\begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ . \\ . \\ . \\ x(n-L) \end{bmatrix} \left[x(n)x(n-1)x(n-2)...x(n-L)\right]\right]$$

$$\underline{\hat{R}}(N) = \frac{N-1}{N}\underline{\hat{R}}(N-1) + \frac{1}{N}\underline{x}(N)\underline{x}^T(N)$$

$$\underline{\hat{p}}(N) = \frac{N-1}{N}\underline{\hat{p}}(N-1) + \frac{1}{N}\underline{x}(N)d(N)$$

THE RLS

$$\underline{\hat{R}}^{-1}(N+1) = \frac{N+1}{N}\left(\underline{\hat{R}}^{-1}(N) - \frac{\left(\underline{\hat{R}}^{-1}(N)\underline{x}(N)\right)\left(\underline{x}^T(N)\underline{\hat{R}}^{-1}(N)\right)}{N + \underline{x}^T(N)\underline{\hat{R}}^{-1}(N)\underline{x}(N)}\right)$$

The solution using k samples is

$$\underline{b}(N) = \underline{\hat{R}}(N)^{-1}\underline{\hat{p}}(N)$$

The (k+1)th sample that enters the process. A new solution is

$$\underline{b}(N+1) = \underline{\hat{R}}(N+1)^{-1}\underline{\hat{p}}(N+1)$$

THE WEIGHTED RLS (WRLS)

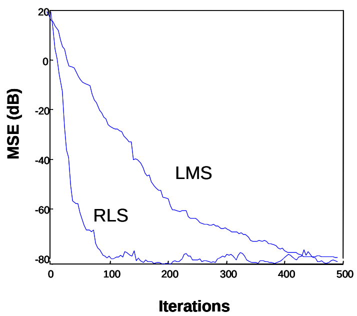
If the estimates of the autocorrelation matrix are modified such that a forgetting factor is introduced, i.e., current (recent) data is emphasized relative to older data then we get a modified time-recursive algorithm called the Weighted RLS (WRLS).

$$\underline{\hat{R}}^w(N+1) = \gamma\underline{\hat{R}}^w(N) + \underline{x}(N+1)\underline{x}^T(N+1)$$

$$\underline{p}^w(N+1) = \gamma\underline{p}^w(N) + \underline{d}(N+1)x(N+1)$$

$$\left[\underline{\hat{R}}_m^{\hat{w}}(N+1)\right]^{-1} = \gamma^{-1}\left(\left[\underline{\hat{R}}^w(N)\right]^{-1} - \frac{\left(\left[\underline{\hat{R}}^w(N)\right]^{-1}\underline{x}(N)\right)\left(\underline{x}^T(N)\left[\underline{\hat{R}}^w(N)\right]^{-1}\right)}{\gamma + \underline{x}^T(N)\left[\underline{\hat{R}}^w(N)\right]^{-1}\underline{x}(N)}\right)$$

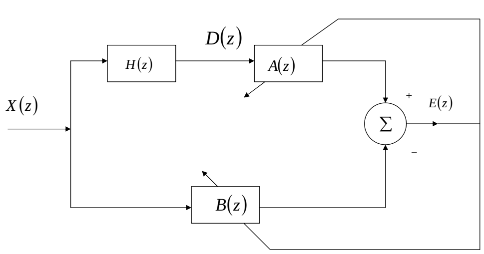
Simulations RLS vs LMS



IIR Adaptive Filtering

Inherent problems in IIR filtering is the stability of the filter during adaptation and the non-linearity of the optimization problem. There are basically two general techniques for adaptive filtering, namely, the equation error model and the output error model. The equation error model involves linear optimization and is stable during adaptation. The output error model involves non-linear optimization. Output error models are discussed by [Shunk]. In the following we describe the equation error model. The structure of the equation error model is given below:

IIR Adaptive Filtering (Cont.)



The Equation Error Model (EEM)

The error equation for the EEM is given by

$$e(n) = d(n) + \sum_{i=1}^{M} a_i d(n-i) - \sum_{i=0}^{L} b_i x(n-i)$$

Let us define the following (L+M+1) x 1 vectors

$$\underline{u}(n) = [-x(n) \text{ - } x(n-1) .. \text{ - } x(n-l)\, d(n-1)\, d(n-2) .. x(d-m)]$$

and

$$\underline{c}(n) = [b_0(n)\ b_1(n-1) .. b_L(n)\, a_1(n)\, a_2(n) .. a_M(n)]$$

$$e(n) = d(n) + \underline{u}^T(n)\underline{c}(n)$$

The Equation Error Model (Cont.)

The MSE

$$\in = E\left[e^2(n)\right] = E\left[d^2(n)\right] - 2\mathrm{E}\left[d(n)\underline{u}^T(n)\right]\underline{c} + \underline{c}^T\ \mathrm{E}\left[\underline{u}(n)\underline{u}^T(n)\right]\underline{c}$$

Minimizing

$$\underline{\nabla}_{\in} = \frac{\partial \in}{\partial \underline{c}} = \underline{0}$$

we get

$$\underline{c}^0 = \underline{r}^{-1}\,\underline{p}$$

$$\underline{p} = E\left[d(n)\underline{u}(n)\right]$$

$$\underline{r} = E\left[\underline{u}(n)\underline{u}^T(n)\right]$$

The LMS Algorithm for the EEM

$$\underline{c}(k+1) = \underline{c}(k) - \mu\hat{\underline{\nabla}}_{\in}(k)$$

where

$$\hat{\underline{\nabla}}_{\in}(k) = -2e(k)\underline{u}(k)$$

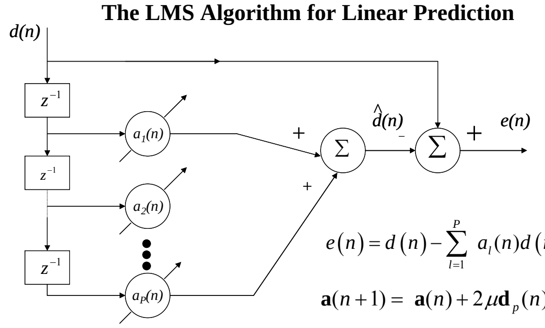
Stable if

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

Misadjustment

$$M \approx \mu tr(\underline{r})$$

The LMS Algorithm for Linear Prediction



$$e(n) = d(n) - \sum_{l=1}^{P} a_l(n)d(n-l)$$

$$\mathbf{a}(n+1) = \mathbf{a}(n) + 2\mu\mathbf{d}_p(n)e(n)$$

$$\mathbf{d}_p(n) = \begin{bmatrix} d(n-1) \\ d(n-2) \\ . \\ . \\ \mathrm{d}(n-p) \end{bmatrix} \qquad \mathbf{a}(\mathrm{n}) = \begin{bmatrix} a_1(n) \\ a_2(n) \\ . \\ . \\ \mathrm{a}_p(n) \end{bmatrix}$$

WEB SITES AND REFERENCES

Web Site for JDSP PROGRAM

- The JDSP program can be found at

http://jdsp.asu.edu

July 2007 XII-73

REFERENCES

DIGITAL SIGNAL PROCESSING

- A. Oppenheim and R. Schafer, Digital Signal Processing, Prentice Hall, Englewood Cliffs, 1975.
- R. Crochiere and L. Rabiner, Multirate Digital Signal Processing, Prentice Hall, 1983.
- Proakis and Manolakis, "Introduction to Digital Signal Processing," Macmillan,1988
- Oppenheim and Schaffer, "Discrete-Time Signal Processing, Ed. 2" Prentice Hall, 1999
- McClellan and Schaffer, "DSP First," Prentice Hall, 1997
- R.W. Ramirez, "the FFT Fundamentals and Concepts," Prentice-Hall, INC.
- Englewood Cliffs, NJ 1985.
- D.J. DeFatta, J.G. Lucas, W.S. Hodgkiss, "Digital Signal Processing: A System Design Approach," Wiley, 1988
- R.A. Gabel and R.A. Roberts, "Signals and Linear Systems," Wiley, 1988
- Johnson, "Introduction to Digital Signal Processing," Prentice Hall, 1988
- Steven Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," 2nd Ed., California Technical Publishing, San Diego California, 1999 (you can download at DSPguide.com)

July 2007 XII-74

REFERENCES (2)

SPECTRAL ESTIMATION

- S.L. Marple, Digital Spectral Analysis with Applications, Prentice Hall, 1987.
- J. Burg, "Maximum Entropy Spectral Analysis," Proc. 37th Meeting of the Society of Exploration Geophysicists, 1967.
- C. Nikias and M. Raghuveer, "Bispectrum Estimation: A Digital Signal Processing Framework," Proc. IEEE, No. 75, pp. 869-891, July 1987.
- Stoica and Moses, Introduction to Spectral Analysis, Prentice Hall 2nd Ed 2004

ADAPTIVE FILTERS

- Haykin, Adaptive Filter Theorty, Prentice Hall, 3rd Ed, 1997
- Cowan and Grant, "Adaptive Filters," Prentice Hall, 1985
- Widrow and Stearns, "Adaptive Signal Processing," Prentice Hall, 1985
- S.J. Elliott, I.M. Stothers, and P.A. Nelson, "A Multiple Error LMS Algorithm and its Application to the Active Control of Noise and Vibration," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-35 (10), 1987, pp. 1423-1434.
- D.R. Morgan, "An analysis of Multiple Correlation Cancellation Loops with a Filter in the Auxiliary Path," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-28(4), pp. 454-467, 1980.
- J. Treichler, C. Johnson, and Larimore, Theory and Design of Adaptive Filters, Prentice Hall 2001

July 2007 XII-75

Other Adaptive Signal Processing References

- [Clar81] G. Clark, S. Mitra, and S. Parker, "Block Implementation of Adaptive Digital Filters," IEEE Trans. on ASSP-29, pp. 744-752, June 1981.
- [Cowa85] C. Cowan and P. Grant, Adaptive Filters, Prentice Hall, 1985.
- [Deis94] M. Deisher and A. S. Spanias, "Practical Considerations in the Implementation Frequency-Domain Adaptive Noise Cancellation," IEEE Trans. on Circuits and Systems, Part II: Analog and Digital Signal Processing, vol. 41, No. 2, pp. 164-168, Feb. 1994.
- [Elio87] S.J. Elliott, I.M. Stothers, and P.A. Nelson, "A Multiple Error LMS Algorithm and its Application to the Active Control of Noise and Vibration," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-35 (10), pp. 1423-1434, 1987..
- [Grif75] L.J. Griffiths, "Rapid Measurements of Digital Instantaneous Frequency," IEEE Trans. on Trans. Acoustics, Speech, and Signal Processing, Vol-23, pp. 207-222, Apr. 1975.
- [Hayk97] S. Haykin, Adaptive Filter Theory, Prentice Hall, 3rd Ed., 1997.
- [Mikh87a] W. Mikhael and A. Spanias, "Comparison of Several Frequency-Domain LMS Algorithms," IEEE Trans. on Circuits and Systems, Vol. CAS-34, No. 5, pp. 586-589, May 1987.
- [Mikh87b] W. Mikhael and F. Wu, "Fast Algorithms for Block FIR Adaptive Digital Filtering," IEEE Trans. on Circuits and Systems, CAS-34, p. 1152, Oct. 1987.
- [Marp87] S.L. Marple, Digital Spectral Analysis with Applications, Prentice Hall, 1987.
- [Shen96] Q. Shen and A. Spanias, "Adaptive Active Sound Reduction, Noise Control Engineering Journal, J44 (6), pp. 281-293, Nov. 1996.
- [Span93] A.S. Spanias, "A Block Time and Frequency Modified Covariance Algorithms for Spectral Analysis," IEEE Transactions on Signal Processing, Vol. 41, no. 11, pp. 3138-3153, Nov. 1993.
- [Stra82] G. Strang, Linear Algebra and its Applications, Academic Press, 1982.
- [Widr75a] B. Widrow et al, "The Complex LMS Algorithm," Proc. IEEE, Vol. 63, pp. 719-720, 1975.
- [Widr75b] B. Widrow et al, "Adaptive Noise Cancelling: Principles and Applications," Proc. IEEE, Vol. 63, pp. 1692-1716, 1975.
- [Widr85] B. Widrow and S. Stearns, Adaptive Signal Processing, Prentice Hall, 1985.

July 2007 XII-76