Further Discussions and Beyond EE630

Electrical & Computer Engineering
University of Maryland, College Park

Acknowledgment: The ENEE630 slides here were made by Prof. Min Wu.
Contact: minwu@umd.edu

End of Semester Logistics

- Project due
- Final exam: two hours, close book/notes
  - Mainly cover Part-2 and Part-3
  - May involve basic multirate concepts from Part-1 (decimation, expansion, basic filter bank)
- Office hours

Higher-Order Signal Analysis: Brief Introduction

- Information contained in the power spectrum
  - Reflect the 2nd-order statistics of a signal (i.e. autocorrelation)
  - => Power spectrum is sufficient for complete statistical description of a Gaussian process, but not so for many other processes
- Motivation for higher-order statistics
  - Higher-order statistics contain additional info. to measure the deviation of a non-Gaussian process from normality
  - Help suppress Gaussian noise of unknown spectral characteristics.
    - *The higher-order spectra may become high SNR domains in which one can perform detection, parameter estimation, or signal reconstruction*
  - Help identify a nonlinear system or to detect and characterize nonlinearities in a time series

m^{th}–order Moments of A Random Variable

- Moments: $m_k = E[X^k]$;
- Central moments: subtract the mean $\gamma_k = E[(X - \mu_X)^k]$
- o Mean: $\mu_X = m_1 = E[X]$
  - Statistical centroid ("center of gravity")
- o Variance: $\sigma_X^2 = \gamma_2 = E[(X - \mu_X)^2]$
  - Describe the spread/dispersion of the p.d.f.
- o 3rd Moment: normalize into $K_3 = \gamma_3 / \sigma_X^3$
  - Represent Skewness of p.d.f. ➔ zero for symmetric p.d.f.
- o 4th Moment: normalize into $K_4 = \gamma_4 / \sigma_X^4 - 3$
  - "Kurtosis" for flat/peakiness deviation from Gaussian p.d.f. (which is zero)

See Manolakis Sec.3.1.2 for further discussions

78
CHAPTER 3
Random Variables, Vectors, and Sequences

First five cumulants for zero-mean r.v.

$$\kappa_x^{(1)} = r_1^{(x)} = \mu_x = 0$$
$$\kappa_x^{(2)} = \gamma_x^{(2)} = \sigma_x^2$$
$$\kappa_x^{(3)} = \gamma_x^{(3)}$$
$$\kappa_x^{(4)} = \gamma_x^{(4)} - 3\sigma_x^4$$
$$\kappa_x^{(5)} = \gamma_x^{(5)} - 10\gamma_x^{(3)}\sigma_x^2$$

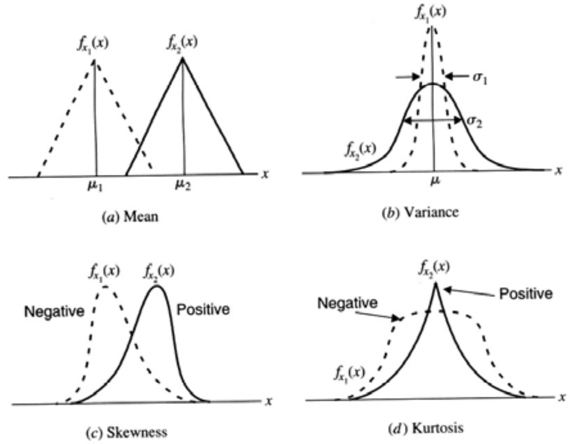


FIGURE 3.2
Illustration of mean, standard deviation, skewness, and kurtosis.

(Figures/Equations are from Manolakis Book Section 3.1; Note – moments of 3rd and above for Gaussian can be expressed in terms of μ and σ.)

Relations Among 3+ Samples of a Random Process

- Generalize from autocorrelation function between a pair of samples for a zero-mean stationary random process
- Triplets of samples: 3rd order cumulant
- Quadruplets of samples: 4th order cumulant

$$\kappa_x^{(1)} = E\{x(n)\} = \mu_x = 0 \quad (12.1.5)$$
$$\kappa_x^{(2)}(l_1) = E\{x^*(n)x(n+l_1)\} = r_x(l_1) \quad (12.1.6)$$
$$\kappa_x^{(3)}(l_1, l_2) = E\{x^*(n)x(n+l_1)x(n+l_2)\} \quad (12.1.7)$$
$$\kappa_x^{(4)}(l_1, l_2, l_3) = E\{x^*(n)x^*(n+l_1)x(n+l_2)x(n+l_3)\} - \kappa_x^{(2)}(l_2)\kappa_x^{(2)}(l_3-l_1) - \kappa_x^{(2)}(l_3)\kappa_x^{(2)}(l_2-l_1) \quad (12.1.8)$$

(complex-valued case)

$$\kappa_x^{(4)}(l_1, l_2, l_3) = E\{x(n)x(n+l_1)x(n+l_2)x(n+l_3)\} - \kappa_x^{(2)}(l_1)\kappa_x^{(2)}(l_3-l_2) - \kappa_x^{(2)}(l_2)\kappa_x^{(2)}(l_3-l_1) - \kappa_x^{(2)}(l_3)\kappa_x^{(2)}(l_2-l_1) \quad (12.1.9)$$

(real-valued case)

[Eq. from Manolakis Book Section 12.1]

High-order Spectra

- Multi-variable DTFT on cumulant functions
  - Bispectrum & Trispectrum: may exhibit patterns in magnitude & phase
- Extend properties under LTI to high-order stats

$$R_x^{(2)}(e^{j\omega}) \triangleq \sum_{l_1=-\infty}^{\infty} \kappa_x^{(2)}(l_1)e^{-j\omega l_1} = R_x(e^{j\omega}) \quad (12.1.15)$$

$$R_x^{(3)}(e^{j\omega_1}, e^{j\omega_2}) \triangleq \sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty} \kappa_x^{(3)}(l_1, l_2)e^{-j(\omega_1 l_1+\omega_2 l_2)} \quad \text{(bispectrum)} \quad (12.1.16)$$

and $$R_x^{(4)}(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) \triangleq \sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty}\sum_{l_3=-\infty}^{\infty} \kappa_x^{(4)}(l_1, l_2, l_3)e^{-j(\omega_1 l_1+\omega_2 l_2)} \quad \text{(trispectrum)} \quad (12.1.17)$$

See Manolakis et al. McGraw Hill book "Statistical & Adaptive S.P." Sec.12.1 High-order statistics for further discussions

[Eq. from Manolakis Book Section 12.1]

Resource on Signal Processing

- IEEE Signal Processing Magazine
  - E-copy on IEEE Xplore; Hard-copy by student membership
- IEEE "Inside Signal Processing eNewsletter"
  http://signalprocessingsociety.org/newsletter/
- Signal Processing related journals/transactions
- Related conferences: ICASSP, ICIP, etc.
- Additional 2-cents beyond courses
  - Attend talks/seminars to broaden your vision
  - Oral communications (oral exams, presentations, etc)

Related Courses Beyond EE630

- Adaptive and space-time signal processing: ENEE634*
- Image/video & audio/speech processing: ENEE631*, 632
- Detection/estimation & information theory: ENEE621*, 627*
  - *See also SP for digital communication in ENEE623*
- Pattern recognition and machine learning: ENEE633
- Special topic courses and seminars in signal processing:
  Occasionally offered.
  E.g. on info forensics & multimedia security, compressive sensing, etc.
- See also related applied math and statistics courses

a
b c

FIGURE 8.20 (a) The prediction error image resulting from Eq. (8.4-9). (b) Gray-level histogram of the original image. (c) Histogram of the prediction error.

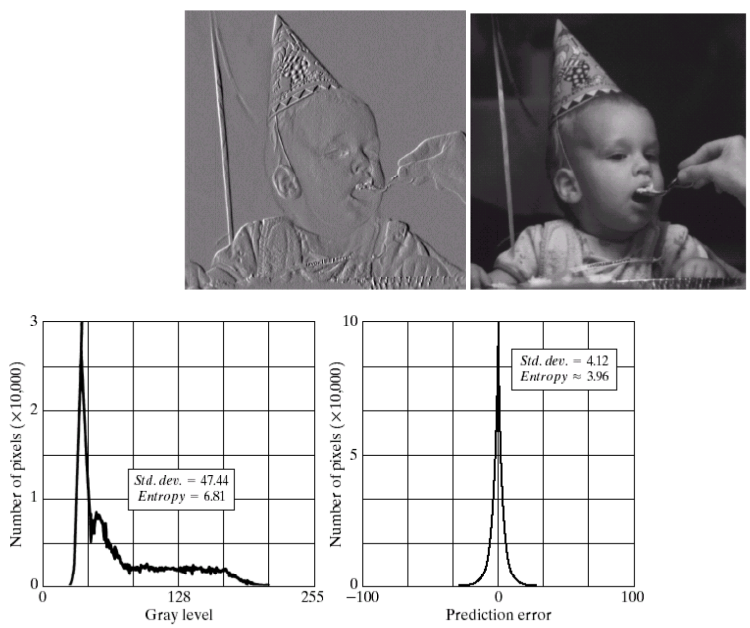


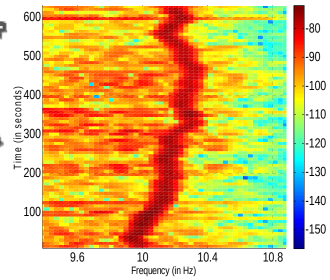
Figure is from slides at Gonzalez/ Woods DIP book website (Chapter 8). Use "previous pixel predictor". Difference image has mid-range gray representing zero and amplifying factor of 8.

Digital Image and Video Processing (ENEE631)

- Human visual perception; color vision
- Image enhancement
- Image restoration
- Image transform, quantization and coding
- Motion analysis and video coding
- Feature extraction and analysis
- Security and forensic issues

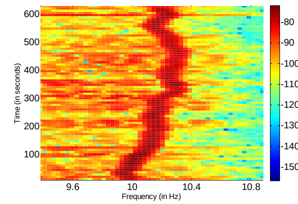
……

Forensic Question on "Time" and "Place"

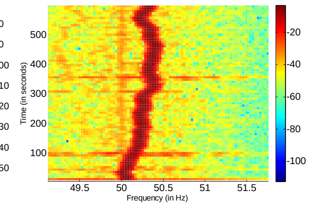


- When was the video actually shot? And where?
- Was the sound track captured at the same time as the picture? Or super-imposed afterward?
- Explore the fingerprint influenced by power grid onto sensor recordings

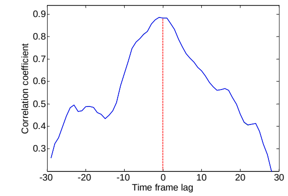
Ubiquitous Forensic Fingerprints from Power Grid



Video ENF signal



Power ENF signal



Normalized correlation

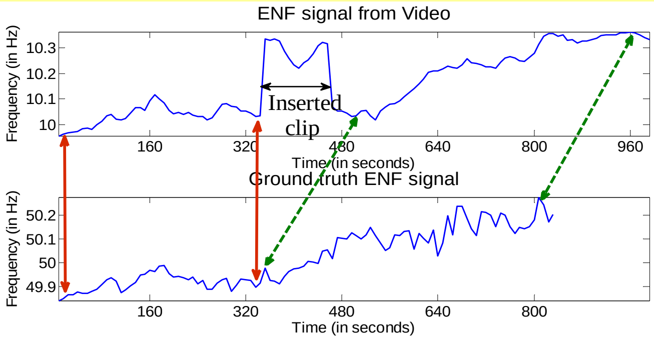
ENF matching result demonstrating similar variations in the ENF signal extracted from video and from power signal recorded in India

- Electric Network Frequency (ENF): 50/60 Hz nominal
  - Varies slightly over time; main trends consistent in same grid
  - Can be "seen" or "heard" in sensor recordings
- Help determine recording time, detect tampering, etc.
- Other potential applications on smart grid & media management

➔ Ref: Garg et al. **ACM Multimedia 2011, CCS 2012 and APSIPA 2012**

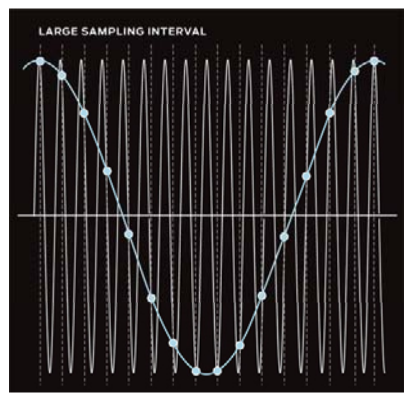
Tampering Detection Using ENF

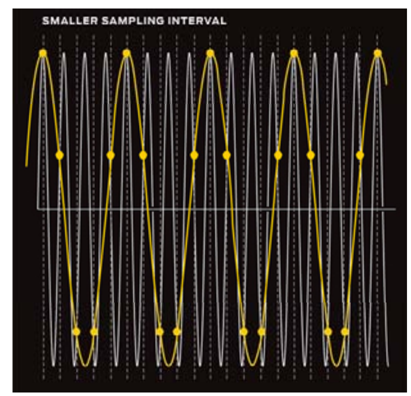
ENF matching result demonstrating the detection of video tampering based on the ENF traces



- Adding a clip between the original video leads to discontinuity in the ENF signal extracted from video
- Clip insertion can also be detected by comparing the video ENF signal with the power ENF signal at corresponding time

Aliasing Revisit: Downsample A Sinusoid





"If the RF signal [white] is not sampled at least twice per cycle, aliasing will occur. But by properly adjusting the sampling interval [indicated by vertical lines], you can down-convert the RF to whatever lower frequency is desired [blue and yellow]."

IEEE Spectrum Magazine April 2009 "Universal Handset" – Alias Harnessed for software-defined radio
http://spectrum.ieee.org/computing/embedded-systems/the-universal-handset/0/cellsb01

ENEE630 Look Ahead

Introduction to Adaptive Filtering

Electrical & Computer Engineering
University of Maryland, College Park

Acknowledgment: the additional overview/introductory slides for beyond ENEE630 were made by Prof. Min Wu and FFP Teaching Fellow Mr. Wei-Hong Chuang, with reference to textbooks by Hayes and Haykins and ENEE634 class notes by Prof. Ray Liu. *Contact: minwu@umd.edu*

Stationarity Assumption in Wiener Filtering

- Wiener filtering is optimum in a stationary environment
  - Unfortunately, most real signals are non-stationary
- One remedy: process the non-stationary signal in blocks, where the signal is assumed to be stationary
- Not always effective
  - For rapidly varying signals, the block length may be too small to estimate relevant parameters
  - Can't accommodate step changes within analysis intervals
  - Solution imposes incorrect data model, i.e., piecewise stationary

=> Try to begin with non-stationarity to develop solutions

Recursive Update of Filter Coefficients

- Wiener Filtering: solve the normal equation

$$\mathbf{R}_x \mathbf{w} = \mathbf{r}_{dx}$$

- If non-stationary, optimal filter coefficients will depend on time n

$$\mathbf{R}_x(n)\mathbf{w}_n = \mathbf{r}_{dx}(n)$$

  - Not always feasible (e.g. high computational complexity)

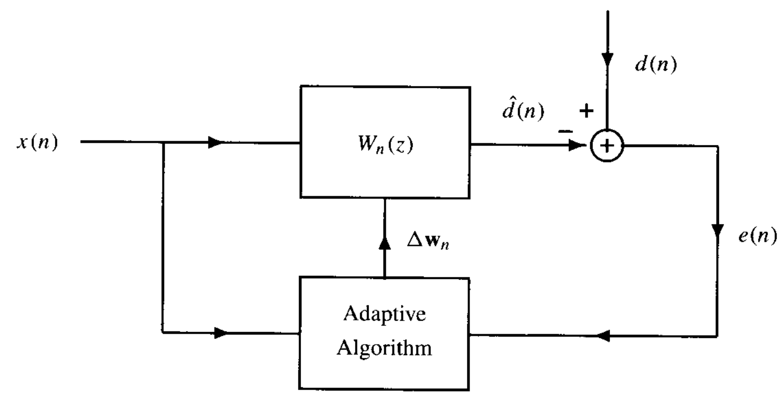
- Can be much simplified with adaptive filtering:

=> Form $\mathbf{w}_{n+1}$ by adding correction $\Delta\mathbf{w}_n$ to $\mathbf{w}_n$ at each iteration

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}_n$$

General Structure of Adaptive Filtering

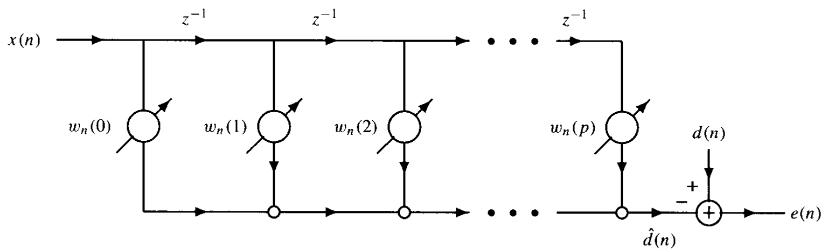
(Fig. from Hayes' book p495)



- Measure the error $e(n)$ at each time n, determine how to update filter coefficients accordingly

(Fig. from Hayes' book Chapter 9)

FIR Adaptive Filter



- Simple & efficient algorithms for coefficient adjustment
- Often perform well enough
- Stability is easily controlled
- Feasible performance analysis

Coefficient Update: Desired Properties

- Corrections should reduce mean-square error

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

- In a stationary environment, $\mathbf{w}_n$ should converge to Wiener-Hopf solution

$$\lim_{n\to\infty} \mathbf{w}_n = \mathbf{R}_x^{-1}\mathbf{r}_{dx}$$

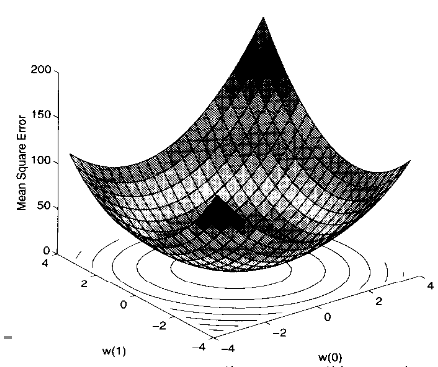
- Avoid explicit signal statistics for $\Delta\mathbf{w}_n$ if possible
  - "Built-in" estimation of statistics
- If non-stationary, filter should track the solution

(Fig. from Hayes' book Chapter 9)

Steepest-Descent Adaptive Filter

- Recall direct approach: minimizing the MSE by setting partial derivative = 0 (this may involve matrix inverse)

=> Alternative: search solution iteratively using a numerical method of steepest descent



- Find the filter coefficients that minimize the error on the error surface
- At every iteration, moves along the direction of the steepest descent of error

(Fig. from Hayes' book Chapter 9)

Method of Steepest Descent

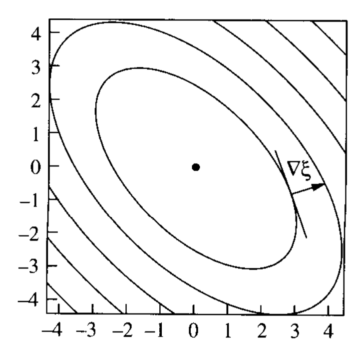
- The steepest direction is given by gradient

$$\nabla\xi(n) = \begin{bmatrix} \dfrac{\partial \xi(n)}{\partial w(0)} \\ \dfrac{\partial \xi(n)}{\partial w(1)} \end{bmatrix}$$

- Update Equation

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla\xi(n)$$

  - μ: step size, controls the rate at which the coefficients move



Method of Steepest Descent

1. **Initialize with initial estimate $\mathbf{w}_0$**
2. **Evaluate gradient $\xi(n)$ at time n**
3. **Update estimate at time n by taking a step of size μ in negative gradient direction**

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla \xi(n)$$

4. **Go back to 2.**

Stability of Steepest Descent

- It can be shown that

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\left\{e(n)\mathbf{x}^*(n)\right\}$$

- If x(n) and d(n) are w.s.s.,

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\left(\mathbf{r}_{dx} - \mathbf{R}_x \mathbf{w}_n\right)$$

  – Correction term is zero if $\mathbf{w}_n = \mathbf{R}_x^{-1}\mathbf{r}_{dx}$ (i.e., a fixed point of the update)

- Does the coefficient update converge?

Convergence of Steepest Descent

- For w.s.s. d(n) and x(n), the steepest descent adaptive filter converges to Wiener-Hopf solution

$$\lim_{n\to\infty} \mathbf{w}_n = \mathbf{R}_x^{-1}\mathbf{r}_{dx}$$

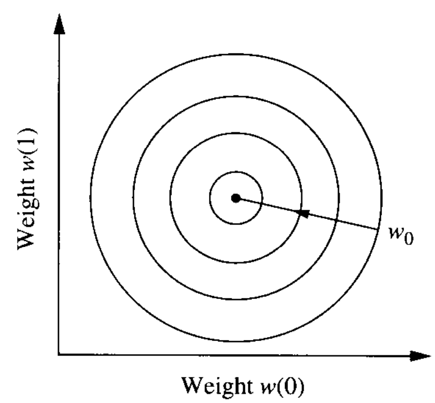
  if the step size satisfies $0 < \mu < \frac{2}{\lambda_{\max}}$

  – λ_{max}: maximal eigenvalue of $\mathbf{R}_x$
  – Can be shown by diagonalizing $\mathbf{R}_x$

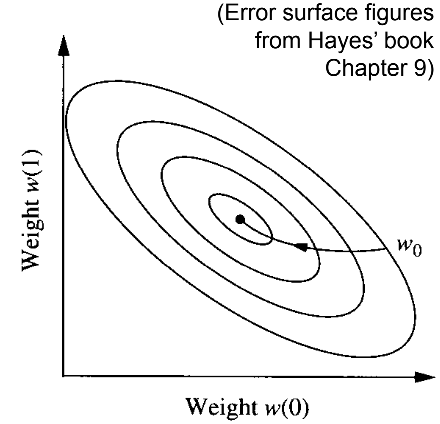
- Convergence rate (how fast the update converges) is determined by the spread of eigenvalues

Effects of Condition Number on Convergence

(Error surface figures from Hayes' book Chapter 9)



small eigenvalue spread



large eigenvalue spread

Least Mean Squares (LMS) Algotithm

- Recall the update in steepest descent:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\left\{e(n)\mathbf{x}^*(n)\right\}$$

  - Practical challenge: the expectation may be unknown or difficult to estimate on the fly

- The LMS replaces the expectation by an "one-shot" estimate

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n)\mathbf{x}^*(n)$$

  - Very crude estimate, but often performs well in practice

LMS Algorithm for pth-order FIR Adaptive Filter

Parameters: p = Filter order
μ = Step size

Initialization: $\mathbf{w}_0 = \mathbf{0}$

Computation: For $n = 0, 1, 2, \ldots$

(a) $y(n) = \mathbf{w}_n^T \mathbf{x}(n)$

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

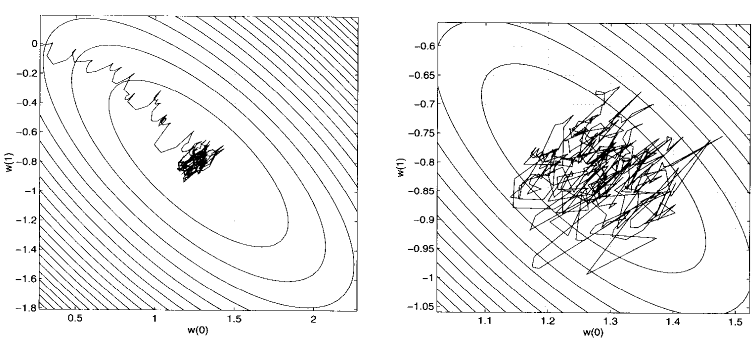
(c) $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n)\mathbf{x}^*(n)$

(Algorithm from Hayes' book Chapter 9)

Randomness of the LMS Algorithm

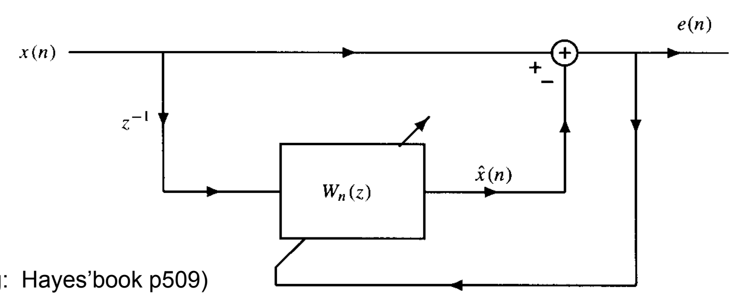
(Fig. from Hayes' book Chapter 9)

- The one-shot estimate approximates the steepest descent direction (i.e., the statistical average)
- The one-shot nature makes $\mathbf{w}_n$ move randomly in a neighborhood, even if initialized from the Wiener solution



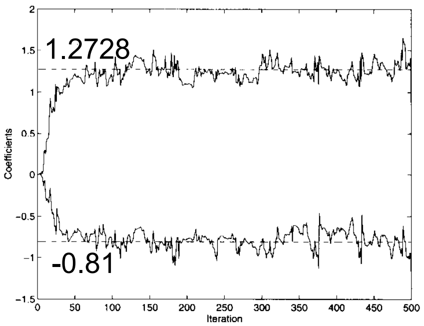
Example: Adaptive Linear Prediction

- x(n) = 1.2728 x(n-1) - 0.81 x(n-2) + v(n)
  - v(n): unit-variance white noise
  - Optimum linear predictor: $\hat{x}$(n) = 1.2728 x(n-1) - 0.81 x(n-2)
- Linear Prediction using LMS

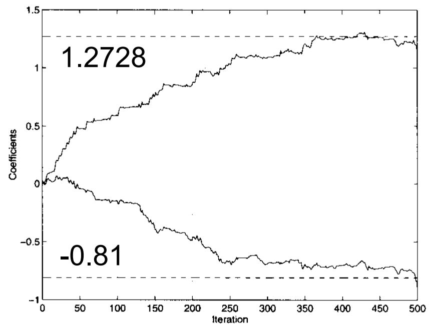


(Fig: Hayes'book p509)

Example: Adaptive Linear Prediction



μ=0.02: faster convergence, less stable



μ=0.004: slower convergence, more stable

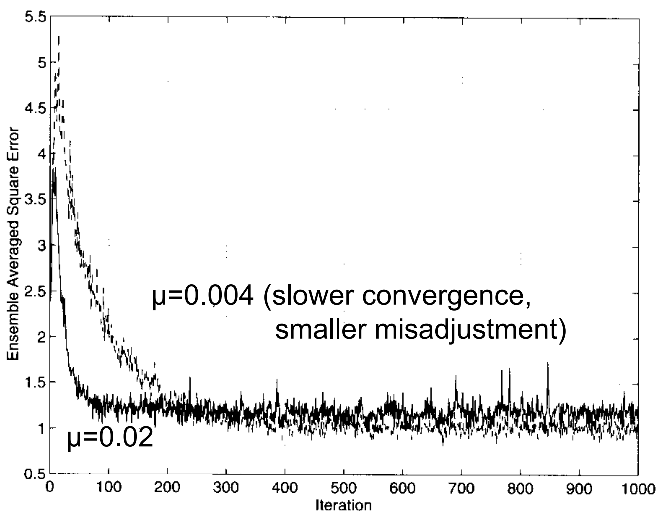
Convergence of the LMS Algorithm

- Examine convergence properties of LMS under a statistical framework
- For w.s.s. d(n) and x(n) the LMS adaptive filter converges in the mean sense if

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

- More stringent condition is required for convergence in mean square sense

Typical Learning Curves: MSE vs. time



Recursive Least Squares (RLS) Algorithm

- Mean-square error v.s. least squares error
  - Mean-square error does not depend on incoming data, but their ensemble statistics
  - Least squares error depends explicitly on x(n) and d(n)
- RLS: minimizes least squares error, where old data are gradually "forgotten"

$$\mathcal{E}(n) = \sum_{i=0}^{n} \lambda^{n-i} |e(i)|^2$$

$0 < \lambda < 1$: "forgetting" factor

Recursive Least Squares (RLS) Algorithm

- Least squares normal equation

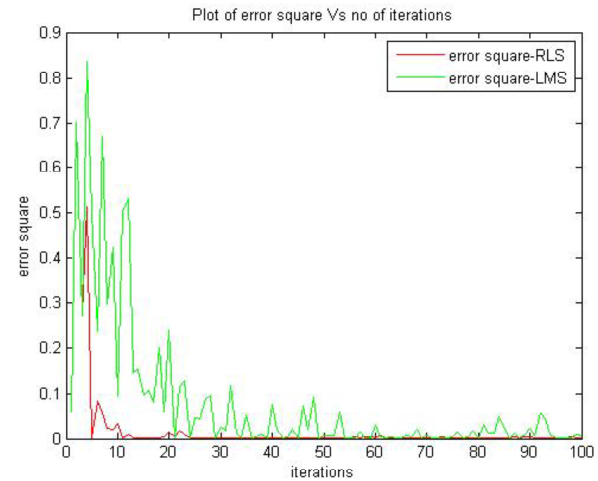
$$\mathbf{R}_x(n)\mathbf{w}_n = \mathbf{r}_{dx}(n)$$

$$\mathbf{R}_x(n) = \sum_{i=0}^{n} \lambda^{n-i}\mathbf{x}^*(i)\mathbf{x}^T(i)$$

$$\mathbf{r}_{dx}(n) = \sum_{i=0}^{n} \lambda^{n-i}d(i)\mathbf{x}^*(i)$$

- $\mathbf{R}_x(n)$ and $\mathbf{r}_{dx}(n)$ can be calculated recursively
- $\mathbf{R}_x^{-1}(n)$ can also be calculated recursively using Matrix Inversion Formula

Learning Rates of RLS and LMS



http://www.mathworks.com/matlabcentral/fileexchange/32498-performance-of-rls-and-lms-in-system-identification

Characteristics of RLS Algorithm

- Convergence rate is an order of magnitude faster than that of LMS, at the cost of higher complexity
- Convergence rate is insensitive to eigenvalue spread
- In theory, RLS produces zero excess error or misadjustment
- RLS can be understood under the unifying framework of Kalman filtering

Summary

- Adaptive filters
  - Address non-stationary signal processing
  - Low computational complexity
  - Recursive update of filter coefficients
- Method of steepest descent
  - Moves in the negative gradient direction
  - converges to Wiener-Hopf if stationary
- LMS algorithm
  - Crude one-shot gradient estimation; reasonable practical performance
- RLS algorithm: recursively minimizes least-squares error

References for Further Explorations

- M. Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, 1996. Chapter 9
  - All figures except one used in this lecture are from the book
- S. Haykin, *Adaptive Filter Theory*, 4th edition, Prentice-Hall, 2002, Chapters 4 & 5

=> See more detailed development in ENEE634 (offered in alternating spring semester)