Adaptive Filtering – An Introduction

José C. M. Bermudez

Department of Electrical Engineering
Federal University of Santa Catarina
Florianópolis – SC
Brazil

ENSEEIHT - Toulouse
May 2009

Regular Linear Digital Filters

- FIR or IIR.
- Most useful for frequency selective applications.
- Desired signal and interference are in different frequency bands.

  Example: To extract high frequency noise (> 20 kHz) from speech.

FIR $\times$ IIR

- FIR Filters
  - Simple to design
  - Can implement linear phase
  - Are always stable
  - Require $\gg$ coefficients than IIR filters
- IIR Filters
  - Harder to design than FIR filters
  - Introduce phase distortion
  - May become unstable
  - Require $\ll$ coefficients than FIR filters

Statistical Linear Filtering

- Many applications $\rightarrow$ more than just frequency band selection
- Signal and interference are frequently within the same freq. band
- Solution is based on the statistical properties of the signals

Basic Operations in Linear Filtering

1. Filtering

   Extraction of information at a given time n by using data measured before and at time n.

2. Smoothing

   Differs from filtering in that data measured *both before and after* n can be used to extract information at time n.

3. Prediction

   Predicts how the quantity of interest will be at time $n + N$ for some $N > 0$ by using data measured up to and including n.

(The filter is linear if its output is a linear function of the observations applied to the input.)

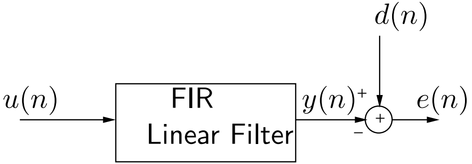
How Do We Do Linear Statistical Filtering?

- We assume some statistical properties of the useful and unwanted signals are available

  Mean Autocorrelation Cross-correlations etc.

- We define some statistical criterion for the optimization of the filter performance

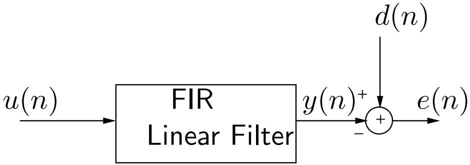
Optimal Linear Filtering
The Wiener Filter - The FIR Case



- $u(n)$ and $d(n)$ stationary and zero-mean
- $d(n)$: information plus some interference
- $y(n)$: estimate of $d(n)$ given some observations of $u(n)$

$$y(n) = \sum w_k u(n-k) = \hat{d}(n|\mathcal{U}_n)$$

- $e(n)$: estimation error



- Performance index: Mean-Square Error

$$\boldsymbol{w} = [w_1, \ldots, w_N]^T \quad \boldsymbol{u}(n) = [u(n), \ldots, u(n-N+1)]^T$$

$$y(n) = \boldsymbol{u}^T(n)\boldsymbol{w}$$

$$J(\boldsymbol{w}) = E\{e^2(n)\}$$ Mean-Square Error

The Wiener Weights

$$e(n) = d(n) - \boldsymbol{u}^T(n)\boldsymbol{w}$$

$$e^2(n) = d^2(n) - 2d(n)\boldsymbol{u}^T(n)\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{u}(n)\boldsymbol{u}^T(n)\boldsymbol{w}$$

$$J(\boldsymbol{w}) = \sigma_d^2(n) - 2\boldsymbol{p}^T(n)\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{R}(n)\boldsymbol{w}$$

where:

$$\boldsymbol{p}(n) = E[d(n)\boldsymbol{u}(n)] \quad : \text{ cross-corr. between } d(n) \text{ and } \boldsymbol{u}(n)$$

$$\boldsymbol{R}(n) = E[\boldsymbol{u}(n)\boldsymbol{u}^T(n)] \quad : \text{ autocorrelation matrix of } \boldsymbol{u}(n)$$

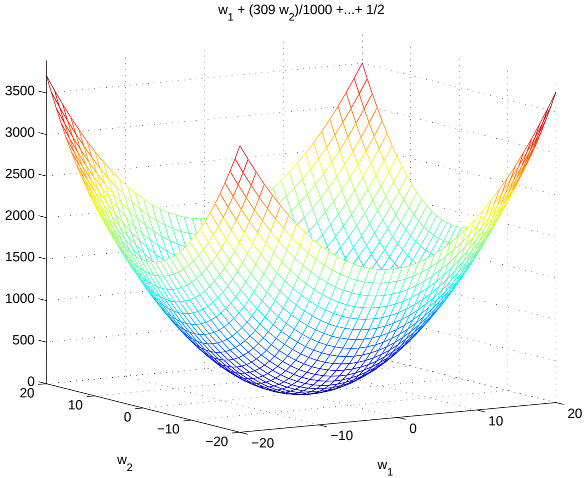
$$\sigma_d^2 = E[d^2(n)]$$

Optimum Weight Vector

$$\boldsymbol{w} = \boldsymbol{w}_o \quad \text{such that} \quad \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{0}$$

$$-2\boldsymbol{p} + 2\boldsymbol{R}\boldsymbol{w}_o = \boldsymbol{0}$$

$$\boldsymbol{R}\boldsymbol{w}_o = \boldsymbol{p} \quad \text{(Normal Equations)}$$



- $\boldsymbol{w}_o$: Minimum of this surface
- Solution of the Normal Equations

  Computationally intensive even with efficient algorithms

What is an Adaptive Filter Anyway?

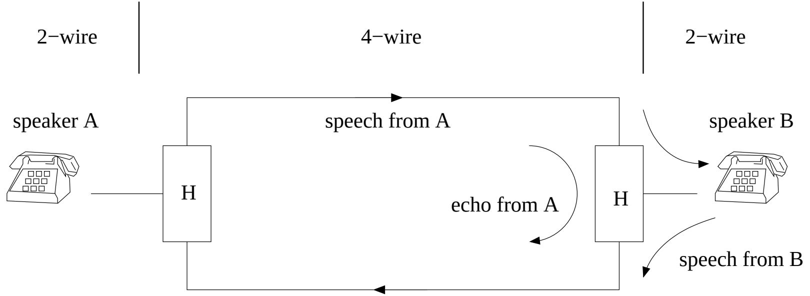
In several practical real-time applications:

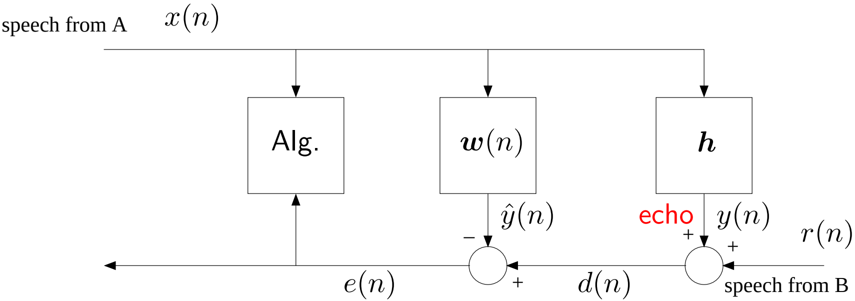
- The signal involved are nonstationary
  - Normal equations must be solved for each n!?!
  - The algorithms used for the stationary case become inefficient
  - What about Kalman Filters?
    - Requires a dynamics model (state-space) for $d(n)$
    - Computationally heavy for real-time
- Signal statistics may be unknown, and there may be no time to estimate them
- Computational complexity between 2 input samples limited by processor speed and by cost

Adaptive Filters

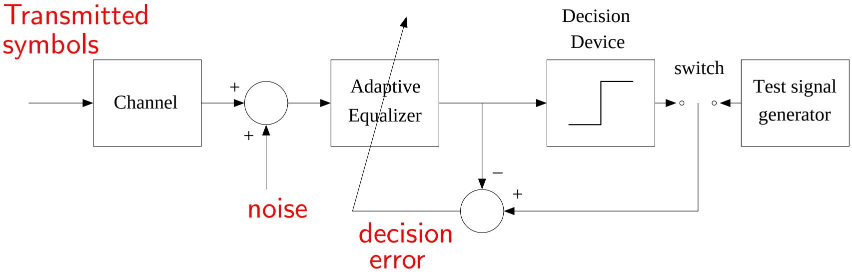
- Change their weights as new input samples arrive
- Weight updating is controlled by an adaptive algorithm
- Optimal solution is approached by improving performance a little bit at each iteration
- Optimal solution is approximated after several iterations (iteration complexity $\times$ convergence time)
- Filter weights become random variables that converge to a region about the optimum weights
- Can track signal and system nonstationarities

Application – Network Echo Cancellation

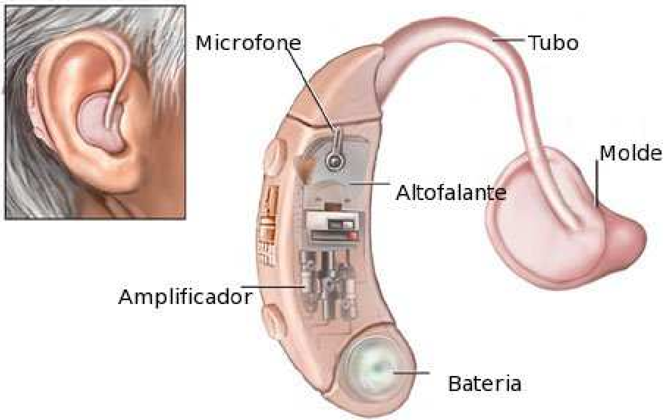




Application – Channel Equalization



Application – Feedback Cancellation in Hearing Aids



Application – Feedback Cancellation in Hearing Aids

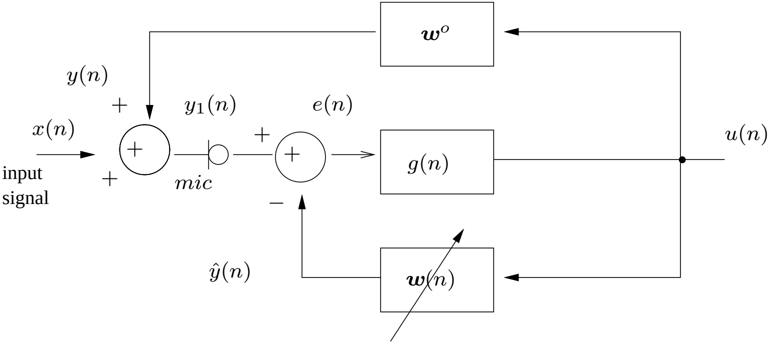


Figure: Basic adaptive feedback cancellation in hearing aids

Adaptive Algorithm - Least Mean Squares (LMS)

Optimization Problem

Determine $\boldsymbol{w}$ that minimizes

$$J(\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}(n)} = \sigma_d^2(n) - 2\boldsymbol{p}^T(n)\boldsymbol{w}(n) + \boldsymbol{w}(n)^T\boldsymbol{R}(n)\boldsymbol{w}(n)$$

Steepest Descent Algorithm

Direction contrary to the gradient $\dfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}$

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2\boldsymbol{p} + 2\boldsymbol{R}\boldsymbol{w}(n)$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu[\boldsymbol{p} - \boldsymbol{R}\boldsymbol{w}(n)]$$

(μ controls de adaptation speed)

Adaptive Algorithm - Least Mean Squares (LMS)

LMS Algorithm

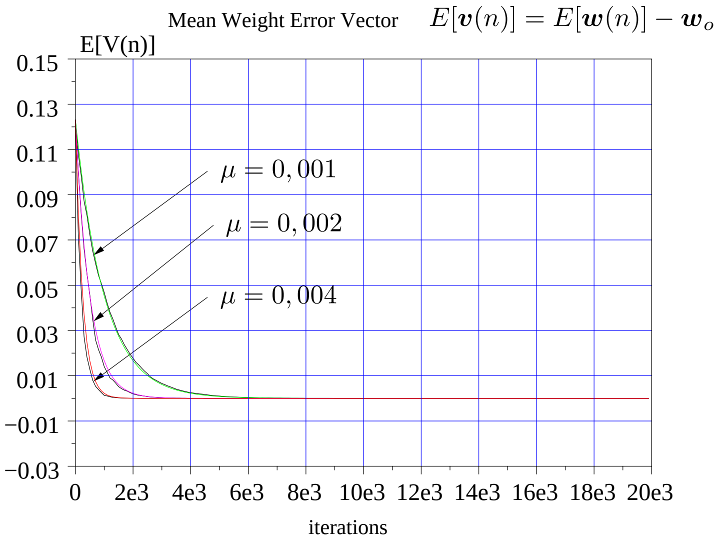
- Steepest descent needs $\boldsymbol{p}$ and $\boldsymbol{R}$!?
- Instantaneous approximation of the gradient

$$\frac{\partial E[e^2(n)]}{\partial \boldsymbol{w}(n)} \approx \frac{\partial e^2(n)}{\partial \boldsymbol{w}(n)} = -2e(n)\boldsymbol{u}(n)$$

- LMS weight updating equation

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu e(n)\boldsymbol{u}(n)$$

LMS – Typical Convergence Behavior



LMS – Typical Convergence Behavior