## Outline

- Adaptive FIR Filters: General Form
- LMS Algorithm
- Recursive Solutions to LMS Problem
- Properties of LMS Algorithm

## Adaptive FIR Filters: General Form

- All of the applications of adaptive FIR filters result in the general form of equations

$$\sum_{k=0}^{M-1} h[k]R_x[l-k] = R_{dx}[l+D] \qquad l = 0, 1, \ldots, M-1$$

- The correlation sequences are not known apriori, so they are estimated from the measured data
- Hence, longer data results in better estimates of correlations, and better estimates of filter coefficients

## Adaptive FIR Filters: General Form (Cont.)

- We also have time-varying random signals now, that is the filter coefficients will be also time-varying
- Time-varying coefficients can be obtained either by sample by sample update, or block by block update
- We consider sample by sample updates in this course
- Sample by sample updates can be obtained in two ways
  - Least mean square algorithm (LMS)
  - Recursive least squares algorithm (RLS)

# LMS Algorithm

- Have a look at the general form of adaptive filters

$$\sum_{k=0}^{M-1} h[k]R_x[l-k] = R_{dx}[l+D] \qquad l = 0, 1, \ldots, M-1$$

- This has the same form as in Wiener filters (except the correlations are not known, but estimated. Hence the solution is

$$\boldsymbol{h}_{\mathrm{opt}} = \Gamma_M^{-1}\boldsymbol{r}_d$$

where $\Gamma_M$ is the matrix with elements from $R_{xx}$ and $\boldsymbol{r}_d$ is the vector elements from $R_{dx}$

## LMS Algorithm (Cont.)

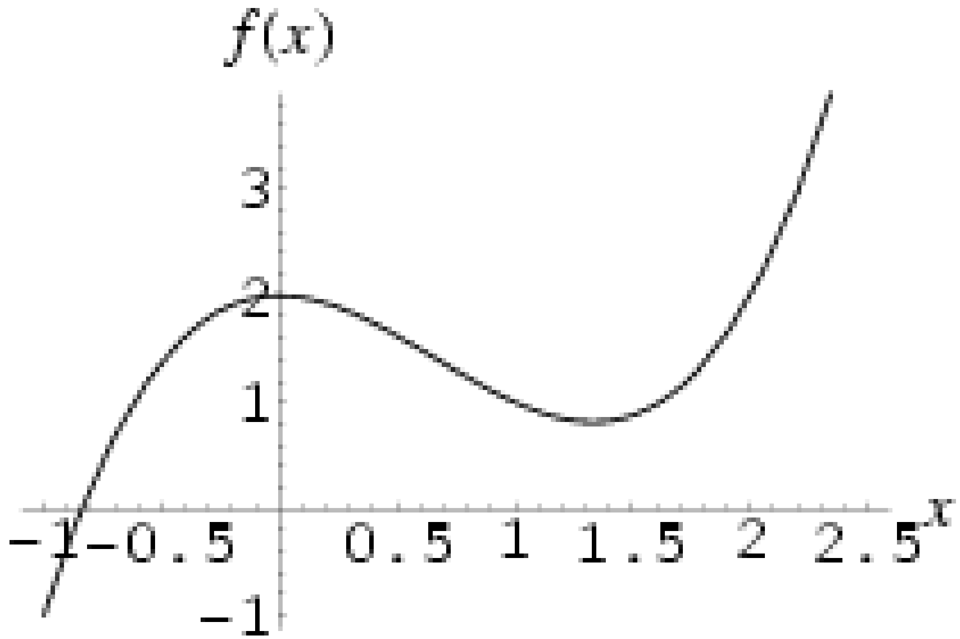
- We also know that the minimum MSE is

$$\mathcal{E}_{\mathrm{M,min}} = \sigma_d^2 - \boldsymbol{r}_d^{\mathrm{H}} \Gamma_M^{-1} \boldsymbol{r}_d$$

- These optimum coefficients can be obtained by matrix inversion or some sort of fast algorithm such as Levinson-Durbin
- But these require that estimates of correlations are obtained beforehand
- Another method is to obtain the solutions iteratively using a gradient search

# Steepest Descent

- A general method to find the minimum of a cost function
- Assume that we want to minimize a cost function $C(x)$
- A local minimum can be found by going into the direction of negative gradient



# LMS Algorithm: Steepest Descent

- Recursive calculation of the optimum filter coefficients are in the form

$$\boldsymbol{h}_M[n+1] = \boldsymbol{h}_M[n] + \frac{1}{2}\Delta[n]S[n]$$

- Here, $S[n]$ is a direction vector, that will take us closer to the solution and $\Delta_n$ is a step size towards that direction
- A simple method of finding minimum is to use the negative of the gradient as the search direction

$$\begin{aligned} S[n] &= -g[n] = -\tfrac{\mathrm{d}\mathcal{E}_M[n]}{\mathrm{d}\boldsymbol{h}_M[n]} \\ &= -2[\Gamma_M \boldsymbol{h}_M[n] - \boldsymbol{r}_d] \end{aligned}$$

- Then using this search direction, the update equation becomes

$$\begin{aligned} \boldsymbol{h}_M[n+1] &= \boldsymbol{h}_M[n] - \Delta[n][\Gamma_M \boldsymbol{h}_M[n] - \boldsymbol{r}_d] \\ &= \boldsymbol{h}_M[n]\{I - \Delta[n]\Gamma_M]\} + \Delta[n]\boldsymbol{r}_d \end{aligned}$$

## LMS Algorithm: Other Methods

- Steepest descent method is slow to converge
- So other gradient-based (but more complex than steepest descent) might be employed
- Such methods could include e.g. conjugate gradient algorithm

## LMS Algorithm: Estimation of Correlations

- The steepest descent update equation assumes that $\Gamma_M$ and $\boldsymbol{r}_d$ are known
- We need to estimate them if they are not known
- Let us have a look at the gradient expression

$$g[n] = 2[\Gamma_M \boldsymbol{h}_M[n] - \boldsymbol{r}_d]$$

- Considering the definitions of $\Gamma_M$ and $\boldsymbol{r}$ we have

$$g[n] = -2\mathrm{E}\{e[n]X_M^*[n]\}$$

where $X_M[n]$ is the vector with elements $x[n-l]$

## LMS Algorithm: Estimation of Correlations

- Although we do not know this expression exactly ($\Gamma_M$ and $\boldsymbol{r}_d$ uknown) we can obtain an unbised estimate of it

$$\hat{g}[n] = -2e[n]X_M^*[n]$$

- Substituting this expression into the steepest descent we obtain

$$\boldsymbol{h}_M[n+1] = \boldsymbol{h}_M[n] + \Delta[n]e[n]X_M^*[n]$$

- If we use a fixed step size we obtain

$$\boldsymbol{h}_M[n+1] = \boldsymbol{h}_M[n] + \Delta e[n]X_M^*[n]$$

# LMS Algorithm: Variations

- If we use more than one sample to change the coefficients, we can obtain a better estimate of the gradient

$$\bar{\hat{g}}[nK] = -\frac{2}{K}\sum_{k=0}^{K-1} e[nK+k]x_M^*[nK+k]$$

- The filter coefficients are updated every $K$th iteration

$$\boldsymbol{h}_M[(n+1)K] = \boldsymbol{h}_M[nK] - \frac{1}{2}\Delta\bar{\hat{g}}[nK]$$

- Since K samples are used, the noise is reduced
- Another method to reduce noise in the estimate of the gradient is to use a low pass filter before using the gradient in the update equation
- Variable step size can be used in case the data has a wide dynamic range

$$\boldsymbol{h}_M[n+1] = \boldsymbol{h}_M[n] + \frac{\Delta}{||X_M[n]||^2} e[n]X_M^*[n]$$

## Properties of the LMS Algorithm

- Convergence and stability
- Noise properties

## Properties of the LMS Algorithm: Convergence

- The expected value of the update equation gives

$$\mathrm{E}\{\boldsymbol{h}_M[n]\} = \{I - \Delta\Gamma_m\}\mathrm{E}\{\boldsymbol{h}_M[n]\} + \Delta\boldsymbol{r}_d$$

- In block form we have

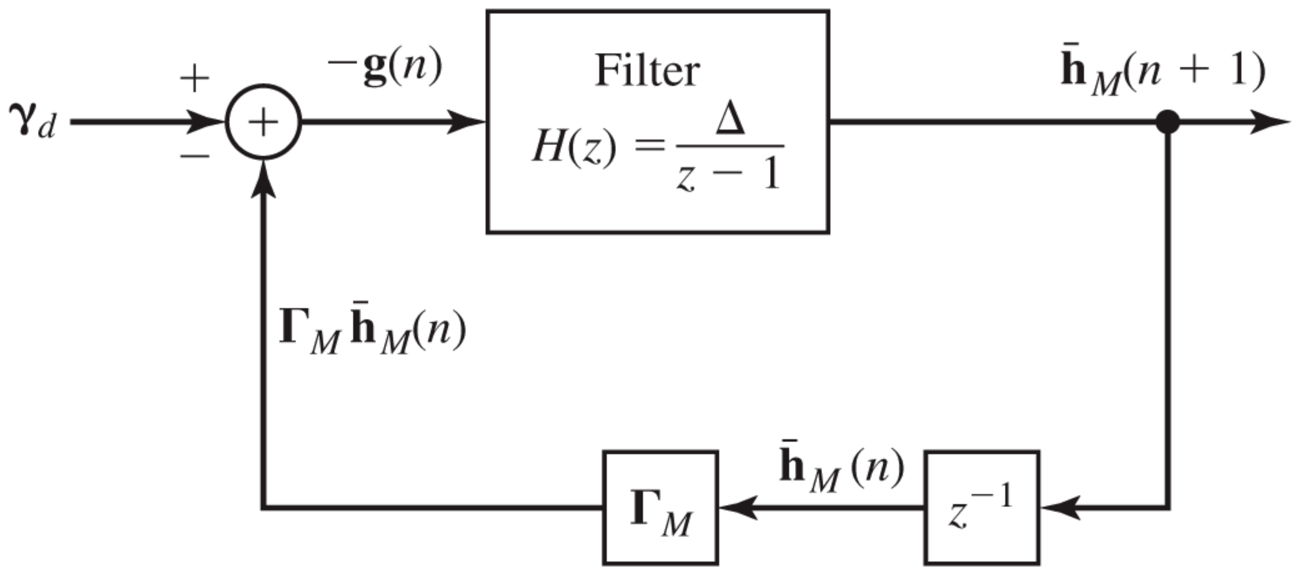



**Figure 13.2.1** Closed-loop control system representation of recursive Equation (13.2.34).

# Properties of the LMS Algorithm: Convergence and Stability

- Let us decouple these difference equations by using diagonalization

$$\Gamma_M = U \Lambda U^{\mathrm{H}}$$

- Substituting we have

$$\boldsymbol{h}_M^o[n+1] = (I - \Delta\Lambda)\boldsymbol{h}_M^o[n]$$

- The stability of the solution to these equations are obviously determined by the step size $\Delta$
- Since equations are now decoupled let us have a look at one of them

$$h^o[k,n] = c(1 - \Delta\lambda_k)^n u[n]$$

## Properties of the LMS Algorithm: Convergence and Stability (Cont.)

- We should have

$$|1 - \lambda_k| < 1$$

or

$$0 < \Delta < \frac{2}{\lambda_k}$$

- Generalizing to all $k$'s we should have

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

# Properties of the LMS Algorithm: Noise Properties

- The error in adaptive filters are larger than the MSE in general because we do not have access to true correlation functions, but just use estimates
- Total MSE is

$$\begin{aligned}\mathcal{E}_T[n] &= \mathcal{E}_{\mathrm{M,min}} + \mathcal{E}_{\mathrm{M,excess}} \\ &= \mathcal{E}_{M,min} + (\boldsymbol{h}_M[n] - \boldsymbol{h}_{\mathrm{opt}})^{\mathrm{T}} \Gamma_M (\boldsymbol{h}_M[n] - \boldsymbol{h}_{\mathrm{opt}})^*\end{aligned}$$

- Under certain approximations, the expression for the excess noise can be calculated as

$$\mathcal{E}_{\mathrm{M,excess}} = \Delta^2 \mathcal{E}_{\mathrm{M,min}} \sum_{k=0}^{M-1} \frac{\lambda_k^2}{1 - (1 - \Delta\lambda_k)^2}$$

- Assuming that we select a small step size so that $\Delta\lambda_k << 1$ we have

$$\mathcal{E}_{\mathrm{M,excess}} = \frac{1}{2} \Delta \mathrm{E}_{\mathrm{M,min}} \sum_{k=0}^{M-1} \lambda_k$$