Outline

- Adaptive FIR Filters: Review
- LMS Algorithm: Review
- RLS Algorithm
- LDU Factorization
- Fast RLS Algorithms
- Properties of RLS Algorithms
- Power Spectrum Estimation: Introduction

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

LMS Algorithm: Review

- In LMS algorithm, we find the optimum filter coefficients using gradient based methods
- Gradient based methods are a general class of methods of finding the minimum of a cost function
- There is only one parameter that we can choose the step size Δ
- There are limits on the step size due to stability requirements
- The convergence can be very slow depending on the eigenvalues of Γ

RLS Algorithm

- Therefore we can develop more complicated methods than LMS that use more number of parameters that we can select
- In the RLS algorithm, we have as many number of parameters as the eigenvalues of Γ
- Algorithm is more complicated (we have to determine M parameters instead of one), but more flexible resulting in faster convergence

RLS Algorithm (Cont.)

- Let us define the following vectors

$$\boldsymbol{h}_M[n] = \begin{bmatrix} h[0,n] \\ h[1,n] \\ . \\ . \\ . \\ h[M-1,n] \end{bmatrix} \qquad \boldsymbol{x}_M[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ . \\ . \\ . \\ x[n-M+1] \end{bmatrix}$$

- Now the problem of finding the optimum filter coefficients is given $\boldsymbol{x}$ what is $\boldsymbol{h}$ that minimizes the weighted least squares error

$$\begin{aligned} \mathcal{E}_M &= \sum_{l=0}^{n} w^{n-l}|d[l] - \hat{d}[l,n]|^2 \\ &= \sum_{l=0}^{n} w^{n-l}|d[l] - \boldsymbol{h}_M^{\mathrm{T}}\boldsymbol{x}_M[l]|^2 \end{aligned}$$

  This is a generalized version of MSE with a weighting factor w

- We can use the weighting filter to put more weight on recent samples

RLS Algorithm (Cont.)

- Minimizing the weighted least squares error results in the set of equations

$$\begin{aligned} R_M[n]\boldsymbol{h}_M[n] &= D_M[n] \\ \sum_{l=0}^{n} w^{n-l}\boldsymbol{x}_M^*[l]\boldsymbol{x}_M^{\mathrm{T}}[l]\boldsymbol{h}_M[n] &= \sum_{l=0}^{n} w^{n-l}\boldsymbol{x}_M^*[l]d[l] \end{aligned}$$

- We can solve this equations for the unknown filter coefficients

$$\boldsymbol{h}_M[n] = R_M^{-1}[n]D_M[n]$$

RLS Algorithm (Cont.)

- Now, R_M is some sort of estimate of the autocorrelation, but not exactly. Hence, it is not Toeplitz (we can use available fast algorithms such as Levinson-Durbin)
- Similarly D_M is related to the estimate of R_{dx}
- Instead of solving this inversion equation from the scratch each time a new sample arrives, we will try to solve it iteratively
- As each sample arrives, the solution will be based on the previous solution

RLS Algorithm (Cont.)

- For recursive inversion of R_M we will make use of a matrix identity

$$R_M^{-1}[n] = \frac{1}{w}\left[R_M^{-1}[n-1] - \frac{R_M^{-1}[n-1]\boldsymbol{x}_M^*[n]\boldsymbol{x}_M^{\mathrm{T}}[n]R_M^{-1}[n-1]}{w + \boldsymbol{x}^{\mathrm{T}}R_M^{-1}[n-1]\boldsymbol{x}_M^*[n]}\right]$$

- Let us simplify notation by a few definitions

$$P_m[n] = \frac{1}{w}\left[P_M[n-1] - K_M[n]\boldsymbol{x}^{\mathrm{T}}[n]P_M[n-1]\right]$$

where

$$K_M[n] = \frac{1}{w + \boldsymbol{x}^{\mathrm{T}}P_M[n-1]\boldsymbol{x}_M^*[n]}P_M[n-1]\boldsymbol{x}_M^*[n]$$

and

$$P_M[n] = R_M^{-1}[n]$$

- Now, let us turn back to calculating the matrix inversion

RLS Algorithm (Cont.)

- We have

$$\boldsymbol{h}_M[n] = P_M[n]D_M[n]$$

- We can calculate $D_M[n]$ recursively as

$$D_M[n] = wD_M[n-1] + d[n]\boldsymbol{x}_M^*[n]$$

- Using the matrix inversion lemma we have

$$\begin{aligned}\boldsymbol{h}_M[n] &= \frac{1}{w}\left[P_M[n-1] - K_M[n]\boldsymbol{x}^{\mathrm{T}}[n]P_M[n-1]\right] \\ &\quad \times [wD_M[n-1] + d[n]\boldsymbol{x}_M^*[n]] \\ &= P_M[n-1]D_M[n-1] + \frac{1}{w}d[n]P_M[n-1]\boldsymbol{x}_M^*[n] \\ &\quad -K_M[n]\boldsymbol{x}_M^{\mathrm{T}}[n]P_M[n-1]D_M[n-1] \\ &\quad -\frac{1}{w}d[n]K_M[n]\boldsymbol{x}^{\mathrm{T}}[n]P_M[n-1]\boldsymbol{x}_M^*[n] \\ &= \boldsymbol{h}_M[n-1] + K_M[n]\left[d[n] - \boldsymbol{x}_M^{\mathrm{T}}[n]\boldsymbol{h}_M[n-1]\right]\end{aligned}$$

RLS Algorithm (Cont.)

- Let us define

$$e_M[n] = d[n] - \boldsymbol{x}_M^{\mathrm{T}} \boldsymbol{h}_M[n-1]$$

- Then the update equation (recursive solution) becomes

$$\boldsymbol{h}_M[n] = \boldsymbol{h}_M[n-1] + K_M[n] e_M[n]$$

- Much more efficient

RLS Algorithm: Summary

The steps of RLS are as follows

- Computer e_M using the optimum coefficients from previous step

$$e_M[n] = d[n] - \boldsymbol{x}^{\mathrm{T}}[n]\boldsymbol{h}_M[n-1]$$

- Compute K_M (called Kalman gain filter)

$$K_M[n] = \frac{P_M[n-1]\boldsymbol{x}_M^*[n]}{w + \boldsymbol{x}^{\mathrm{T}}[n]P_M[n-1]\boldsymbol{x}_M^*[n]}$$

- Update P_M

$$P_M[n] = \frac{1}{w}\left[P_M[n-1] - K_M[n]\boldsymbol{x}_M^{\mathrm{T}}[n]P_M[n-1]\right]$$

- Calculate optimum filter coefficient

$$\boldsymbol{h}_M[n] = \boldsymbol{h}_M[n-1] + K_M[n]e_M[n]$$

- Note that the each coefficient is updated independently using M free parameters (elements of K_M)

LDU factorization

- The RLS algorithm involves the calculation of squares of $\boldsymbol{x}$. This operation might result in increased round-off errors
- Using LDU decomposition helps to decrease these errors
- For example, let us use LDU decomposition of $P_M[n]$

$$P_M[n] = L_M[n]\gamma_M[n]L_M^{\mathrm{H}}[n]$$

LDU factorization (Cont.)

- Substituting this into the update equation of P_M in RLS algorithm we can obtain the following

$$
\begin{aligned}
& L_M[n]\gamma_M[n]L_M^{\mathrm{H}}[n] \\
& = \tfrac{1}{w}\left[L_M[n-1]\hat{L}_M[n-1]\hat{\gamma}[n-1]\hat{L}_M^{\mathrm{H}}[n-1]L_M^{\mathrm{H}}[n-1]\right]
\end{aligned}
$$

- That is the update equations are

$$L_M[n] = L_M[n-1]\hat{L}_M[n-1]$$

$$\gamma_M[n] = \frac{1}{w}\hat{\gamma}_M[n-1]$$

- These update equations depend on directly $\boldsymbol{x}$ not the square of it, hence the error due to rounding is significantly smaller

Fast RLS Algorithms

- The most time-consuming part of the RLS algorithm is the computation of K_M where matrix multiplications are involved
- Fast RLS algorithms avoid these matrix multiplications by using forward and backward prediction formulae

Properties of RLS Algorithms

- Superior convergence rate, especially important for fast changing signals
- An example of channel equalization

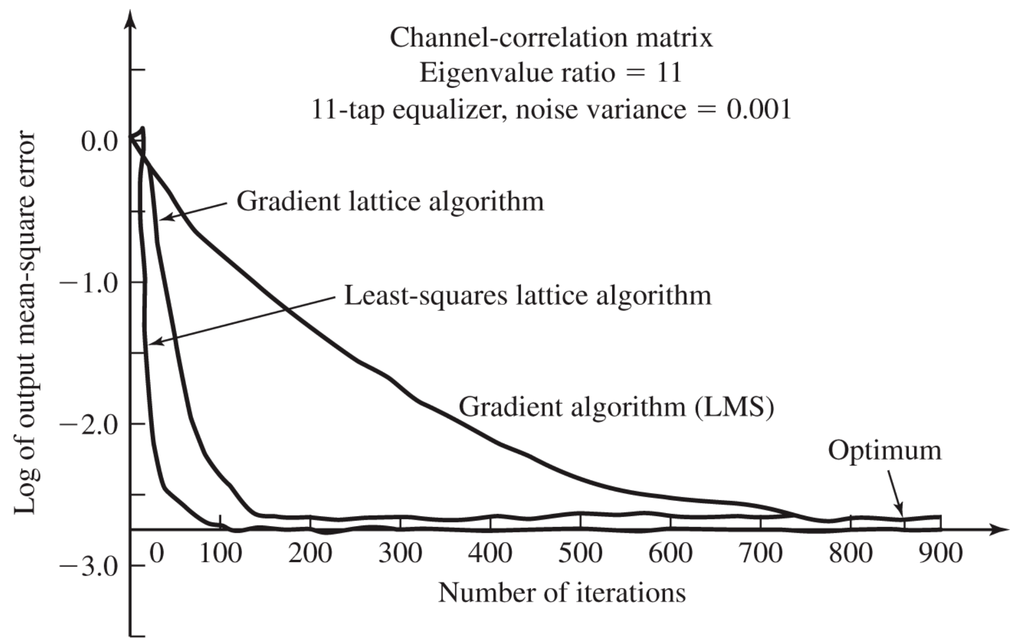

Figure 13.4.3 Learning curves for RLS lattice, gradient lattice, and LMS algorithms for adaptive equalizer of length $M = 11$. (From *Digital Communications* by John G. Proakis. ©1989 by McGraw-Hill Book Company. Reprinted with permission of the publisher.)

Properties of RLS Algorithms

- Computationally more complex than LMS
- Direct RLS $O(M^2)$
- Fast RLS around $O(M)$ but still with higher cost than LMS
- Another disadvantage: accumulation of roundoff errors during iterations

Power Spectrum Estimation: Introduction

- The goal is to estimate the power density spectrum of a signal given a finite length observation of it
- When the signal is stationary the longer the data the better the estimate is
- When the signal is non-stationary longer data do not guarantee better estimates
- For non-stationary case, we must use data of sufficient length that would result in a reliable estimate, but not too long that would result in ignoring the time-varying characteristics
- Power spectrum estimation methods can be categorized into two groups
  - Non-parametric estimation: no prior model is assumed, the power density spectrum samples are estimated directly
  - Parametric estimation: prior knowledge is used to model the power density spectrum using a few parameters, these parameters are estimated that yield a final estimate of power density spectrum