AVD623: Communication Systems-II

Vineeth B. S.
Dept. of Avionics

Lecture 6

Figures are taken from “Communication Systems” by Simon Haykin, “Software receiver design” by Sethares and Johnson.

Review

- Multipath interference and models for dispersive channels
- Least Squares (minimum squared error) Linear Equalization

Multipath interference

- If $u(t)$ is transmitted we receive $y(t)$ where

$$y(t) = a_1 u(t-\Delta_1) + a_2 u(t-\Delta_2) + \cdots + a_n u(t-\Delta_n).$$

- The difference $\Delta_n - \Delta_1$ is called the delay spread of the channel
- Recall the digital transmission system block diagram from last class
- Since we sample the received signal we have

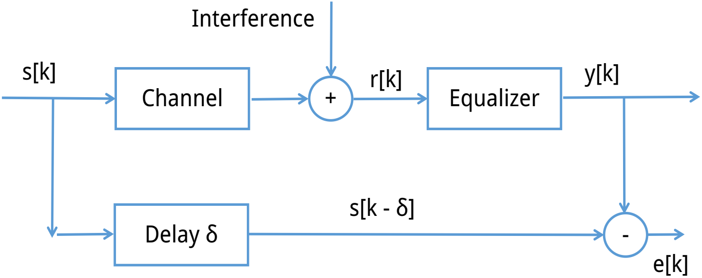
$$y(kT_b) = \alpha_1 u(kT_b) + \alpha_2 u((k-1)T_b) + \cdots + \alpha_N u((k-N)T_b)$$

- In case we have noise or additive interference we have an extra term

$$y(kT_b) = \alpha_1 u(kT_b) + \alpha_2 u((k-1)T_b) + \cdots + \alpha_N u((k-N)T_b) + \eta(kT_b)$$

- Note that in this sampled model, the parameter N should be such that $NT_b \geq \Delta_n$

Least squares linear equalization



- Suppose both source and receiver have access to a predetermined sequence of bits
- Then how can we design an equalizer to mitigate ISI

Least squares linear equalization

- Assume that the equalizer is of the linear transversal form
- Let

$$y[k] = \sum_{j=0}^{n} f_j r[k-j]$$

- For example,

$$y[n+1] = [r[n+1], r[n], \ldots, r[1]] \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Least squares linear equalization

- We can write

$$\begin{bmatrix} y[n+1] \\ y[n+2] \\ \vdots \\ y[p] \end{bmatrix} = \begin{bmatrix} r[n+1] & r[n] & r[n-1] & \dots & r[1] \\ r[n+2] & r[n+1] & r[n] & \dots & r[2] \\ r[n+3] & r[n+2] & r[n+1] & \dots & r[3] \\ & & \dots & & \\ r[p] & r[p-1] & r[p-2] & \dots & r[p-n] \end{bmatrix} \times \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

- $Y = RF$
- The matrix R has a special structure - Toepliz

Least squares linear equalization

- The error is $e[k] = s[k-\delta] - y[k]$
- Or $E = S - Y = S - RF$
- We define an error metric $J_{LS} = \sum_{i=n+1}^{p} e[k]^2$
- Our objective is then to choose the f_i such that J_{LS} is minimized

Least squares linear equalization

- The error metric $J_{LS} = \sum_{i=n+1}^{p} e[k]^2$
- Or we have that $J_{LS} = E^T E$
- $J_{LS} = (S - RF)^T (S - RF)$
- Writing this out we have $J_{LS} = S^T S - 2S^T RF + (RF)^T RF$
- We use a mathematical trick here to solve for the optimal F
- Suppose $\Psi = [F - (R^T R)^{-1} R^T S]^T (R^T R)[F - (R^T R)^{-1} R^T S]$
- Then $J_{LS} = \Psi + S^T [I - R(R^T R)^{-1} R^T] S$
- Since only Ψ depends on F we minimize Ψ
- But Ψ is minimized at $F^* = (R^T R)^{-1} R^T S$
- We note that this solution depends on the specification of δ

Adaptive form of Least squares linear equalization

- If we have a training sequence then the filter coefficients F can be computed as $(R^T R)^{-1} R^T S$
- Since this involves an inverse, it is computationally hard
- Let us use a gradient approach to approximate the solution - instead of analytically finding the optimal F
- We define

$$J_{LS} = \frac{1}{2} avg \left\{ e[k]^2 \right\}$$

- Remember that

$$e[k] = s[k - \delta] - y[k] = s[k - \delta] - \sum_{j=0}^{n} f_j r[k - j]$$

- Since we are adapting our filter coefficients they change with time. So we denote the filter coefficients as

$$f_i[k]$$

- Suppose $\mu > 0$, then we can update the filter coefficients $f_i[k]$ according to

$$f_i[k+1] = f_i[k] - \mu \frac{\partial J_{LS}}{\partial f_i}\Big|_{f_i = f_i[k]}$$

Adaptive form of Least squares linear equalization

- Remember that

$$e[k] = s[k-\delta] - y[k] = s[k-\delta] - \sum_{j=0}^{n} f_j r[k-j]$$

- Suppose $\mu > 0$, then we can update the filter coefficients $f_i[k]$ according to

$$f_i[k+1] = f_i[k] - \mu \frac{\partial J_{LS}}{\partial f_i}\Big|_{f_i = f_i[k]}$$

- We proceed as follows

$$\begin{aligned} \frac{\partial J_{LS}}{\partial f_i} &= \frac{\partial \frac{1}{2} avg\left\{e^2[k]\right\}}{\partial f_i} \\ &\approx avg\left\{\frac{\partial \frac{1}{2} e^2[k]}{\partial f_i}\right\} \\ &= avg\left\{-e[k]r[k-i]\right\} \end{aligned}$$

Adaptive form of Least squares Linear Equalization

- Approximately

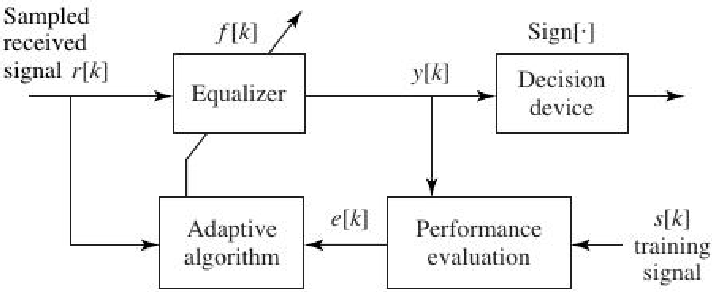
$$\frac{\partial J_{LS}}{\partial f_i} = avg\,\{-e[k]r[k-i]\}$$

- So we obtain that

$$f_i[k+1] = f_i[k] + \mu\, avg\,\{e[k]r[k-i]\}$$

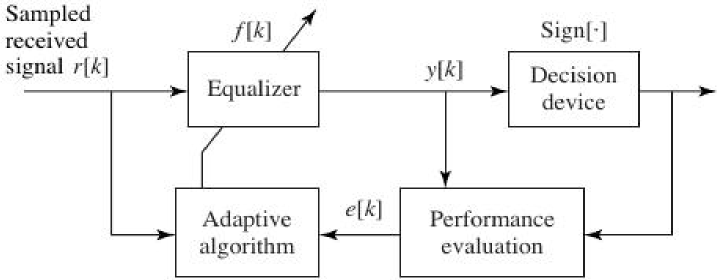
- We will again approximate this as

$$f_i[k+1] = f_i[k] + \mu\,\{e[k]r[k-i]\}$$



Decision Directed Linear Equalization

- How to reduce the overhead due to sending training symbols?
- Use the estimated symbols themselves for training - the training is directed by the decisions taken



- The update step becomes

$$f_i[k+1] = f_i[k] + \mu \left[(sign(y[k]) - y[k]) r[k-i] \right]$$

Dispersion Minimizing Linear Equalization

- We consider an alternative performance metric - one that does not depend on the actual symbols $s[k-\delta]$
- For binary data note that $s[k]^2 = 1$ if $s[k] \in \{1, -1\}$
- So we consider a performance metric

$$J_{DM} = \frac{1}{4} avg \left\{ (1 - y^2[k])^2 \right\}$$

  This measures the dispersion of the equalizer output from the ideal value of 1
- The adaptation algorithm in this case is

$$f_i[k+1] = f_i[k] - \mu \frac{\partial J_{DM}}{\partial f_i} \Big|_{f_i = f_i[k]}$$

- Exercise: Using the sequence of steps used for deriving adaptive least squares show that the adaptation step can be written as

$$f_i[k+1] = f_i[k] + \mu (1 - y^2[k]) y[k] r[k-i]$$