EE4140

Simulation Assignment - II

NAME : Ayush Mukund Jamdar
ROLL : EE20B018

- This document is the project report for SA2.
- This report shall describe the assignment problem, the approach and the results.

NOTE : Section numbers in this report follow the exact same indexing as the Python code notebook.

[Section - I]

- Linear FIR Equalizer Design :

INTRODUCTION & THEORY

- We minimize the Mean Squared Error

$I(k)$ ——— $r(k)$ ——— $d(k) = I(k-\Delta)$

transmitted symbols — L-tap — linear equalizer — estimated symbols

- Equalizer filter (Weiner Filter)

$$\bar{w} = [w_0 \; w_1 \cdots w_{N-1}]^T$$

and,

$$\bar{R}\,\bar{w} = \bar{p}$$

where $\bar{R}$ = autocorrelation matrix → toeplitz symmetric

$$= E[\bar{r}(k)\,\bar{r}(k)^T]$$

and $\bar{p} = E[\bar{r}(k)\,d(k)]$
cross correlation

$$\bar{r}(k) = [r(k) \;\; r(k-1) \;\; \cdots \;\; r(k-N+1)]^T$$

- We find $r(k)$ using convolution

$$r(k) = \sum_{l=0}^{L-1} f_l \, I(k-l) + v(k)$$

- Within linear filters, there are two ways to approach the $\bar{R}$ and $\bar{p}$ matrices —

A] **Statistical WF** :

(a) Auto correlation : using statistical nature of expectation,

$$R_{vv}(n) = \sum^{L-1} f_l f_{l+n} \sigma_I^2 + \sigma_v^2 \delta(n)$$

$$\sum_{\ell=0} \qquad \in \mathbb{R}^{N\times N} \qquad n = 0, 1, \ldots, N-1$$

- Since R_{yy} is Toeplitz (+ symmetric), we only need to find one row. 'n' here indices row elements.
- We assume that $I(k)$ is an IID RV with variance σ_I^2 and Gaussian white noise with variance σ_V^2
- This came from =

$$R_{yy}(n) = \mathbb{E}\left[r(k) \cdot r(k-n)\right]$$

(b) Cross Correlation : Similarly, we can write,

$$p(n) = \mathbb{E}\left[\left(\sum_{\ell=0}^{L-1} f_\ell I(k-n+1-\ell) + V(k-n+1)\right) I(k-\Delta)\right]$$

$$p(n) = \sigma_I^2 f(\Delta+1-n) \qquad p \in \mathbb{R}^{N\times 1}$$

$$n = 0, 1, \ldots, N-1$$

NOTE: this explains the indexing used in code to find these matrices.

Thus, the statistical linear filter exploits the problem statistics and finds an analytical solution for the equalizer vector.

B] Time Averaged Weiner Filter:

- Time ergodicity

- uses time average to calculate $\bar{R}$ and $\bar{P}$

(a) Auto correlation $R = \sum_{k=1}^{N} \frac{\bar{r}(k) \cdot \bar{r}(k)^T}{N}$

(b) Cross correlation $\bar{p} = \sum_{k=1}^{N} \frac{\bar{r}(k)\, d(k)}{N}$

* Assignment Questions

NOTE: Indexing is in sync with code so that it is easy to follow.

1.1] For all parts of section 1,

(a1) $F(z) = \frac{1}{\sqrt{2}}\left(0.8 - z^{-1} + 0.6 z^{-2}\right)$

We need to find the best estimator $\bar{w}$ for $N=3$, $\Delta = 0$, SNR = 10 dB

$\Rightarrow \sigma_I^2 = 1 \quad \sigma_V^2 = 0.1$

Result: $\bar{w}_{opt} = \begin{bmatrix} 0.95 \\ 0.8 \\ 0.3 \end{bmatrix}$ $J_{min} = 0.457$

1.2] $N = 10$, $\Delta = 0$, SNR = 10 dB

(a2)

$W_{opt} = [\,0.98 \quad 0.84 \quad 0.31 \quad -0.08 \quad -0.2 \quad -0.13$

$-0.03 \quad 0.02 \quad 0.03 \quad 0.016]^T$

$J_{min} = 0.445$

1.3] $N = 10$, $\Delta = 5$, SNR = 10dB

(a.3) $J_{min} = 0.337$

1.4] To find the pair (N^*, Δ^*) that minimizes

(a.4) J_{min}, we iterate over all possible values.

to get (See code)

$N^* = 24$

$\Delta^* = 12$ $\quad$ $J_{min}(N^*, \Delta^*) = 0.3314$

1.5] Evaluate SER for 4 PAM

(a.5) We use a total of

$$N_{samp} = 50{,}000 + N + L - 2$$

samples of I_k

- In the get SER function, we run through the blocks f(n) the w(n) to get $\hat{I}(k-\Delta)$.
- In decoding, we compare $I(k)$ with $\hat{I}(k-(N+L+\Delta-2))$
- Plot attached after (1.7)

1.6] TAWF: for $N = 10$, $\Delta = 5$, we run the (a6) time averaging for different number of snapshots used to construct R and P.

We get

	J_{min}
$P = 20$	0.64
$P = 100$	0.31
$P = 500$	0.339

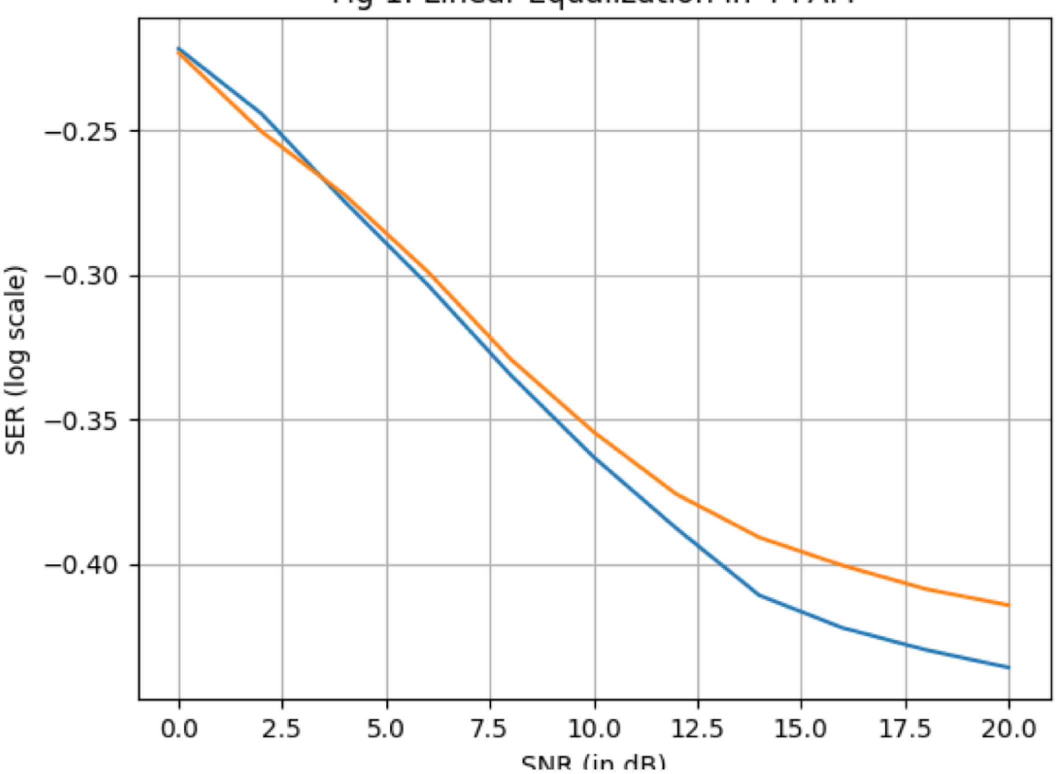
1.7]

(a7) Now we plot SER for SWF and TAWF together in Fig. 1.

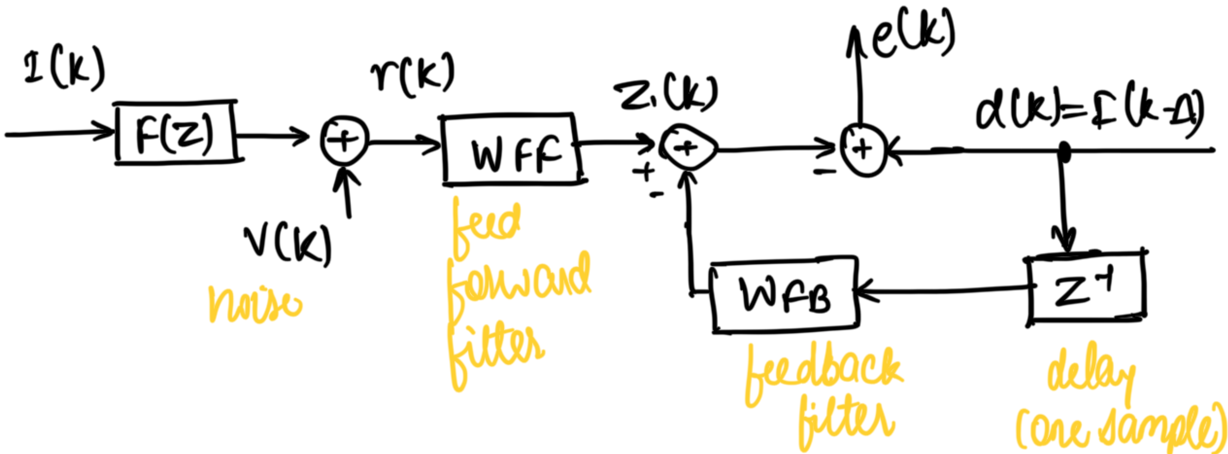
Comments:

(1) TAWF closely approximates the statistical WF. Hence, the assumption of ergodic and stationary are valid.

Fig 1. Linear Equalization in 4-PAM



[Section – II]
Decision Feedback Equalizer



- Here, we use a 6-tap filter

$$f(z) = [1, -0.95, 0.5, 0.15, -0.2, -0.1] / c$$

$$c = 1.4916$$

2.1] SWF Linear Equalizer

(b1) $J_{min} = 0.264$

2.2] DFE $N_1 = 15$ $N_2 = 5$

(b2) We have the new expression for W

$\bar{W} = [W_{FF_0}, W_{FF_1}, \ldots, W_{FF_{N_1}}, W_{FB_1}, \ldots, W_{FB_{N_2}}]$

$W = [w_{f0}\ w_{f1} \cdots w_{fN_1-1}\ w_{b0} \cdots w_{bN_2-1}]$

Thus the Weiner solution remains the same

$$\bar{W} = R^{-1} p$$

- Define

$$e(k) = d(k) - W \begin{bmatrix} r(k) \\ r(k-1) \\ \vdots \\ r(k-N_1+1) \\ -d(k-1) \\ \vdots \\ -d(k-N_2) \end{bmatrix}$$

$d(k)$ = decoded sequence

$$b(k) = [r(k)\ r(k-1) \dots r(k-N_1+1)\ -d(k-1)\ -d(k-2) \cdots -d(k-N_2)]^T$$

then we can define

$$R = E[b(k)\, b(k)^T]$$
$$p = E[b(k)\, d(k)]$$

for

$$r(k) = [r(k)\ r(k-1) \dots r(k-N_1+1)]^T$$
$$d(k) = -[d(k-1) \cdots d(k-N_2)]^T$$
$$R = \begin{bmatrix} E[r(k)\, r(k)^T] & E[r(k)\, d(k)^T] \\ E[d(k)\, r(k)^T] & E[d(k)\, d(k)^T] \end{bmatrix}$$

this is done by numpy 'block'

Each of these matrices are explained

1. $E[r_k r_k^T] = \sum_{l=0}^{L-1} f_l f_{l-n} \sigma_I^2 + \sigma_v^2 \delta(j-i)$

2. $E[d_k d_k^T] = \sigma_I^2 \, I(N_2)$ identity matrix

3. $$E[r_k d_k^T] = -\sigma_I^2 \begin{bmatrix} f_{\Delta+1} & f_{\Delta+2} & \cdots & f_{\Delta+N_2-1} & f_{\Delta+N_2} \\ f_{\Delta} & \vdots & & \vdots & \vdots \\ \vdots & & & & \vdots \\ f_{\Delta-N_1+2} & & & f_{\Delta-N_1+N_2} & f_{\Delta-N_1+N_2+1} \end{bmatrix}$$

4. $$\bar{p} = \sigma_I^2 \begin{bmatrix} f_{\Delta} \\ f_{\Delta-1} \\ \vdots \\ f_{\Delta-N_1+1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

- using this results, we run the simulation to get $J_{min} = 0.16$
- comparing results,
  - LE $J_{min} = 0.26$
  - DFE $J_{min} = 0.16$

Thus, DFE promises a better decoding. (even with a lower delay of 1)
$\longrightarrow$ this is due to the feedback filter
$\longrightarrow$ thus a lower MSE

2.3] and 2.4]
(b3) and (b4)

- Now we run the simulation for a Binary

Now we run the simulation for Binary PAM (BPAM) and calculate SER using three methods

(1) SWF Linear Equalizer

(2) Decision feedback Equalizer

(3) Viterbi Algorithm

[See code]

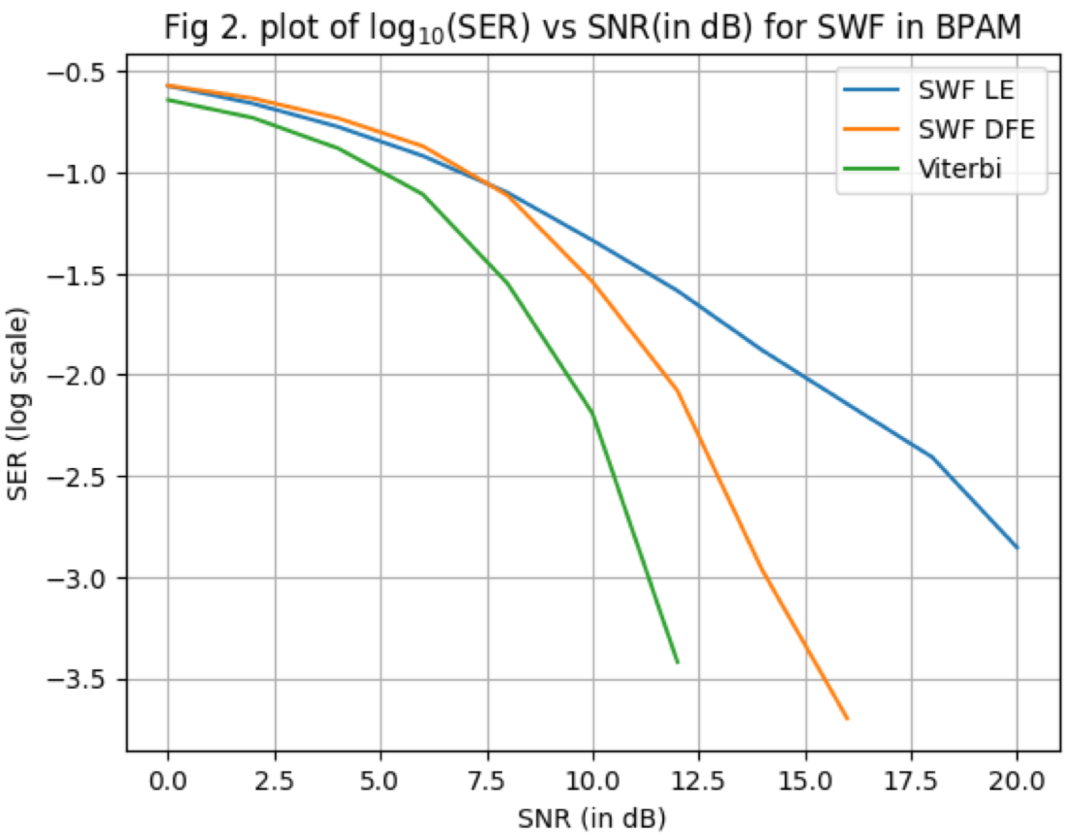


Fig. 2

(1) Clearly, VA performs better than equalizers since VA is the optimal approach for a linear channel model

(2) DFE, as analysed earlier, performs better than LE due to feedback that handles post cursors

(3) At low SNR, noise $\approx$ signal power, error rates are the highest.

the difference between SERs of three methods becomes significant at high SNR