Problem Set: Adaptive Filtering

U B Desai*

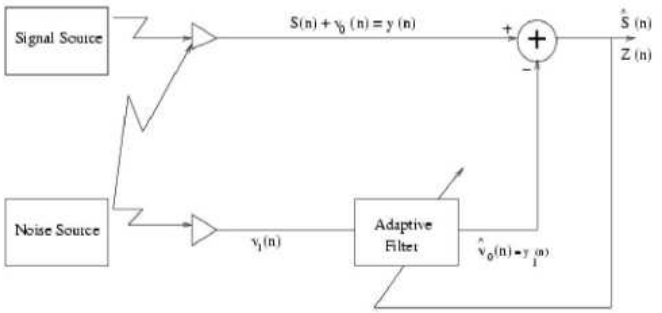

Fig. 1.0

***Abstract*—This problem set covers all the basic concepts in Adaptive Filtering.**

1 Estimation

1. Consider the following M-th order FIR adaptive structure for noise cancellation shown in Fig. 1.0. Assume that $v_i(n)$ and s(n) are statistically independent. Let $e(n) = s(n) + v_0(n) - \hat{v}_0(n)$. Now suppose the adaptation rule is chosen such that $E[e^2(n)]$ is minimized. Then show that $E[e^2(n)] = E[(s(n) - \hat{s}(n))^2]$. Thus minimizing $E[e^2(n)]$ will indeed yield $\hat{s}(n)$ as the minimum mean square estimate of s(n).
2. Let $\{y_k\}_{k=1}^N$ be N measurements of unknown constant c. The problem is to derive the expression for the l_2,l_1, and l_∞ estimate of the constant c using the measurements $\{y_k\}_{k=1}^N$. We model the measurements as sample values of a discrete time random variable Y having the density function

$$f_Y(y) = \sum_{k=1}^{N} \alpha_k \delta(y - y_k) \tag{1.1}$$

where α_k are such that

$$\sum_{k=1}^{N} \alpha_k = 1. \tag{1.2}$$

Show that

(a) The l_2 estimate of c is given by

$$\hat{c} = \sum_{k=1}^{N} \alpha_k y_k \tag{1.3}$$

(b) The l_1 estimate of c is given by $\hat{c} = y_p$, for $1 \le p \le N$ such that

$$\sum_{y_k<\hat{c}} \alpha_k = \sum_{y_k>\hat{c}} \alpha_k \tag{1.4}$$

(c) The l_∞ estimate is given by

$$\hat{c} = \frac{\min\{y_k\} + \max\{y_k\}}{2} \tag{1.5}$$

3. Let

$$X_k, k = 1 \dots N$$

be N independent samples of a stationary Gaussian distribution.Thus

$$f_{X_k|M,V}\left(x_k|m,\sigma^2\right) = \frac{1}{2\pi\sigma^2} e^{-(x_k-m)^2/(2\sigma^2)} \tag{1.6}$$

and letting $Z = \{X_1, ..., X_N\}, z = \{x_1, ..., x_n\}$

$$f_{Z|M,V^{(z|m,\sigma^2)}} = \prod_{k=1}^{N} f_{X_k|M,V^{(x_1|m,\sigma^2)}}$$

$$= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\sum_{k=1}^{N} (x_k - m)^2/2\sigma^2\right] \tag{1.7}$$

(a) Assume σ^2 is known. Show that the Maximum Likelihood Estimate (MLE) of m is given by

$$\hat{m}_{MLE} = \frac{1}{N} \sum_{k=1}^{N} x_k \tag{1.8}$$

and is an unbiased estimate.

(b) Assume m is known. Show that MLE of $\hat{\sigma}^2$ is given by

$$\hat{\sigma}^2_{MLE} = \frac{1}{N} \sum_{k=1}^{N} (x_k - m)^2 \tag{1.9}$$

and is unbiased.

*The author is with the Department of Electrical Engineering, IIT Hyderabad, Kandi 502285.

(c) *Neither m nor σ^2 is known.* Show that

$$\hat{m}_{MLE} = \frac{1}{N}\sum_{k=1}^{N} x_k, \tag{1.10}$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{k=1}^{N}(x_k - \hat{m}_{MLE})^2 \tag{1.11}$$

Furthermore, show that, $\hat{m}_{MLE}$ is unbiased, while $\hat{\sigma}^2_{MLE}$ is biased with

$$E[\hat{\sigma}^2_{MLE}] = \frac{N-1}{N}\sigma^2 \tag{1.12}$$

(d) Recursive computation of $\hat{m}_{MLE}$ and $\hat{\sigma}^2_{MLE}$. Show that if

$$\hat{m}_{MLE} = \hat{m}_N \tag{1.13}$$

$$\hat{\sigma}^2_{MLE} = \hat{\sigma}^2_N, \tag{1.14}$$

then

$$\hat{m}_N = \frac{1}{N}\left[(N-1)\hat{m}_{N-1} + x_N\right] \tag{1.15}$$

$$\hat{\sigma}^2_N = \frac{1}{N}\left[(N-1)\hat{\sigma}^2_{N-1} + \frac{N}{N-1}(x_N - m_N)^2\right] \tag{1.16}$$

4. Solve the following l_2 minimization problems. Let A be an $m \times n$ matrix, x an $n \times 1$ vector, and y an $m \times 1$ vector. Find x such that

(a)

$$\|Ax - y\|^2 = (Ax - y)^T(Ax - y) \tag{1.17}$$

is minimized.

(b)

$$\min_{subj.Ax=b} \|x\|^2 \tag{1.18}$$

Note: These are two fundamentals finite dimensional l_2 approximation problems which can serve as prototypes for any other finite dimensional l_2 approximation problem.

5. **On QR Decomposition**

*Definition:*LetA be an m×n matrix with m>n and rank p; then the QR decompositon of A is defined by

$$A = \begin{bmatrix} Q & N \end{bmatrix}\begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} QR_1 & QR_2 \end{bmatrix} \triangleq QR$$

where

(a) the dimensions of various matrices are Q:m×p, N:m×$(m-p)$,$R_1 : p\times$p,$R_2 : p \times (n-p)$. $R = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$ is p× n.

(b) $\begin{bmatrix} Q & N \end{bmatrix}$ is an orthogonal matrix,namely ,

$$\begin{bmatrix} Q & N \end{bmatrix}\begin{bmatrix} Q^T \\ N^T \end{bmatrix} = I_m = \begin{bmatrix} Q^T \\ N^T \end{bmatrix}\begin{bmatrix} Q & N \end{bmatrix}$$

Note the obvious $QQ^T \neq I$ and $NN^T \neq I$. But $Q^TQ = I_p$ and $N^TN = I_{m-p}$.

(c) R_1 is a p×p upper triangular matrix. Note, if p=n then $R_2 = 0$.

(a) Show that the columns of Q provide an orthogonal basis for the range space of A, while the columns of N provide an orthogonal basis for the null space of A^T.

(b) **Algorithm for obtaining the QR Decomposition**

Consider a 2×1 column vector $\begin{bmatrix} x & y \end{bmatrix}^T$. Now it is shown earlier that the orthogonal Givens rotation

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, c = \frac{x}{\sqrt{x^2+y^2}}$$

$$s = \frac{y}{\sqrt{x^2+y^2}}$$

when applied to $\begin{bmatrix} x & y \end{bmatrix}^T$,yields

$$G\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ 0 \end{bmatrix}$$

First figure out how the above method can be used to selectively zero elements of any column vector.Using this idea develop an algorithm for computing the QR decomposition of an m×n $(m > n)$ matrix A. You can illustrate your algorithm by considering a full rank 4×3 matrix A.

6. One possible norm for an m×n matrix A is defined as

$$\|A\| \triangleq \max_{x^Tx=1, x\epsilon\Re^n} (x^TA^TAx)$$

Now let the SVD of an m×n matrix A be A=UΣV^T.Show that $\|A\| = \sigma_{max}(A)$.

7. Let the SVD of an m×n matrix A be A=UΣV^T.Now define the pseudo inverse of A as

$$A^\dagger = V\Sigma^{-1}U^T$$

verify that the above $A^\dagger$ satisfies the following identities for the matrix to be a pseudo inverse.

(i) $(A^\dagger)^\dagger = A$, (v) $(A^\dagger A)^T = A^\dagger A$,

(ii) $(A^T)^\dagger = (A^\dagger)^T$, (vi) $A^\dagger = (A^T A)^\dagger A^T$,

(iii) $A^\dagger A A^\dagger = A^\dagger$, (vii) $A^\dagger = A^T (A A^T)^\dagger$.

(iv) $A A^\dagger A = A$,

2 Levinson's Algorithm

1) **Computing the Sample Covariance Matrix**
Let $Y(k)$ be a discrete time stochastic process, and $\{y(k)\}_0^{N-1}$ be a realization of this process.

A typical problem in practice is to compute the second order statistics of the process.
The procedure for computing the mean is well known; in fact it is the answer obtained earlier using the maximum likelihood estimation procedure under Gaussian assumption, namely

$$\hat{m}_N = \frac{1}{N} \sum_0^{N-1} Y(k)$$

To avoid a new notation we use the *computer programming definition*
$Y(k) := Y(k) - \hat{m}_N \rightarrow y(k) := y(k) - \hat{m}_N$

Now we look for procedures for computing the covariance of Y(k) using $\{y(k)\}_0^{N-1}$.
(a) Let

$$\bar{r}_N(k) = \frac{1}{N-k}[\sum_{j=0}^{N-k-1} y(j+k)y(j)], k = 0, 1, ...N-1$$

show that

i. $\bar{r}_N(k)$ is an unbiased estimate of the true covariance parameter $r(k) = E[(j+k)Y(j)]$
ii. the covariance matrix constructed using $r_N(k)$ will not be non-negative definite. Because of this problem one looks for alternative methods for computing the sample covariance matrix.

(b) Alternately let the sample covariance matrix be defined as

$$r_N(k) = \frac{1}{N-k}[\sum_{j=0}^{N-k-1} y(j+k)y(j)], k = 0, 1, ...N-1$$

i. Show that $r_N(k)$ is not an unbiased estimate of the true covariance parameter; nevertheless it is an asymptotically unbiased estimate.
ii. Show that the covariance matrix constructed using $\bar{r}_N(k)$ is non-negative definite.
(Hint: Express the sample covariance matrix as a product of data matrix with its transpose).

2) Let
(a) $\hat{E}[x(n)|y(n-1), ...y(n-M)]$

(b) $\hat{E}[x(n+N)|y(n+N-1), ...y(n+N-M)]$
*Note:*This simple result shall be used often.

3) **Determining $\hat{x}(k|n)$ using Levinson's Algorithm**
In class we developed the Levinson's Algorithm for the one step prediction problem and for generating the innovations. In this problem you are required to develop a recursive scheme for computing $\hat{x}(k|n)$ in the framework of Levinson's algorithm.
Show that
$\hat{x}(k|n+1) = \hat{x}(k|n) + \Gamma(k,n)\epsilon(n)$

Determine the expression for the gain $\Gamma(k,n)$ and an expression for recursively computing it. The above is referred to as the measurement update equation.

4) In the Levinson Algorithm derive the expression

$$p(n) = r(0) + \sum_{i-0}^{n-1} a_n(n-i)r(i-n)$$

where $p(n)$ is the innovations (error) variance

$$p(n) = E[\{y(n) - \hat{y}(n|n-1)\}^2]$$

5) Let $y(\cdot)$ be a zero mean process with

$$E[y(k+m)y(m)] = (\frac{1}{4})^{|k|}$$

and $x(\cdot)$ be another process with

$$E[y(k+m)y(m)] = \begin{cases} (2/3)^k & k \geq 0 \\ (1/3)^k & k \leq 0 \end{cases}$$

$$E[x(k+m)x(m)] = (\tfrac{1}{2})^{|k|}$$

   (a) Find the optimal filter for computing the linear least squares estimate of $x(n+N), N > 0$, based on $\{y(k), k \leq n\}$.

6) **Simulation:**Consider the sample values of a stationary process y(Â·) on the home page for the
ASP course:in http: //sharada.ee.iitb.ac.in/ee608/, file name is levinson.dat
   (a) Determine the first 50 covariance lags $r(n)$ $0 \leq n \leq 50$.
   (b) Using Levinson's algorithm determine an appropriate AR model. Comment on the goodness of this model. If there are any deficiencies or problems give reasons for them.

3

1) Consider the state space model

$$X(k+1) = F(k)X(k) + G(k)u(k), x(0) = x_0$$

with $E[X_0] = 0, E[u(k)] = 0, E[x_0 {x_0}^T] = \Pi_0, E[u(k)u^T(j)] = Q(k)\delta(k-j)$, and for $j \geq k, E[x(k)u^T(j)] = 0$. Let $\Pi(k) = E[x(k)x^T(k)]$.

   (a) Show that

$$\Pi(k+1) = F(k)\Pi(k)F^T(k) + G(k)Q(k)G^T(k)$$

   (b) Show that

$$\Pi(k) - P(k|k-1) \geq 0$$

Give an interpresentation to this result.

2) Verify the matrix inversion identity

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

3) Show that the filtered error covariance matrix is given by

$$P(k|k) = P(k|k-1) - P(k|k-1)H^T {R_\epsilon}^{-1}(k)HP(k|k-1)$$

4) **Measurement Update and Time Update in Kalman Filtering**
Let the filtered estimate be $\hat{x}(k|k) \triangleq \mathcal{P}(k)|\{y(0), ...y(k)\}]$. Using the kalman Filter (one step predictoer) equations developed in class and associated manipulations, show that the filtered and one step predicted estimate can be computed through the following equations.
Measurement Update

$$\begin{aligned} \hat{x}(k|k) &= \hat{x}(k|(k-1) + K_f(k)[y(k) - H\hat{x}(k|k-1)], \hat{x}(0|-1) = \\ k_f(k) &= P(k|k-1)H^T R\epsilon^{-1}(k) \\ &= P(k|k)H^T R^{-1} \\ R_\epsilon(k) &= R + HP(k|k-1)H^T \\ P(k|k) &= P(k|k-1) - P(k|k-1)H^T {R_\epsilon}^{-1} HP(k|k-1), P(0|- \end{aligned}$$

Time Update

$$\begin{aligned} \hat{x}(k+1|k) &= F\hat{x}(k|k) \\ P(k+1|k) &= FP(k|k)F^T + GQG^T \end{aligned}$$

Note, form the above equation it should be obvious as to why the computation of the filtered estimate from the one step predicted estimate is referred to as the Measurement

Update, while the computation of the one step predicted estimate from the filtered estimate is referred to as the Time Update.

4

1) Find an expression for J_{min}, the Wiener optimal cost.

2) *Method of Steepest Descent:*
In class we derived the LMS algorithm using the method of, what was referred to as, *gradient descent*.In literature, the method of steepest descent is talked about very often. Most of the steps in the method of gradient descent and steepest descent are the same, except for the selection of the step size parameter μ. In the method of steepest descent, one solves another minimization problem for selecting μ. This is illustrated in the present problem.
Consider the Problem

$$\min_W J(n) = \min_W \{E[(d(n) - W^T X(n))^2]\}$$

with notations and assumptions as defined in class. This problem can be solved non-iteratively, by using the orthogonality principle or simply differentiating the above. On can obtain an iterative algorithm using the method of steepest descent. We define the steepest descent based iterative equation as

$$W(n+1) = W(n) + \mu(n)(-\nabla_W[J(n)])$$

Now μ is selected such that $\bar{J}(n) = [J(n)]_{W=W(n+1)}$ is minimized w.r.t μ. Show that that the optimal value of Show that that the optimal value of μ is given by is given by

$$\mu(n) = \frac{(\nabla_{W_{(n)}}[\bar{J}(n)])^T(\nabla_{W_{(n)}}[\bar{J}(n)])}{(\nabla_{W_{(n)}}[\bar{J}(n)])^T R(\nabla_{W_{(n)}}[\bar{J}(n)])}$$

Note that, here μ is time varying.

3) In class we showed that the LMS algorithm converges in the mean if μ is selected such that $0 < \mu < 2\lambda_{max}$.
Show that this condition is satisfied if μ is selected such that $0 < \mu < 2/(M$ (signal power)).

Simulations

4) Using the data given to you in Problem Set 2, implement the LMS algorithm in the predictive mode to whiten the given process. You will need to select an appropriate order for the filter.
What is the eigenvalue spread for this data set?
Remark: The data given earlier in Problem Set 2 was generated from a four pole - three zero model:

$$H(z) = \frac{1 + 0.5z^{-1} + 0.007z^{-2} - 0.013z^{-3}}{1 - 1.2z^{-1} + 0.575z^{-2} - 0.1375z^{-3} - 0.0125z^{-4}}$$

driven by uniform white noise. This was deliberately done to violate various assumptions that we have been making and still see the validity of different adaptive algorithms.

(a) Generate the data for LMS algorithm using the model

$$H(z) = \frac{(z - 0.8)(z + 0.7)}{(z - 0.9)(z + 0.8)(z + 65)}$$

To generate the data you will drive the

above system by a Gaussian white process with zero mean and variance preselected by you. Determine how long you need to run the system, so that it has reached steady state. Once in steady state, start collecting the data. Collect, at least, 512 points (you may need more in case your LMS algorithm is going to take a long time for convergence).

(b) Get an estimate for the signal energy for the above data, and using this estimate determine the range for μ. Select two values for μ in this range.

(c) Run the LMS algorithm in the predictive mode for the data that you have generated and for the two choices of μ.

(d) Do a validation test. You should use the following for the purpose of comparison

i. Learning curve (i.e mean square error curve)

ii. Convergent values of $W(n)$

iii. Whiteness of the error
Comment on which choice of μ gives better results, and why.

5

1) *Complex LMS Algorithm*
Let all the variables in the algorithm be complex. Now, wherever you have transpose terms like A^T, it will get replaced by $A^H = (A^*)^T$. Let

Tap input vector be X(n) = $X_R(n) + jX_I(n)$
Desired Response be d(n) = $d_R(n) + jd_I(n)$
Tap Weight Vector be W(n) = $W_R(n) + jW_I(n)$
Estimation Error be e(n) = $e_R(n) + je_I(n)$

Show that

$$W_R(n+1) = W_R(n) + \mu[e_R(n)X_R(n) - e_I(n)X_I(n)]$$
$$W_I(n+1) = W_I(n) + \mu[e_R(n)X_I(n) + e_I(n)X_R(n)]$$

where

$$e_R(n) = d_R(n) - W_R^T X_R(n) + W_I^T(n)X_I(n)$$
$$e_I(n) = d_I(n) - W_R^T X_I(n) - W_I^T(n)X_R(n)$$

Note: You will need to use derivative of complex quantities w.r.t to real and imaginary parts. Let $f(W)$ be a scalar complex function of a complex column vector $W = W_R + jW_I$,and $W_R = [w_R^1 ... w_R^M]^T, W_I = [w_I^1 ... w_I^M]^T$. Now the derivative of $f(W)$ w.r.t W is defined as:

$$\frac{\partial f(W)}{\partial W} = \begin{bmatrix} \frac{\partial f(W)}{\partial w_R^1} - j\frac{\partial f(W)}{\partial w_I^1} \\ \frac{\partial f(W)}{\partial w_R^2} - j\frac{\partial f(W)}{\partial w_I^2} \\ \vdots \\ \frac{\partial f(W)}{\partial w_R^M} - j\frac{\partial f(W)}{w_I^M} \end{bmatrix}$$

and the conjugate derivative is defined as

$$\frac{\partial f(W)}{\partial W^*} = \begin{bmatrix} \frac{\partial f(W)}{\partial w_R^1} + j\frac{\partial f(W)}{\partial w_I^1} \\ \frac{\partial f(W)}{\partial w_R^2} + j\frac{\partial f(W)}{\partial w_I^2} \\ \vdots \\ \frac{\partial f(W)}{\partial w_R^M} + j\frac{\partial f(W)}{w_I^M} \end{bmatrix}$$

2) **Simulation:**This problem deals with *adaptive noise cancellation* using the LMS algorithm. You will need two data streams: one corresponding to signal plus noise and the other corresponding to noise. You could synthesize such data but this would be too far from the real situation. Thus

(a) Use the speech data files provided on the course web page.
The files are for speech plus noise, and only noise. You will notice that you cannot decipher the speech when you play the speech plus noise file; but upon proper implementation of the noise cancellation algorithm you will be able to decipher the speech.

i. noise3.wav and signal noise3.wav form one par
ii. snp.wav and n.wav from one pair
iii. look for other such pairs on the web; these can also be included in the noise cancellation simulation data set.

(b) OR (optional) your group should capture its own data streams.
You need to generate one stream as: speech data plus noise, and the other stream as only noise. You could use two microphones to capture the data streams. The main catch is in capturing both the data streams simultaneously.
Have at least two set of data; each with a different SNR (one with low SNR and the other with high SNR).

6

1) *An alternate aspect of the convergence analysis for the RLS algorithm*

Suppose there exists a true model for $d(n)$ of the form

$$d(n) = X^T(n)W_o + v(n)$$

where $v(n)$ is some unknown error, and W_o the true weight vector which is also unknown.
The question we want to investigate is whether $W(n) \rightarrow W_o$ in the mean and the mean square sense. Note this is different from what we did in class where we were considering the convergence of $W(n)$ to W_* the Wiener optimal solution.
In this problem we shall only investigate the convergence of $W(n) to W_o$ in the mean. Also consider the RLS algorithm based on

$$W(n) = [\quad \sum_{k=0}^{n} X(k)X^T(k)]^{-1}[\quad \sum_{k=0}^{n} X(k)d(k)]$$

(a) Show that, if $v(i) \perp\!\!\!\perp X(k)$for $k \leq i$, then $E[W(n)] = W_o$.

(b) In a deterministic sense, show that $W(n) \rightarrow W_o$ if $\lim_{n\rightarrow\infty} \frac{1}{n} \sum_{k=0}^{n} X(k)v(k) = 0$.

2) Start with the following non-recursive equations and derive the RLS algorithm having the necessary initial condition $(P(0) = \delta^{-1})I$.

$$W(n) = [\delta I + \quad \sum_{i=1}^{n} X(i)X^T(i)]^{-1}[\quad \sum_{i=1}^{n} X(i)d(i)]$$

$$P(n) = [\delta I + \quad \sum_{i=1}^{n} X(i)X^T(i)]^{-1}$$

3) Simulate the normalized LMS algorithm and compare with the LMS algorithm. For this you should use the
   (a) data generated by the model in the previous problem.
   (b) generate the data for an AR model
   assume that that $v(n)$ is zero mean, unit intensity white sequence. You will need to determine the AR parameters for the given pole locations. Note, you should generate at least 100 data sets, each data set having large enough points so that you can get a reasonable assessment of the mean squared error. The length of the data set will depend on how many iterations are required for convergence
   Use the following form of the normalized LMS algorithm.

$$W(n+1) = W(n) + \frac{aX(n+1)}{c + X^T(n+1)X(n+1)}(d(n) - X^T(n+1)W$$

   You should use the following for the purpose of comparison and validation

   (a) Learning curve (i.e mean square error curve)
   (b) Convergent values of $W(n)$
   (c) Whiteness of the error
4) Simulate the RLS algorithm using
   (a) The data generated during the simulations for the LMS and N-LMS algorithms in the earlier home works.
   (b) Compare the results with those obtained for the LMS and N-LMS algorithms (rate of convergence, excess mean square error, weight values, etc.)
5) Apply the above RLS algorithm for the noise cancellation problem. For this use the data that was used in the earlier noise cancellation simulation. Compare your noise cancellation results using LMS, NLMS and RLS.
6) **Divergence of RLS Algorithm**
   (a) Simulate the divergence in the RLS algorithm. One way to achieve this is to is to round-off every variable to, say, the third or second significant digit, at the end of each iteration (if divergence still does not occur, you may have to round-off to first significant digit).This way, we artificially introduce round-off errors and examine its accumulation effect.
   (b) For the the procedure of part (a) which simulates divergence, now implement the reinitialization scheme for the RLS algorithm and check whether it is able to overcome the problem of divergence. Thus, in

the reinitialization algorithm too, you will be rounding off each variable to third or fourth significant digit.

7

1) As shown in class the decision feedback equalizer is obtained by finding α_k and β_k such that $E[|\ I_k - \hat{I}_k\ |^2]$ is minimized
subject to

$$\hat{I}_k = \sum_{k=-M_1}^{0} \alpha_k y(k-n) + \sum_{k=1}^{M_2} \beta_k I_{k-n}$$

Show that $\{\alpha_k, \beta_k\}$ are obtained by solving

$$\sum_{j=-M_1}^{0} \alpha_j r(l_1 - j) + \sum_{j=1}^{M_2} \beta_j q^*(j - l_1) = q^*(-l_1)$$

$$\sum_{j=-M_1}^{0} \alpha_j q(-j + l_2) 1(L + j - l_2) + \beta_{l_2} = 0$$

where $M_1 \le l_1 \le 0$ and $1 \le l_1 \le M_2$.

In the above $q(\cdot)$ is the impulse response sequence such that the preprocessed received signal (preprocessed in the sense that the received signal has gone through a matched filter and a whitening filter) is given by

$$y(k) = \sum_{n=0}^{L} q(n) I_{k-n} + \eta(k)$$

where we assume that $I_n \perp\!\!\!\perp I_m$ for $m \neq n$, and

$E[I_n I_m^*] = \delta_{m,n}$. Also $I_n \perp\!\!\!\perp \eta(k)$ for all n and k, and $E_\eta(k)\eta(l) = \sigma_n \delta_{k,l}$.

2) Derive the update equation for the the training based LMS Algorithm for adaptive equalization, namely
$W(n+1) = W(n) + \mu Y^*(n)[I(n) - Y^T(n)W(n)], W(0) = 0$

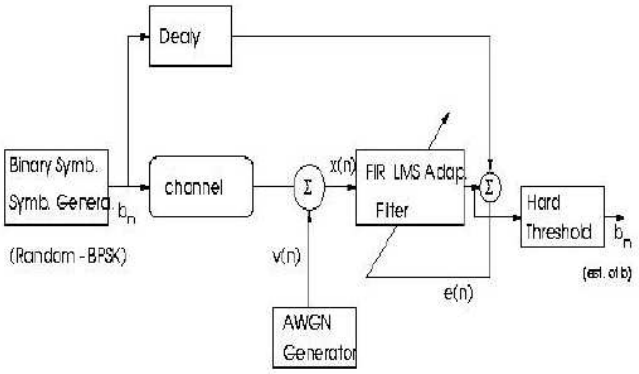

Figure 1: For problem no. 4

Fig. 7.0

3) Derive the CM algorithm of Goddard for the case for
$J_{12} = (|\hat{I}_k| - D)^2$
Derive the weight update equation as well as the carrier tracking equation.
Simulations
4) **LMS algorithm for channel equalization using training signals:**
Consider the following block diagram for the adaptive channel equalizer using training sequences.
In the above we are assuming that the transmitted symbols are binary (bpsk) with zero mean $b_n = \pm 1$. These are passed through a channel with impulse response

$$h(n) = \begin{cases} \frac{1}{2}[1 + cos(\frac{2\pi}{F}(n-2))] & n = 1, 2, 3 \\ 0 & otherwise \end{cases}$$

and corrupted by additive white Gaussian noise (AWGN) with zero mean and variance σ^2. The delayed symbols that are fed at the output of the adaptive filter act as training signals.The channel impulse response is an 3-point FIR filter with a raised cosine type of structure.Parameter F controls the eigen value spread of $R = E[X(n)X^T(n)$.The structure of this problem is very similar to the one given in Haykin.
Note:

$$x(n) = \sum_{k=1}^{3} h(k) b_{n-k} + v(n)$$

You will need two random number generators; one for the symbols bn and the other for the AWGN.For AWGN assume $\sigma^2 = 0.01$. For b_n use a binary random number generator where 1 and -1 are generated with equal probability. You could do this by using a uniform random number generator, uniform over [-0.5, 0.5]. Whenever you get a negative number, set it to -1 and whenever you get a positive number set it to +1.

(a) Experiment with different number of tap weights in the FIR LMS adaptive filter. For example you could consider 7, 9, or 11 tap weights.
(b) Experiment with different variance for AWGN.
(c) Estimate the amount of delay you need based on the number of tap weights you are using.
(d) Run the experiment for two vales of F, say $F = 3$ and $F = 3.2$.
(e) Plot the mean square error curves for each case.
(f) Give a table for the values for $W(\infty)$.
(g) At the output of the adaptive filter have a hard threshold unit to classify the output as 1 or -1. This is done since we know that the transmitted symbols were 1 or -1. Now compute the percentage bits that are in error at convergence (this will give you the bit error rate (BER)).

5) Repeat the above channel equalization problem for the following cases:

(a) The channel is no longer as given by the cosine function. But, now let the channel be modeled by an FIR filter of length 10, and each FIR coefficient be of unit magnitude. This way we will be considering the performance of the equalizer under a narrow band channel.
(b) The data symbols are not binary (BPSK) but are QPSK. You can generalize the method used for generating BPSK to a method for generating QPSK. Plot the bit error rate. Do this for both the cases: (a) the FIR channel specified above, and (b) any other narrow band channel of your choice. Your channel could be FIR, IIR or non-linear.

8

1) Implement Sato's blind channel equalizer for:

(a) BPSK data symbols and channel as specified

$$h(n) = \begin{cases} \frac{1}{2}[1 + cos(\frac{2\pi}{F}(n-2))] & n = 1, 2, 3 \\ 0 & otherwise \end{cases}$$

with $F = 4.5$ or greater.
(b) QPSK data, with the channel as above.
(c) BPSK data, but with a narrow band channel $h(n) = \begin{cases} 1 & n = 0, 1, ...9 \\ 0 & otherwise \end{cases}$
(d) QPSK data with narrow band channel.
(e) A channel of your choice.
In all cases specify the variance of additive noise. Compare the results of the Sato's blind algorithm with the previous simulation where one had used the knowledge of the channel.

2) Implement the *Transform Based* LMS algorithm in the one step predictive mode. Use the data set for the LMS algorithm from Home Work No.3, Problem 8. Use any real transform; DCT or DWT. Comment on the convergence and MSE performance of the transform based LMS algorithm.