Name: Chad Williams

Problem	Points	Score
1(a)	15	
1(b)	15	
1(c)	15	
2(a)	15	
2(b)	15	
3(a)	15	
3(b)	10	
Total	100	

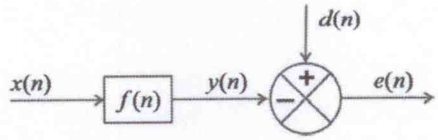
Notes:

(1) The exam is closed books and notes except for one double-sided sheet of notes.

(2) Please indicate clearly your answer to the problem.

(3) The details of your solutions are more important than the answers. Please explain your solutions clearly and include as many details as possible.

1. For the adaptive system shown to the right, assuming $f(n)$ is a linear, time-invariant moving average filter:



$$y(n) = \sum_{i=0}^{L-1} f(i)x(n-i)$$

(a) Derive the normal equations for the minimum least squares error estimate of the filter coefficients.

(b) Develop the concept of a linear prediction filter based on this model (explain how this model is modified to produce a linear prediction estimate of the filter coefficients).

(c) Derive the expression for the autocorrelation estimate of the linear prediction coefficients.

2. Modify the block diagram shown above to produce the basic LMS adaptive filter.

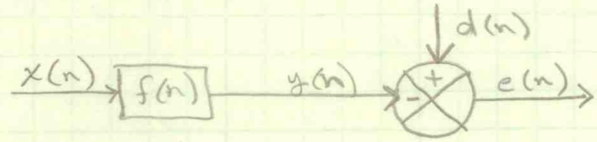
(a) Derive an expression for estimation of the filter coefficients using an iterative-in-time approach.

(b) Compare and contrast this to the approach in Prob. 1.

3. Suppose the input signal to the adaptive filter shown above is as follows: $x(n) = ax(n-1) + w(n)$. Assume $w(n)$ is zero-mean white Gaussian noise.

(a) Explain how successful the filter in no. 2 will be at correctly estimating the underlying parameters of this signal. Be as specific as possible and use terms such as the bias, variance, and convergence.

(b) Explain under what conditions we prefer approach no. 2 over approach no. 1. Again, be as specific as possible in your explanation, discussing issues such as, but not limited to, computational complexity.

1) 

$$y(n) = \sum_{i=0}^{L-1} f(i)\, x(n-i)$$

1a) Define the objective as such:

$$J = E\{e^2(n)\} = \sum_n e^2(n)$$

The goal is to design the filter, $f(n)$, to minimize the squared error output of the system.

From the diagram above we can see that:

$$e(n) = d(n) - y(n) = d(n) - f(n) * x(n)$$

$$= d(n) - \sum_i f(i)\, x(n-i)$$

We want to minimize the error by optimizing the filter coefficients thus we need to differentiate J w.r.t. the filter coefficients.

$$\frac{\partial J}{\partial f(j)} = \frac{\partial E\{e^2(n)\}}{\partial f(j)} = 2E\Big\{e(n) \underbrace{\frac{\partial e(n)}{\partial f(j)}}_{\text{lets find this expression}}\Big\}$$

$$\frac{\partial e(n)}{\partial f(j)} = \frac{\partial}{\partial f(j)}\left(d(n) - \sum_i f(i)\, x(n-i)\right)$$

$$= \cancel{\frac{\partial d(n)}{\partial f(j)}}^{\,0} - \frac{\partial}{\partial f(j)}\left(\sum_i f(i)\, x(n-i)\right)$$

$$\frac{\partial e(n)}{\partial f(j)} = -x(n-j)$$

Combining results we obtain:

$$\frac{\partial J}{\partial f(j)} = 2E\{e(n)(-x(n-j))\}$$

: Continue on next page

1a) continued

$$\frac{\partial J}{\partial f(j)} = 2E\left\{\left(d(n) - \sum_i f(i)x(n-i)\right)\left(-x(n-j)\right)\right\}$$

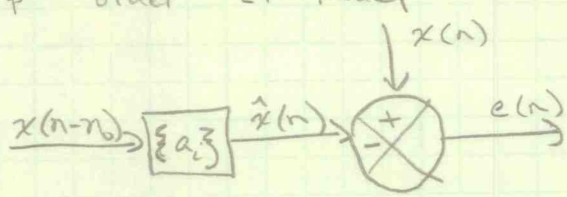
$$= -2\underbrace{E\{d(n)x(n-j)\}}_{\text{cross correlation}} + 2E\Big\{\sum_i f(i)\underbrace{x(n-i)x(n-j)}_{\text{Auto correlation}}\Big\}$$

$$\frac{\partial J}{\partial f(j)} = 2\left(\sum_i R(n-i,n-j)f(i) - g(n-j,n)\right)$$

Set equal to zero to minimize:

$$\boxed{\sum_i R(n-i,n-j)f(i) = g(n-j,n)}$$ Normal Equations

1b) p^{th} order LP model



$$\hat{x}(n) = \sum_{i=1}^{p} a_i\, x(n-n_0-i)$$

The linear prediction filter implements the concept of predicting a signal using a linear combination of past samples of the signal. For a zero mean stationary signal $x(n)$ a linear prediction takes the form of $\hat{x}(n)$ shown above. The coefficients, a_i, of the filter are called predictor coefficients. These predictor coefficients are determined via least squares minimization as explained in problem 1a. The objective function is formed, differentiated w.r.t the predictor coefficients, and equated to zero in order to minimize. This minimization results in a set of normal equations which are solved to find the predictor coefficients. In the generalized normal equations shown for least squares in 1a), $\bar{g}$ is a cross correlation vector. However, for the LP filter $\bar{g}$ consists of autocorrelation coefficients.

1c) p^{th} order LP model

$$\hat{x}(n) = \sum_{i=1}^{p} a_i x(n-n_0-i) \qquad \text{let } n_0 = 0$$

– prediction error :

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i)$$

– Objective function:

$$J = E\{e^2(n)\} = E\left\{\left(x(n) - \sum_{i=1}^{p} a_i x(n-i)\right)^2\right\}$$

$$J = E\{x^2(n)\} - E\left\{2x(n)\sum_{i=1}^{p} a_i x(n-i)\right\} + E\left\{\left(\sum_{i=1}^{p} a_i x(n-i)\right)^2\right\}$$

$$J = E\{x^2(n)\} - 2\sum_{i=1}^{p} a_i \left(E\{x(n)x(n-i)\}\right) + E\left\{\left(\sum_{i=1}^{p} a_i x(n-i)\right)^2\right\}$$

– Now differentiate J w.r.t. a_ℓ and equate to zero:

$$\frac{\partial J}{\partial a_\ell} = -2E\{x(n)x(n-\ell)\} + 2E\left\{\sum_{i=1}^{p} a_i x(n-i)\right\} x(n-\ell) = 0$$

– Rearrange terms to match covariance function definition:

$$\sum_{i=1}^{p} a_i E\{x(n-i)x(n-\ell)\} = E\{x(n)x(n-\ell)\}$$

$$\Rightarrow \sum_{i=1}^{p} a_i c(i,\ell) = c(0,\ell)$$ → This is the covariance method for LP

In matrix form $C\bar{a} = \bar{c}$

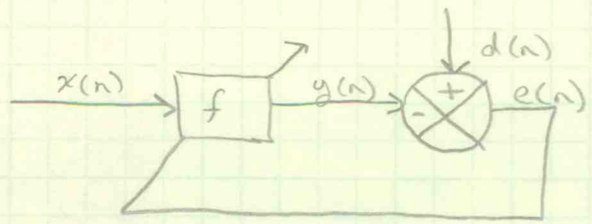
$$\bar{a} = C^{-1}\bar{c}$$

– The covariances are converted to correlations when we assume stationary inputs. This is known as the autocorrelation method.

$$\Rightarrow \bar{a} = R^{-1}\bar{r}$$

$$\bar{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \qquad R = \begin{bmatrix} r(0) & r(1) & \cdots & r(p-1) \\ r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix} \qquad \bar{r} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix}$$

2)



2a) In the least squares case the optimal solution to the normal equations is:

$$\bar{f}^* = R^{-1}\bar{g}$$

A generalized iterative solution is

$$\bar{f}^*_{n+1} = \bar{f}_n - \alpha_n \bar{P}_n$$

; the subscript denotes the update
α_n scales the step size
$\bar{P}_n$ is the search direction vector

Use steepest descent to determine P_n.

$$\bar{P}_n = \nabla J_n = \frac{\partial E\{e^2(n)\}}{\partial \bar{f}_n}$$

Now,

$$\bar{f}_{n+1} = \bar{f}_n - \alpha_n \nabla J_n$$

As shown in Prob. 1 $\nabla J = 2(R\bar{f} - \bar{g})$

Thus,

$$\bar{f}_{n+1} = \bar{f}_n - 2\alpha_n (R\bar{f} - \bar{g})$$

2b) The LP approach in Prob 1 directly solves the linear system of normal equations in order to determine the filter coefficients. The LMS approach is an iterative approach that uses feedback from the instantaneous prediction error. Since LMS is a iterative technique the filter must be initialized. This is typically done by setting $f_0 = \bar{0}$. However, if prior knowledge exists speed of convergence is increased if the selection of f_0 is close to the optimal solution. Also the addaptation constant, α, greatly effects the performance of the filter. Increasing the magnitude of α increases the iteration step size and thus potentially increases speed of convergence. However, increasing α also increases the algorithm's ability to react to noise, or be "too adaptive." Also if α is increased increased beyond certain limits divergence can occur. In short, while LMS has advantages in adaptation, it is not "idiot proof."

Chad Williams

ECE 8423 Exam #1 (5)

3) $x(n) = a x(n-1) + w(n)$; $w(n)$: zero mean white Gaussian

3a) The LMS adaptive filter will perform well as long as it is "set up" properly. Convergence and stability as well as the mean squared error are major performance concerns of the filter. The "magic" parameter appears to be the update step size, α_n. If α_n is set too small, convergence will be slow. However if α_n is set too large instability and divergence can occur. Over sized α_n's can also introduce bias in the mean squared error. In practice the step size should be bounded by the following relation in order to ensure stability as well as minimize error bias:

$$0 < \alpha < \frac{2}{L\,(\text{power of the input})}$$

3b) The main advantage of an LMS adaptive filter is in the aspect that the filter adapts or "reoptimizes" with each new data point. This is essential for input signals that are not statistically stationary. A LP filter requires inputs to be stationary or at a minimum stationary within the window of data that is being processed. In order to process a continually time varying signal (statistically) using a LP, data window length must be selected appropriately and a new least squares operator (filter coefficients) must be calculated. Thus at each interval a linear system of equations must be solved. The LP requires approximately $2L^2$ operations if the Levinson recursion is used. On the other hand LMS only requires $2L$ operations per step. In short, LP is more efficient for stationary signals, and LMS is better for non stationary signals.