Name: Danny Parker

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:

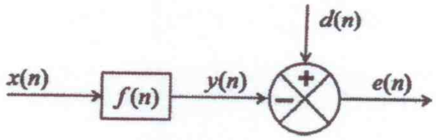
(1) The exam is closed books and notes except for one double-sided sheet of notes.

(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.

Danny Parker

1.) a) derive LMS filter coefficients

define objective function

$J = E\{e^2(n)\}$ where $e(n) = d(n) - y(n) = d(n) - f^T x(n)$

$E\{(d(n) - f^T x(n))(d(n) - f^T x(n))^T\}$

$E\{d^2(n) - 2d(n) f^T x(n) + f^T x(n) x^T(n) f\}$

$J = \sigma_d^2 - 2f^T g + f^T R f$ where $g = E\{d(n)x(n)\}$

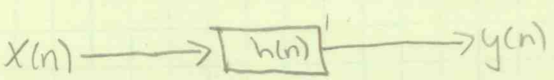
$R = E\{x(n)x^T(n)\}$

To find optimal solution $\frac{\partial J}{\partial f} = 0$

$\frac{\partial J}{\partial f} = 0 - 2g + 2Rf = 0$

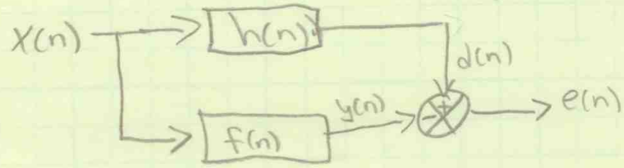
solve for f^* $\boxed{f^* = R^{-1} g}$

b) The goal of a linear prediction filter is to predict the next value based on the previous values. So in other words you need a model that given the input you can predict the output.



where h(n) is unknown and LTI but a linear predictor is needed to approximate it.

so to use this concept to estimate h(n)



for $f^* = R^{-1} g$ (part a)

then f^* will be a linear predictor for $h(n) * X(n)$

c) assume $X(n)$ is an iid zero mean process

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

$= E\left\{\left[\sum_{i=0}^{L-1} f(n) x(n-i)\right]\left[\sum_{j=0}^{L-1} f(n-j) X(n-i-j)\right]\right\}$

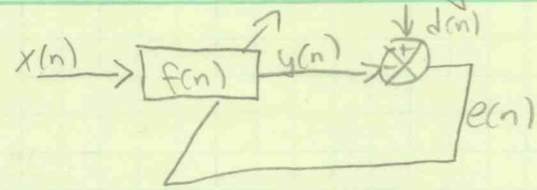
$= E\left\{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} f(n) f(n-j) X(n-i) X(n-i-j)\right\}$ only correlated when $i=j$

$r(n) = \sum_{i=0}^{L-1} f(n) r_{XX}(n-i)$

$\boxed{\text{since } \mu = 0 \text{ iid then } r(n) = \sigma_X^2 \sum_{i=0}^{L-1} h(i) h(i+n)}$

Danny Parker

2



a) assume filter update of form

$$f_{n+1} = f_n - c\,P$$

basically computing the next set of coeff as the previous ones plus some weighted change.

Since the defined design surface, $J = E\{e^2(n)\}$, is concave with one minima a good choice for P would be $\nabla J = 2E\{e(n)\frac{\partial e(n)}{\partial f}\}$

Which would lead to the solution from question 1, $\nabla J = 2(Rf - g)$

However since this is an 'iterative-in-time' approach the whole signal is not available so R and g can't be estimates

So replace ∇J with an instantaneous $\nabla \hat{J} = -2e(n)x_n$

for conviènence choose $c = \frac{\alpha}{2}$ and $P = \nabla \hat{J}$

This leads to $\boxed{f_{n+1} = f_n - \alpha\, e(n)x(n)}$

b) The obvious differences is computing f^* in one step vs many iterations.

In the iterative method you also have to worry about choosing α.

Choose too small and convergence will be slow; too large and it might not converge at all. Since the problem is concave the inital guess for f_0 will not matter much but since you are approximating the overall error at each step the final solution can have some bias.

Danny Parker

3. $x(n) = ax(n-1) + w(n)$ $\quad w = \mu = \phi \; \sigma_w^2$

a) If you want the filter to estimate $x(n)$ then the length of the filter needs to be at least 1 to fully estimate $x(n)$

$f^* = R^{-1} g \qquad R = \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix}$

$r(0) = E\{[ax(n-1) + w(n)][ax(n-1) + w_n]\}$

$= E\{a^2 x(n-1)x(n-1) + w(n)w(n) + 2ax(n-1)w(n)\}$

$= a^2 + \sigma_w^2 + \phi$

assume $E\{x(n)x(m)\} = \delta(n-m)$

$r(1) = E\{[ax(n-1) + w(n)][ax(n-2) + w(n-1)]\}$

$= E\{a\,x(n-1)x(n-2) + ax(n-1)w(n-1) + w(n)ax(n-2) + w(n)w(n-1)\}$

$= 0$

steady state $J_\infty = J_{min}(1 + \frac{\alpha}{2} L r(0)) = J_{min}(1 + \frac{\alpha}{2}(a^2 + \sigma_w^2))$

convergence $0 < \alpha < \frac{2}{L r(0)} = \frac{2}{a^2 + \sigma_w^2}$

time to converge $t = \frac{1}{\alpha \lambda_{min}} = \frac{1}{\alpha a^2 + \alpha \sigma_w^2}$

error $u_{n+1} = (I - \alpha R) u_n = (1 - a^2 + \sigma_w^2) u_n$

so as long as $0 < \alpha < \frac{2}{a^2 + \sigma_w^2}$ and $(a^2 + \sigma_w^2) < 1$ it will converge and have no bias

b) we prefer approach 2 over 1 when the whole data set is not known a priori or when the statistics of the data are changing over time (non-stationary). Method 2 has to worry about convergance issues but that can be mitigated by some effects when using Newton or RLS methods. Method 2 can be less computationally complex especially with large data sets. In some cases like RLS it's only $O(L^2)$.